MODEL ORDER REDUCTION IN POROUS MEDIA FLOW SIMULATION AND OPTIMIZATION

A Dissertation

by

MOHAMMADREZA GHASEMI

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,	A. Daniel Hill

May 2015

Major Subject: Petroleum Engineering

ABSTRACT

Subsurface flow modeling and simulation is ubiquitous in many energy related processes, including oil and gas production. These models are usually large scale and simulating them can be very computationally demanding, particularly in workflows that require hundreds, if not thousands, runs of a model to achieve the optimal production solution. The primary objective of this study is to reduce the complexity of reservoir simulation, and to accelerate production optimization via model order reduction (MOR) by proposing two novel strategies, Proper Orthogonal Decomposition with Discrete Empirical Interpolation Method (POD-DEIM), and Quadratic Bilinear Formulation (QBLF). While the former is a training-based approach whereby one runs several reservoir models for different input strategies before reducing the model, the latter is a training-free approach.

Model order reduction by POD has been shown to be a viable way to reduce the computational cost of flow simulation. However, in the case of porous media flow models, this type of MOR scheme does not immediately yield a computationally efficient reduced system. The main difficulty arises in evaluating nonlinear terms on a reduced subspace. One way to overcome this difficulty is to apply DEIM onto the nonlinear functions (fractional flow, for instance) and to select a small set of grid blocks based on a greedy algorithm. The nonlinear terms are evaluated at these few grid blocks and interpolation based on projection is used for the rest of them. Furthermore, to reduce the number of POD-DEIM basis and the error, a new approach is integrated in this study to update the basis online. In the regular POD-DEIM workflow all the snapshots are used to find one single reduced subspace, whereas in the new technique, namely the localized POD-DEIM, the snapshots are

clustered into different groups by means of clustering techniques (*k-means*), and the reduced subspaces are computed for each cluster in the offline (pre-processing) phase. In the online phase, at each time step, the reduced states are used in a classifier to find the most representative basis and to update the reduced subspace.

In the second approach in order to overcome the issue of nonlinearity, the QBLF of the original nonlinear porous media flow system is introduced, yielding a system that is linear in the input and linear in the state, but not in both input and state jointly. Primarily, a new set of variables is used to change the problem into QBLF. To highlight the superiority of this approach, the new formulation is compared with a Taylor's series expansion of the system. At this initial phase of development, a POD-based model reduction is integrated with the QBLF in this study in order to reduce the computational costs. This new reduced model has the same form as the original high fidelity model and thus preserves the properties such as stability and passivity. This new form also facilitates the investigation of systematic MOR, where no training or snapshot is required.

We test these MOR algorithms on the SPE10 and the results suggest twofold runtime speedups for a case study with more than 60,000 grid blocks. In the case of the QBLF, the results suggests moderate speedups, but more investigation is needed to accommodate an efficient implementation. Finally, MOR is integrated in the optimization workflow for accelerating it. The gradient based optimization framework is used due to its efficiency and fast convergence. This workflow is modified to include the reduced order model and consequently to reduce the computational cost. The water flooding optimization is applied to an offshore reservoir benchmark model, UNISIM-I-D, which has around 38,000 active grid blocks and 25 wells. The numerical solutions demonstrate that the POD-based model order reduction can reproduce accurate optimization results while providing reasonable speedups.

DEDICATION

To my beloved parents for their endless support and kindness throughout my life

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my advisor, Dr. Eduardo Gildin, for his excellent guidance, patience, and support. His continuous encouragement and believing in me in this research gave me the strength and inspiration to fulfill this work.

Special thanks to Dr. Efendiev who gave me a unique insight and helped me in understanding and solving the mathematical problems that I encountered during my research.

I would like to also thank my other committee members, Dr. King and Dr. Datta-Gupta for providing me with their constructive feedback and suggestions to improve this study.

I cordially express my gratitude to all my colleagues in our research group who supported me through this long journey. I want to give special thanks to Yanfang Yang, who was working with me on part of this project, for useful discussions and knowledge sharing regarding POD-DEIM model reduction.

I would like to acknowledge Exxon-Mobil for the financial support at the beginning of this project.

NOMENCLATURE

Abbreviations

BHP	bottom hole pressure
DEIM	discrete empirical interpolation method
MOR	model order reduction
MRST	MATLAB reservoir simulation toolbox
NPV	net present value
POD	proper orthogonal decomposition
QBDAE	quadratic bilinear differential algebraic equation
SVD	singular value decomposition
TPWL	trajectory piecewise linearization

Variables

f_w	Buckley-Leverett fractional flow function	[–]
J	Jacobian matrix	[–]
K	permeability	[ft^2]
p	pressure	[psia]
q	volumetric flow rate	[bbl/day]
Q	sink/source term	[–]
R	residual vector	[–]
s	water saturation	[–]
t	time	[day]
T	transmissibility matrix	[–]
u	phase velocity	[ft/day]

Greek

δ	solution update
λ	mobility or eigenvalues
μ	viscosity
ϕ	porosity
Φ	POD basis
Ψ	DEIM basis

subscripts

o	oil
w	water
ro	relative to oil
rw	relative to water

subscripts

n	nth time step
k	kth nonlinear iteration

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

My research focuses on petroleum engineering, economics, and mathematics. The primary objective of this study is to reduce the complexity of reservoir simulation, as well as accelerating production optimization. This is certainly fundamental for future of reservoir simulation, since it would facilitate the creation of new levels of large scale reservoir model simulation and optimization.

1.1 Background

Every industry is dependent on energy; from education, medical, transport to corporate industries. Consequently, there is an essential need for secure and sustainable energy resources. Hydrocarbons, such as oil and gas, have been one of the main sources of energy for the last 150 years. Petroleum engineers play significant roles in recovering hydrocarbons efficiently while reducing environmental impacts.

According to recent *energy outlook* surveys and publications [32, 40], projections for fossil fuel extraction are on the rise. The World, and in particular, the US growing domestic production in unconventional reservoirs (e.g., from tight oil and shale formations) will continue to reshape the energy outlook for years to come [5, 4]. In order to meet the increasing demand for energy, the Oil and Gas industry needs to continue to perform discovery/exploration campaigns, but most importantly, will have to look for better reservoir management practices for fields that have already been developed. Achieving these goals require understanding the physics behind subsurface flow and different parameters that facilitate the production of hydrocarbons. One of the main tools is having a representative model, usually called reservoir model, and being able to simulate different scenarios.

The reservoir model is derived by considering the interaction between a porous

rock and underground fluid flow in a high pressure and temperature media. This phenomenon can be described by partial differential equations (PDEs) that need to be solved for analysis and development. Generally, these equations need to be solved numerically using proper time and spatial discretization techniques [29].

Reservoir simulation is an area of reservoir engineering in which computer models are used to simulate and predict the flow of fluids (typically, oil, water, and gas) through porous media. It involves a mathematical model governed by partial differential equations (PDEs) describing subsurface flow. Reservoir simulation assists us to take long term critical decisions in developing a field, e.g. well placement and injection/production rate. It also helps us in uncertainty assessment, sensitivity analysis, history matching and optimizing control inputs to have higher reservoir recovery factors (often called production optimization) [116].

For some petroleum fields, optimization of production operations can be a major factor on increasing production rates and reducing production cost. However, these reservoir models often times have large number of grid blocks and include multi-physics and multi-components. This results in a large scale model that is computationally expensive to simulate. Despite the great advances in reservoir modeling and the advent of high-performance computing (HPC), high-fidelity physics-based numerical simulation still remains challenging due to its complexity and computational time. This problem is revealed in more computationally intensive frameworks, history matching [87, 3], production optimization problems [108, 33] and uncertainty quantifications [67].. Also, the computational time of such large-scale models become the bottleneck of fast turnarounds in the decision-making process and assimilating real-time data into the model and closed-loop reservoir management [69, 53, 49].

Over the past decade, numerous techniques have been applied in the context of porous media flow simulation to reduce the computational effort associated with the

solution of the underlying coupled nonlinear partial differential equations. These techniques vary from heuristic approaches [83, 76], to more rigorous mathematical techniques [8],which explore the idea of determining a *reduced complexity* model that can approximate the full nonlinear system of equations with certain accuracy. In many cases, reduced-order modeling techniques have shown to be a viable way of mitigating computational complexity in simulation of the large-scale model, while they maintain high level of accuracy when compared with high fidelity models.

There are different techniques for model order reduction (MOR) and constructing lower complexity models. For example, upscaling is used to coarsen the reservoir description and make it suitable for flow computation [35, 66]. Multi-scale method is another approach accounting for different scales on the discretization of the underlying partial differential equations [37, 36]. These methods yield approximations that can preserve local features of the porous media flow model. However, in this thesis I focus on MOR techniques that uphold global characteristics of the reservoir model.

1.2 Research Scope

The objectives of this thesis are,

- Assessing the application of model order reduction to reduce the computational cost of the reservoir model.
- Addressing the existing issues in MOR application to nonlinear system and develop new techniques to remove/mitigate these problems.
- Discussing an efficient procedure for optimization of a petroleum field under physical and economic constraints is considered.
- Integrating the MOR technique in optimization workflow to accelerate produc-

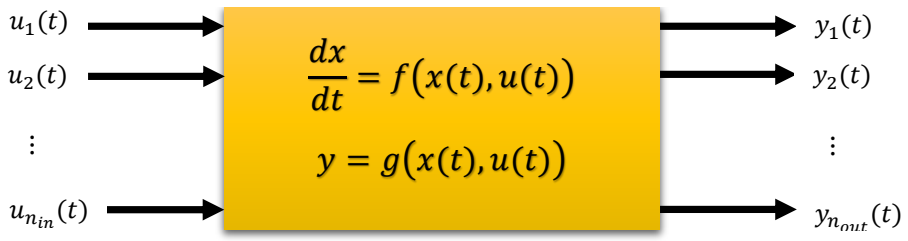


Figure 1.1: Input-output system description

tion optimization.

The impact of this work is also to facilitate simulating various scenarios for increasing production and improving reservoir management. Preliminarily, I describe my research on model order reduction, before discussing my work on optimization, and finally make deductions on the findings, as well as my future research plans.

1.3 Literature Review

1.3.1 Model Order Reduction

A dynamical system can be viewed as an input-output system, where different set of control variables yield unique responses. In this perspective a system can generally have multiple inputs and multiple outputs as shown in Figure 1.1.

Assuming that there is n states (grid blocks) in a model, The general form of nonlinear systems with n_{in} number of inputs and n_{out} number of outputs is,

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t)), \quad t \geq 0 \\ y(t) &= h(x(t), u(t)), \quad x(0) = x_0, \end{aligned} \tag{1.1}$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^{n_{in}}$, $f : \mathbb{R}^n \to \mathbb{R}^n$, and $h : \mathbb{R}^n \to \mathbb{R}^{n_{out}}$.

Model order reduction (MOR) is basically a technique aimed at reducing the computational complexity of mathematical models used in the context of numerical

simulations. Model order reduction is a systematic approach to approximate a high dimensional model with a meaningful lower dimension model such that it becomes computationally faster to simulate and generate similar results in terms of input-output behavior as described in system of equations 1.2,

$$\begin{aligned} \dot{\hat{x}}(t) &= \hat{f}(\hat{x}(t), u(t)), \quad t \geq 0 \\ \hat{y}(t) &= \hat{h}(\hat{x}(t), u(t)), \quad \hat{x}(0) = \hat{x}_0, \end{aligned} \tag{1.2}$$

where $\hat{x} \in \mathbb{R}^r$, $\hat{f} : \mathbb{R}^r \to \mathbb{R}^r$, $\hat{h} : \mathbb{R}^r \to \mathbb{R}^{n_{out}}$, and $r \ll n$.

In order to have a reliable input-output approximation of the original model, the following conditions should hold [106],

- The norm of approximation error should be small,
- The crucial properties of the original system, such as stability, can be preserved.
- The reduction procedure should be computational efficient to avoid large overhead.

Model order reduction has been successfully applied to large class of problems in diverse areas such as control engineering [7], micro electro mechanical systems [109, 30], power electronics [91, 24], signal processing, video and image compression [107], and fluid mechanics [97]. Also, it has been presented in reservoir simulation applications in recent years [65, 52]. However, there are still many challenges left to be addressed mainly in nonlinear systems and in preserving stability and robustness of the original system.

Model order reduction in reservoir simulation is akin to upscaling method in the sense that both techniques approximate the original large scale model with a reduced one that is computationally faster and yields similar results in terms of

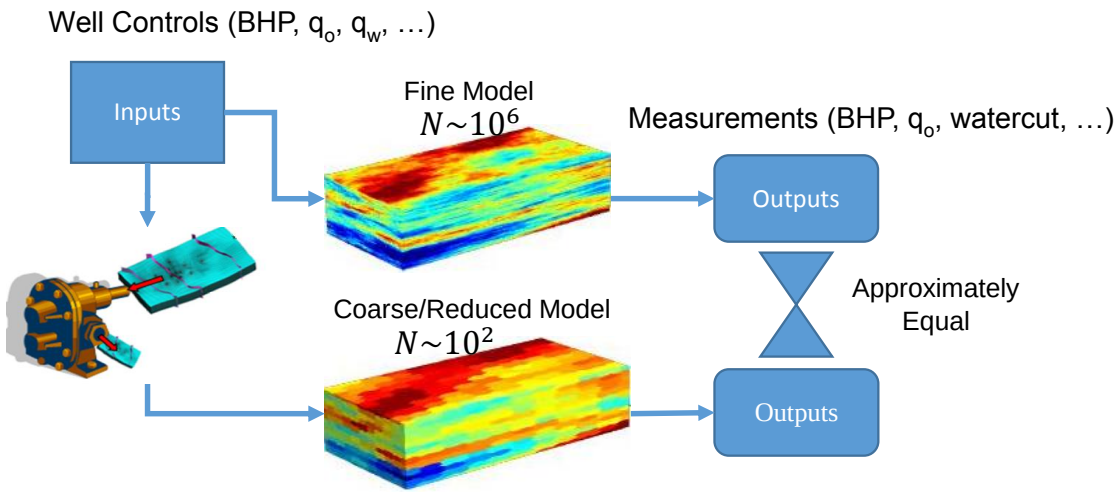


Figure 1.2: Model order reduction concept

oil production, water cut, oil saturation among other measurements as described in Figure 1.2. However, these techniques are significantly different because MOR reduces them based on the dynamic response of the entire model, whereas upscaling is relatively local and is applied to the adjacent grid blocks.

MOR techniques vary from non-intrusive methods, i.e., do not depend on modifications of a reservoir simulation code, to a more intrusive and sophisticated methods that depend on several modifications of legacy code or the development of new simulator codes [9]. In the case of non-intrusive methods, data-driven model reduction has been the choice in material balance-type modeling such as the capacitance-resistance model (CRM) [114], application of artificial intelligence and neural networks [83], network flow model [76], and dynamic mode decomposition techniques [50, 51]. These models are simple, easily implementable and computationally much faster than original simulation models. However, they are mainly mathematical models and are not based on the underlying physics of the problem. Thus, a lot of trainings are required

to get a suitable model, which may often times fail in forecasting the production [12].

The main focus of this work will be on intrusive schemes, whereby one will modify the governing equations of the model to achieve a new system of equations with less unknown parameters. These methods are still capable of capturing some of the properties of the original model. Different MOR methods are compared in [64], assuming reservoir simulation model both as linear and nonlinear system. These techniques, derived from control theory, are proper orthogonal decomposition (POD), balanced truncation, and subspace identification. It was shown that POD gave the best approximation for reservoir simulation models. There are other methods based on frequency domain e.g. Krylov subspace [14], however they are more applicable to linear systems and also there is no error bound on the reduced model. Most of MOR methods are projection based techniques, meaning that the high-dimensional state-variables (i.e. oil pressures and water saturations in two-phase flow) are projected into a much smaller subspace so that the input/output relationship is preserved with certain accuracy [52].

In this work, POD is considered as a good candidate for MOR. POD is a mathematical procedure that utilizes an orthogonal transformation to convert a given set of observations of conceivably correlated variables into a set of variables that are linearly independent and are referred to as basis [75]. These basis are the eigenvectors corresponding to the most significant eigenvalues of a covariance matrix. In many cases the covariance matrix can be approximated by using a set of solutions at different time steps, referred as snapshots [81, 111]. In other areas of research, POD is referred to as principal component analysis (PCA) as well as discrete Karhunen-Loeve (K-L) transformation. POD is the optimal solution to minimize the energy of the error between the exact model and reduced one [112]. The computational cost of generating the basis are low and its implementation is relatively easy. However,

if the input control deviates too much from the training inputs, the reduced order model may not provide an accurate results [97].

In order to apply POD to reservoir simulation, it is necessary to run the fine scale model with a specific training schedule while saving the results at each time step, and this is referred to as snapshot. These snapshots are used in the POD framework for finding the basis of the reduced subspace and constructing a projection matrix. This is usually referred to as offline stage. Consequently, with a new sets of input parameters and schedule one can run the reduced order model instead of original large scale one. This phase is usually called online stage.

In spite of being computationally simple and easily implementable, POD in nonlinear systems is not computationally very efficient. In linear systems the reduced order model can be computed offline (all the matrices can be computed before simulating the model). Thus, less number of states need to be solved online compared to high dimensional model, and there is no need to project back to fine scale. However, in nonlinear models the parameters are state dependent and usually the resulting system matrices are varying. Also to evaluate the function properties, the reduced state need to be projected back to fine scale in the online phase [55]. These nonlinear functions are used to evaluate the fluid and rock properties at different pressure, saturation [38]. This may result in a model that is computationally even more expensive than the original model. The most important thing in this context is to approximate a nonlinear function to avoid projecting back to large scale space. There are different remedies to resolve this issue.

One of the approaches is Trajectory-Piecewise Linear (TPWL) [94, 56], which is based on approximating the nonlinear functions by a weighted sum of linearized models at several known trajectories derived from training inputs. When the input parameters changes, the closest linearized model will be selected for simulation in-

stead of the original high order nonlinear model. Even though this methodology has shown to yield good approximation and also reduction of several orders of magnitude in the size of the problem [22], there are still many nonlinear functions that may not be approximated well by linearization. Thus, it might result in instability and wrong solution if the new trajectory is not close to the linearization point. Also, several state trajectories and Jacobian matrices need to be saved, which may take a lot of disk space even for relatively small size problems.

Another approach is missing point estimation (MPE) [10], which is based upon the theory of Gappy POD [39]. The key idea of MPE is to evaluate nonlinear terms only at few points and therefore it is considerably faster than projecting back by inner products and evaluation of the large scale. Moreover, it can be shown that under mild conditions, this computational acceleration does not effect the accuracy of results significantly. An optimal selection of points is done by minimize aliasing effects in using only partial spatial points. It is shown in [10] that since this problem is combinatorial optimization, one can solve a suboptimal problem based on greedy algorithm, where a point that yields the lowest value of conditional number of $\phi^T \phi$ (where ϕ is the projection matrix) is added to the selected set of points, and the process is repeated. In this way, the subset is constructed by choosing one point at a time. The main application of MPE is in the context of linear time varying systems.

In this thesis, I used a discrete empirical interpolation method (DEIM) [27, 55], a variant of empirical interpolation method (EIM), that is suitable for reducing the dimension of systems of ordinary differential equations arising from finite difference discretization of time dependent PDEs and/or parametrically dependent steady state problems. This method approximates a nonlinear function by combining projection with interpolation. DEIM is similar to MPE in the sense that both methods employ a smaller set of spatial grid points for evaluating the nonlinear terms to avoid projecting

back (L_2 inner product) to large scale space and evaluate nonlinearities. However, the fundamental procedures for selecting a set of spatial grid points are different. While MPE focuses on reducing the number of equations and considers only a POD basis for state variables, DEIM focuses on approximating each nonlinear term individually and considers both POD basis for state variables and each nonlinear term [28].

DEIM is also similar to best point interpolation method (BPIM) [86] in the context of selecting a subset of grid points and interpolate the nonlinear terms in the rest of the points. However, they have different selection method. While the interpolation points in BPIM are determined from a least-squares minimization problem, DEIM uses greedy algorithm to select the spatial grid points.

I considered the POD-DEIM for model reduction in porous media flow. Thus, the nonlinear terms are approximated by some form of interpolation, and therefore great reductions can be achieved in computational and the problem can be solved for all the time steps in the reduced space. In this case, the nonlinearities are also approximated based on interpolation of a POD-like basis obtained from the snapshots of nonlinear terms. It is possible to reduce the size of reservoir model few orders of magnitude in reservoir simulation and optimization by applying POD-DEIM [48]. Note that the global MOR methods, e.g. POD-DEIM, and the local MOR ones, e.g. multi-scale, can be combined to form global-local model reduction technique [48, 37]. However, this approach is not discussed in this thesis and I only consider global MOR.

If a system exhibits dynamic results with a wide array of changes, however, many of the POD and DEIM basis would be required to accurately approximate the state of the system and nonlinear terms. An intervention or solution to this is to search for not only one global subspace, but also multiple local basis and subspaces to capture particular dynamics [89]. With localization in perspective, the dimension of the reduced space could be decreased even further and in return the computational

cost can be brought down.

In localized MOR, rather than just one global reduced subspace, multiple local subspaces are constructed for the reduced system. The main idea is, in the offline stage,to divide the snapshots into different subgroups and apply POD to each domain to obtain local POD basis. This will require less number of basis for each region compared to global POD basis. This will also give us a better approximation of the solution trajectory in each subdomain, because more representative basis will be used. In the online stage, based on the state of system at each time step the proper subspace that has the best approximation will be selected. The localization idea was introduced for POD method in [6] and for DEIM in [89]. The localization is conducted based on machine learning techniques and using efficient classification algorithm [68]. We extend this approach to POD and introduced localized POD (LPOD). The LPOD as well as localized DEIM (LDEIM) is applied to reduce the computational cost of porous media flow simulation.

There are different techniques to split the solution snapshots space into different regions. In most of these methods, the domain is split into subdomains recursively [34], but this method might results in a large number of subdomains in practice. The unsupervised learning methods was proposed in [6] that can results in unstable clustering behavior if the clustering method and its parameters are not carefully chosen [113]. Furthermore, the given procedure requires precomputing auxiliary quantities to ensure an online selection procedure that is independent of the dimension of the original system; however, the number of these auxiliary quantities, which are computed in the offline phase, scales cubically with the number of clusters. This can be an issue especially in optimization workflow, where one updates basis of reduced model periodically to obtain more stable and accurate solution.

Here, I use machine learning as suggested in [89] to classify the snapshots into

different clusters and for each cluster a local reduced-order model is constructed . A local reduced-order model is then selected with respect to the current state of the system. I also propose modified algorithm for localized POD that only uses few number of indexes for declassification in the online phase to find the corresponding cluster. This also reduces the storage required for the auxiliary parameters. I applied localized POD (LPOD) as well as localized DEIM (LDEIM) to reduce the computational cost of porous media flow simulation [47].

Additionally, for nonlinear systems with analytical function of state and input under small perturbation inputs, the input-output response can be approximated closely by Volterra series [99]. This family of nonlinear systems are referred to as *weekly nonlinear systems*. One can use the first terms in this series and end up with a linear model. Even though the resulted linear model is simple, in many applications it is not sufficient to describe the behavior of physical processes. Thus, the bilinear systems are established to adequately describe practical processes and provide deeper insight into the properties of the original systems.

Bilinear systems are a special class of nonlinear systems that are linear in input and linear in state but not jointly linear in state and input. This type of systems arises as natural models for a variety of physical and biomedical processes, e.g. the modeling of nonlinear RC circuits, and micro electromechanical systems (MEMS) [15]. The study of such systems goes back to early 1970s [72, 84].

The first step into the bilinear model reduction framework is to reformulate the problem based on Volterra's series expansion of nonlinear terms and to find the system matrices to be used into the reduction algorithm. Modeling the problem in bilinear form will primarily increase the number of states. Although this might seems counter intuitive, increasing the size of the system before applying model reduction can help us find better basis and reduced subspace. However, the size of the problem

in bilinear form based on Volterra's series increases quadratically. Thus, these models usually have very large dimension, requiring an efficient model order reduction for general applicability.

Generalized balanced truncation has been applied for bilinear model reduction [19]. These methods give satisfying reduced model, but have the computational challenges in finding projection matrices as in the linear case. Recently two new approaches for MOR of bilinear systems have been proposed in [42] based on interpolation methods, optimal H_2 MOR applying multi-point interpolation, and constructing bilinear system realization from sampled data.

An alternative approach is to reformulate the problem as a quadratically bilinear system. There is no Taylor expansion or approximation in converting original model to this form of systems [58]. Most of the systems with nonlinear terms can be converted into this form by defining a new state and some algebraic equation. Thus, usually the nonlinear ODE will be modified and results in a quadratical differential algebraic equation. As there is no approximation in the quadratic bilinear systems, it captures the dynamic of the original nonlinear system accurately. Also, this formulation allows certain properties of the original models to be preserved even in the reduced order models, such as stability [17].

In this thesis, first bilinear iterative rational Krylov algorithm (BIRKA), as proposed in [17], is applied successfully to bilinear model reduction of single phase flow [46]. In Krylov method the basis are generated to approximate a large scale model around specified frequencies. These basis are applied to project the high dimensional state of the model to a lower one. Krylov subspace preserve stability and passivity of the system. However, the main drawback of this approach is that there is no guaranteed on the error bound [57]. I also reformulate the two-phase flow models to quadratic bilinear system and compare it with the bilinear form based on Volterra's

series expansion [54]. It will be shown that this method can be generalized for different nonlinear terms as well, and it can be applied it to large heterogeneous reservoir models.

1.3.2 Production Optimization

It is important to produce the existing oil and gas reservoir efficiently. To achieve this goal many optimization workflows can be considered based on certain type of functional to be maximized/minimized, for example sweeping efficiency, maximizing net present value (NPV), and cumulative oil production among others. The production optimization problem addresses these challenges as a sequences of control inputs. This optimization is performed on a simulation model that captures the physics of the real reservoir. Our main focus here is on water flooding optimization, determining fluid injection rates and production targets.

The optimized well controls can be determined iteratively as shown in Fig. 1.3 applying gradient based method. The control parameters are updated iteratively in the negative direction of the gradient of the cost function until the optimal values are found (at which the gradient is close to zero).

The gradient of the objective function with respect to input parameters can be computed either numerically or by solving adjoint equations [20]. The numerical gradient is calculated by perturbing each control input. Although this approach is easily applicable to any existing simulator, it is not efficient for a problem with large number of control inputs. The total number of simulation model runs required is as many as number of control inputs plus one [22]. Thus, in a problem with large number of control parameters, it is more efficient to use adjoint formulation for computing gradient.

In gradient based optimization, adjoint methods are widely used for calculating

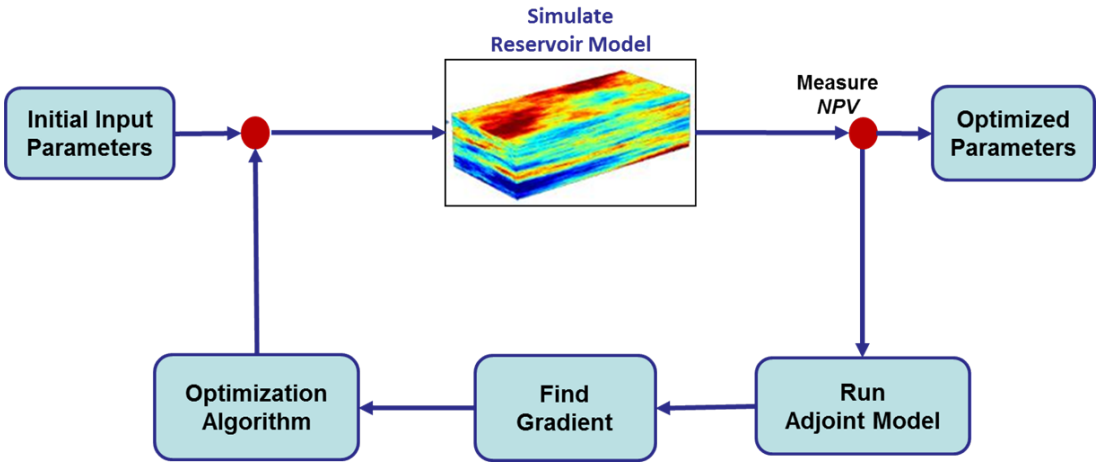


Figure 1.3: Optimization framework

the gradient in problems with large number of variable parameters. The first step is to solve the forward model, e.g. reservoir simulator, for all time steps with given initial conditions and control inputs. At this stage all the dynamic states, e.g. pressure and saturation as well as Jacobians should be stored for all the time steps. The cost function will be evaluated with results of the forward model simulation.

Next, the adjoint model can be solved using the stored states to calculated the Lagrange multipliers. The gradients will be calculated using the Lagrange multipliers and forward model results as explained in [103, 105]. The gradient will be used to update the control input. This process should be repeated until the gradient is sufficiently close to zero.

Water flooding optimization problem has been primarily considered in [33]. They proposed adjoint based gradient to solve the problem and also applied POD model reduction on both forward and adjoint model to reduce the computational time. They applied this method to a relatively small problem (45x45) with many input controls (2x45). Even though their approach found the maximum NPV, they observed modest speedup. The main reason is that to evaluate fluid and rock properties, it is required

to have the pressure and saturation in the fine scale at each iteration.

The production optimization under nonlinear constraints was considered in [104]. They formulated the problem as constrained nonlinear programing (NLP) and solved it based on penalty functions.They proposed to find an approximate feasible direction and then projecting in onto active constraints and solving them during the forward model evaluation. The production optimization (without nonlinear constraints) were solved based application of POD-TPWL model reduction technique in [22]. However, they used perturbation method for calculating the gradient because the adjoint model was not implemented in their simulator. Also, after the initial training the reduced model would be used for the entire optimization with updating the basis or the reduced model. This can be problematic in an optimization problems where the solution is far from the initial training schedule.

For systematic retraining of reduced model during production optimization, [108] used trust-region methodology. They also applied POD-DEIM model reduction to accelerate the production optimization. [63] solved the production optimization based on a gradient free method (pattern search), however this approach usually needs extensive number of iterations to find the optimal solution.

1.4 Thesis Outline

In this thesis, primarily I review the underlying partial differential equations describing the flow in subsurface porous media and discuss different possible methods to discretize these equations and solve them efficiently. Next, the POD-DEIM as well as localized POD-DEIM model reduction will be applied to porous media flow and it will be verified in some examples and case studies. This chapter will be followed by introducing bilinear systems and quadratic bilinear formulation. These two methods will be compared and it will be shown how quadratic bilinear form can

be used to reformulate two phase flow. The model reduction will also be applied to reduce the size of the problem. Then, the model order reduction in the context of optimization will be considered and a case study of water flooding optimization on an offshore reservoir model will be presented. Ultimately, this thesis will be concluded by overview of results and future work in the last chapter.

2. RESERVOIR SIMULATION

The flow in porous media can be described by a set of partial differential equations representing conservation of mass, momentum, and energy as a function of pressure, temperature, and saturation/fraction of each phase/component [13]. In this chapter, I summarize the underlying partial differential equations related to porous media flow simulation. In particular, two-phase oil-water reservoirs will be briefly discussed. This type of reservoirs is usually common in water flooding cases, where the pressure support is provided by injecting water into reservoir. It is assumed that the reservoir is above bubble point and the oil component is only in the liquid phase.

2.1 An Incompressible Two-Phase Flow Model

The two-phase flow is considered in a reservoir domain (denoted by Ω) under the assumption that the displacement is dominated by viscous effects; i.e., we neglect gravity, compressibility, and capillary pressure. The two phases are water and oil, and they are assumed to be immiscible. We write Darcy's law for each phase as follows:

$$u_l = -\frac{k_{rl}(s)}{\mu_l} K \nabla p, \tag{2.1}$$

where u_l is the phase velocity, K is the permeability tensor, k_{rl} is the relative permeability to phase l ($l = o, w$), s is the water saturation (we use s instead of s_w for simplicity) and p is pressure. Throughout this thesis, we will assume that a single set of relative permeability curves is used. Combining Darcy's law with a statement of conservation of mass allows us to express the governing equations in terms of the

so-called pressure and saturation equations:

$$
\begin{aligned}
-\nabla \cdot (\lambda(s) K \nabla p) &= q_w + q_o \quad \text{in} \quad \Omega, && (2.2)\\
\phi \frac{\partial s}{\partial t} + \nabla \cdot (f_w(s) u) &= \frac{q_w}{\rho_w} \quad \text{in} \quad \Omega. && (2.3)\\
u \cdot n &= 0 \quad \text{on} \quad \partial\Omega \quad : \text{no flow at boundary} && (2.4)\\
s(t=0) &= s_0 \quad \text{in} \quad \Omega \quad : \text{initial known saturation.} && (2.5)
\end{aligned}
$$

where ϕ is the porosity, $u = u_w + u_o$ is the total velocity. Moreover, q_w and q_o are volumetric source terms for water and oil and λ is the total mobility defined as,

$$\lambda(s) = \lambda_w(s) + \lambda_o(s) = \frac{k_{rw}(s)}{\mu_w} + \frac{k_{ro}(s)}{\mu_o}, \tag{2.6}$$

$f_w(s)$ is the fractional flow function,

$$f_w(s) = \frac{\lambda_w(s)}{\lambda(s)} = \frac{k_{rw}(s)}{k_{rw}(s) + \frac{\mu_w}{\mu_o} k_{ro}(s)}. \tag{2.7}$$

The above descriptions are referred to as the fine or high fidelity model of the two-phase flow problem. The Eqs. (2.2) and (2.3) are nearly elliptic, see [1], and I follow the sequential formulation. Thus, one solves for pressure first and then use the result to solve for saturation at each time step.

2.1.1 Pressure Equation: Mixed Formulation

The pressure equation is solved based on mixed finite element method (FEM) [43]. By introducing the flux variable $u = -\lambda(s) K \nabla p$, we get the following first-

order system

$$\begin{aligned} (\lambda(s)K)^{-1}u + \nabla p &= 0, \\ \nabla \cdot u &= q. \end{aligned} \tag{2.8}$$

To drive the mixed FEM, we primarily define the following Sobolev space with imposing no flow at the boundary,

$$H_0^{1,div} = \{u \in (L^2(\Omega)) : \nabla \cdot u \in L^2(\Omega) \,\text{and}\, u \cdot n = 0\}. \tag{2.9}$$

The mixed formulation of Eq. (2.8) is to find $(p, u) \in L^2(\Omega) \times H_0^{1,div}(\Omega)$ such that,

$$\begin{aligned} ((\lambda K)^{-1}u, v) + (\nabla p, v) &= 0 \qquad \forall v \in H_0^{1,div}(\Omega) \\ (\nabla \cdot u, l) &= (q, l) \quad \forall l \in L^2(\Omega) \end{aligned} \tag{2.10}$$

where $(\cdot, \cdot)$ is the L^2 inner product. To make this system of equation well posed, one should add an extra constraint, e.g. $\int_\Omega p \, dx = 0$.

In a mixed FEM discretization we replace the pressure and velocity solution space ($L^2(\Omega)$ and $H_0^{1,div}$) by finite dimensional subspaces that typically consists of low order piecewise polynomials (triangles, quadrilateral,etc). For further discussion one can refer to [1]. Defining the finite element spaces Q and V,

$$\begin{aligned} Q &= \{p \in L^2(\Omega) : p|_{\Omega_i} \quad \text{is constant} \quad \forall \Omega_i \in \Omega\} \\ V &= \{v \in H_0^{1,\text{div}}(\Omega) : v|_{\Omega_i} \quad \text{have linear components} \quad \forall \Omega_i \in \Omega, \\ &\quad (v \cdot n_{ij})|_{\gamma_{ij}} \text{ is constant } \forall\ \gamma_{ij} \in \Omega, \text{ and } v \cdot n_{ij} \text{ is continuous across } \gamma_{ij}\} \end{aligned} \tag{2.11}$$

where n_{ij} is the unit normal to γ_{ij} pointing from Ω_i to Ω_j. The corresponding

Raviart-Thomas mixed FEM is to find $(p, u) \in Q \times V$ such that

$$\int_\Omega (\lambda(s)K)^{-1} u \cdot v \ dx \ - \ \int_\Omega p \nabla \cdot v \ dx = 0, \tag{2.12}$$

$$\int_\Omega (\nabla \cdot u) \ l \ dx = \int_\Omega q \ l \ dx, \tag{2.13}$$

for all $v \in V$ and $l \in Q$. After discretization and writing $p = \sum_{\Omega_m} p_m \ \phi_m$ and $v = \sum_{\gamma_{ij}} v_{ij} \psi_{ij}$, where $\phi_m \in Q, \psi_{ij} \in V$ are the basis functions; the system of equations (2.12) takes the form,

$$\begin{pmatrix} B(\lambda(s)) & -C^T \\ C & 0 \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} 0 \\ g \end{pmatrix} \tag{2.14}$$

where $B = [b_{ij,kl}]$ and $C = [c_{m,ij}]$ defined as,

$$b_{ij,kl} = \int_\Omega (\lambda(s)K)^{-1} \psi_{ij} \cdot \psi_{kl} \ dx, \quad c_{m,ij} = \int_{\Omega_m} \nabla \cdot (\psi_{ij}) dx \tag{2.15}$$

and g results from the sink/source terms. Here, we keep the same notations for discrete and continuous variables for simplicity.

2.1.2 Saturation Equation

Generally, an implicit time (backward Euler) discretization will be followed to solve for saturation profile, while a mass conservative finite volume is used for the spatial derivative discretization. Consider a cell Ω_i with edges γ_{ij} and associated normal vectors n_{ij} pointing out of Ω_i, the saturation Eq. (2.3) will be discretized as,

$$s_i^{n+1} = s_i^n + \frac{\Delta t}{|\Omega_i|} \left(q^+ - \sum_j F_{ij}(s^{n+1}) u_{ij} + f_w(s_i^{n+1}) q^- \right). \tag{2.16}$$

where s_i^n is the cell-average of the water saturation at time $t = t_n$, $q^+ = \max(q_i, 0)$ and $q^- = \min(q_i, 0)$, u_{ij} is the total flux (for oil and water) over the edge γ_{ij} between the two adjacent cells and F_{ij} is a numerical approximation of the flux over edge γ_{ij} defined as,

$$F_{ij} \approx \int_{\gamma_{ij}} (f_{ij}(s)u_{ij}) \, .n_{ij} \, dv. \tag{2.17}$$

There are different schemes to evaluate the integrand in (2.17). It is common to use first order approximation, known as upstream weighting, defined as,

$$f_{ij}(s) = \begin{cases} f_w(s_i) & \text{if } \; u_{ij} \cdot n_{ij} \geq 0; \\ f_w(s_j) & \text{if } \; u_{ij} \cdot n_{ij} < 0. \end{cases} \tag{2.18}$$

Note that in a sequential approach, first the pressure equation is solved on a fine grid. Next, the saturation equation is solved implicitly. Note that the pressure and velocity equations in (2.14) is linear system, whereas the saturation equation in (2.16) is nonlinear system and can be solved for s^{n+1} by iterative methods such as Newton-Raphson efficiently. The residual as a function of saturation is defined as follows,

$$R(s) \triangleq s - s^n - A(u) \; f_w(s) - \frac{\Delta t}{|\Omega|} q^+, \tag{2.19}$$

where s^n is the vector of cell-saturations at time instance t_n and $f_w(s)$ is the vector of fractional flow values including $f_w(s_i)$, A is a matrix implementing $\frac{\Delta t}{|\Omega|}[-\sum_j F_{ij}(s^{n+1})u_{ij} + f_w(s_i^{n+1})q^-]$ on a cell by cell basis. The derivative of Residual in (2.19) is called Jacobian defined as,

$$J(s) \triangleq \frac{\partial R}{\partial s} = \mathrm{I} - A \; f_w'(s). \tag{2.20}$$

where f'_w is the explicit derivative of fractional function with respect to saturation. At each iteration of Newton method, the current state is updated as,

$$s^{k+1} = s^k - \alpha\ J(s^k)^{-1}\ R(s^k), \tag{2.21}$$

and the iterations are stopped at each time step when the norm of the update or Jacobian are smaller than a tolerance. Note that we can have an outer iteration loop as well, meaning that after finding the saturation, one can repeat these steps for the current time step until the changes in pressure is smaller than a small tolerance. However, in this thesis we assumed that the time steps are small and there is no need to have an outer iteration loop. The flowchart of sequential reservoir simulation for fine scale models is shown in figure 2.1.

2.1.3 Boundary Conditions

In this thesis, it is assumed that there is no flow at the boundary of the reservoir. Thus, the boundary conditions are only located at wells. In the pressure equation (2.2) and saturation equation (2.3), q represents wells that are rate controlled, the injector/producer or source/sink terms. The Peaceman equation [88] is used for the wells that are pressure controlled as,

$$q = -\lambda\,(s)\,WI\,(p - p_{bhp})\,, \tag{2.22}$$

where WI is the well index depending on the well-bore geometry and p_{bhp} is the bottom-hole pressure of a well.

In the next chapter, our goal is to present a reduced-order model for both pressure and saturation equations and reduce the computational runtime.

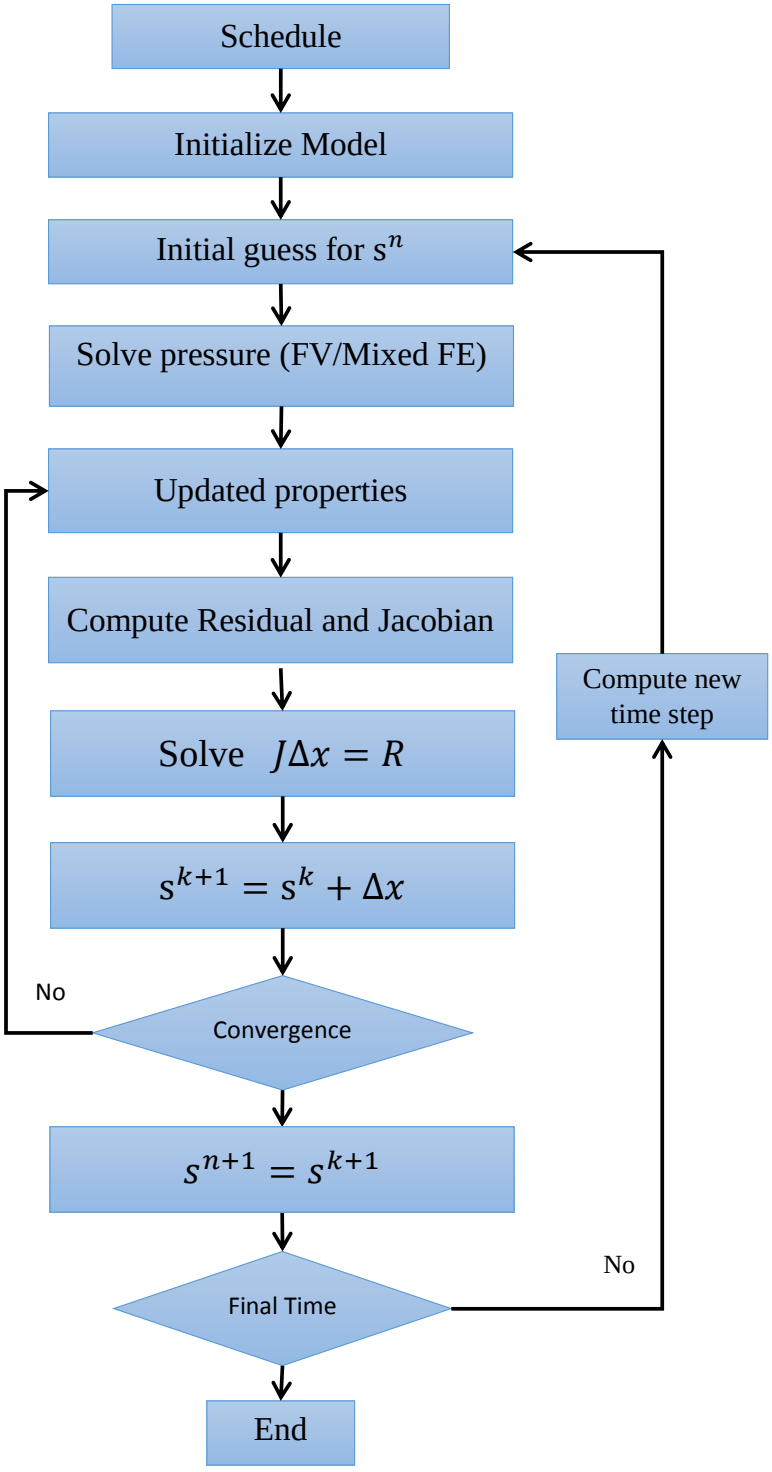


Figure 2.1: Flowchart of sequential solver in reservoir simulation.

3. REDUCED ORDER MODELING PROCEDURE*

Model order reduction has been shown to be a viable way of mitigating the large-scale nature of the simulation models and accelerate the computations and optimizations in subsurface applications [64]. It also facilitates the possibility of near real-time analysis [80, 53]. The goal is to represent a system with a minimum number of necessary parameters. Generally, one seeks linear and/or nonlinear models that approximate the full-order system with certain accuracy so that they preserve some of the properties of the original system and be computational more efficient to run. Note that the complexity of a model is generally represented by the number of states of the large scale model, which can be translated into the number of gridblocks after discretization of PDE equations.

There are various techniques for MOR depending on whether the model is linear or nonlinear. Owing to the fact that reservoir simulation consists of solving a set of nonlinear equations at every time step, I opted to focus on MOR that is applicable to this type of systems. One of the well-established MOR methods is proper orthogonal decomposition (POD). POD is arguably one of the most efficient methodologies used in model order reduction context [64]. This is because of its computational simplicity, ease of implementation and good approximation

3.1 Proper Orthogonal Decomposition

POD is a mathematical procedure that utilizes an orthogonal transformation to convert a given set of observations of conceivably correlated variables into a set of variables that are linearly independent and are referred to as basis. This method is

*Part of this section is reprinted with permission from "Fast Multiscale Reservoir Simulations Using POD-DEIM Model Reduction" by M. Ghasemi, et al., 2015, SPE Reservoir Simulation Symposium, Copyright[2015] SPE.

extensively discussed in the literature (see [25, 78]), and I briefly mention some basic concepts here.

POD is typically performed on the state solutions. The methodology is such that one runs high fidelity model with training inputs and save the output/states at each time step. The saved solution states are put next to each other as,

$$\mathbb{S}_x = [x_1,\, x_2,\, \cdots,\, x_{ns}] \in \mathbb{R}^{N \times ns} \tag{3.1}$$

where ns is the total number of snapshots, x_i can represent pressure, saturation, or other state solution at time instance i and the states are vectorized and stacked in a matrix $\mathbb{S}$ (referred as the matrix of snapshots), N is the total number of grid-blocks. The goal is to find a set of basis to re-parametrize the state of the system. In POD, one finds an orthonormal $\{\phi_i\}_{i=1}^{r}$ basis such that

$$\min_{\phi_i} \sum_{j}^{ns} \left\| x_j - \sum_{i}^{r} \left(x_j^T \phi_i\right) \phi_i \right\|. \tag{3.2}$$

It can be shown (see [111]) that the solution of this minimization problem is found by applying singular value decomposition (*svd*) to the snapshots matrix,

$$\mathbb{S}_x = V\, \Lambda\, W^T \tag{3.3}$$

where matrices V and W are left and right projection matrices and Λ is a matrix containing the singular values. The projection matrix $\phi \in \mathbb{R}^{N \times r}$ can be obtained by selecting the first r columns of the matrix V.

We denote the snapshots of the pressure, the velocity, and and saturation by $\mathbb{S}_p$, $\mathbb{S}_u$, $\mathbb{S}_s$, respectively. After applying a singular value decomposition on these

matrices, we select the projection matrix by indicating the fraction of total energy to be captured. The fractional energy is defined as,

$$E = \frac{\sum_{i=1}^{r} \sigma_i}{\sum_{i=1}^{ns} \sigma_i} \tag{3.4}$$

where σ_i is the ith diagonal element of matrix Λ. The number of basis selected such that the captured energy is $0.9 < E < 1$ (see discussions in [111] for selecting modes). After obtaining the projection matrix, the pressure, velocity and saturation are projected into reduced subspace as,

$$p = \Phi_p \, p_r, \quad u = \Phi_u \, u_r, \quad s = \Phi_s \, s_r \tag{3.5}$$

Computational savings achieved by POD-Galerkin depend on several factors and in particular nonlinear function evaluations can pose a challenge and significantly affect the speed-up that might be gained by using POD [93]. There are different approaches to reduce the complexity of nonlinear terms. One can sub-sample nonlinear terms in specific components and reconstruct the rest of the terms by gappy POD [11]. The Gauss-Newon method with approximated tensors (GNAT) is another technique to represent the nonlinear term with sparse sampling [23]. We apply DEIM, which constructs a separate subspace for nonlinear terms, selects interpolation points via a greedy strategy, and then combines the interpolation and projection to approximate the nonlinear terms in the subspace.

3.2 Discrete Empirical Interpolation Method

Discrete empirical interpolation method (DEIM) is a modification of POD that reduced the computational complexity of evaluating nonlinear terms. This method is a simplified description of empirical interpolation method (EIM) [16]. DEIM ap-

proximates a nonlinear function by combining projection and interpolation. This method is similar to missing point estimation (MPE) [11] in the sense that both methods employ few selected gridblocks to avoid the expensive fine scale nonlinear function evaluation at each time step. However, the gridblocks are selected differently in these two methods. DEIM approximates each nonlinear function so that the complexity of evaluating nonlinear terms becomes proportional to the small number of selected gridblocks, whereas MPE focusing on reducing the number of equations. Therefore, MPE considers POD basis only for state variables, but DEIM procedure considered POD basis for both state variables and the nonlinear terms. DEIM is reviewed briefly in this section, and one can refer to [26] for further explanation.

Let $f(\tau) \in \mathbb{R}^n$ denotes a nonlinear function where τ refers to time t or any control parameter μ. We approximate the function f by projecting it into a subspace spanned by the basis functions $\mathrm{U} = (u_1, \cdots, u_m) \in \mathbb{R}^{n \times m}$ as

$$f(\tau) \approx \mathrm{U}\ c(\tau) \tag{3.6}$$

The basis functions are calculated similar to POD. To compute the coefficient vector c, we need to define a selection matrix as,

$$\mathrm{P} = [\mathbf{e}_{\wp_1}, \cdots, \mathbf{e}_{\wp_m}] \in \mathbb{R}^{n \times m}$$

where $\mathbf{e}_{\wp_i} = [0, \cdots, 0, 1, 0, \cdots, 0]^T \in \mathbb{R}^n$ is the $\wp_i^{\text{th}}$ column of the identity matrix $\mathrm{I}_n \in \mathbb{R}^{n \times n}$ for $i = 1, \cdots, m$. Multiplying (3.6) by P^T and assuming that the matrix $\mathrm{P}^T\mathrm{U}$ is nonsingular, we obtain

$$f(\tau) \approx \mathrm{U}\ c(\tau) = \mathrm{U}\ (\mathrm{P}^T\mathrm{U})^{-1}\mathrm{P}^T f(\tau). \tag{3.7}$$

DEIM approximate the nonlinear function f in (3.7) with a single subspace of dimension m spanned by columns of projection matrix U and will reduce the computational runtime.

To summarize, the approximation of the nonlinear function $f(\tau)$, as given by (3.7), requires the followings:

- Computation of the projection basis U
- Identification of the indices $\{\wp_1, \cdots, \wp_m\}$

To determine the projection basis U, we assemble the function evaluations in a matrix $\mathbb{S}_f \in \mathbb{R}^{n \times n_s}$ and employ the singular value decomposition (svd) to compute the proper orthogonal decomposition (POD) modes. These modes are used as the projection basis in the approximation given by (3.6). As for the interpolation indices $\{\wp_1, \cdots, \wp_m\}$, they are selected using greedy algorithm as given in Algorithm 1.

Algorithm 1 Discrete empirical interpolation method (DEIM) procedure [26]

procedure DEIM($\mathrm{U} = (u_1, \cdots, u_m) \in \mathbb{R}^{n \times m}$)
  $[| \rho |, \wp_1] = \max\{| u_1 |\}$
  $\mathrm{U} = (u_1)$, $\mathrm{P} = (\mathbf{e}_{\rho_1})$
  for $\ell = 2, \cdots, m$ **do**
    $(\mathrm{P}^T\mathrm{U})c = \mathrm{P}^T u_\ell$
    $r = u_\ell - \mathrm{U}c$
    $[| \rho |, \wp_\ell] = \max\{| r |\}$
    $\mathrm{U} \leftarrow (\mathrm{U}\ u_\ell)$, $\mathrm{P} \leftarrow (\mathrm{P}\ \mathbf{e}_{\rho_\ell})$
  end for
  return $\overrightarrow{\wp} = (\wp_1, \cdots, \wp_m)^T$
end procedure

A set of indices are constructed inductively on the nonlinear function basis in DEIM procedure as outline in Algorithm 1. In this algorithm, the first interpolation

Type of System	Total Complexity
Fine Scale (sparse)	$\mathcal{O}(n^2)$
POD	$\mathcal{O}(r^3 + nr^2)$
POD-DEIM	$\mathcal{O}(r^3 + mr^2)$

Table 3.1: Comparing computational complexity of different models, n: number of gridblocks, r: number of POD basis, and m: number of DEIM basis.

point (gridblock) is selected form the entries of the first input basis, u_1 with largest magnitude. The remaining gridblocks are selected such that each of them corresponds to the index of the largest value in the residual $r = u_\ell - \mathrm{U}c$. Note that linear independence of input basis gurantees that in each iteration the residual is nonzero vector, and thus $\mathrm{P}^T\mathrm{U}$ is nonsingular and DEIM is well defined.

The POD-DEIM procedure can be summarized as follow,

1. Run the fine scale model with training schedules (variation of inputs in their possible range) and save the pressure, saturation and the nonlinear terms at each time step

2. Select the appropriate number of basis based on *svd* to capture the necessary energy of each variable

3. Apply *deim* function as explained in Algorithm 1 to basis matrix of nonlinear functions to select grid points based on the greedy algorithm

4. Approximate all the nonlinear functions in a formulation with DEIM

The complexity of the fine scale model and reduced model based on POD and POD-DEIM are compared in the Table 3.1.

In reservoir simulation the rock and fluid properties are nonlinear functions of pressure or saturation. Also, depending on the problem formulation, one may deal

with mobility or fractional flow functions as nonlinear terms of pressure and saturation. Evaluating these functions (and also their derivatives) at all gridblocks is computationally expensive. Specifically in the case of applying POD, it requires projecting back to fine scale model at each iteration. In this case, one can employ DEIM procedure to reduced the computational cost corresponding to these nonlinear terms.

Since this concept will be important later on, we will study a simple example which underscores the main idea. In the following example, we apply DEIM to evaluate a nonlinear function at very few gridblocks and find this property at the rest of the grid points by interpolation.

3.2.1 Numerical Example: Application of DEIM on Mobility Function

In this example, the mobility function is considered as a nonlinear function to demonstrate the application of DEIM for reservoir simulation. It is assumed that the fluid is slightly compressible, and the mobility function is defined as,

$$\lambda_l = \frac{k_{rl}\left(S_w\right)}{\mu_l\left(P\right) B_l\left(P\right)} \tag{3.8}$$

A 5-spot two-phase flow (oil-water) reservoir model in two dimensions is considered. It is assumed that the reservoir is under water flooding recovery process. The fluid properties are shown in Table 3.2 and the relative permeability are depicted in Figure 3.1b. The reservoir is 450x450x10 meters with a constant porosity of 0.2 and heterogeneous permeability as shown in Figure 3.1a. This reservoir is discretized by the finite volume method using Cartesian grid of size 45x45x1. The producers are controlled by constant bottom hole pressure of 2900 [psi] and the injection rate is constant 900 [bbl/d]. The initial water saturation and pressure are assumed to be 0.1 and 3000 psia respectively. The capillary pressure and gravity are neglected.

In order to apply POD-DEIM method, the reservoir model was simulated using

a fully implicit two phase flow simulator developed in MATLAB [82]. The model was run for 1100 days and the snapshots of pressures, water saturations, and the nonlinear residual terms were saved every 10 days. In all cases, we worked with separated snapshot matrices for the water and oil equations.

The singular values of pressure and saturation, shown in Figure 3.2, reveal that most of the energy of the system can be captured in few dominant modes and the fine scale model can be projected into a reduced space of low dimension. Also the first four dominant pressure and saturation basis are shown in Figure 3.3 to illustrate that they capture the pressure and saturation distribution.

Similar to POD, the basis and the corresponding singular values of the nonlinear function snapshots are found by *svd* . The singular values of oil and water mobility functions have the same trend as pressure and saturation singular values. The first four dominant basis are shown in Figure 3.4 illustrating that they capture the mobility distribution.

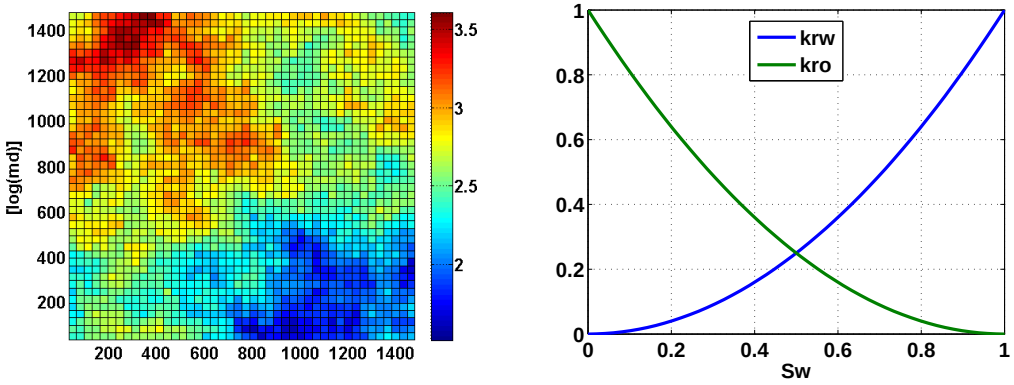
After selecting first 20 basis, the selection matrix was found based on DEIM algorithm. The selected gridblocks are shown in Figure 3.5a. The results of DEIM and exact nonlinear function evaluation are also compared in Figure 3.5b for one time instance revealing that the error in interpolation is in the order of $\mathcal{O}(10^{-5})$. Other properties that are nonlinear functions of states can be interpolated in the same manner.

3.3 Global Model Order Reduction for Pressure Equation

After running the training schedule and saving $\mathbb{S}_p$, $\mathbb{S}_u$ and $\mathbb{S}_s$; using *svd* yields POD basis matrices $\Phi_u \in \mathbb{R}^{N_u \times r_u}$ and $\Phi_p \in \mathbb{R}^{N \times r_p}$ and $\Phi_s \in \mathbb{R}^{N \times r_s}$ with $r_u \ll N_u$ and $r_p \ll N$ and $r_s \ll N$. Through Galerkin projection onto the subspace spanned by POD basis and substituting the new defined variables in (3.5) into (2.14), one

Symbol	Value	Unit
ϕ	0.2	[-]
B_o	1.012-0.955	[rbl/stb]
B_w	1	[rbl/stb]
c_o	1e-5	[1/psi]
c_w	3e-6	[1/psi]
μ_o	1	[cp]
μ_w	1.17-0.89	[cp]

Table 3.2: Rock and fluid properties



(a) Permeability distribution (b) Relative permeability

Figure 3.1: Inverted 5 spot water flooding pattern

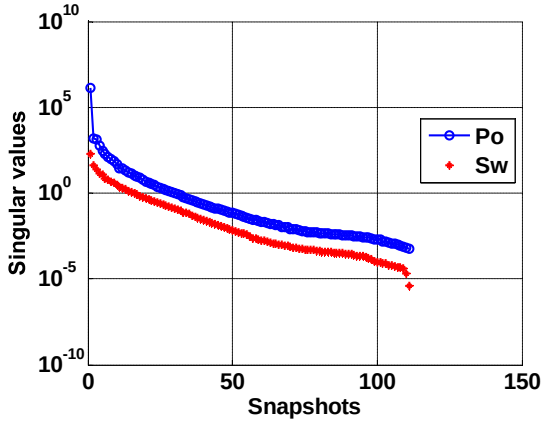
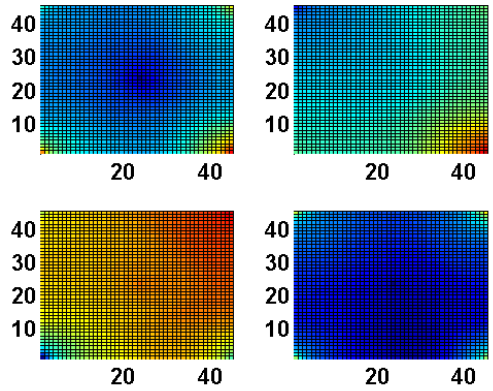
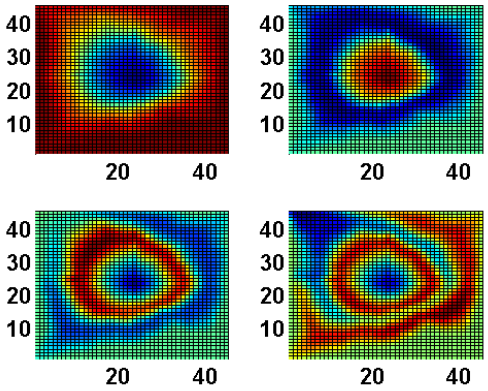


Figure 3.2: Singular values of pressure and saturation snapshots

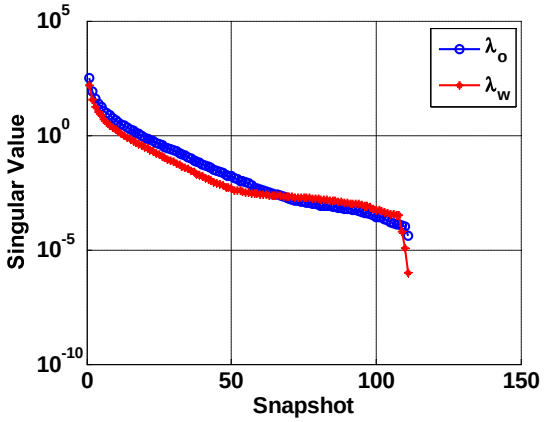


(a) pressure basis

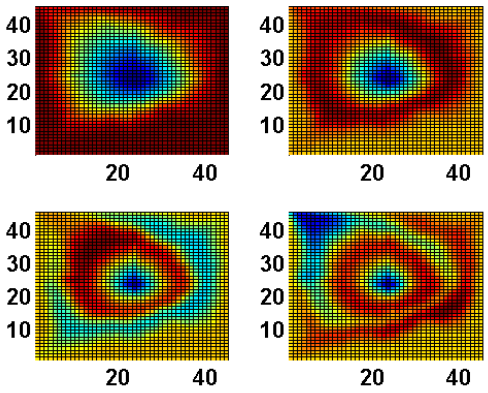


(b) saturation basis

Figure 3.3: First four dominant POD basis

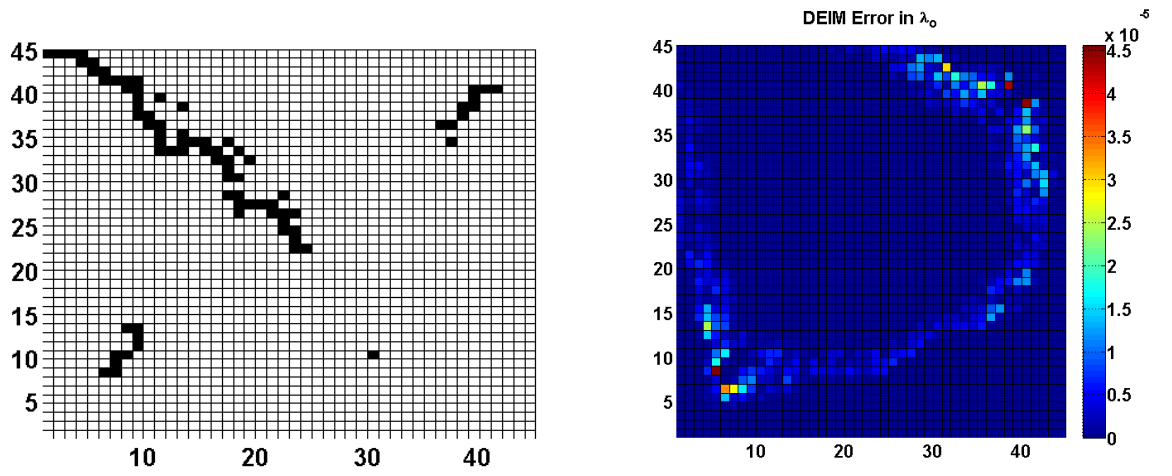


(a) Singular values of the snapshots of mobility



(b) Four dominent DEIM basis of $\lambda_o(S_w, P_o)$

Figure 3.4: First four dominant DEIM basis



(a) Selected gridblocks (b) Error between the exact and DEIM mobility

Figure 3.5: Application of DEIM on oil mobility

gets the reduced pressure equation as,

$$\begin{bmatrix} \Phi_u^T B \Phi_u & -\Phi_u^T C^T \Phi_p \\ \Phi_p^T C \Phi_u & 0 \end{bmatrix} \begin{bmatrix} u_r \\ p_r \end{bmatrix} = \begin{bmatrix} 0 \\ \Phi_p^T g \end{bmatrix}.$$

Note that the matrix B is of the form (2.15), which is a nonlinear function of saturation. Our goal is to solve the problem in a reduced space through the entire simulation time, and avoid projecting back to fine scale solution. Thus, one need to use an approximation for the mass matrix. Here, we apply DEIM method to approximate the nonlinear function as it will be explained. The nonlinear function $f_\lambda(s) = (\lambda(s)K)^{-1}$ is projected onto a smaller subspace, i.e.

$$f_\lambda(s) \approx \sum_{l=1}^{m} c_l \, U_l = \mathrm{U}_\lambda \, (\mathrm{P}_\lambda^T \mathrm{U}_\lambda)^{-1} \mathrm{P}_\lambda^T f_\lambda(\Phi_s s_r), \tag{3.9}$$

where m is the number of DEIM basis for approximating this nonlinear function, and s_r is the saturation the reduced space (of size $r_s \ll N_s$). Substituting (3.9) in

the definition of matrix B in (2.15), we get

$$\int_\Omega f_\lambda(s)\psi_i\psi_j dx = \sum_{l=1}^{m} c_l \int_\Omega U_l\psi_i\psi_j dx, \tag{3.10}$$

Most of the matrices in Eqs (3.9) and (3.10), e.g. $\mathrm{U}_\lambda(\mathrm{P}_\lambda^T\mathrm{U}_\lambda)^{-1}$, $\mathrm{P}_\lambda^T\Phi_s$, and

$$M_l = \int_\Omega U_l\psi_i\psi_j dx, \quad l = 1, \ldots, m \tag{3.11}$$

can be precomputed in the offline stage. In the online stage, we only need to find the coefficients and do the summation

$$M = \sum_{l=1}^{m} c_l M_l. \tag{3.12}$$

3.4 Global Model Order Reduction for Saturation Equation

To employ POD-DEIM, one needs to substitute the reduced saturation from equation (3.5) into equation (2.3). By applying POD to the saturation equation, the residual becomes an overdetermined system of reduced saturation state. This system of equations can be changed into determined by projecting it into a reduced space as,

$$\mathcal{R}_r(s_r) \triangleq \Phi_s^T W \mathcal{J}^{-1}\mathcal{R}(\Phi_s s_r, s^n, u^{n+1}) \tag{3.13}$$

Certain properties of this system can be preserved by selecting a proper W. Two common ones are Gakerkin projection and Petrov-Galerkin projection, where $W = \mathcal{J}$ and $W = \mathcal{J}^T\mathcal{J}$ respectively. In this thesis, we only consider Galerkin projection as it has shown to yield a good performance. Thus, the reduced residual is defined as,

$$\mathcal{R}_r(s_r) \triangleq {\Phi_s}^T\mathcal{R}(\Phi_s s_r, s^n, u^{n+1}) \tag{3.14}$$

This equation can be expanded as,

$$\mathcal{R}_r(s_r^{n+1}) \triangleq s_r^{n+1} - s_r^n - (\Phi_s)^T \left(A\ f_w(\Phi_s\ s_r^{n+1}) + \frac{\Delta t}{|\Omega|} q^+ \right) \tag{3.15}$$

Next, one can apply DEIM as it was explained in the previous section to the fractional flow function $f_w(s)$ for projecting it to a lower dimensional subspace. Thus, instead of evaluating this nonlinear term at fine scale, it will be calculated at a reduced subspace and interpolated for the rest of the grid points, i.e.

$$f_w(s) \approx \mathrm{U}_w\ (\mathrm{P}_w^T \mathrm{U}_w)^{-1} \mathrm{P}_w^T f_w(\Phi_s s_r) \tag{3.16}$$

Substituting Eq (3.16) into (3.15) results in,

$$\Phi_s^T A f_w(s) = \Phi_s^T A \mathrm{U}_w\ \left(\mathrm{P}_w^T \mathrm{U}_w\right)^{-1} f_w\left(\mathrm{P}_w^T \Phi_s s_r\right) \tag{3.17}$$

Here $\Phi_s^T A \mathrm{U}_w\ \left(\mathrm{P}_w^T \mathrm{U}_w\right)^{-1} \in \mathbb{R}^{r_s \times m_f}$, and $\mathrm{P}_w^T \Phi_s \in \mathbb{R}^{m_f \times r_s}$ can be precomputed in offline stage; r_s is the number of POD basis for saturation, m_f is the number of DEIM points for flux.

One can also calculate the corresponding Jacobian of the reduced residual defined in (3.14) as,

$$\mathcal{J}_r \triangleq \frac{\partial \mathcal{R}_r}{\partial s_r} = {\Phi_s}^T \frac{\partial \mathcal{R}}{\partial s_r} = \Phi^T \frac{\partial \mathcal{R}}{\partial s} \frac{\partial s}{\partial s_r} = {\Phi_s}^T \mathcal{J} \Phi_s \tag{3.18}$$

Substituting the definition of fine scale Jacobian from equation (2.20) into (3.18) results in,

$$\mathcal{J}_r \triangleq \frac{\partial \mathcal{R}_r}{\partial s_r} = \mathrm{I}_{r_s} - \Phi_s^T A \mathrm{U}_w\ \left(\mathrm{P}_w^T \mathrm{U}_w\right)^{-1} f_w'\left(\mathrm{P}_w^T \Phi_s s_r\right) \tag{3.19}$$

Newton-Raphson iterations are used to solve for the reduced saturation at each

time instance as following,

$$s_r^{k+1} = s_r^k - \alpha \, \mathcal{J}_r(s^k)^{-1} \, \mathcal{R}_r(s^k), \tag{3.20}$$

Thus, at each time step the saturation will be solved in the reduced space without the need to project back to the fine scale as shown in the POD-DEIM flowchart for saturation equation in Figure 3.6.

Note that the matrix A is derived from upwind indices and depends on the velocity. Thus, one needs to address the updating of upwinding matrix during simulation. One approach is to project back to fine scale solution at each time step after the pressure equation is solved, and update the upwinding matrix based on new velocity solution. Other alternative method is to update the upwinding matrix based on reduced velocity vector, i.e. finding regions in the reduced subspace where the velocity direction has changed using the reduced velocity and only updates those regions. These issues are still open and will be left for future research.

In the following section, the POD-DEIM is verified in some numerical examples to reduced the complexity of models and reduce the computational time.

3.5 Numerical Examples

In this section, some examples related to flow through porous media within POD-DEIM framework will be discussed and the results are compared with high fidelity solution to verify the applicability of this method.

3.5.1 POD-DEIM Basis Analysis

In this example, the relative error between reduced order model and the high fidelity model is compared. A 5-spot two-phase flow (oil-water) reservoir model in two dimensions is considered, with one injector in the center of the reservoir and four

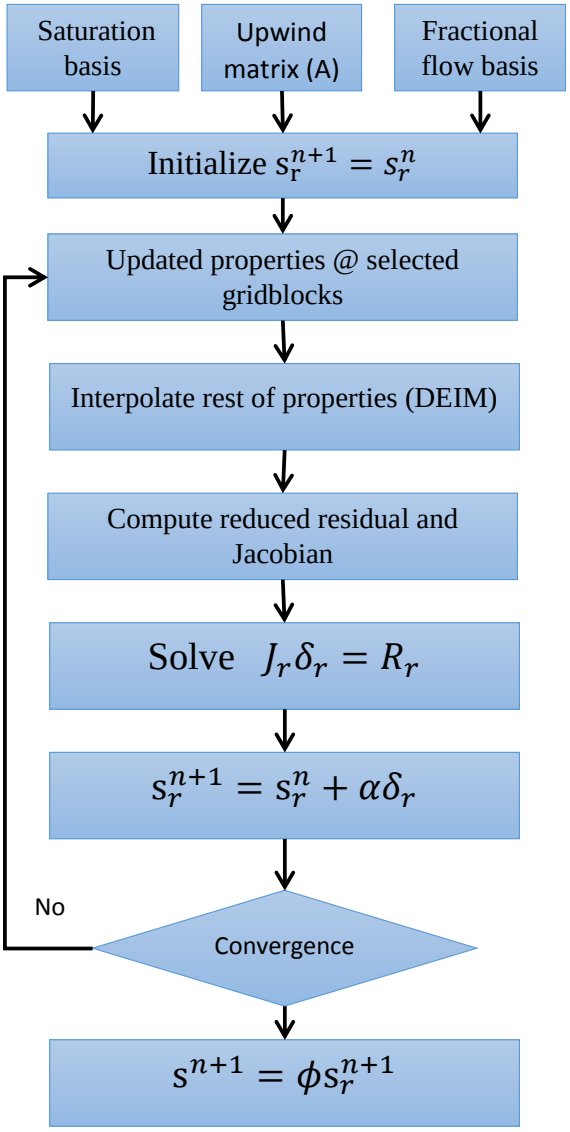


Figure 3.6: POD-DEIM flowchart on saturation equation.

producers in the corners. The permeability of this model is the layer 10th of SPE10 comparative model [31], which has large heterogeneity as shown in Figure 3.7.

This reservoir is discretized using Cartesian grid of size $20ft \times 10ft \times 2ft$. Overall the reservoir model has $60 \times 220 \times 1 = 13200$ active cells. The fluid viscosity ratio is $\mu_w/\mu_o = 0.1$. The relative permeability is a quadratic function of saturation similar to previous example. We assumed a constant porosity of 0.2 for entire model.

For the training schedule, the producers are controlled by bottom hole pressure and the injector by injection rate as shown in Figure 3.8. Note that this amount of injection was selected to assure one pore volume will be injected throughout simulation time (1000 days). The initial water saturation and pressure are assumed to be 0.0 and 2500 psia respectively.

The computational time and the total number of nonlinear iterations for the fine scale model is shown in Table 3.3. The number of POD and DEIM basis are varied as it is shown in the Table 3.4. Note that all the cases were run using the snapshots obtained from the Training schedule in Figure 3.8. This analysis indicates that the pressure and saturation error decreases as the number of basis for them increases.

Note that by increasing the number of basis from one to two basis for pressure equation the error in both pressure and saturation equation reduced significantly. Also, the number of total iteration in reduced model is significantly less than the high fidelity model because few number of unknowns is need to be solved for in the reduced subspace. The speed-up is mainly in pressure equation, however on can see that there is a reasonable reduction in computational time of saturation equation as well.

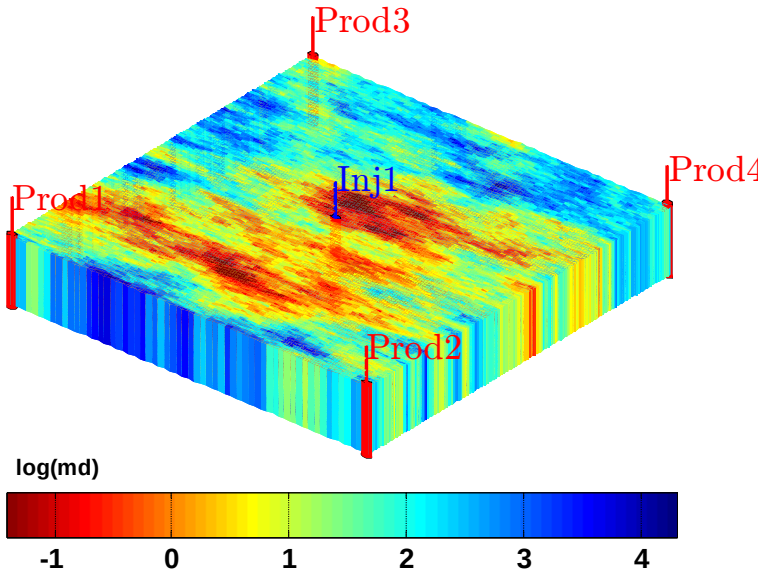
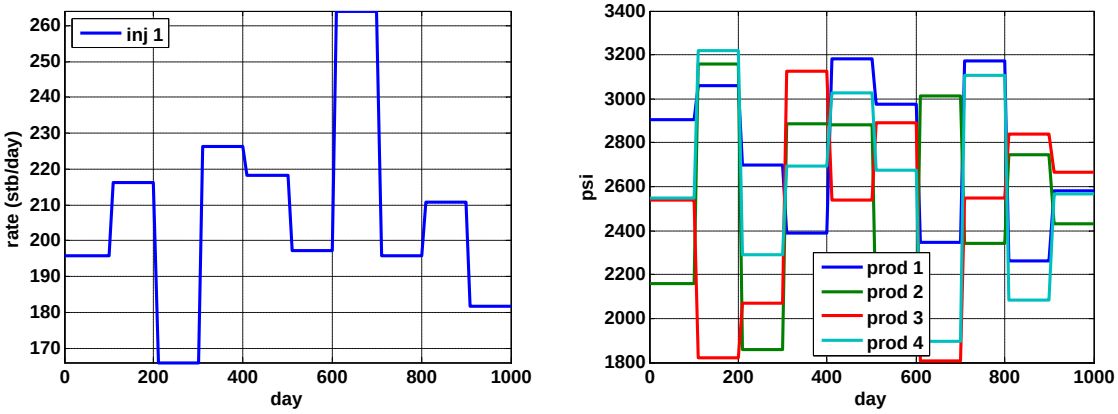


Figure 3.7: Permeability, layer 10th of SPE10



(a) injection rate

(b) bottom hole pressure

Figure 3.8: Training schedule of example 3.5.1

	Press Eq. Elapsed Time (s)	Sat. Eq. Elapsed Time (s)	Total Number of Iteration
Fine Scale Model	44	5.5	1254

Table 3.3: Computational time of the fine scale model in the example 3.5.1

POD-DEIM Model														
Energy Truncation				# Basis Functions										
Velocity	Pressure	Water Sat	Fractional Flow	Velocity	Pressure	Water Sat	Fractional Flow	Elapsed Time Pressure	Elapsed Time Saturation	Total Simulation Time	Overall Speed-Up	# Total Iter	Err Water Sat	Err Pressure
0.99	0.9788	0.9	0.94	12	1	5	10	Not Converged	--	--	--	--	--	--
		0.95	0.96			10	15	7	1.8	8.92	5.66	261	**2.1**	**0.64**
		0.97	0.98			15	20	7.1	2.2	9.11	5.5	265	**3.1**	**0.64**
	0.991	0.9	0.94		2	5	10	7	1.7	8.8	5.7	242	**0.033**	**0.015**
		0.95	0.96			10	15	7.2	1.9	9.1	5.6	288	**0.022**	**0.025**
		0.97	0.98			15	20	7	2.2	9.3	5.4	271	**0.018**	**0.02**
	0.995	0.9	0.94		3	5	10	7.2	1.7	8.9	5.7	242	**0.033**	**0.02**
		0.95	0.96			10	15	7	1.9	9.1	5.5	267	**0.019**	**0.015**
		0.97	0.98			15	20	7.2	2.1	9.4	5.4	271	**0.014**	**0.013**

Table 3.4: Relative error in pressure and saturation by changing the number of POD-DEIM basis

3.5.2 Case Study: 5 Layers of SPE10

In this section we apply the model reduction methodologies for a two-phase flow (oil-water) reservoir model under the water flooding recovery process with the structure of a 5-spot. An injector is in the center of the reservoir and four producers are in the corners. It is also assumed that all of them are perforated only at the bottom layer. The reservoir is SPE10 comparative model [31] (five layers of 10th-14th). This model is synthetic but can be representative of a real reservoir with large heterogeneity as shown in Figure 3.9.

This reservoir is discretized using Cartesian grid of size $20ft \times 10ft \times 2ft$. Overall the reservoir model has $60 \times 220 \times 5 = 66000$ active cells. The fluid viscosity ratio is $\mu_w/\mu_o = 0.1$. The relative permeability curves are quadratic as depicted in Figure 3.1b. We assumed a constant porosity of 0.2 for entire model.

For the training schedule, the producers are controlled by bottom hole pressure and the injector by injection rate as shown in Fig. 3.10. Note that this amount of injection was selected to assure one pore volume will be injected throughout simulation time (1000 days). The initial water saturation and pressure are assumed to be 0.0 and 2500 psia respectively.

In order to apply the POD-DEIM methods, the reservoir was simulated for 1000 days and the snapshots of pressure (Po), velocity (Vel), water saturation (Sw) and fractional flow of water (fw) were saved every 10 days. Thus, there are 100 snapshots for each variables. Each snapshot is reshaped to a column vector and is stacked in a snapshot matrix. After applying *svd* to each matrix, one can find the basis as explained in previous sections. The singular values of the snapshot matrices are shown in Figure 3.11a and 3.11b. These figures reveal that there is a faster decay in the singular values for the pressure and velocity compared to saturation and fractional

function. Thus, more basis for saturation and nonlinear terms are needed to capture its dynamic behavior and in turn to yield small error.

The selection criteria here was to capture at least 99% of the energy of snapshots. The number of basis is compared for reduced model to the original fine scale one in Table 3.5. It is obvious that several orders of magnitude in model order reduction is attained in this example. The pressure equation runtime reduced more than 90 times and the saturation equation around 18 times. Overall the reduced model can be run 77 times faster than the original fine scale one.

The relative error of saturation is calculated at the time t_i as,

$$s_{err}(t_i) = \|s_{red}(t_i) - s_{ref}(t_i)\|_2 / \|s_{ref}(t_i)\|_2, \quad t_i \in [0 \;\; T_f] \tag{3.21}$$

where $s_{red}(t_i)$ is the solution from model reduction, $s_{ref}(t_i)$ is the high fidelity solution, and T_f is the final time. This error is less than 5% for both POD and POD-DEIM approach in most of the simulation time as shown in Figure 3.13. This indicates that in spite of interpolating the nonlinear term in POD-DEIM, the error is still close to POD and overall the reduced model is a good approximation.

The final water saturation at the bottom layer and water cut for all the producers after 1000 days of simulation is shown in Figure 3.12 and compared with fine scale model. Figures 3.14b and 3.14a show the spatial relative error at the final time in the pressure and saturation, respectively. The error in pressure is $\mathcal{O}(10^{-3})$, and in saturation $\mathcal{O}(10^{-1})$. Note that the error is usually larger in the cells around injector due high dynamical fluid flow.

We run the reduced model with a new test schedule as shown in Figure 3.15, to make sure that the POD-DEIM model reduction is robust to input variation. This schedule is obtained by $\pm 20\%$ random perturbation of the training schedule. Note

	Fine Scale	POD-DEIM	Final Relative Error
# pressure basis	66000	2	–
# velocity basis	183400	12	–
# saturation basis	66000	20	–
# fractional flow basis	66000	25	–
pressure Eq. elapsed time (s)	8309	91.8	0.01
saturation Eq. elapsed time (s)	315	17.5	0.05
total simulation elapsed time (s)	8627	112	–
speed-up	–	77	–

Table 3.5: Compare fine and reduced scale model

that the basis of the reduced model are not updated and we used the same basis obtained from training snapshots.

The final water saturation at the bottom layer and water cut for all the producers after 1000 days of simulation with the test schedule are shown in Figure 3.16 and compared with fine scale model. Figures 3.18b and 3.18a show the spatial relative error at the final time in the pressure and saturation, respectively. The error in pressure is still $\mathcal{O}(10^{-3})$, and in saturation $\mathcal{O}(10^{-1})$. Note that even though the error is around 0.1 in some of the cells around injector due high dynamical fluid flow, the average saturation error is smaller than 5% for most of the simulation time as shown in Figure 3.17.

In the next section, the mixed FEM will be compared with finite volume regarding the discretization of pressure equation. In particular, the mass conservation and the number of basis required to capture the nonlinear dynamic will be discussed.

3.6 Mass Conservation in POD with Finite Volume Discretization

Because POD is a Galerkin projection, it will not necessarily honor mass conservation property. If we assume that the snapshots η_i are obtained with saturation

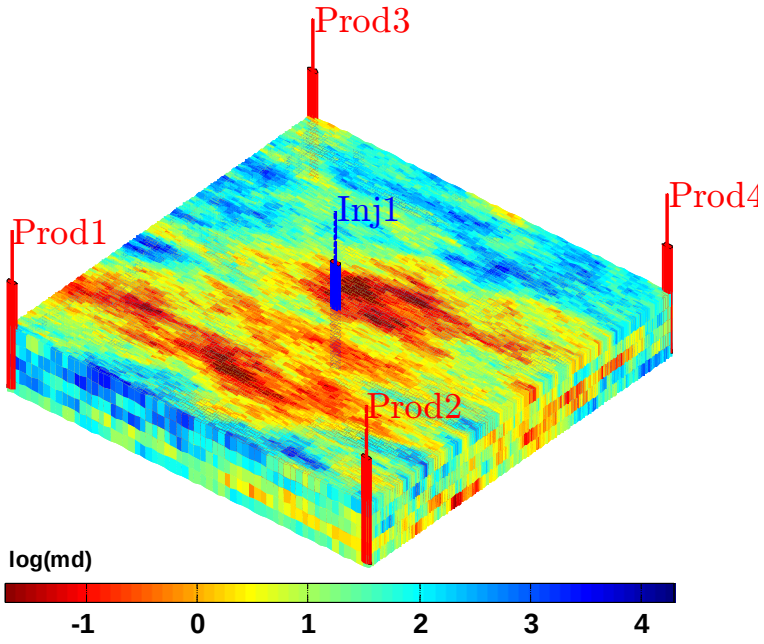
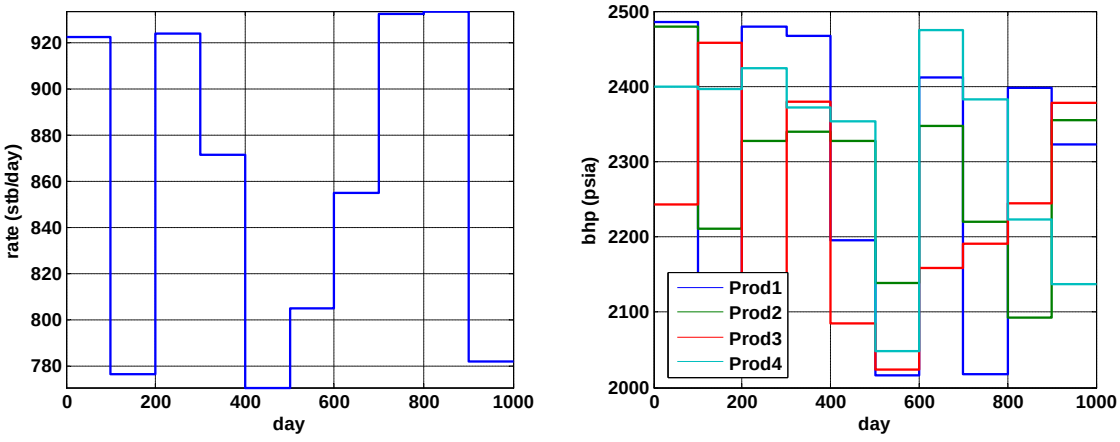


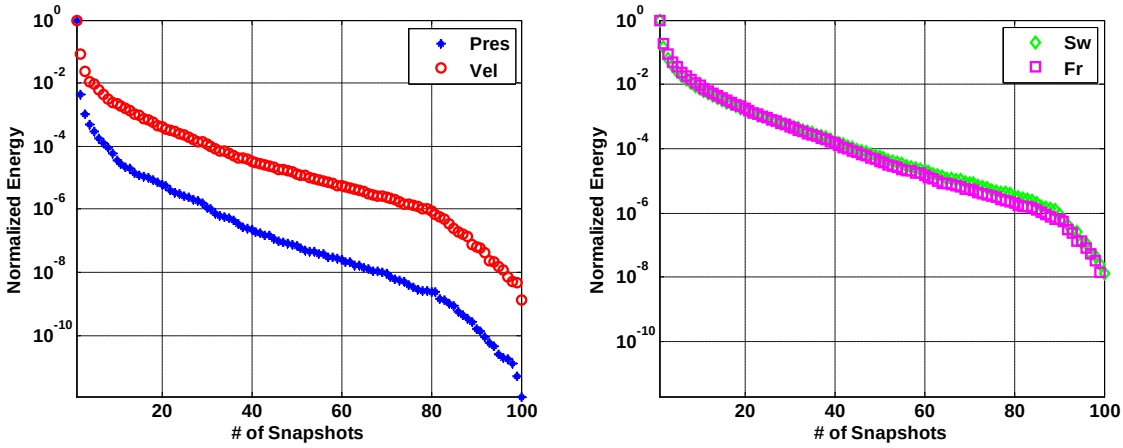
Figure 3.9: Permeability, SPE10 - 5 layers (10th-14th)



(a) injection rate

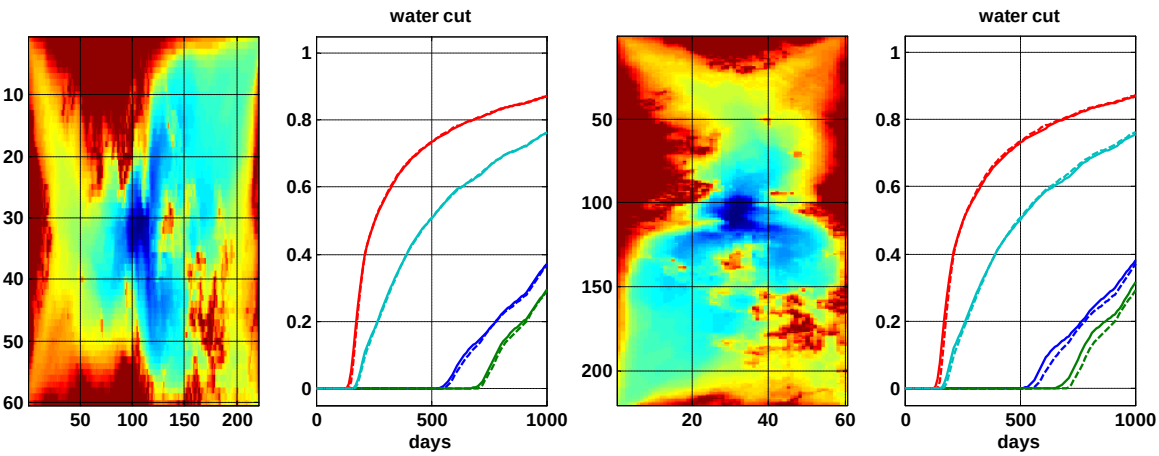
(b) bottom hole pressure

Figure 3.10: Training schedule of example 3.5.2



(a) pressure (Pres) and velocity (Vel) (b) saturation (Sw) and fractional function (Fr)

Figure 3.11: Singular values of snapshot matrix



(a) POD (b) POD-DEIM

Figure 3.12: Final water saturation and water cut for reduced (solid) and high fidelity (dashed) model for both POD and POD-DEIM approaches under Training Schedule

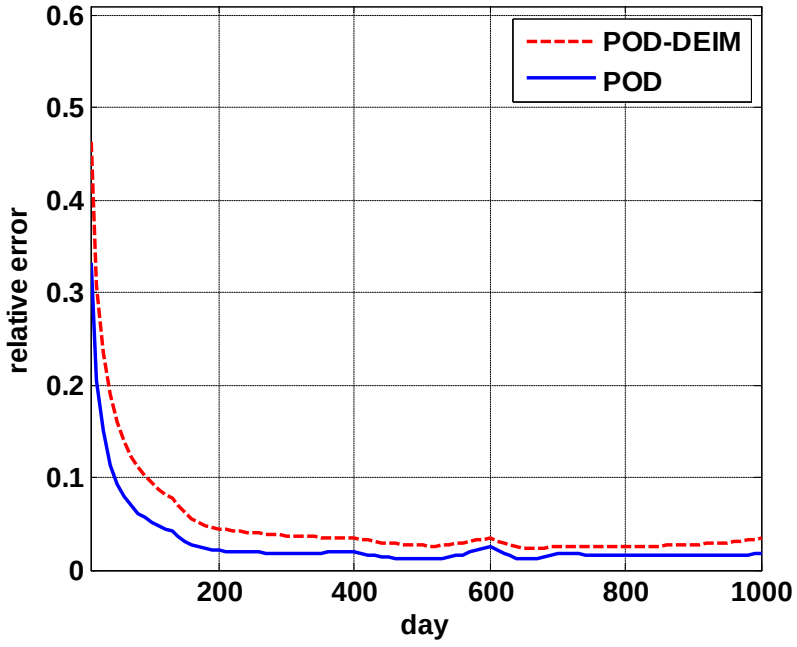
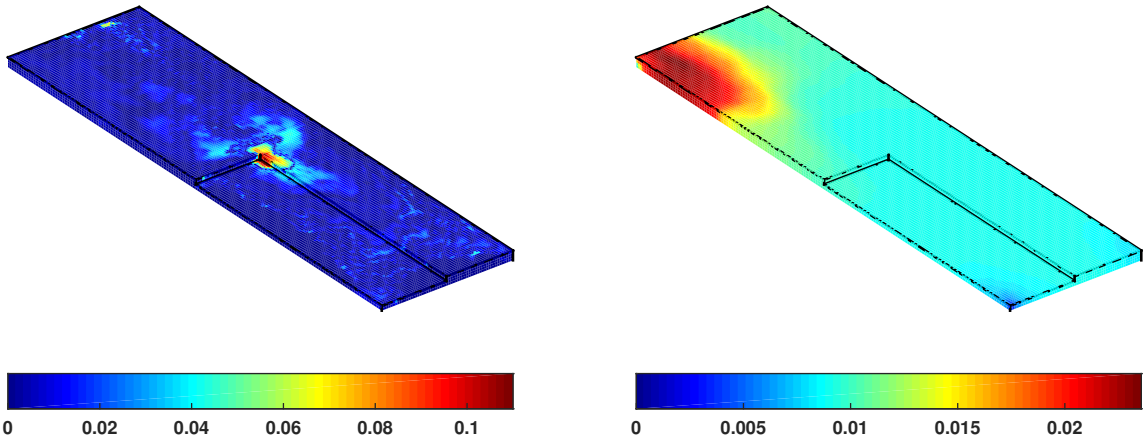


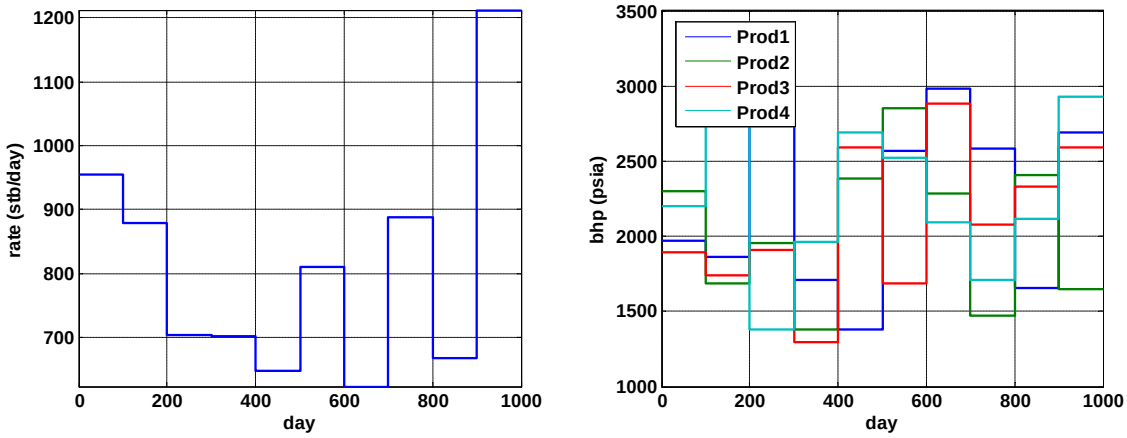
Figure 3.13: Temporal saturation error calculated based on Eq. 3.21 for the training Schedule



(a) saturation error

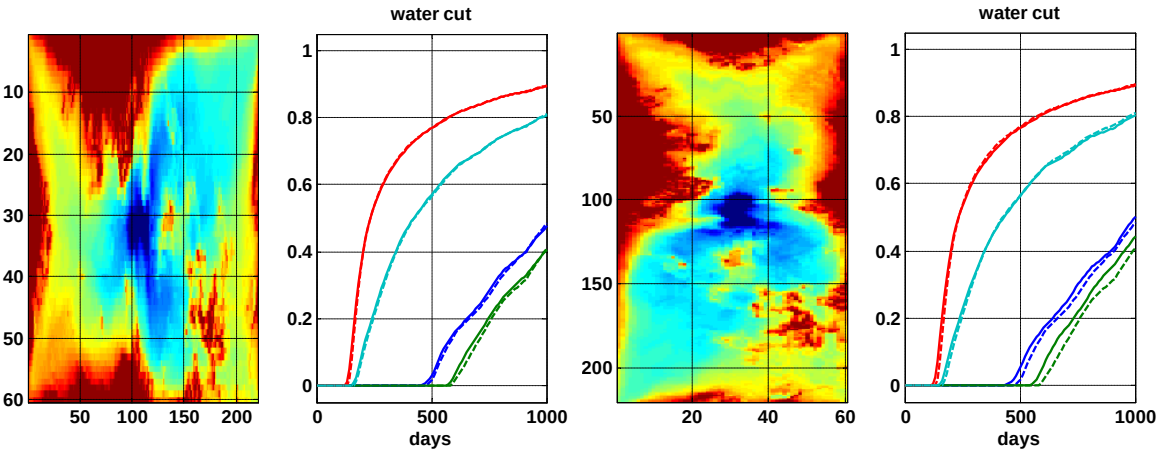
(b) pressure error

Figure 3.14: Final relative error for the training schedule



(a) injection rate (b) bottom hole pressure

Figure 3.15: Test schedule obtained by ±20% random variation in training



(a) POD (b) POD-DEIM

Figure 3.16: Final water saturation and water cut for reduced (solid) and high fidelity (dashed) model for both POD and POD-DEIM approaches for the test schedule

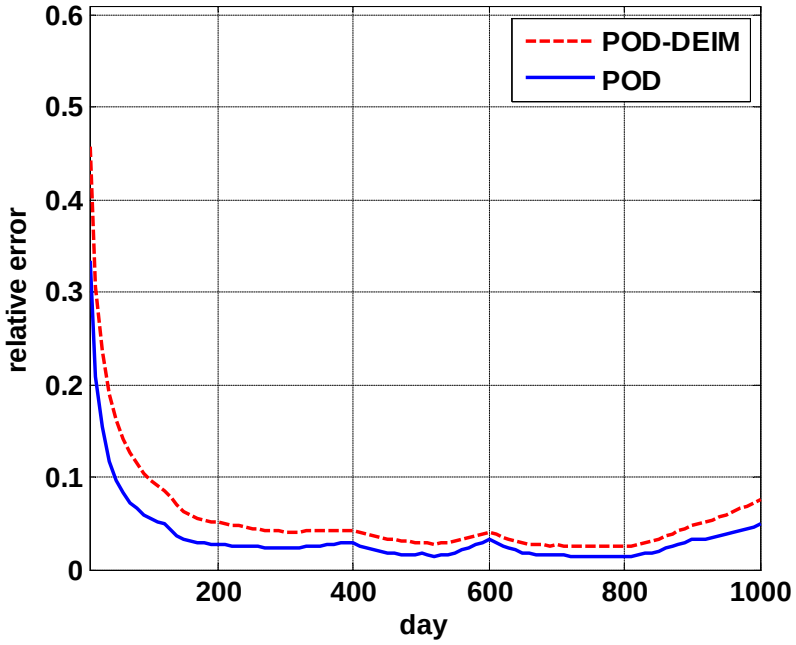
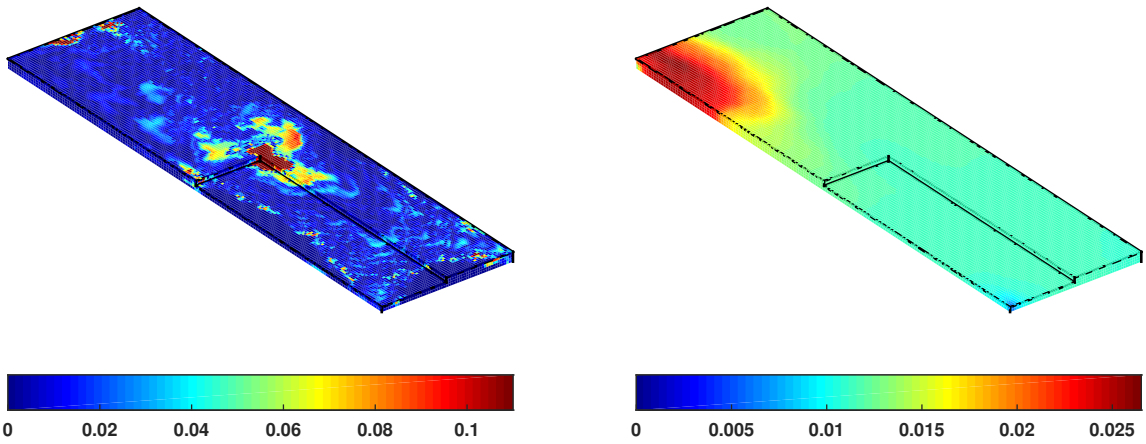


Figure 3.17: Temporal saturation error for the test schedule



(a) saturation error

(b) pressure error

Figure 3.18: Final relative error for the test schedule

fields s_i, then the mass conservation holds if one of the snapshots, η_l, coincides with the current solution; however even if the projection error is zero, i.e., the solution at the current time can be exactly represented by pressure snapshots, $p = \sum_i d_i \eta_i$, for some d_i, it does not guarantee that the current solution will be mass conservative.

One way to achieve a mass conservation is using an auxiliary variable, velocity field. By constructing POD basis functions for the velocity field, we can guarantee that the velocity field is conservative because it consists of a linear combination of velocity basis functions [48]. This allows us to achieve higher degree of reduction, as it is demonstrated in the following example.

3.6.1 Example

In this example, the reservoir model is a two-phase flow (oil-water) model under the water flooding recovery process with the structure of a 5-spot. The reservoir model is discretized using Cartesian grid of size $10ft \times 10ft \times 10ft$. Overall the reservoir model has $45 \times 45 \times 1 = 2025$ active cells. The permeability of the reservoir is 10 (md) homogeneous and the porosity is 0.2. The relative permeability curves is quadratic as shown in Figure 3.1b.

All the producers have constant bottom hole pressure at 2500 (psia). For training input, the injector bottom hole pressure is 3750 (psia). The reservoir model was simulated for 1000 days and the snapshots of the nonlinear fractional function was saved every 10 days. After applying *svd* on the snapshot matrices, the pressure and saturation basis are obtained. We selected 13 pressure basis to preserve 0.9999 energy and 13 basis for saturation to preserve 0.99 of its energy. These basis used to construct the projection matrices to project fine scale states to the reduced subspaces. When the reduced model was run with the same exact input and boundary conditions, the error was small and the results were close to fine scale solution. However, the

reduced models are used in frameworks with different inputs other than training one. Thus, we perturbed the bottom hole pressure of the injector as $\pm 5\%$, as shown in Figure 3.19. Although, this is a small perturbation and it is only in one of the input variables, the results of reduced model is far from high fidelity solution as it is shown in Figure 3.20 for a reduced model based on finite volume discretization. All the producers have the same water cut as shown in Figure 3.20a, due to the symmetry in the problem. Note that it the water cut is very different towards end of simulation, because the mass conservation is violated in most of the gridblocks, as shown in Figure 3.21.

We also applied POD on a mixed finite element formulation of this example. After running the same training and saving the snapshots, we selected 2, 8, and 13 basis for pressure, velocity, and saturation to preserve 99.5%, 99%, and 99% of the energy of the corresponding snapshots, respectively. The reduced model on mixed formulation not only result in small error for exact input, but also replicates a very similar results to high fidelity model for test input as shown in Figure 3.22. The saturation error is smaller than 2% for most of simulation time.

This example showed that MOR techniques, which only use pressure field to construct a reduced model, are sensitive to changes in boundary conditions. Thus, one need to have many basis to keep the error small or to reformulate the problem (as in the mixed finite volume) and solve for the velocity at the same time.

In the next section clustering methods are discussed. One of these techniques is used to classify the snapshots into different groups and improve the basis selection and the performance of the reduced model.

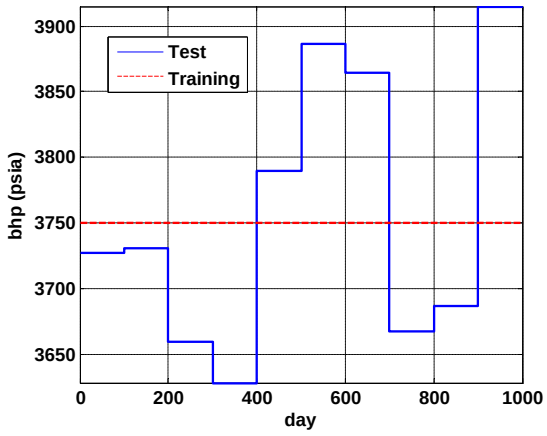
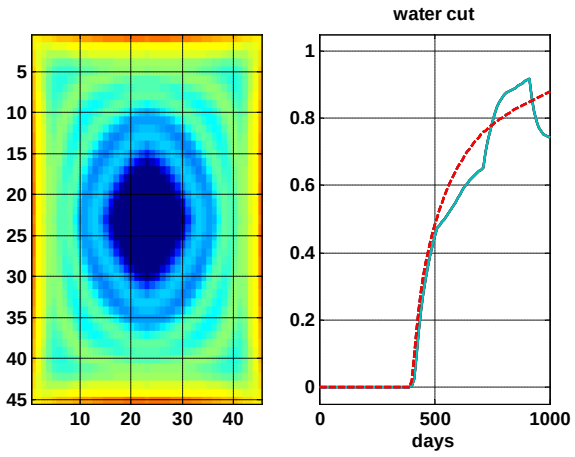
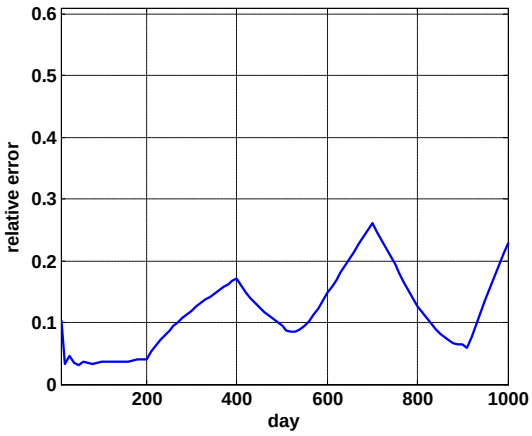


Figure 3.19: Training and test (perturbed) bottom hole pressure of the injector

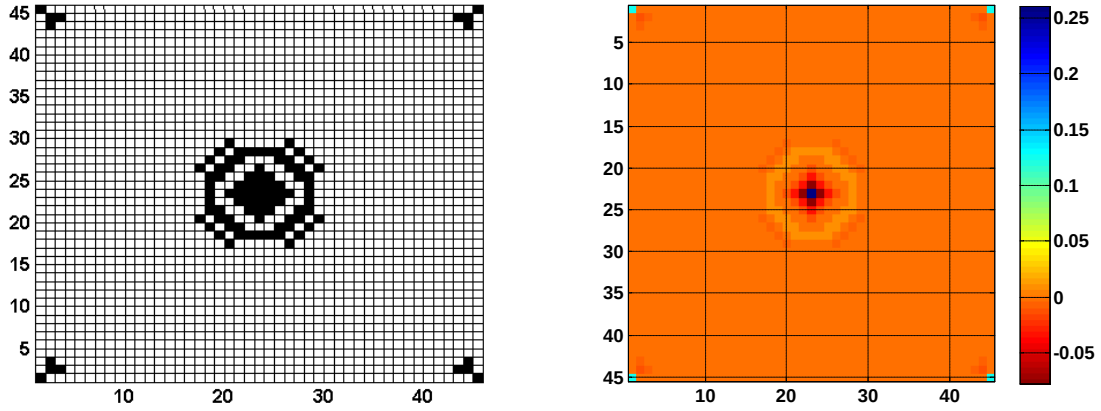


(a) final water saturation and water cut in reduced (——) and high fidelity (- - -) model



(b) water saturation error in reduced model

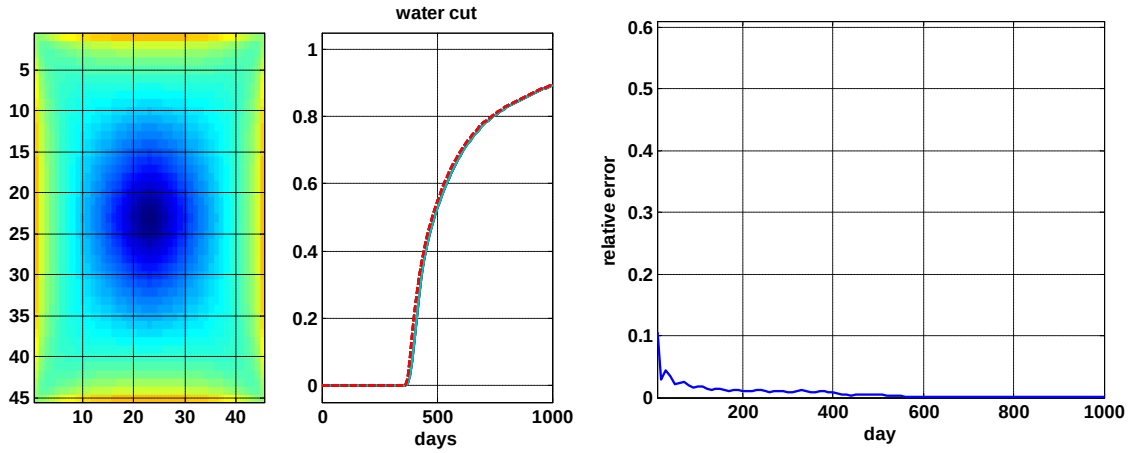
Figure 3.20: POD model reduction on finite volume formulation



(a) Gridblocks where mass conservation does not hold

(b) Mass conservation voilation scaled as PVI

Figure 3.21: Evaluation of mass conservation at final time



(a) final water saturation and water cut in reduced (——) and high fidelity (- - -) model

(b) water saturation error in reduced model

Figure 3.22: POD model reduction on mixed finite element formulation

3.7 Clustering

Clustering or classification is the task of grouping a set of objects such that the ones with a similar feature are in the same group, called cluster. Cluster analysis is one of the main techniques in statistical data analysis, used in many fields such as machine learning and pattern recognition among others.

There are various algorithms for cluster analysis of a data set. They differ in their notion of what constitutes a cluster and how to find it efficiently based on measures of clustering, which are distances among a cluster members, density in a data space, and statistical distributions.

Clustering can be formulated as a multi-objective optimization problem. The appropriate clustering algorithm and the parameter settings such as density threshold or the number of clusters depend on the individual data set. Thus, clustering is not usually an automatic task, but rather an iterative process.

There are mainly four types of clustering, namely hierarchical, centroid based, distribution based, and density based clustering. Here we briefly overview them.

Hierarchical clustering connects objects to form a cluster based on their distance. Thus, a cluster is described mainly by the maximum distance that is allowed to connect parts. As a cluster consists of multiple objects, one can choose the minimum, maximum, or the average of an object distance to the members of a group for clustering [70]. This approach is not very robust towards outliers and it is computationally very demanding and can be slow for large data sets.

Centroid based clustering represents a cluster by a central vector, which may not be necessarily a member of the data set. If the number of clusters is fixed, then this clustering can be formulated as an optimization problem, to find the required number of cluster centers (centroids) and assigning the objects to the nearest cluster.

This optimization problem is NP-hard and the common approach is to find the approximate suboptimal solution. One of the popular approximative method is *k-means* [61]. This classification method partitions the data space into a structure known as a Voronoi diagram, and it is close to nearest neighbor classification.

Distribution based clustering group the objects that most likely belong to the same distribution. Even though this method has an excellent theory, it suffers from overfitting problem. Thus, one usually can fix the number of distributions and classify the data set iteratively [115].

Density based clustering defines areas with higher density in the data set. The most popular method based on density clustering is DBSCAN. Similar to linkage based classification, this method connect points within certain distance [73]. However, these points also satisfy a density criterion. A key drawback of this approach is detecting cluster borders and it may fail to find the intrinsic cluster structures which are common in the real data.

In this thesis, *k-means* with squared Euclidean distance is used to cluster the snapshots. This algorithm is easy to implement and has a good performance [68]. In this method, given a set of data points in n-dimentional space and an integer k corresponding to the number of clusters, the problem is to determine k points, called centroids, so as to minimize the distance from each data point to its nearest center. There are iterative scheme to solve this problem efficiently [71].

In the next section a localized approach is introduced for constructing an efficient reduced order model for fluid flow in porous media simulation. This technique will not only reduced the number of basis required for capturing the dynamic of the system, but also will improve the performance of the reduced model.

3.8 Localized Model Reduction in Porous Media Flow

Up to this point POD-DEIM was applied to reduce the size of a reservoir model with nonlinear terms. Whereas regular POD-DEIM approximates the fine scale model with just one single reduced subspace, the localized POD (LPOD) and localized DEIM (LDEIM) that are introduced in this section compute several local subspaces.

Each subspace characterize a region of solutions and all together they not only approximates the high fidelity model better, but also reduces the computational cost of simulation. Both LPOD and LDEIM use classification approach to partition the solution space into different regions in the offline computational phase. After obtaining each subdomain, POD and DEIM is applied to construct the local reduced-order basis.

In the online phase, at each time step, the reduced states and nonlinear functions will be used in a classifier to find the most representative basis for POD and DEIM. This method is still computationally efficient for realtime applications because the selection is based on reduced order dimension [47]. The advantages of LPOD and LDEIM are shown in a numerical example of two phase flow at the end of this section.

3.8.1 Localized POD

Even though the regular POD reduces the dimension of a system significantly, many POD basis might be required to accurately approximate the system if it exhibits a wide range of dynamical solutions. Thus, The main idea is to divide the snapshots into different subgroups and apply POD to each domain to obtain local POD basis. This will require less number of basis for each region compared to global POD basis. Also, it will give us a better approximation of the solution trajectory in each subdomain, because more representative basis will be used.

Basically, LPOD is a variant of regular POD with multiple set of basis. In POD algorithm, *svd* is applied to the entire set of snapshots to attain the global set of basis for reduced model, whereas in LPOD the snapshots are divided into smaller sets based on a criteria or characteristics, and then the basis are found for each subsets. This approach can be used by applying Algorithm 2.

Note that this method is different than the method proposed in [21], where one apply clustering to divide all the snapshots into smaller subsets and then apply *svd* to the mean of each cluster. Although this method may help to reduce the number of required basis, the results may become poor due to averaging effect and using the centroids that are not necessarily any of the snapshots.

There are different techniques to split the solution snapshots space into different regions. In most of these methods, the domain is split into subdomains recursively [34], but this method might results in a large number of subdomains in practice. Here, we classify the snapshots into different clusters with *k-means* and for each cluster a local reduced-order model is constructed as suggested in [89].

In the online phase, a local reduced-order model is selected with respect to the current state of the system. This localization approach was introduced for POD method in [6]. However, they proposed unsupervised learning methods that can results in unstable clustering behavior if the clustering method and its parameters are not carefully chosen [113].

Furthermore, the given procedure in [6] requires precomputing auxiliary quantities to ensure an online selection procedure that is independent of the dimension of the original system. However, the number of these auxiliary quantities, which are computed in the offline phase, scales cubically with the number of clusters. Also, the procedure to select a local reduced-order model scales quadratically with the number of clusters. This can be an issue especially in optimization workflows, whereby

one updates basis of reduced model periodically to obtain more stable and accurate solution.

Thus, the modified Algorithm 2 for localized POD was introduced in this study, where only few number of indecies for classification is used in the offline phase and also these few indecies are employed in the online phase to find the corresponding cluster. This modification also reduces the storage required for the auxiliary parameters [47].

The inputs of Algorithm 2 are the snapshots matrix $\mathbb{S}$ and the number of clusters k and the number of POD basis in each cluster r. The subscript s refers to saturation. Note that in the first step of this algorithm *deim* function is used, which select grid points that are most representative in the solution space based on the greedy algorithm as explained in Algorithm 1. At Line 2 and 3 of this algorithm, a small sets of indices, instead of fine scale dimension, will be used for clustering in the *k-means* algorithm. This not only accelerates clustering, but also helps finding the proper set of basis in the online phase to become independent of fine scale solution. Also, the auxiliary variables are smaller and need less storage. It should be clear that this will not introduce any extra error in the LPOD, if enough indices are selected based on the greedy algorithm, because the selected grid points are only used for clustering and classification.

The output of the *k-means* algorithm is the centroid of each cluster and the corresponding index. These centroids and the indecies are used to define a classifier function as explained in Line 4. This classifier will be used in the online phase to distinguish the proper index of a cluster where the current state belongs to. After clustering all the snapshots one can find the POD basis for each group as explained in Lines 5-8. Note that in the Line 7 of this algorithm a small subset of this basis are selected based on greedy algorithm to be used in the classifier in the online phase.

The POD basis for each group, the classifier and the small subset of these basis are return from this algorithm.

Algorithm 2 Classification procedure of LPOD

1: **procedure** LPOD($\mathbb{S}_s, r, k$)
2: $\quad \mathrm{P}_g = deim(\mathbb{S}_s)$
3: $\quad (centroids, \{S_1, \cdots, S_k\}) \leftarrow kmeans(\mathbb{S}_s^T\ \mathrm{P}_g, k)$
4: $\quad c^s(x) = knnclassify(x, centroids, 1:k)$
5: $\quad$ **for** $i = 1:k$ **do**
6: $\quad\quad \phi_i \leftarrow svd(S_i, r)$
7: $\quad\quad w_i^s \leftarrow \mathrm{P}_g^T\ \phi_i$
8: $\quad$ **end for**
9: $\quad$ **return** $c^s(x), \phi_i, w_i^s$
10: **end procedure**

If LPOD is applied to the saturation equation, in the online phase at each Newton-Raphson iteration, the index of a cluster that current state is most likely belongs to can be found as follows,

$$i \leftarrow c^s(w_i^s\ s_r) \tag{3.22}$$

where $c^s(.)$ and w^s are the output of the Algorithm 2, and s_r is the reduced saturation state. After obtaining the cluster index, one can choose the corresponding POD basis and solve the problem in proper reduced subspace.

3.8.2 Localized DEIM

In this section we discuss the general description of LDEIM and an efficient method to select each subspace in the online phase without fine scale calculation. LDEIM constructs several local DEIM subspaces as, (U_1, P_1),...,(U_k, P_k) in the offline phase and then select the best one in the online phase. Two main questions are how to cluster the snapshots into subsets based on a feature and how to efficiently

select the proper set in the online phase. Here, we follow the approach suggested in [89] regarding these questions as explained in Algorithm 3.

Algorithm 3 LDEIM Procedure [89]

1: **procedure** LDEIM($\mathbb{S}_f, m, k$)
2: $\quad \mathrm{P}_g = deim(\mathbb{S}_f)$
3: $\quad (centroids, \{S_1, \cdots, S_k\}) \leftarrow kmeans(\mathbb{S}_f^T\ \mathrm{P}_g, k)$
4: $\quad c^f(x) = knnclassify(x, centroids, 1:k)$
5: $\quad$ **for** $i = 1:k$ **do**
6: $\quad\quad \psi_i \leftarrow svd(S_i, m)$
7: $\quad\quad \mathrm{P}_i \leftarrow deim(\psi_i, m)$
8: $\quad\quad w_i^f \leftarrow \mathrm{P}_g^T\ \psi_i\ (\mathrm{P}_i^T \psi_i)^{-1}$
9: $\quad$ **end for**
10: $\quad$ **return** $c^f(x), \psi_i, w_i^f$
11: **end procedure**

The inputs of Algorithm 3 are a snapshots matrix of nonlinear functions $\mathbb{S}_f$, the number of clusters k, and the number of DEIM basis in each cluster m. At the first step a subset of all snapshots, which consists a few gridblocks is selected to be used in *k-means* clusterings as shown in Line 2 and 3 of this algorithm. These steps accelerate the clustering and also make the online classification independent of fine scale calculations.

The output of the *k-means* algorithm is the centroid of each cluster and the corresponding index. These centroids and the indecies are used to define a classifier function as explained in Line 4. This classifier will be used in the online phase to find the proper index of a cluster where the current state belongs to. After clustering all the snapshots one can find the DEIM basis for each group as explained in Lines 5-9. Note that in the Line 8 of this algorithm a small subset of this basis are selected based on greedy algorithm to be used in the classifier in the online phase [89]. The

DEIM basis for each group, the classifier and the small subset of these basis are return from this algorithm.

If LDEIM is applied to the saturation equation, in the online phase at each Newton-Raphson iteration, the index of a cluster that current state is most likely belongs to can be found as follows,

$$i \leftarrow c^f(w_i^f \ \tilde{f}) \tag{3.23}$$

where $c^f(.)$ and w^f are the output of the Algorithm 3, and $\tilde{f}$ is the approximated nonlinear function based on DEIM approach. Here we select following point-based feature extraction,

$$\tilde{f} = f(\mathrm{P}_i^T \ s) \quad \in \mathbb{R}^m \tag{3.24}$$

After obtaining the cluster index, one can choose the corresponding DEIM basis and solve the problem in the reduced subspace.

3.8.3 Application of LPOD-LDEIM on Homogeneous Model

In this example we will apply LPOD and LDEIM on a simple model and compare the results with the regular POD and DEIM. The reservoir model is a two-phase flow (oil-water) model under the water flooding recovery process with the structure of a 5-spot as shown in Figure 3.23. The reservoir model is discretized using Cartesian grid of size $10ft \times 10ft \times 10ft$. Overall the reservoir model has $45 \times 45 \times 1 = 2025$ active cells. The permeability of the reservoir is 10 (md) homogeneous and the porosity is 0.2. The relative permeability curves is quadratic as shown in Figure 3.1b.

The reservoir model was simulated for 1000 days under constant injection rate of one pore volume and constant BHP at all the producers. The snapshots of pressure, velocity, water saturation, and fractional flow was saved every 10 days, resulting in

100 snapshots. Primarily, the POD basis for pressure and velocity were found as before, by applying *svd* on their snapshots and selecting the dominant basis. Next, LPOD was applied as instructed in Algorithm 2 on the 100 snapshots of saturation to cluster them into 3 subsets with corresponding basis. Note that in the online phase, the index corresponding to each cluster is found based on the equation (3.22).

The mean of all the snapshots of saturation and the first 4 dominant POD basis are shown in Figure 3.24. Also, the mean of three clusters resulted from *kmean* and the first 4 dominant POD basis for each cluster is demonstrated in Figure 3.25. comparison of first basis in all the clusters reveals that they are similar to the mean of each cluster, whereas the fourth basis are dynamic and have high frequency. Since the global POD average out all the snapshots, there is a chance that part of a local dynamic might be ignored in the global dominant basis that are selected. Thus, LPOD is expected to give better result as it is shown at the end of this example.

Next, one can apply DEIM on all of the fractional flow snapshots and obtain the global DEIM grid points as shown in Figure 3.26. LDEIM was also applied on the fractional flow snapshots as explained in Algorithm 3, and 3 clusters were attained. 10 grid points were selected per cluster based on DEIM. The mean of each cluster and the selected grid points are shown in Figure 3.27.

Comparing the global selected grid blocks and the local ones in each cluster reveals that the clustering allows LDEIM to concentrate the interpolation points in only a certain parts of the reservoir, i.e. close to the front of water saturation. This is one of the reasons that LDEIM can give us better results compared to regular DEIM, because the selected points are more into dynamic part of the model.

Figure 3.28 indicates the average error in saturation by increasing the number of basis in both DEIM and LDEIM. This figure reveals that for few number of basis LDEIM results in a smaller error compared to regular DEIM. However, for high

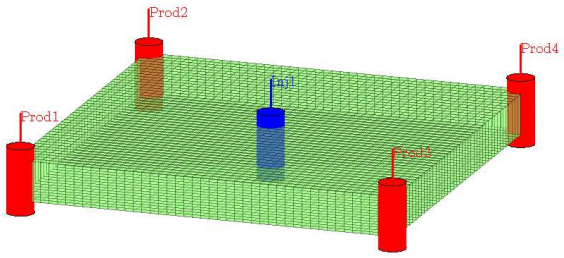


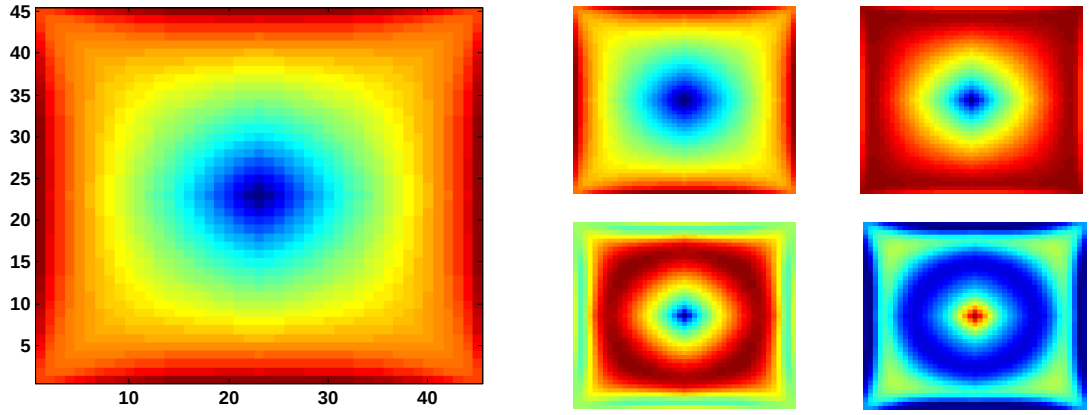
Figure 3.23: Homogeneous reservoir model with the 5-spot pattern

number of basis the error from LDEIM approach is slightly higher than DEIM.

For a fixed number of basis Figure 3.29 illustrate the effect of increasing the number of clusters on the average saturation error. This figure indicates that 4 cluster gives us the minimum error. Silhouette plot is another validation method for clustering of data. This technique provides a concise graphical representation of how well each snapshot lies within its cluster [96]. The silhouette plot of the snapshots corresponding to this example with four clusters is shown in Figure 3.30. The squared Euclidean distance were used as a metric to generate this plot. Silhouette value of 0.7-1.0 indicates that strong structure has been found as it is the case for this example.

Now to verify the localized model order reduction on this simple model, a training schedule is used for the injector (rate controlled) and all the producers (BHP controlled) as shown in **??** and Figures **??**, respectively. Note that this amount of injection was selected to assure one pore volume is injected throughout the simulation time. The initial water saturation and pressure are assumed to be 0.0 and 2500 psia, respectively.

The final water saturation and the water cut at all the producers are shown in Figures. 3.32a and 3.32b for POD and LPOD, respectively. As illustrated in these



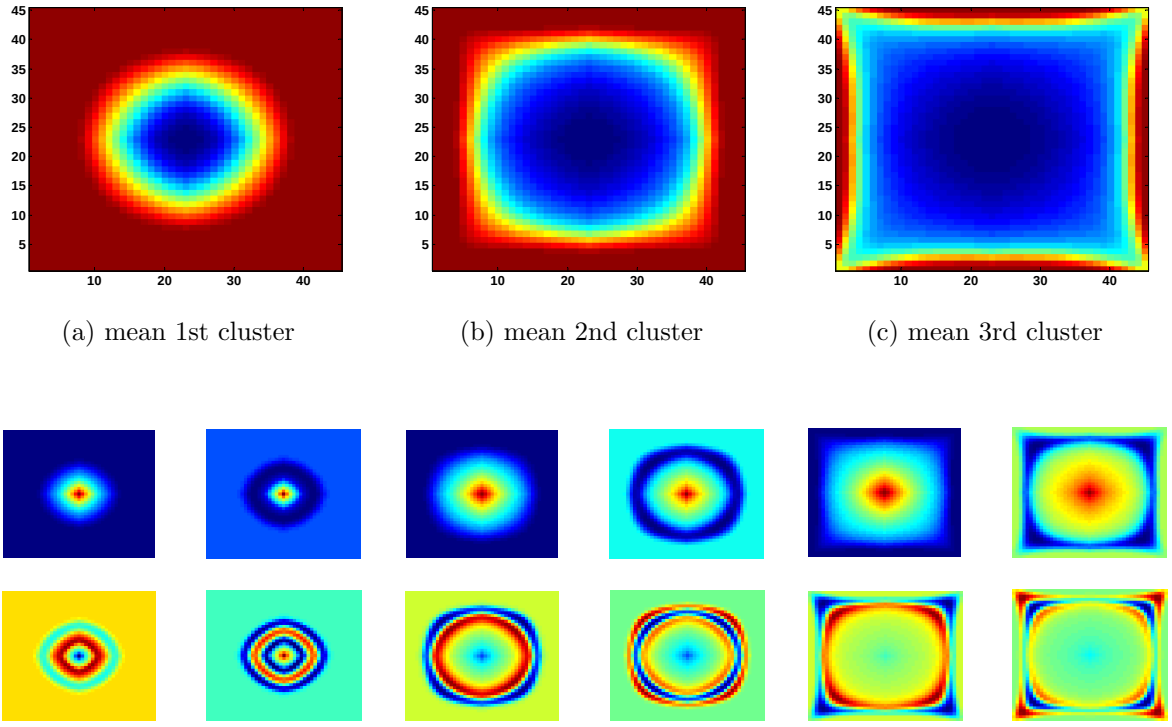
(a) mean of all saturation snapshots (b) four dominant POD basis of all snapshots

Figure 3.24: Application of global POD on all snapshots and the first four dominant basis

figures the LPOD outperforms the regular POD, yielding better approximation of the high fidelity model with almost exact water-cuts. Figures 3.34a and 3.34a compare the same results for DEIM and LDEIM, whereas LDEIM outperforms regular DEIM. The final water saturation error is also compared between POD and LPOD in Figure 3.32 and between DEIM and LDEIM in Figure 3.35. The relative error in saturation is calculated based on 3.21 and is compared for POD and LPOD in Figure 3.36a and for DEIM and LDEIM in Figure 3.36b. All of these results confirm the superiority of local model reduction over global one.

One needs to test the localized model reduction with a different input to verify its robustness we used the test schedule in Figure 3.37, which is 10 % variation with respect to the training schedule.

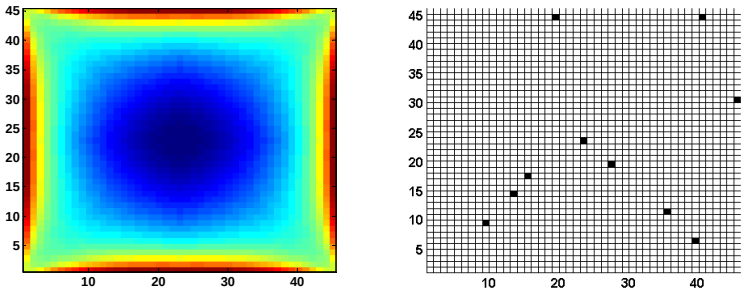
The final water saturation and the water cut at all the producers are shown in Figures. 3.38 for POD and LPOD, respectively. As illustrated in these figures the LPOD outperforms the regular POD, yielding better approximation of the high



(a) mean 1st cluster (b) mean 2nd cluster (c) mean 3rd cluster

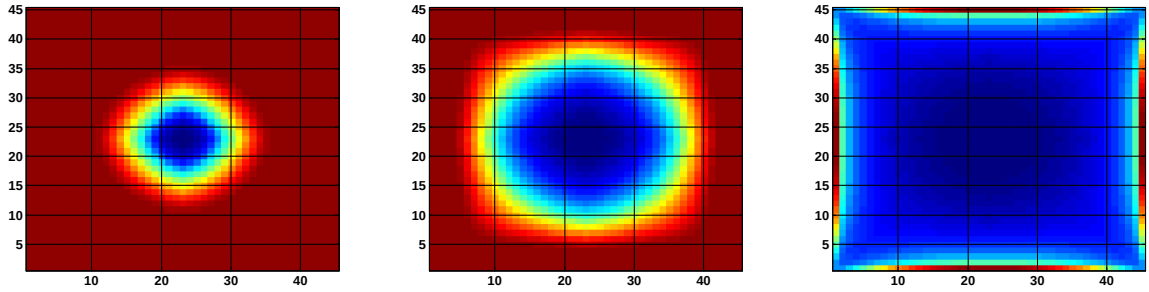
(d) 1st cluster POD basis (e) 2st cluster POD basis (f) 3st cluster POD basis

Figure 3.25: Illustration of the four clusters mean and the first four dominant basis of each cluster found based on localized POD

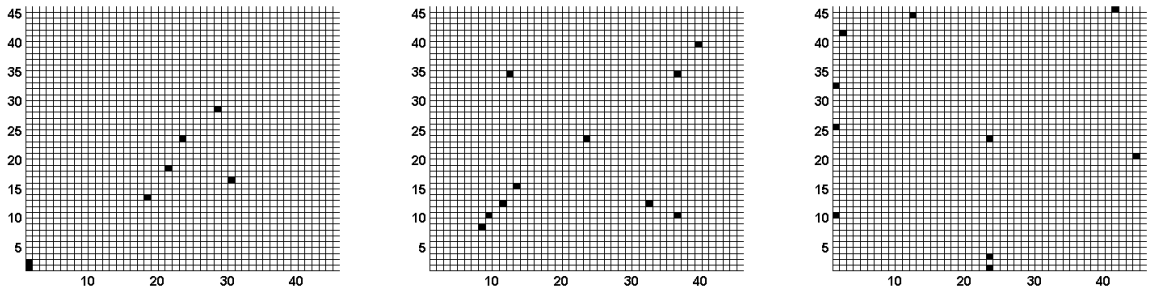


(a) mean of all snapshots (b) DEIM on all snapshots

Figure 3.26: Application of global DEIM on all snapshots and the selected grid blocks



(a) mean 1st cluster (b) mean 2nd cluster (c) mean 3rd cluster



(d) DEIM 1st cluster (e) DEIM 2st cluster (f) DEIM 3st cluster

Figure 3.27: Application of localized DEIM on four clusters and 10 selected grid points for each cluster

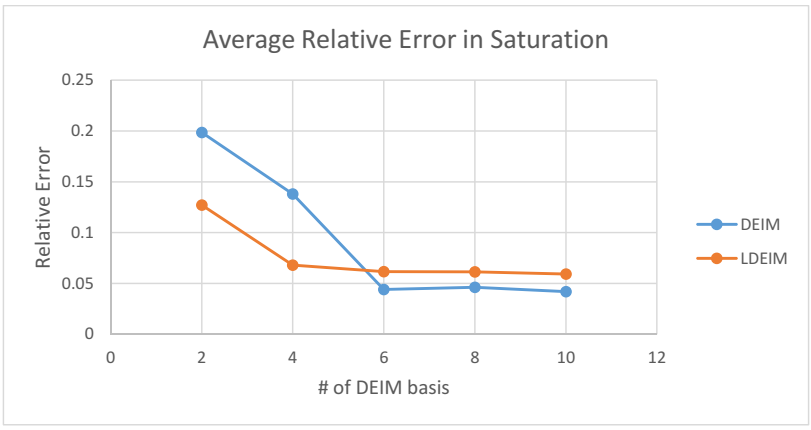


Figure 3.28: Average error with respect to increasing localized DEIM basis

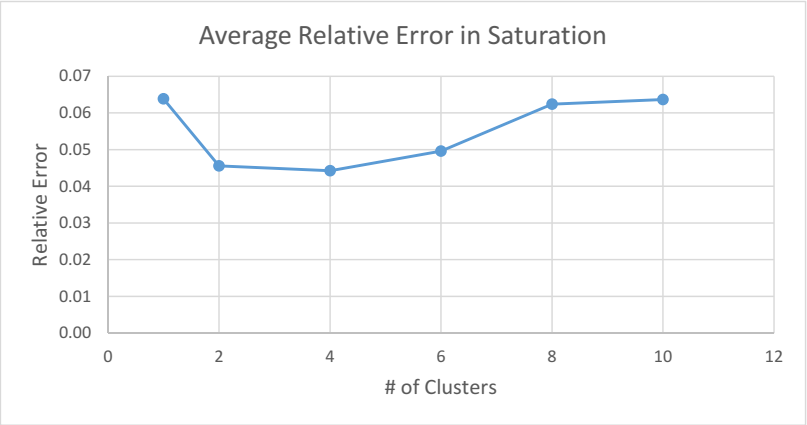


Figure 3.29: Average saturation error with respect to increasing the number of localized DEIM clusters

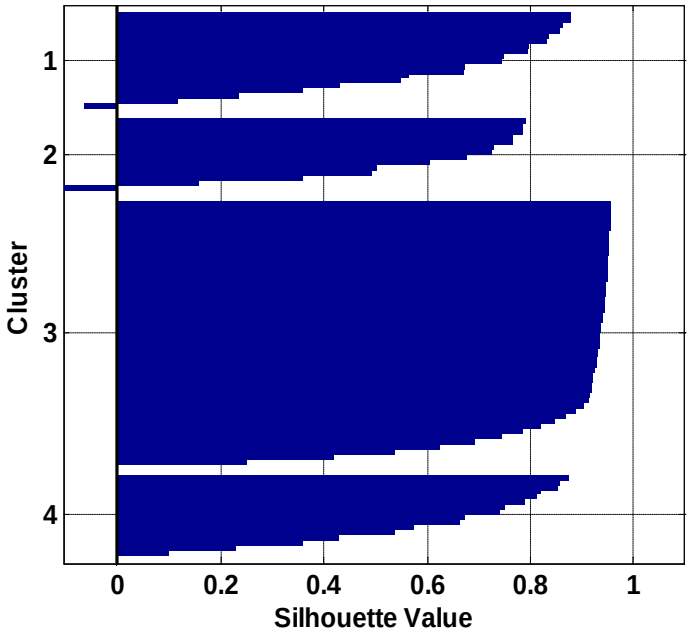
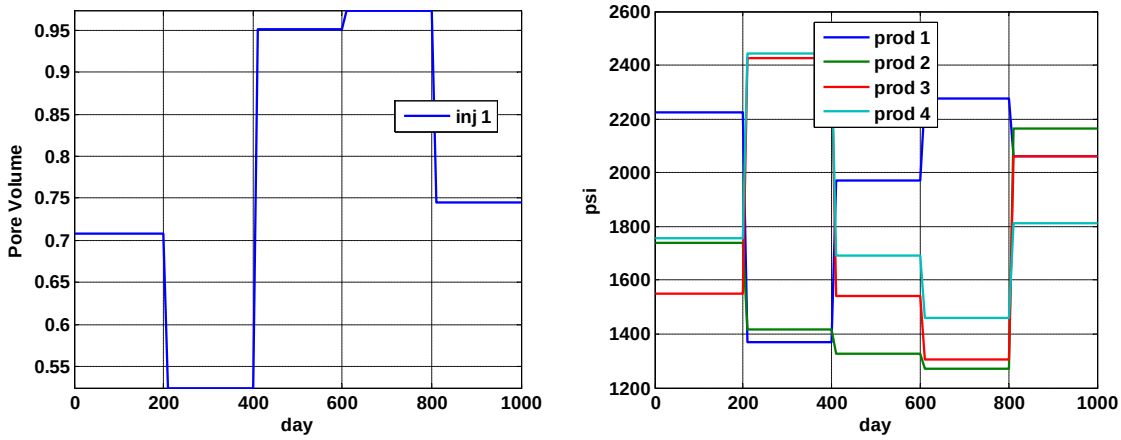
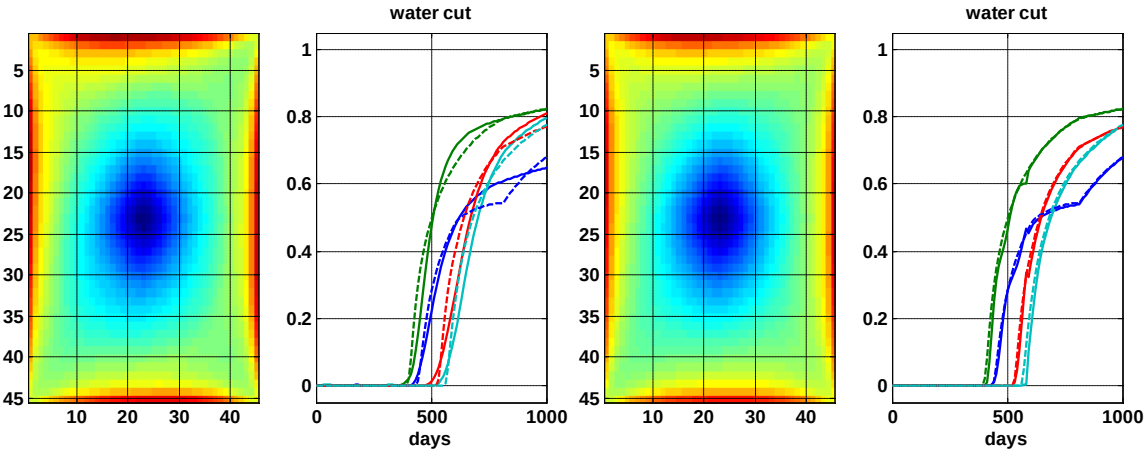


Figure 3.30: Localized DEIM cluster analysis based on silhouette plot



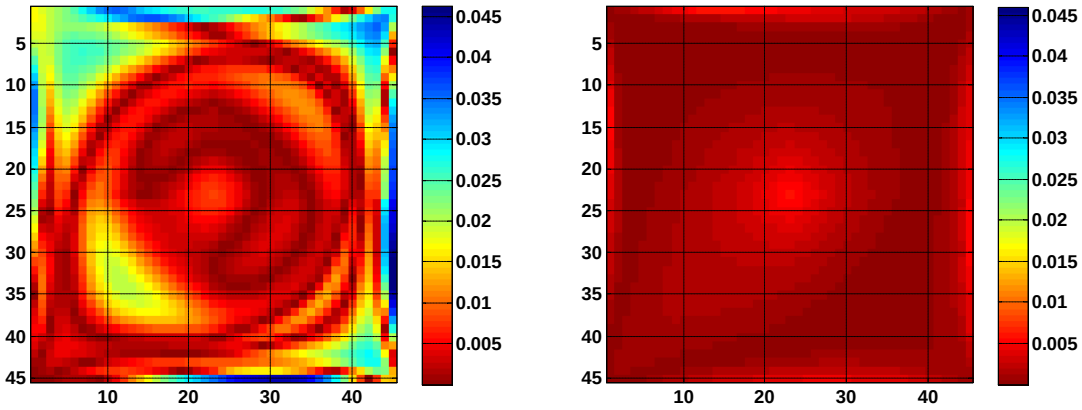
(a) injection rate (b) producers BHP

Figure 3.31: Training schedule of the homogeneous model



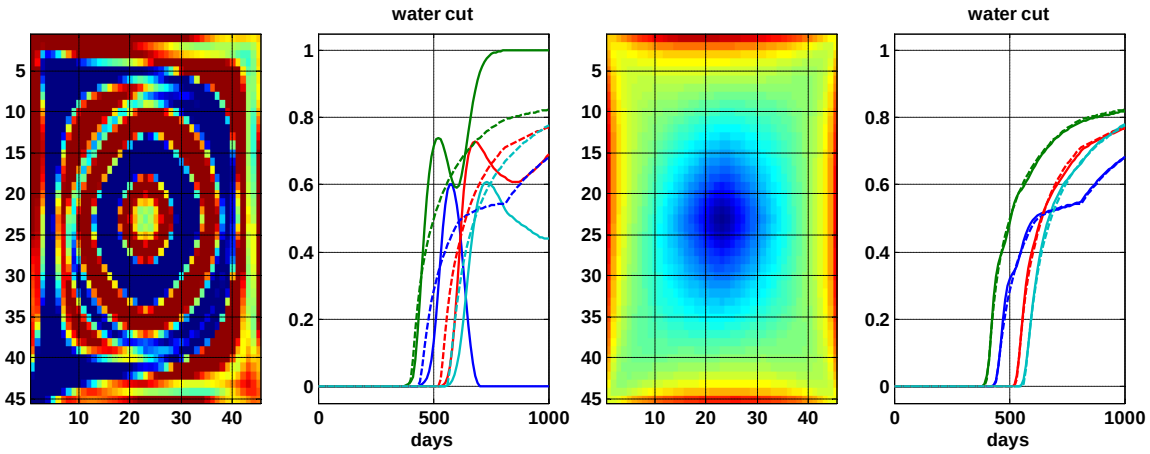
(a) POD (5 basis) (b) LPOD (5 basis)

Figure 3.32: Final water saturation and water cut at producers by applying POD and LPOD on the homogeneous model under the training schedule



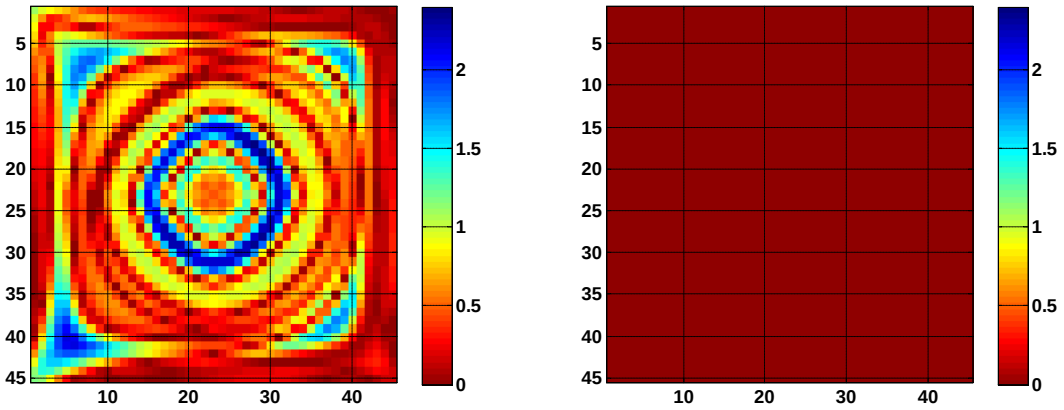
(a) POD (5 basis) (b) LPOD (5 basis)

Figure 3.33: Final water saturation error by applying POD and LPOD on the homogeneous model with the training schedule



(a) DEIM (10 basis) (b) LDEIM (10 basis)

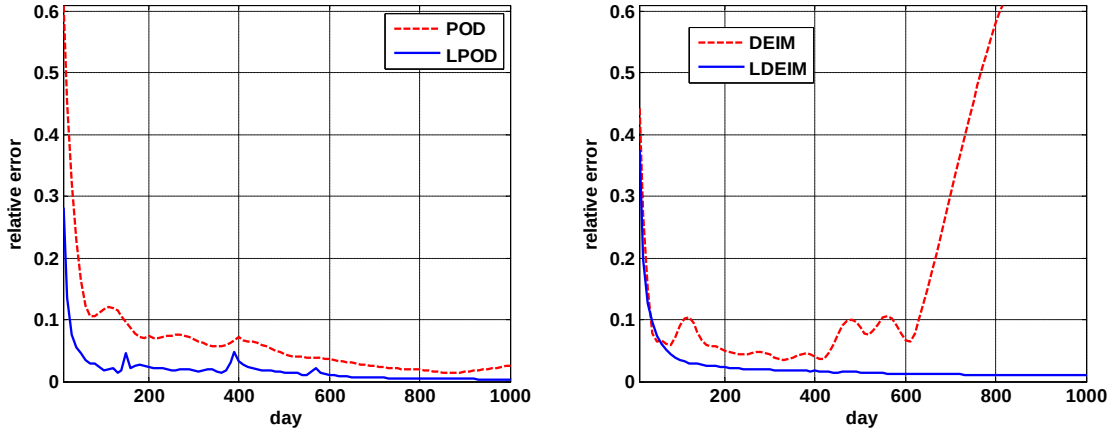
Figure 3.34: Final water saturation and water cut at producers by applying DEIM and LDEIM on the homogeneous model under the training schedule



(a) DEIM (10 basis)

(b) LDEIM (10 basis)

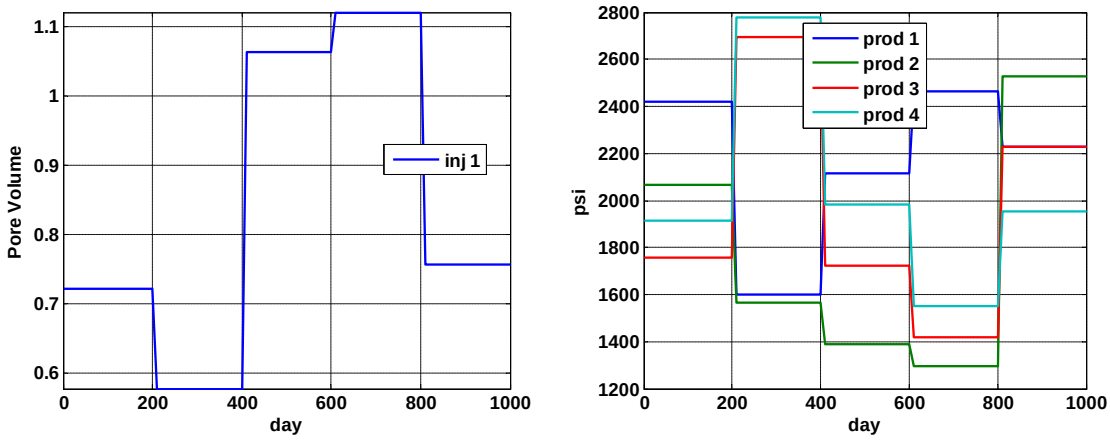
Figure 3.35: Final water saturation error by applying DEIM and LDEIM on the homogeneous model with the training schedule



(a) POD and LPOD

(b) DEIM and LDEIM

Figure 3.36: Temporal saturation error for the homogeneous model under the training schedule



(a) injection rate (b) producers BHP

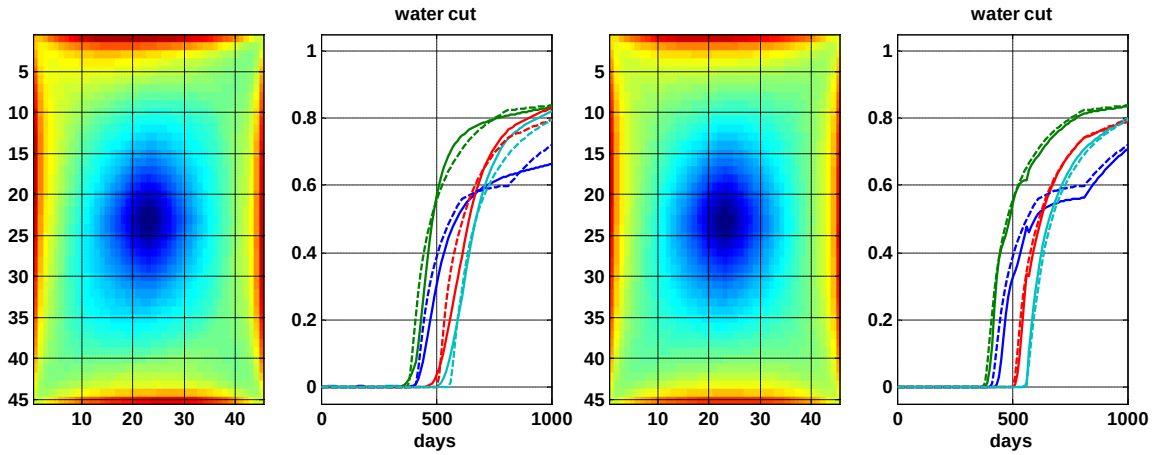
Figure 3.37: Test schedule of the homogeneous model

fidelity model with almost exact water-cuts. Figures 3.39 compares the same results for DEIM and LDEIM, whereas LDEIM outperforms regular DEIM.

The temporal saturation error is calculated based on (3.21) and is compared for POD and LPOD in Figure 3.40a and for DEIM and LDEIM in Figure 3.40b. All of these results also confirm the superiority of localized model reduction over global method with test schedule.

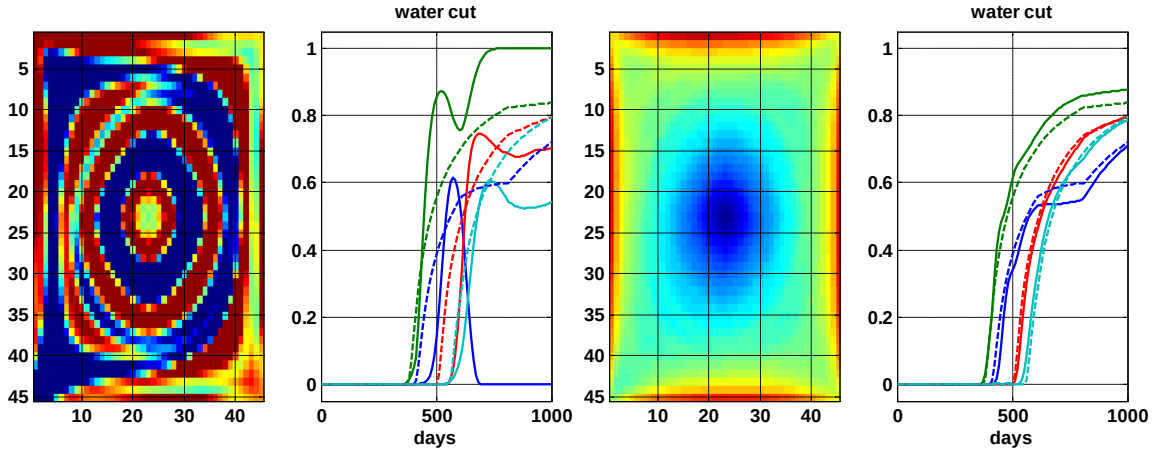
3.8.4 Application of LPOD-LDEIM on Heterogeneous Model

In this example the localized model reduction is applied to a two-phase flow (oil-water) reservoir model under the water flooding recovery process with the structure of a 5-spot as shown in Fig. 3.41. The permeability of the reservoir is taken from SPE10 comparative model [31] (layer 10th). The reservoir model is discretized using Cartesian grid of size $20ft \times 10ft \times 2ft$. Overall the reservoir model has $60 \times 220 \times 1 = 13200$ active cells. The fluid viscosity ratio is $\mu_w/\mu_o = 0.1$. The relative permeability curves is quadratic as depicted in Fig. 4.8b. We assumed a constant porosity of 0.2



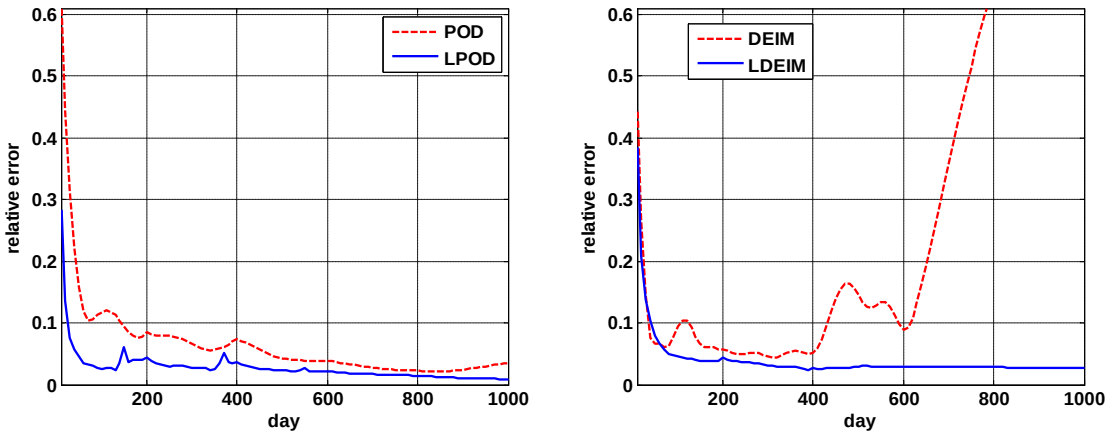
(a) POD (5 basis) (b) LPOD (5 basis)

Figure 3.38: Final water saturation and water cut at producers by applying POD and LPOD on the homogeneous model under the training schedule



(a) DEIM (10 basis) (b) LDEIM (10 basis)

Figure 3.39: Final water saturation and water cut at producers by applying DEIM and LDEIM on the homogeneous model under the training schedule



(a) POD and LPOD (b) DEIM and LDEIM

Figure 3.40: Temporal saturation error for the homogeneous model under the test schedule

for entire model.

The producers are controlled by bottom hole pressure and the injector by rate. The input schedule is changed every 200 days as shown in Figs. 3.42. This figure includes both the training and the test schedule. Note that injection volume is at least one pore volume throughout simulation time (1000 days). The initial water saturation and pressure are assumed to be 0.0 and 2500 psia, respectively.

In order to apply the LPOD-LDEIM methods, we simulated the reservoir for 1000 days and saved the snapshots of pressures, velocity, water saturations and the nonlinear fractional function every 10 days. Thus, we have 100 snapshots for each variables. Each snapshot is reshaped to a column vector and is stacked in a snapshot matrix. After applying *svd* to each matrix, one can find the basis for reduced model. The singular values of the snapshot matrices are shown in Figure 3.43. As it can be seen, there is a faster decay in the singular values for the pressure and velocity com-

pared to saturation and fractional function. Thus, we need more basis for saturation and fractional function to have a reliable reduced model.

Here, we apply regular (global) POD to velocity and pressure, and LPOD to saturation because it is obtained from a nonlinear equation solved by Netwon-Raphson iterations. The selection criteria for pressure and velocity basis was to capture at least 99% of the energy of snapshots. Throughout this example we compare the LPOD and POD as well as DEIM and LDEIM on saturation equation.

The number of basis is compared between different reduced order models and the original fine scale one in Table 3.6. The snapshots were classified into 4 clusters. The overall speedup is around 5.5 because this is still relatively small model. Also, even though localized method has more overhead during the online phase, it does not have significantly higher computational time compared to regular POD-DEIM.

Note that, when we apply DEIM or LDEIM, the saturation is assumed to be projected to reduced subspace by regular (global) POD basis. Combining LPOD-LDEIM will be considered in future work. In this example, we select 7 basis for both POD and LPOD and 9 basis for both DEIM and LDEIM.

The final water saturation and the water cut at all the producers are shown in Figure 3.44 for POD and LPOD. As illustrated in these figures the LPOD approximate the original fine scale model much better, with almost exact water cut. Figure 3.46 compares the same results for DEIM and LDEIM, whereas LDEIM outperforms regular DEIM.

The spatial error in final water saturation is also compared between POD and LPOD in Figure 3.45 and between DEIM and LDEIM in Figure 3.47.

The relative temporal saturation error is calculated based on (3.21) and is compared for POD and LPOD in Figure 3.48a and for DEIM and LDEIM in Figure 3.48b. All of these results confirm the superiority of local model reduction over global one.

Num # of	Fine Scale	POD/LPOD	DEIM/LDEIM
pressure basis	13200	3	–
velocity basis	26120	12	–
saturation basis	13200	7/7	–
fractional basis	13200	–	9/9
iterations	712	260/281	320/335
Total elapsed time	108.7 (s)	19.6/20.2 (s)	18.7/19.2 (s)

Table 3.6: Compare localized and global model order reduction

Note in these figures, at some time instances the saturation error increases suddenly in the localized MOR methods, before it decreases again. These points correspond to switching from one reduced subspace to another one. More investigation is required to figure out how to make this transition smoother.

The reduced model is simulated with a new test schedule as shown in Figure 3.49, to ensure that the localized model order reduction is robust to input variation. This schedule is obtained by $\pm 5\%$ random perturbation of the training schedule. Note that the basis of the reduced model are not updated and we used the same basis obtained from training snapshots.

The final water saturation and water cut for all the producers after 1000 days of simulation with the test schedule are shown in Figure 3.50 for POD and LPOD, and in Figure 3.52 for DEIM and LDEIM. These figures verify that localized MOR techniques outperform the regular MOR methods.

The spatial error in final water saturation is also compared between POD and LPOD in Figure 3.51 and between DEIM and LDEIM in Figure 3.53. The relative temporal saturation error is compared for POD and LPOD in Figure 3.54a and for DEIM and LDEIM in Figure 3.54b. All of these results confirm that localized MOR is robust to input changes and also it is superior to global MOR techniques.

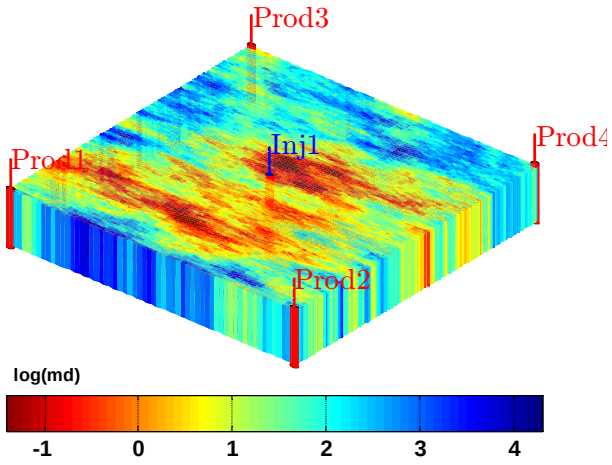
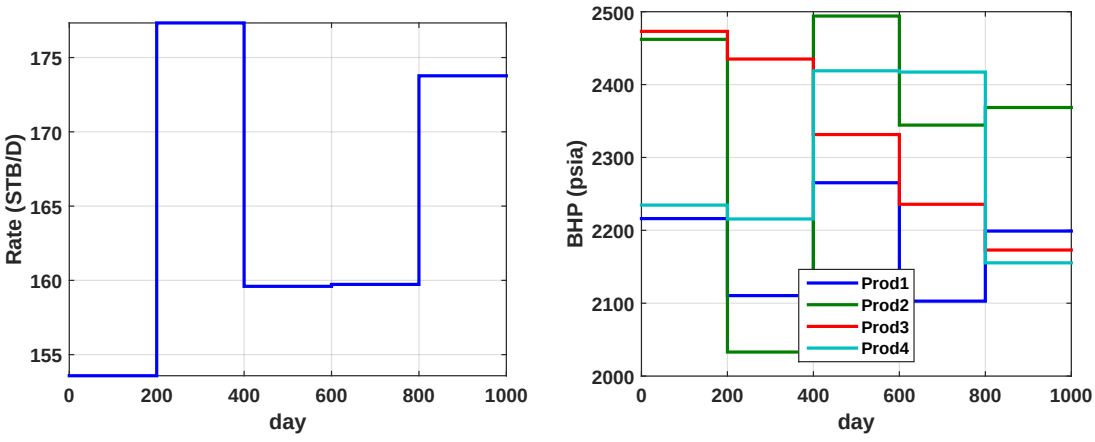


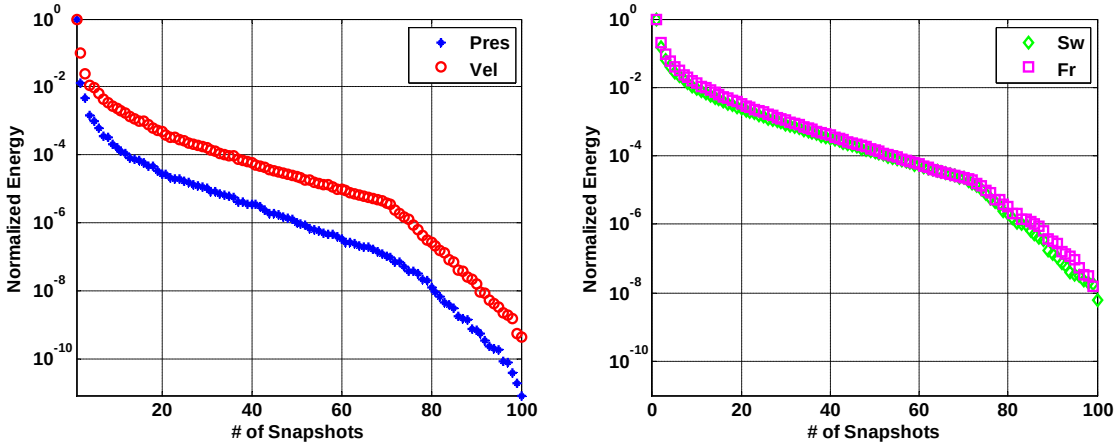
Figure 3.41: 5-spot pattern on Layer 10th of SPE10



(a) injection rate

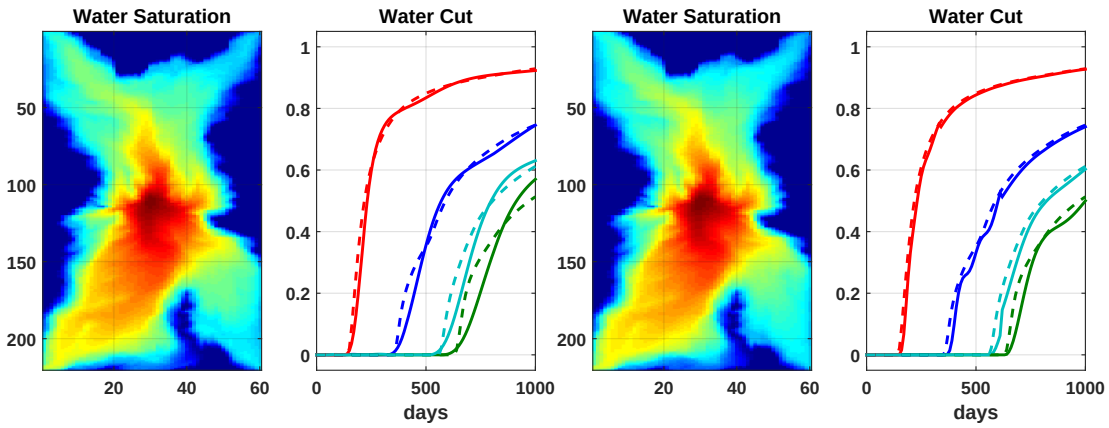
(b) producers BHP

Figure 3.42: Training schedule of the heterogeneous model



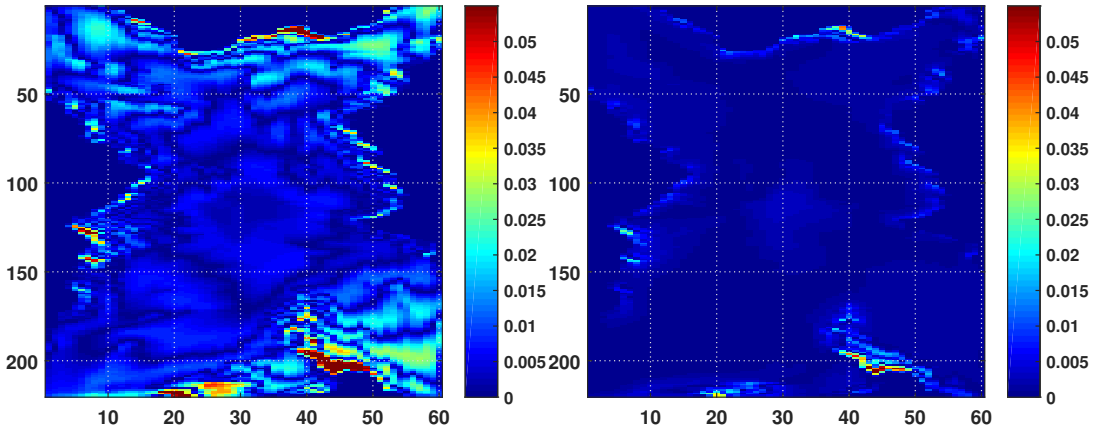
(a) pressure and velocity (b) saturation and fractional function

Figure 3.43: Singular values of snapshot matrices



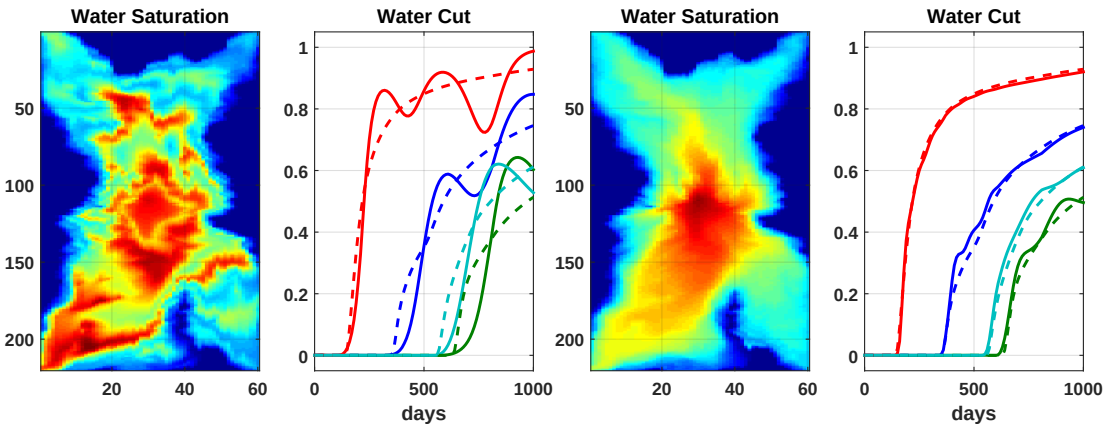
(a) POD (7 basis) (b) LPOD (7 basis)

Figure 3.44: Final water saturation and water cut at producers after applying POD and LPOD on heterogeneous model under the training schedule



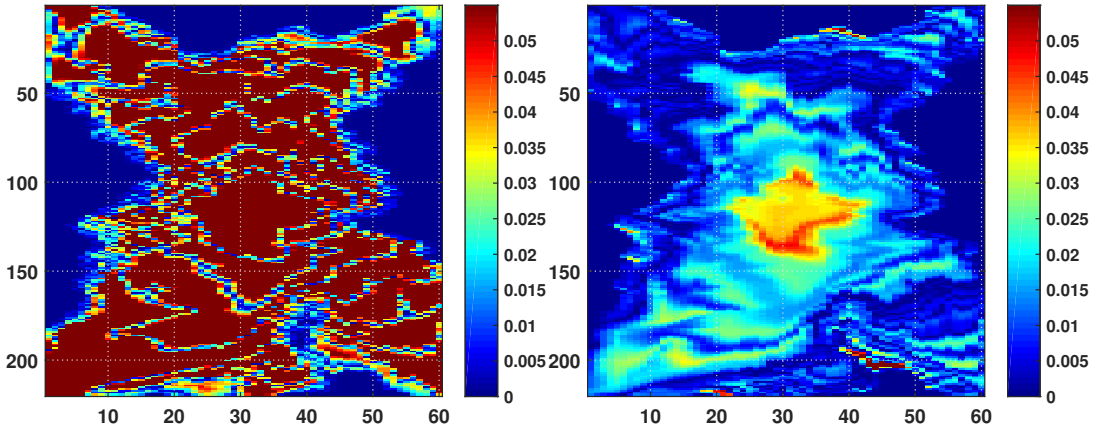
(a) POD (7 basis) (b) LPOD (7 basis)

Figure 3.45: Final water saturation error after applying POD and LPOD on heterogeneous model under the training schedule



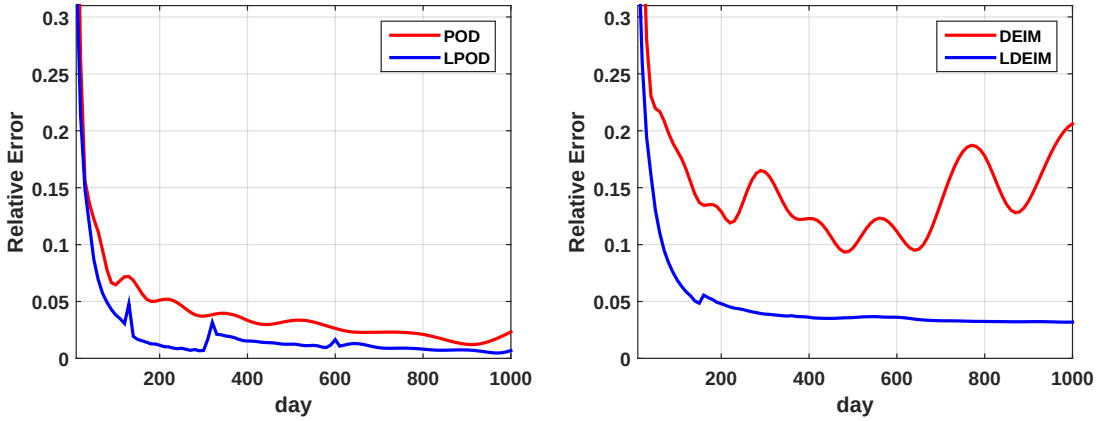
(a) DEIM (9 basis) (b) LDEIM (9 basis)

Figure 3.46: Final water saturation and water cut at producers after applying DEIM and LDEIM on heterogeneous model under the training schedule



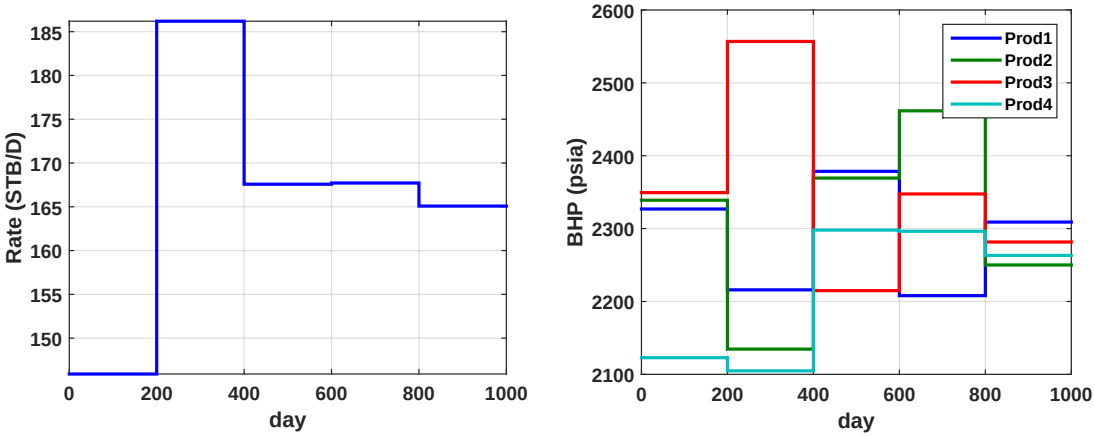
(a) DEIM (9 basis) (b) LDEIM (9 basis)

Figure 3.47: Final water saturation error after applying DEIM and LDEIM on heterogeneous model under training schedule



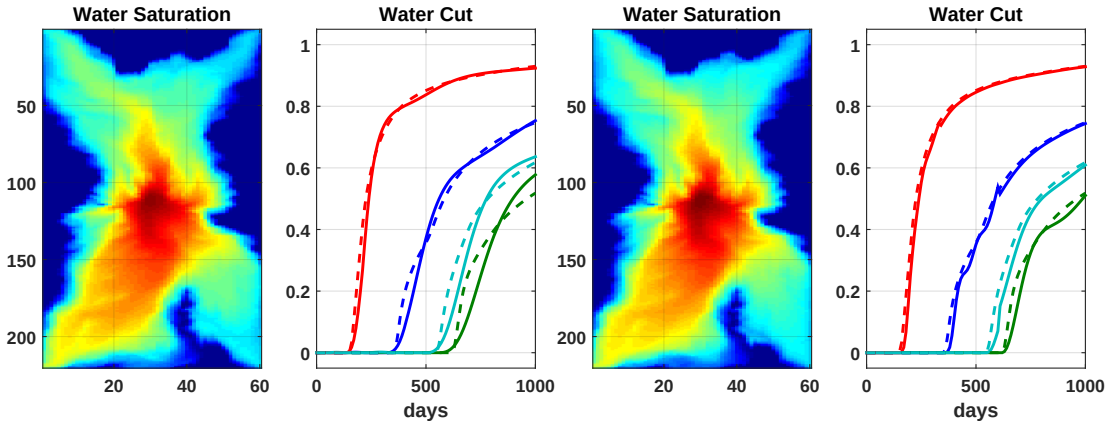
(a) POD and LPOD (b) DEIM and LDEIM

Figure 3.48: Temporal saturation error after applying localized MOR on heterogeneous model under training schedule



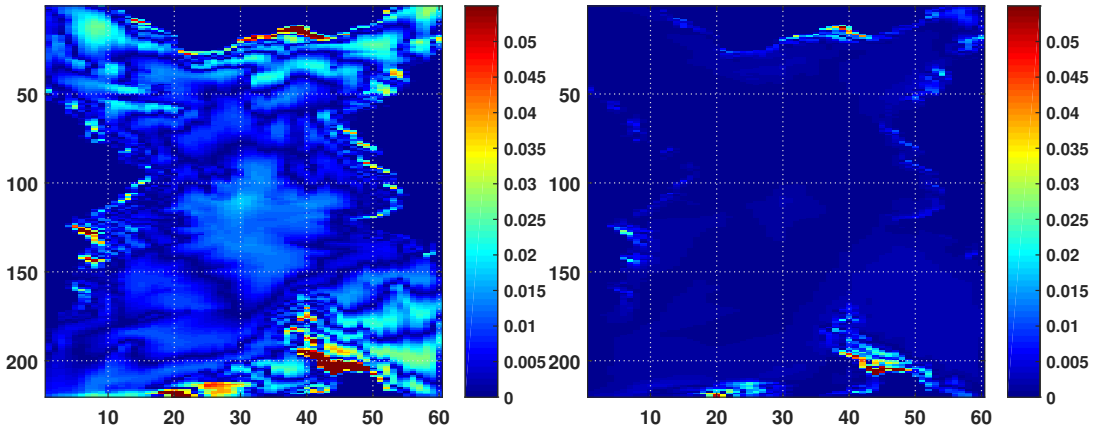
(a) injection rate (b) producers BHP

Figure 3.49: Test schedule resulted from ±5% variation in the training schedule



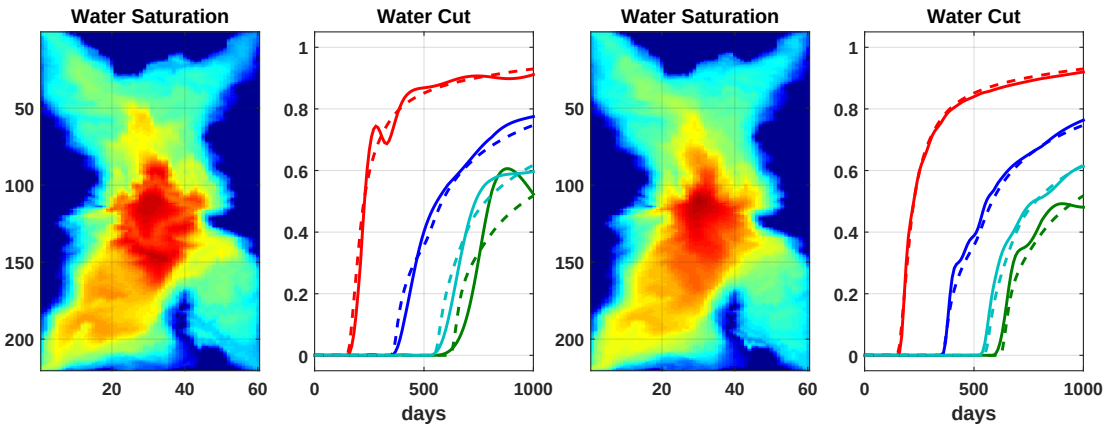
(a) POD (7 basis) (b) LPOD (7 basis)

Figure 3.50: Final water saturation and water cut after producers after applying POD and LPOD on heterogeneous model under the test schedule



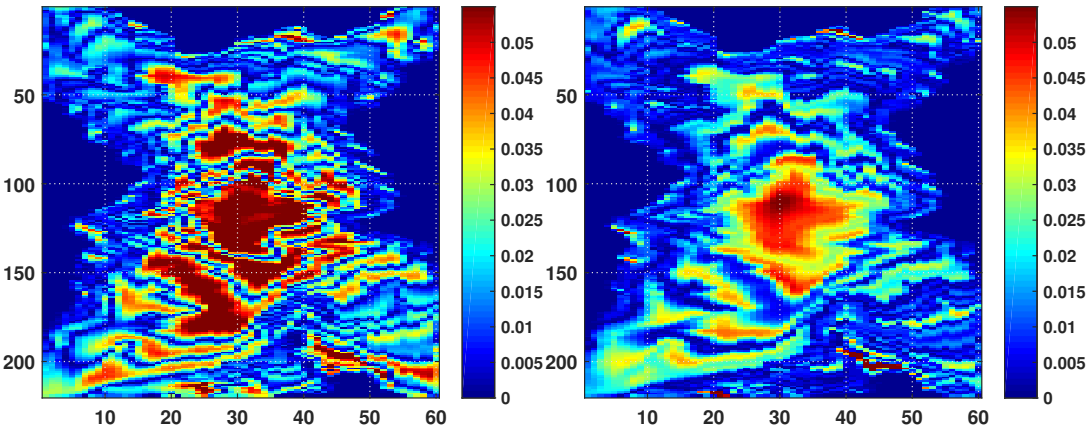
(a) POD (7 basis) (b) LPOD (7 basis)

Figure 3.51: Final water saturation error after applying POD and LPOD on heterogeneous model under the test schedule



(a) DEIM (9 basis) (b) LDEIM (9 basis)

Figure 3.52: Final water saturation and water cut at producers after applying DEIM and LDEIM on heterogeneous model under the test schedule



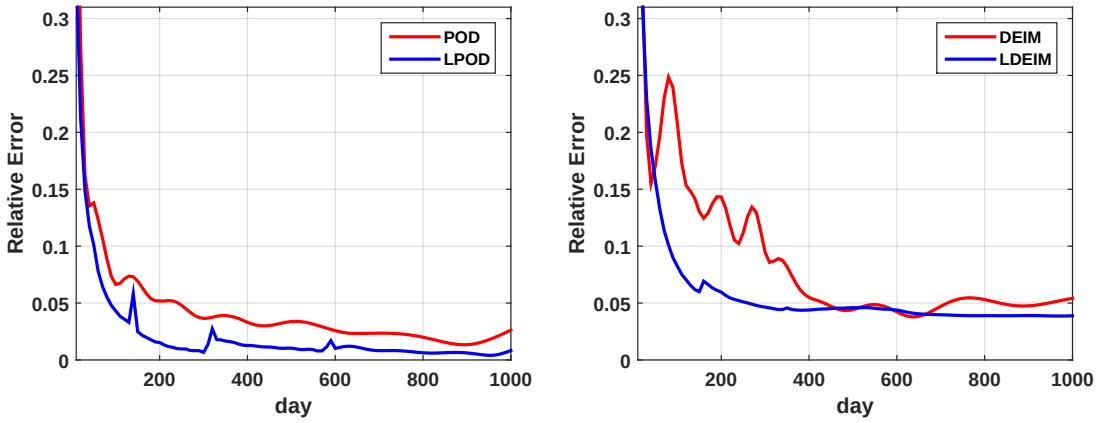
(a) DEIM (9 basis) (b) LDEIM (9 basis)

Figure 3.53: Final water saturation error after applying DEIM and LDEIM on heterogeneous model under the test schedule

3.9 Chapter Summary

I this chapter, reduced order modeling based on POD was addressed. After reviewing the POD procedure, its drawback in nonlinear systems was discussed and a new technique, namely DEIM, was applied to alleviate the issue. This new method mitigate the computational time by reducing the number of required nonlinear function evaluations. The POD-DEIM was verified in a numerical example whereby it was applied to SPE-10 benchmark and produced reliable results with respect to high fidelity model.

Next, to improve the performance of POD-DEIM and to have more representative basis, a localized POD-DEIM model reduction scheme was introduced for the solution of the two-phase flow in heterogeneous porous media. It has been our experience that performing localization with multiple snapshot yield a more accurate and stable reduced order models than a single snapshot set. More work still need to be done in the selection the appropriate number of basis and the specific clustering technique



(a) POD and LPOD

(b) DEIM and LDEIM

Figure 3.54: Temporal saturation error after applying localized MOR on heterogeneous model under the test schedule

used.

In the next chapter, a new formulation is developed for porous media flow equations, called bilinear, which can improve the performance of reduced order models in terms of stability and validity.

4. MODEL ORDER REDUCTION ON BILINEAR RESERVOIR MODELS

The model order reduction (MOR) techniques that were discussed in the last chapter, are input dependent, meaning that in order to construct a reduced order model one needs snapshots of a computed solution of the original model. Thus, if the input function is changed, which is common in optimization, no rigorous error bound can be declared for the new solution.

There are well-established MOR procedures for linear systems to construct the reduced model without the need for training and simulation of fine scale models based on input-output transfer function description of the model. For example balanced truncation, which preserves the most controllable and observable states of the system [7]. However, one of the main challenges in nonlinear systems is the lack of a general input-output representation analogous to transfer function in linear systems.

Nevertheless, if a nonlinear system can be transformed into a new form, namely a bilinear system, higher order transfer functions can be constructed to capture the input-output behavior of the underlying system [98]. The bilinear systems are linear in the input and linear in the state but not in both jointly. The main advantage of having a nonlinear system in this form is that the reduced order model can be constructed without the snapshot, meaning that there is no need to simulate the original large scale model [18, 41]. In this chapter, two methods for transforming the nonlinear systems into bilinear form are addressed.

In the first approach to convert a nonlinear system into a bilinear form, the nonlinear system is approximated using higher order terms of the Taylor expansion. Usually these expansion terms are truncated at second order and yield a bilinear system. This process is also known as Carleman bilinearizations. This new form

of the system has higher dimension than the original model, yet the generalized moment matching technique based on iterative rational Krylov algorithm [59, 17] is applied to find the projection matrices and to reduce the size of the problem, without depending on a particular input trajectory or training. This method is applied to a single phase flow model (a highly compressible gas reservoir) and numerical results are presented to verify the systematic MOR [46].

The second approach transforms the nonlinear system of two phase flow into a new dynamical system called quadratic bilinear differential algebraic equations (QBDAEs). The author in [58] showed that a large class of smooth nonlinear affine systems can be transformed into QBDAE. The main idea is to introduce new variables and rewrite the equations as new algebraic or differential equations that are at most quadratic with respect to states. The dimension of the new system of equations increases linearly, whereas the previous method based on Carleman bilinearization increases the size of problem quadratically. These two techniques for bilinearization of nonlinear systems are compared in [17].

Although the idea behind QBDAE of two phase flow is to reduced the size of the problem without simulation and training, there are still some issues that need to be addressed such as multiple inputs and outputs nature of reservoir models and non-affine inputs. As this study is the pioneer work in considering these new techniques for porous media flow simulation, only POD model order reduction on QBDAE form of two phase flow was applied to prove the applicability of new formulation. The systematic MOR approaches without training and snapshots for the QBDAE system of two phase flow should be addressed in the future work.

4.1 Carleman Bilinearization

In this section the bilinear form is computed using Carleman bilinearization approach, which is truncated multidimensional Taylor's expansion of the nonlinear terms. Consider a state space equations of a model as,

$$\begin{aligned} \dot{x}(t) &= a(x(t),t) + b(x(t),t)u(t), \quad t \geq 0 \\ y(t) &= c(t)x(t) + y_0, \quad x(0) = x_0, \end{aligned} \tag{4.1}$$

where $x(t)$ is the state vector, $u(t)$ and $y(t)$ are the input and the output of the system, respectively. We assume that the system (4.1) is linear-analytic, i.e. the functions $a(x,t)$ and $b(x,t)$ are analytic in x and $a(x,t), b(x,t)$ and $c(t)$ are continuous in t.

Without loss of generality, we assume that initial condition is zero, and also $a(0,t) = 0$ and $u(t) = 0$ implies $x(t) = 0$ and $y(t) = y_0$. Nevertheless, if this is not the case, one can always transform the system to 4.1 by introducing a new state variable $\hat{x} = x - x_0$.

Using multivariate Taylor's expansion around $x_0 = 0$, we obtain the following power series representations

$$\begin{aligned} a(x,t) &= a_1(t)x^{(1)} + a_2(t)x^{(2)} + \cdots + a_N(t)x^{(N)} + \ldots \\ b(x,t) &= b_0(t) + b_1(t)x^{(1)} + \cdots + b_{N-1}(t)x^{(N-1)} + \ldots \end{aligned} \tag{4.2}$$

where $x^{(i)} = x^{\otimes i} = \overbrace{x \otimes x \otimes \cdots \otimes x}^{i \text{ times}}$ and $\otimes$ denotes the Kroncker's product, for example $x^{(2)} = x \otimes x$, and $a_i(t) = \frac{\partial^i a(x,t)}{\partial x^i}|_{x_0}$ and $b_i(t) = \frac{\partial^i b(x,t)}{\partial x^i}|_{x_0}$.

The system (4.1) can be written in the form

$$\begin{aligned} \dot{x}(t) &= \sum_{i=1}^{N} a_i(t)x^{(i)} + \sum_{i=0}^{N-1} b_i(t)x^{(i)}u(t) + \ldots, \quad t \geq 0 \\ y(t) &= c(t)x(t) + y_0, \quad x(0) = x_0, \end{aligned} \tag{4.3}$$

where the dots represent higher order terms. To determine the first N kernels corresponding to the system (4.3), differential equations are developed for $x^{(i)}$, $2 \leq i \leq N$. The differential equation of $x^{(i)}$ takes the form

$$\begin{aligned} \frac{d}{dt}[x^{(i)}(t)] = &\sum_{i=1}^{N-i+1} a_{i,k}(t)x^{(k+i-1)} + \sum_{i=0}^{N-1} b_{i,k}(t)x^{(k+i-1)}u(t) + \ldots, \quad t \geq 0 \\ &x^{(i)}(0) = 0, \quad i = 1, 2, \ldots, N. \end{aligned} \tag{4.4}$$

The coefficients $a_{i,k}$ are given by $a_{1,k}(t) = a_k(t)$, and for $i > 1$

$$\begin{aligned} a_{i,k}(t) = &a_k(t) \otimes I_n \otimes \cdots \otimes I_n + I_n \otimes a_k(t) \otimes I_n \otimes \cdots \otimes I_n \\ &+ \cdots + I_n \otimes \cdots \otimes I_n \otimes a_k(t) \end{aligned} \tag{4.5}$$

where the number of terms is i and there are $i - 1$ Kronecker products. A similar notation is used for $b_{i,k}$. If we write $x^{\otimes} = (x^{(1)}, x^{(2)}, \ldots, x^{(N)})^{\intercal}$, then Carleman bilinearization of system (4.1) is written in the form of

$$\frac{d}{dt}x^{\otimes} = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,N} \\ 0 & a_{2,1} & \cdots & a_{2,N-1} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{N,1} \end{pmatrix} x^{\otimes}$$
$$+ \begin{pmatrix} b_{1,1} & b_{1,2} & \cdots & b_{1,N-1} & 0 \\ b_{2,0} & b_{2,1} & \cdots & b_{2,N-2} & 0 \\ 0 & b_{3,0} & \cdots & b_{3,N-3} & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & b_{N,0} & 0 \end{pmatrix} x^{\otimes}u + \begin{pmatrix} b_{1,0} \\ 0 \\ \vdots \\ 0 \end{pmatrix} u + \ldots \tag{4.6}$$
$$y(t) = [c(t)\, 0 \ldots\, 0]x^{\otimes}(t) + y_0(t) + \ldots, \quad x^{\otimes}(0) = 0.$$

The bilinear form of (4.6) can be posed in matrix-vector notation as,

$$\frac{d}{dt}x^{\otimes}(t) = Ax^{\otimes}(t) + \sum_{k=1}^{m} N_k x^{\otimes}(t)u_k(t) + Bu(t) \tag{4.7}$$
$$y(t) = Cx^{\otimes}(t), \tag{4.8}$$

where A, B, C and N_k are matrices derived from the bilineariztion process as are indicated in Eq. (4.6).

4.2 Model Order Reduction of Bilinear Systems

In this section, we briefly introduce the concept of model reduction for bilinear systems. The idea is to construct the projector $P = VW^T$ where $V, W \in \mathbb{R}^{n\times r}$ with $W^TV = \mathbf{I}_r$, such that the reduced-order model can be obtained by projecting the state-space equations into a much smaller subspace by

$$x^{\otimes} = Vx_r^{\otimes} \tag{4.9}$$

In the model order reduction framework, one seeks to construct a reduced bilinear system as

$$\frac{d}{dt}x_r^{\otimes}(t) = \hat{A}x_r^{\otimes}(t) + \sum_{k=1}^{m} \hat{N}_k x_r^{\otimes}(t)u_k(t) + \hat{B}u(t) \tag{4.10}$$

$$y(t) = \hat{C}x_r^{\otimes}(t), \tag{4.11}$$

where

$$\hat{A} = W^T A V, \tag{4.12}$$

$$\hat{N} = W^T N_k V, \quad k = 1, \cdots, m \tag{4.13}$$

$$\hat{B} = W^T B, \quad \hat{C} = CV \tag{4.14}$$

The main question is how to find the projection matrices, V, W. Krylov subspace techniques have been addressed frequently in recent years for model order reduction (MOR) of large scale systems. Krylov method approximate a large scale model by moment matching. This requires that the reduced system response matches in some certain frequencies with the response of high fidelity model. This problem can be solved efficiently by means of iterative methods, such as Arnoldi and Lanczos factorization [14].

Here the generalization of the rational Krylov interpolation method that is known for linear system [60] is employed to match a set of points in the transfer function domain. Selecting these points is very crucial to have a reasonable reduced model. Considering the $\mathbb{H}_2$-norm of the error and minimizing it results in the Bilinear Iterative Rational Krylov Algorithm (BIRKA) 4. One can refer to [17] for the derivation and explanation of this algorithm.

Algorithm 4 Bilinear Iterative Rational Krylov Algorithm (BIRKA) [17]

input A, r, q
make an initial selection $\sigma_1, \cdot, \sigma_q$
while change in $\sigma_i > \epsilon$ **do**
    Compute $V = [V_1, \cdots, V_r]$ and $W = [W_1, \cdots, W_r] \in \mathbb{R}^{n \times (q + \cdots + q^r)}$
    Compute truncated SVD of V_q and W_q of V and W.
    $\hat{A} = (W_q^T \, V_q)^{-1} W_q^T \, A \, V_q$
    $\sigma_i \leftarrow -\lambda_i(\hat{A})$
end while
return $V = V_q, \quad W^T = (W_q^T \, V_q)^{-1} W_q^T$

4.3 Carleman Bilinearization of Single Phase Flow Model

The single phase flow equation in porous media without gravity is expressed as follows,

$$\frac{\partial(\rho\phi)}{\partial t} = \nabla . \left(\frac{\rho}{\mu} K \nabla p \right) + q \tag{4.15}$$

where ρ is the fluid density, ϕ is porosity of the rock, μ is the viscosity, K is the permeability, and p is the pressure. Assume that the fluid and rock properties are a function of pressure as,

$$\rho = \rho_{ref} \mathrm{e}^{c_o \left(P - P_{ref} \right)} \Rightarrow \frac{\partial \rho}{\partial p} = c_o \rho \tag{4.16}$$

$$\mu = \mu_{ref} \mathrm{e}^{-c_\mu \left(P - P_{ref} \right)} \Rightarrow \frac{\partial \mu}{\partial p} = -c_\mu \mu \tag{4.17}$$

$$\phi = \phi_{ref} \mathrm{e}^{c_r \left(P - P_{ref} \right)} \Rightarrow \frac{\partial \phi}{\partial p} = c_r \phi \tag{4.18}$$

Thus Eq. (4.15) will be simplified to,

$$\rho \phi c_t \frac{\partial p}{\partial t} = \nabla . \left(\frac{\rho}{\mu} K \nabla p \right) + q \tag{4.19}$$

$$\text{Accumulation} = \text{Flux} + \text{Sink/Source} \tag{4.20}$$

After applying finite volume and spatial discretization of Eq. (4.20), the final equation can be written as,

$$A\left(p\right)\frac{dp}{dt} = T(p)\ p - q\left(p\right)p_{wf} \tag{4.21}$$

where A is an accumulation matrix that is diagonal, T is a transmissiblity matrix that is hepta diagonal in general case, q is a vector that is non-zero at the well locations, and p_{wf} is a bottom-hole pressure of the well that is the control input of the systems.

We consider the following form of Eq. (4.21)

$$\begin{aligned}\frac{dp}{dt} &= A(p)^{-1}\,T(p)p - A(p)^{-1}\,q(p)p_{wf} &(4.22)\\ &= \hat{T}(p)p - \hat{q}(p)p_{wf}, \quad p(0) = 0. &(4.23)\end{aligned}$$

Writing $f(p) = \hat{T}(p)p$ one can define $f_0 = 0$, $f_1 = \frac{\partial f(P)}{\partial P^\intercal}|_{P=0}$, and $f_2 = \frac{\partial^2 f(P)}{\partial P^{\intercal 2}}|_{P=0}$. Thus to write the system in the form of Eq. 4.6 the following terms are calculated,

$$\begin{aligned}f_{1,1} &= f_1, \qquad f_{1,2} = f_2 &(4.24)\\ f_{2,1} &= f_1 \otimes I_n + I_n \otimes f_1 &(4.25)\end{aligned}$$

In order to find Taylor's series terms, one needs to compute the first and second derivatives of the transmissibility term $f(p)$ as,

$$\begin{aligned}\frac{\partial \hat{T}(p)p}{\partial p^\intercal} &= \frac{\partial \hat{T}(p)}{\partial p^\intercal}(\mathrm{I}_n \otimes p) + \hat{T}(p)\frac{\partial p}{\partial p^\intercal} &(4.26)\\ &= \frac{\partial \hat{T}(p)}{\partial p^\intercal}(\mathrm{I}_n \otimes p) + \hat{T}(p). &(4.27)\end{aligned}$$

Also, the second derivative is given by

$$\frac{\partial^2 \hat{T}(p)p}{\partial p^{\intercal 2}} = \frac{\partial}{\partial p^{\intercal}}\Big(\frac{\partial \hat{T}(p)}{\partial p^{\intercal}}(\mathrm{I}_n \otimes p) + \hat{T}(p)\Big) \tag{4.28}$$

$$= \frac{\partial^2 \hat{T}(p)}{\partial p^{\intercal 2}}(\mathrm{I}_{n^2} \otimes p) + \frac{\partial \hat{T}(p)}{\partial p^{\intercal}}\mathrm{K}_{(n,n)} + \frac{\partial \hat{T}(p)}{\partial p^{\intercal}}, \tag{4.29}$$

where $\mathrm{K}_{(n,n)}$ is a permutation matrix defined by

$$\mathrm{K}_{(n,n)} = \sum_{i,j} \mathrm{U}_{ij} \otimes \mathrm{U}_{ij}^{\intercal}$$

and U_{ij} is the $n \times m$ matrix whose ijth element is 1 and all other elements are 0.

Similarly, one can define $g(p) = \hat{q}(p)$ and $g_0 = \hat{q}(0)$, $g_1 = \frac{\partial g(p)}{\partial p^{\intercal}}|_{p=0}$ and $g_2 = \frac{\partial^2 g(p)}{\partial p^{\intercal 2}}|_{p=0}$ hence writing,

$$g_{1,0} = g_0, \quad g_{1,1} = g_1, \quad g_{1,2} = g_2 \tag{4.30}$$

$$g_{2,0} = g_0 \otimes I_n + I_n \otimes g_0. \tag{4.31}$$

Assuming n is the number of gridblocks and defining $p^{\otimes} = (p^{(1)}, p^{(2)}) \in \mathbb{R}^{n+n^2}$, then Carleman bilinearization of system (4.21) is written in the form of,

$$\frac{d}{dt}p^{\otimes} = \begin{pmatrix} f_{1,1} & f_{1,2} \\ 0 & f_{2,1} \end{pmatrix} p^{\otimes} + \begin{pmatrix} g_{1,1} & 0 \\ g_{2,0} & 0 \end{pmatrix} p^{\otimes} p_{wf} + \begin{pmatrix} g_{1,0} \\ 0 \\ \vdots \\ 0 \end{pmatrix} p_{wf}. \tag{4.32}$$

4.4 Numerical Example

In the following example, the bilinear modeling is applied to a single phase flow. In order to show the bilinear approximation, and discuss the derivation, a numerical example is set up that includes a simple, yet representative, nonlinear single-phase gas reservoir. The reservoir is 210x210x30 ft with uniform griding as each grid block is 30x30x30 ft. The gas isothermal compressibility is assumed to be 5E-3, and the rock compressibility is 3E-5. There is one well that is located in the middle of the reservoir and the draw-down is 500 psi. Also, the permeability is heterogeneous field as shown in Figure (4.1). This model is simulated for 1000 days of production.

First, we formulate all of the matrices involved into the bilinear transformation as shown in Eq. (4.32). Using the reservoir description given above, three different cases were run using the fully nonlinear system, the linear approximated system, and the bilinear approximated one. In both approximations, we take the initial condition, i.e. initial pressure of the reservoir, as the operating point for performing the Taylor series expansions.

The pressure decline at the well-bore and the boundary of the reservoir are shown in Figure (4.2). As can be seen, the bilinear approximation is more representative and describes the nonlinearities better than the simple linearization around the initial point. Note that $p \in \mathbb{R}^{49}$, meaning that there are 49 states in the nonlinear system corresponding to 49 griblocks of the model. Similarly, the linear approximation of the model has the same number of states.

However, the number of states in the bilinear approximated model is $p^{\otimes} \in \mathbb{R}^{49+49^2}$. Thus, it is obvious that we need to reduce the size of bilinear approximated system to make it computationally efficient. We expect that increasing the size of the problem before applying model reduction and introducing new terms corresponding to the

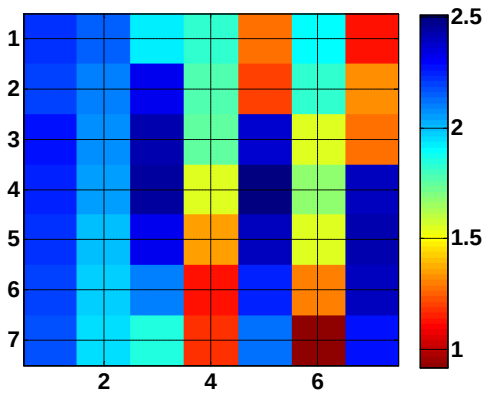


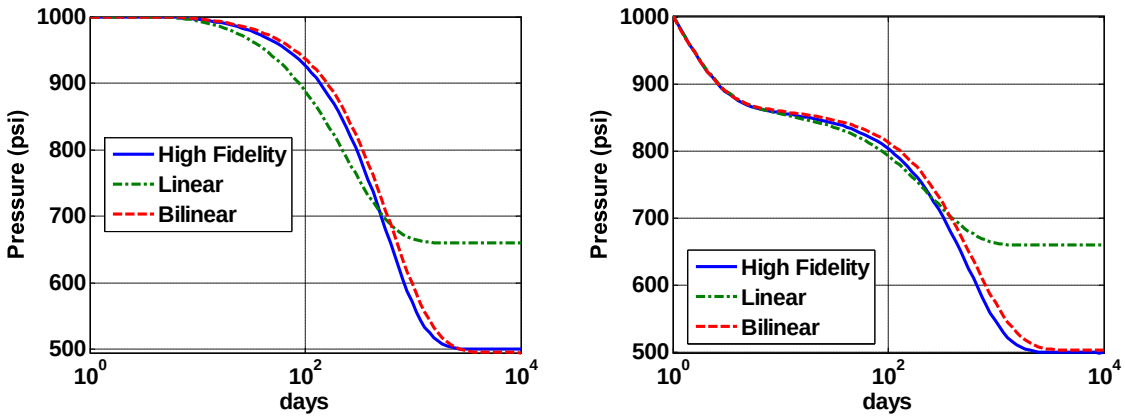
Figure 4.1: Permeability distribution of 7x7 model (in Log(md))

nonlinearity in the system results in a better reduced order model.

In order to reduce the size of the bilinear system, BIRKA in Algorithm 4 is applied to find the basis and construct the projection matrix V and W. These matrices are used to set up the reduced order model of Eq. (4.11). As it is shown in Figure (4.3) the reduced bilinear model with 14 basis will follow the same trend as fine scale nonlinear model with a negligible error. On the other hand, the linear approximation of the model yield large error and divergence in long time simulation of the model. Note that here the original model was linearized at the initial condition, however one might get better results by choosing different linearization point.

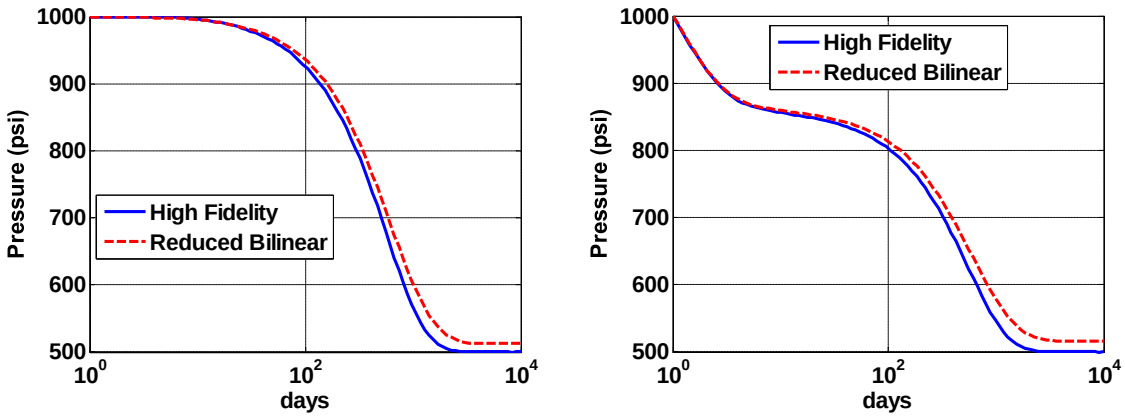
4.5 Quadratic Bilinear Model of Two Phase Flow Model

In this section, the goal is to apply bilinear formulation to the two phase flow in porous media. It is assumed that the rock and fluid are incompressible and the fluid is moving only under viscous force. A sequential discretization formulation is considered as explained in 2, in which at each iteration the pressure is solved and the results are used to solve for the saturation iteratively by Newton-Raphson method.



(a) Pressure at the boundry (b) Pressure at wellbore

Figure 4.2: Temporal pressure changes for different models



(a) Pressure at the boundry (b) Pressure at wellbore

Figure 4.3: Temporal pressure changes for a reduced model with 14 basis

Two distinct methods are considered for transforming the saturation equation into bilinear form, Carleman bilinearization and quadratic bilinear formulation. These two methods are compared in a numerical example. Next, we discuss the model reduction technique applied to the quadratic bilinear form of two-phase flow system. Finally, we demonstrate the proposed methodology in a numerical example of two-phase flow reservoir model under water flooding.

As the pressure equation is linear, one can apply regular POD or other MOR methods to reduce the dimension of the system as discussed in 3.3. Here, I focus on saturation equation since it is nonlinear and the proposed method in this chapter are more applicable to it.

4.5.1 Saturation Equation in State Space Representation

As it was explained in subsection 2.1.2, an implicit time (backward Euler) discretization of the saturation Eq. (2.3) results in,

$$s_i^{n+1} = s_i^n + \frac{\Delta t}{|\Omega_i|}\left(q^+ - \sum_j F_{ij}(s^{n+1})u_{ij} + f_w(s_i^{n+1})q^-\right). \tag{4.33}$$

where s_i^n is the cell-average of the water saturation at time $t = t_n$, $q^+ = \max(q_i, 0)$ and $q^- = \min(q_i, 0)$, u_{ij} is the total flux (for oil and water) over the edge γ_{ij} between the two adjacent cells and F_{ij} is a numerical approximation of the flux over edge γ_{ij} defined as,

$$F_{ij} \approx \int_{\gamma_{ij}} (f_{ij}(s)u_{ij})\,.n_{ij}\; dv. \tag{4.34}$$

There are different schemes to evaluate the integrand in (4.34). It is common to use

first order approximation, known as upstream weighting, defined as,

$$f_{ij}(s) = \begin{cases} f_w(s_i) & \text{if } u_{ij} \cdot n_{ij} \geq 0; \\ f_w(s_j) & \text{if } u_{ij} \cdot n_{ij} < 0. \end{cases} \tag{4.35}$$

Equation (4.33) can be expressed in continuous time domain as,

$$\frac{ds}{dt} = A\, f\,(s) + \frac{1}{|\Omega|} q^{+} \tag{4.36}$$

where A is a matrix implementing $\frac{-1}{|\Omega|}[\sum_j F_{ij}(s)u_{ij} - f_w(s_i)q^{-}]$ on a cell by cell basis. In order to write the nonlinear Eq. (4.36) in the standard state space form, one needs to find the relationship between the input variable, u, to the states of the system, namely the matrix B. To this end, one can write,

$$q^{+} = B\,(s)\,u, \tag{4.37}$$

where u is the inputs to the system. In this case, u represents the production rate at producer wells or the injection rate at injectors. Here we discuss the derivation of matrix B for 5-spot pattern, however other cases can be found similarly. As there are four producers and one injector, $u \in \mathbb{R}^{5\times 1}$ and $B \in \mathbb{R}^{n\times 5}$.

In the diagonal elements of the matrix B corresponding to the producer locations, the fractional flow should be calculated by using the saturation of the same gridblock. As it is assumed that the water is injected, the diagonal element of matrix B corresponds to injector should be one. This results in the following input matrix,

$$B = \text{diag}\,(f_w(s_{\wp_1}), \cdots, f_w(s_{\wp_2}), \cdots, 1, \cdots, f_w(s_{\wp_3}), \cdots, f_w(s_{\wp_4})) \tag{4.38}$$

$$u = (q_{prod1}, \cdots, q_{prod2}, \cdots, q_{inj}, \cdots, q_{prod3}, \cdots, q_{prod4})^T \tag{4.39}$$

where $\wp_i$ for $i = 1, \cdots, 4$ corresponds to the gridblocks containing wells. The final state space representation of the saturation equation is,

$$\frac{ds}{dt} = Af(s) + B(s)u \tag{4.40}$$

4.5.2 Carleman Bilinear Approximation

As discussed in the previous section, one can use Volterra series and multi-variable Taylor expansion [99] to derive the bilinear systems of equation here.

Assuming $a(s,t) = Af(s(t))$ and $b_i(s,t) = b_i(s(t))u_i$, the saturation Eq. (4.40) is written as,

$$\frac{ds}{dt} = a(s,t) + b_1(s,t)u_1 + b_2(s,t)u_2 + \cdots \tag{4.41}$$

where b_i is the ith column of matrix B and u_i is the ith input. Using multivariate Taylor's expansion (around $s_0 = 0$) one obtains the following power series representations

$$\begin{aligned} a(s,t) &= a_0(t) + a_1(t)s^{(1)} + a_2(t)s^{(2)} + \cdots + a_N(t)s^{(N)} + \ldots \\ b(s,t) &= b_0(t) + b_1(t)s^{(1)} + \cdots + b_{N-1}(t)s^{(N-1)} + \ldots \end{aligned} \tag{4.42}$$

where $s^{(i)} = s^{\otimes i} = \overbrace{s \otimes s \otimes \cdots \otimes s}^{i \text{ times}}$ and $\otimes$ denotes the Kroncker's product. Defining new variable x as

$$x^{\otimes} = (s^{(1)}, s^{(2)}) \in \mathbb{R}^{n+n^2} \tag{4.43}$$

Assuming there is n gridblocks and $m = n_{prod} + n_{inj}$ (total number of producers and injectors) inputs, Carleman bilinearization of system (4.40) is written in the

form of,

$$\dot{x}(t) = Ax(t) + \sum_{k=1}^{m} N_k x(t) u_k(t) + Bu(t) \tag{4.44}$$

$$y(t) = Cx(t), \tag{4.45}$$

where $A, N_K \in \mathbb{R}^{(n+n^2)\times(n+n^2)}$ for $k = 1, ..., m$, and $B \in \mathbb{R}^{(n+n^2)\times m}$ and $C \in \mathbb{R}^{p\times(n+n^2)}$ are matrices derived from the bilineariztion process similar to the bilinearization of single phase flow as explained in the previous section.

Since Carleman bilinearization is based on Taylor expansion it is applicable to weakly nonlinear systems around linearization point. Also, it increases the number of states quadratically as shown in Eq. (4.43), before applying any model reduction. This can be a problem for large scale systems. As these models have high dimension, an efficient model order reduction method is required.

An alternative and more accurate approach is to reformulate the problem as an equivalent quadratic bilinear system. There is no Taylor expansion or approximation in converting original model to this form of systems [58]. This technique is also applicable to a broad range of nonlinear systems and also the size of the system increases linearly before applying model order reduction.

4.5.3 Quadratic Bilinear Formulation

In order to derive a polynomial system from a general nonlinear system, one can approximate it around its steady state solution based on Taylor series [90]. However this method is only applicable for a weakly nonlinear systems due to the locality approximation of Taylor series. Moreover, This technique may lead to instability and also can result in a system that is computationally more expensive to run. These issues preclude the applicability of Carleman bilinearization as it is based on Taylor

series.

To alleviate the problems corresponding to Taylor expansion, a class of nonlinear model order reduction were established based on trajectory piece-wise linear (TPWL) [95]. The term trajectory refers to solution path when the model is run by a given input. Along each section of the trajectory the system is linearized and also reduced. These reduced models are joined together to form the overall nonlinear reduced system. However, the evaluation of this system can be still expensive as it depends on the number of local approximation and also the order of the polynomial.

In this section, a new method is applied to nonlinear two phase flow, namely model order reduction via quadratic bilinear systems, to reduce the dimension of it. In this approach, the nonlinear equations are converted to quadratic bilinear differential algebraic equations (QBDAE). The key difference between the QBDAE and the Carleman bilinearization is that the original system is re-written in an exact equivalent representation without any approximation in QBDAE [58]. This will be explained more in a numerical example later to compare these two bilinearization techniques.

In this section, a two phase flow simulation problem is formulated as a quadratic bilinear form. The model order reduction is also applied to reduce the size of the system. This technique is general and is applicable for a wide range of reservoir simulation models with different properties. This concept was applied to the case of two-phase flow models and some preliminary results were shown in [54]. The quadratic bilinear formulation is extended to more general cases and is applied to a large heterogeneous reservoir model in this study.

4.5.4 Saturation Equation in QBDAE

We reformulate the problem by introducing new states to represents nonlinear terms. As it has been discussed in [58], for most of the nonlinear differential equations it is possible to reformulate them in quadratic bilinear differential algebraic equation (QBDAE). We derive the QBDAE of the saturation equation, as it is nonlinear.

The main nonlinearity in saturation Eq. (4.40) is the fractional flow,

$$f_w(s) = \frac{1}{1 + \lambda_o(s)/\lambda_w(s)} = \frac{1}{1 + \frac{\mu_w}{\mu_o}\frac{k_{ro}(s)}{k_{rw}(s)}} \tag{4.46}$$

If Corey's correlation is used for relative permeability, then

$$\frac{s - s_{wi}}{1 - s_{wi} - s_{or}} = \gamma s - \beta \tag{4.47}$$

$$k_{rw}(s) = (\gamma s - \beta)^N \tag{4.48}$$

$$k_{ro}(s) = (1 - (\gamma s - \beta))^N \tag{4.49}$$

Here we explain the derivation for $N = 2$, yet it can be generalized in the same manner for higher values of N. One can re-parametrize the fractional flow as,

$$x = (\gamma s - \beta)^2 \tag{4.50}$$

$$y = \frac{x}{x + \alpha(1 + x + 2(\beta - \gamma s))}. \tag{4.51}$$

where $\alpha = \mu_w/\mu_o$. After expanding the above equations and writing them in algebraic form one obtains,

$$\gamma^2 s^2 - 2\beta\gamma s + \beta^2 - x = 0 \tag{4.52}$$

$$(1 + \alpha)xy + \alpha(1 + 2\beta)y - 2\alpha\gamma s y - x = 0. \tag{4.53}$$

Also, note that one can represent $B(s)\ u$ in equation (4.40) as,

$$B(s)\ u = \sum_{i=1}^{n_{prod}} \tilde{D}_i f\,(s)\, u_i^{prod} + \sum_{j=1}^{n_{inj}} \tilde{d}_j u_j^{inj} \tag{4.54}$$

where $\tilde{D}_i$ is a zero matrix except at the diagonal element corresponding to the location of each producer and $\tilde{d}_j$ is a zero vector except at the index corresponds to the location of the injector.

Thus, the state space representation of the saturation equation with respect to newly defined variables is,

$$\begin{aligned}
\frac{ds}{dt} &= A\ y + \sum_{i=1}^{n_{prod}} \tilde{D}_i\ y\ u_i^{prod} + \sum_{j=1}^{n_{inj}} \tilde{d}_j u_j^{inj} && (4.55)\\
0 &= -2\beta\gamma s - x + \gamma^2 s^2 + \beta^2 && (4.56)\\
0 &= \alpha\,(1+2\beta)\,y - x + (1+\alpha)\,x\ y - 2\alpha\gamma s\ y && (4.57)
\end{aligned}$$

Note that this equation can still be solved based on Newton-Raphson iterations. The new residual is defined as , $R(s,y,x) =$

$$\begin{pmatrix}
(s^n - s)/\Delta t + A\ y + \sum_{i=1}^{n_{prod}} \tilde{D}_i\ y\ u_i^{prod} + \sum_{j=1}^{n_{inj}} \tilde{d}_j u_j^{inj} \\
-2\beta\gamma s - x + \gamma^2 s^2 + \beta^2 \\
\alpha\,(1+2\beta)\,y - x + (1+\alpha)\,x\ y - 2\alpha\gamma s\ y
\end{pmatrix}$$

and the Jacobian of the right hand side of this equation is, $J(s,y,x) =$

$$\begin{pmatrix}
-\mathbf{I}/\Delta t & A + \sum_{i=1}^{n_{prod}} D_i u_i & 0 \\
-2\beta\gamma\mathbf{I} + 2\gamma^2 s & 0 & -\mathbf{I} \\
-2\alpha\gamma\mathrm{diag}(y) & \alpha\,(1+2\beta)\,\mathbf{I} + (1+\alpha)\,\mathrm{diag}(x) - 2\alpha\gamma\mathrm{diag}(s) & -\mathbf{I} + (1+\alpha)\,\mathrm{diag}(y)
\end{pmatrix}$$

Assuming that the number of grid blocks is n, one can define a new state variable as,

$$x_q = \left(s^T, y^T, x^T\right)^T \in \mathbb{R}^{3n}, \tag{4.58}$$

and write the saturation equations of (4.55) to (4.57) in the form of quadratic bilinear differential algebraic equation (QBDAE) as follows,

$$E\frac{d}{dt}x_q = G_1 x_q + G_2 x_q \otimes x_q + \sum_{i=1}^{n_{prod}} D_i \; x_q \; u_i^{prod} + \sum_{j=1}^{n_{inj}} d_j u_j^{inj} \tag{4.59}$$

where

$$E = \begin{pmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad G_1 = \begin{pmatrix} 0 & A & 0 \\ -2\beta\gamma \; I & 0 & -I \\ 0 & \alpha\,(1+2\beta) \; I & -I \end{pmatrix},$$

$$G_2 = \begin{pmatrix} Z \\ \gamma^2 \; I_{s2} \\ -2\alpha\gamma \; I_{sy} + (1+\alpha) \; I_{yx} \end{pmatrix},$$

$$D_i = \begin{pmatrix} 0 & \tilde{D}_i & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad d_i = \begin{pmatrix} \tilde{d}_i \\ 0 \\ 0 \end{pmatrix}.$$

Note that all the matrices in G_2 are calculated based on Algorithm (5).

Despite we derived the QBDAE for specific mobility function, this procedure can be extended to higher polynomial functions as well. It can be shown that the number of states increases linearly compared to Carleman bilinearization that is quadratic. QBDAE formulations of a system is not unique, but they are all equivalent, meaning that the solution of the QBDAE systems are exactly equal to the original system

Algorithm 5 Compute matrix G_2 in MATLAB for QBDAE

procedure $G_2(n)$
  $Z = spalloc(n, 9n^2, 0)$
  $e = speye(1, 3n)$
  for $i = 1 : n$ **do**
    $I_{s2}(i,:) \leftarrow kron(circshift(e,[1, i-1]), circshift(e,[1, i-1]))$
    $I_{sy}(i,:) \leftarrow kron(circshift(e,[1, i-1]), circshift(e,[1, i-1+n]))$
    $I_{yx}(i,:) \leftarrow kron(circshift(e,[1, i-1+n]), circshift(e,[1, i-1+2n]))$
  end for
  return $Z, I_{s2}, I_{sy}, I_{yx}$
end procedure

if the initial conditions are consistent to each other [58]. Thus there is no approximation involved in this step, whereas Carleman bilinearization is a Taylor series approximation. These two approaches to convert a nonlinear system into bilinear form are compared in Figure 4.4.

In the next section, our goal is to present a reduced-order model for both pressure and saturation equations and reduce the computation runtime.

4.5.5 Numerical Example: Comparison of Carleman Bilinear Approximation and QBDAE Formulation

In this section the Carleman bilinear approximation and quadratic bilinear formulation are both applied on a simple two phase flow and the results are compared. A quarter 5-spot two-phase flow (oil-water) reservoir model in two dimensions is considered. It is assumed that the reservoir is under water flooding recovery process. It is assumed that the fluid viscosity of oil is ten times higher than water, and the porosity is 0.2 for the entire reservoir. The heterogeneous permeability and the relative permeability are depicted in Figure 4.5. The reservoir is 450x450x30 (ft) that is discretized by the finite volume method using Cartesian grid of size 15x15x1. The producers are controlled by constant bottom hole pressure of 2500 [psi] and the

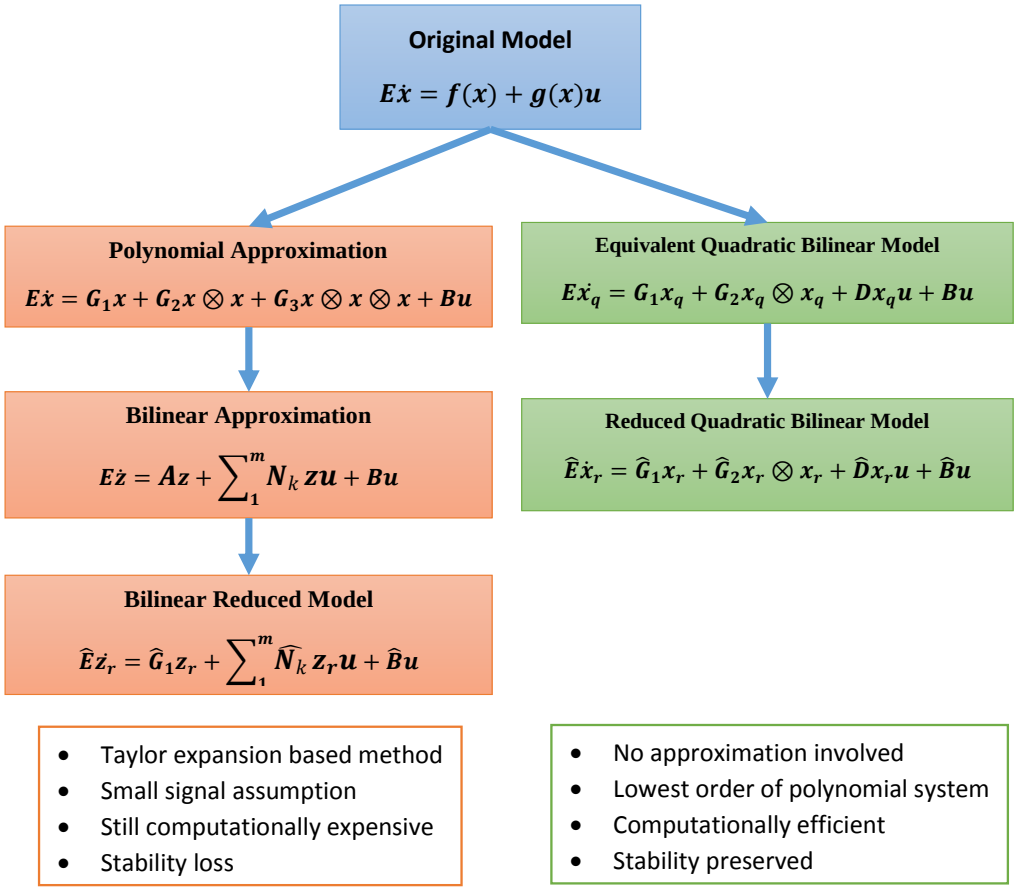
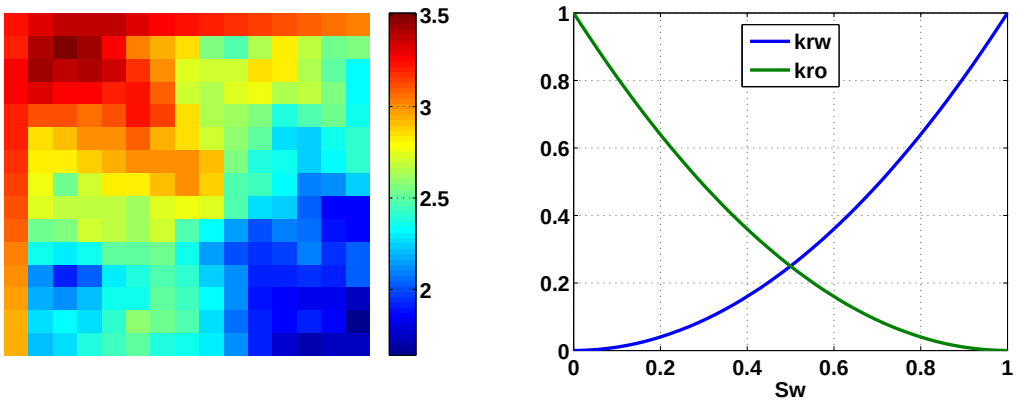


Figure 4.4: Comparing Carleman bilinear approximation and equivalent quadratic bilinear formulation (referred as QBDAE in this chapter)



(a) permeability (log(md)) (b) relative permeability

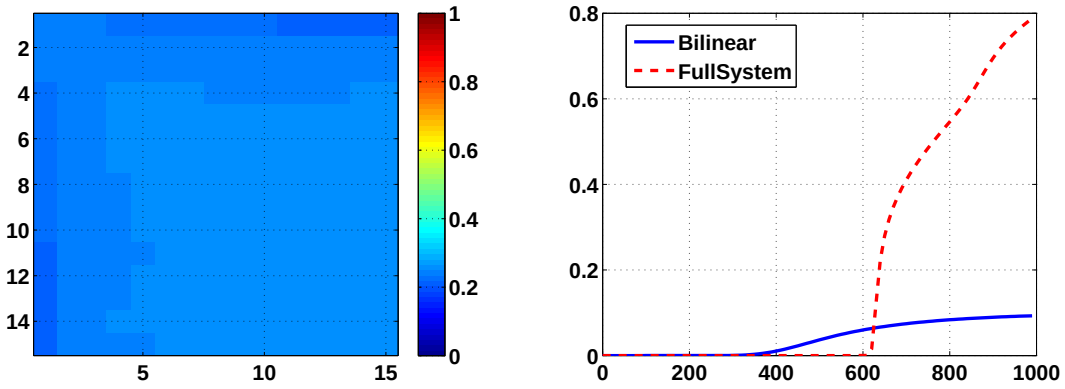
Figure 4.5: 15x15 quarter spot model

injection rate is constant at one PVI. The initial water saturation and pressure are assumed to be 0 and 3000 psia respectively.

The reservoir model is simulated for 1000 days. The final water saturation and water cut (ratio of produced water over total produced fluid) for Carleman bilinear approximation and QLDAE formulation are shown in Figure 4.6 and Figure 4.7, respectively. These results are compared with exact solution and it reveals that Taylor expansion of fractional flow could not capture dynamic of the model. On the contrary, there is no approximation in QLDAE and its solution exactly matching the original system.

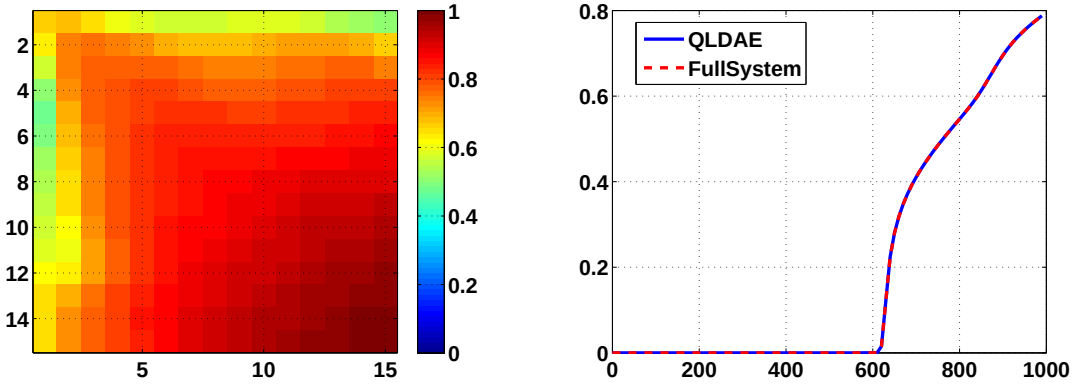
4.5.6 Model Order Reduction of Saturation Equation Formulated as QBDAE

In this section, we introduce the concept of model order reduction for QBDAE systems. The idea is to construct the projector $P = VW^T$ where $V, W \in \mathbb{R}^{n\times r}$ with $W^TV = \mathbf{I}_r$, such that $r \ll n$ and the reduced-order model can be obtained by



(a) final water saturation (b) water cut

Figure 4.6: Carleman bilinear approximation



(a) final water saturation (b) water cut

Figure 4.7: Quadratic linear/bilinear formulation (QLDAE)

projecting the state-space equations into a much smaller subspace by

$$x_q = V\ x_r \tag{4.60}$$

The first step in model reduction framework is to construct the projection matrix. One can apply iterative rational Krylov method to bilinear systems to find the projection matrix [17]. However, the application of the existing algorithms to QBDAE systems is hindered as they are only suitable for single input output, affine systems. In this study the projection matrix for QBDAE systems is found based on POD method, meaning that the model is run with training inputs and the saved snapshots are used to find the basis. This will show the applicability of the new formulation and the possibility of reducing the dimension of the model. The model order reduction based on the Krylov subspace without the need for training and snapshots need to be address in the future work.

To obtain the projection matrix based on POD approach, one needs to run the model with a training inputs and save the snapshots of saturation, saturation squared and fractional flow. After applying *svd* on each snapshot matrix, the projection basis are obtained and can be used to project the original high dimensional state to a reduced one as Eq. 4.60. The matrices in Eq. 4.59 are used to construct a reduced QBDAE form of saturation equation.

$$\hat{E}\frac{d}{dt}x_r = \hat{G}_1 x_r + \hat{G}_2 x_r \otimes x_r + \sum_{i=1}^{n_{prod}} \hat{D}_i\ x_r\ u_i^{prod} + \sum_{j=1}^{n_{inj}} \hat{d}_j u_j^{inj} \tag{4.61}$$

where

$$\hat{E} = V^T E V \tag{4.62}$$

$$\hat{G}_1 = V^T G_1 V, \quad \hat{G}_2 = V^T G_2 V \otimes V \tag{4.63}$$

$$\hat{D}_i = V^T D_i V, \quad \hat{d}_j = V^T d_j \tag{4.64}$$

Note that the resulted reduced system in Eq. 4.61 is also a QBDAE and preserves some important features of the original system, such as stability.

4.6 Numerical Results

In this section we apply the model order reduction methodologies for a two-phase flow (oil-water) reservoir model under the water flooding recovery process with the structure of a 5-spot. Here we have an injector in the center of the reservoir and four producers in the corners. The reservoir is SPE10 comparative model [31] (layer 10th). This model is synthetic but can be representative of a real reservoir with large heterogeneity.

This reservoir model is discretized using Cartesian grid of size $20ft \times 10ft \times 2ft$. Overall the reservoir model has $60 \times 220 \times 1 = 13200$ active cells. The fluid viscosity ratio is $\mu_w/\mu_o = 0.1$. The absolute heterogeneous permeability and the relative permeability curves are depicted in Figures. 4.8a and 4.8b, respectively. The porosity is assumed to be constant at 0.2 for entire model.

For the training schedule, the producers are controlled by bottom hole pressure and the injector by injection rate as shown in Figure 4.9. Note that this amount of injection was selected to ensure at least one pore volume injection throughout simulation time (1000 days). The initial water saturation is assumed to be 0.0 .

In order to apply the POD on QBDAE, we simulated the reservoir for 1000

	Fine Scale (QBDAE)	QBDAE+POD
# pressure basis	13200	3
# velocity basis	26120	24
# saturation basis	13200	35
# saturation squared basis	13200	78
# fractional flow basis	13200	78

Table 4.1: Compare fine and reduced scale quadratic bilinear model

days and saved the snapshots of pressure (Po), velocity (Vel), water saturation (Sw) and its squared and fractional flow function (fw) every 5 days. Thus, we have 200 snapshots for each variables. Each snapshot is reshaped to a column vector and is stacked in a snapshot matrix. After applying *svd* to each matrix, one can find the basis as explained in previous sections. The singular values of the snapshot matrices are shown in Figure. 4.10. As can be seen, there is a faster decay in the singular values for the pressure and velocity compared to saturation and its related functions. Thus, we need more basis for the saturation and the nonlinear functions to capture most of the energy and have small error.

The selection criteria here was to capture at least 99% of the energy of snapshots. The number of basis is compared for reduced model to the original fine scale one in Table 4.1. It is obvious that several orders of magnitude in model order reduction is obtained in this example.

The final water saturation and the water cut for all the producers after 1000 days of simulation is shown in Figure 4.11 and compared with fine scale model. Note that the water cut resulted from high fidelity model (dashed line) and the reduced model (solid line) are almost identical.

Figures 4.12a and 4.12b show the spatial relative error at the final time in the saturation and pressure, respectively. The error in both pressure and saturation

are $\mathcal{O}(10^{-1})$. In Figure 4.13, the temporal variation of the relative error is plotted between the fully-resolved and approximate solutions, as defined in the following equation,

$$E(t^k) = \|x(t^k) - \tilde{x}(t^k)\|_2 / \|x(t^k)\|_2, \quad t^k \in [0\ T_f] \tag{4.65}$$

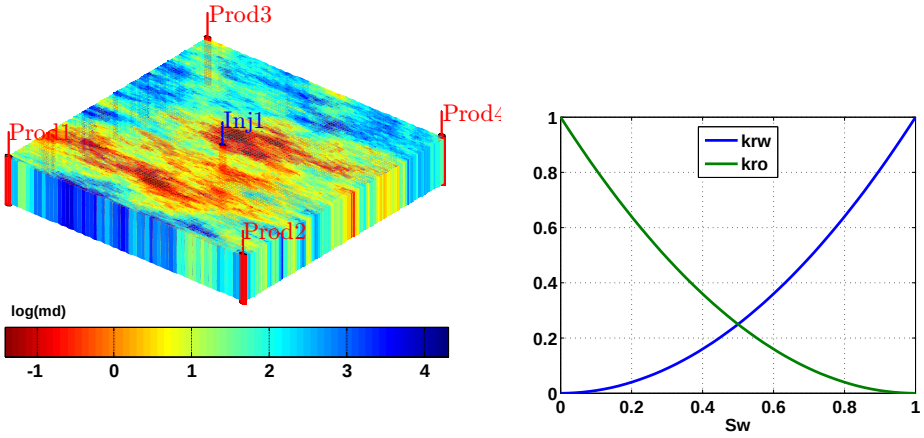
where x and $\tilde{x}$ are the saturation/pressure results from the high fidelity model and the reduced model, respectively and T_f is the final time of simulation. The error is less than 3% for most of the simulation time.

We run the reduced model with a new test schedule as shown in Figure 4.14, to make sure that the reduced QBDAE based on POD is robust to input variation. This schedule is obtained by $\pm 10\%$ random perturbation of the training schedule. Note that the basis of the reduced model are not updated and we used the same basis obtained from training snapshots.

The final water saturation and the water cut for all the producers after 1000 days of simulation with the Test schedule is shown in Figure 4.15 and compared with fine scale model. Again it is obvious that the reduced model replicates almost exactly the same water cut as high fidelity model. Figures 4.17a and 4.17b show the spatial relative error at the final time in the saturation and pressure, respectively. The errors are still $\mathcal{O}(10^{-1})$ in both pressure and saturation. In Figure 4.16, the temporal variation of the relative error is plotted between the fully-resolved and approximate solutions, and the error is less than 5% for most of the simulation time.

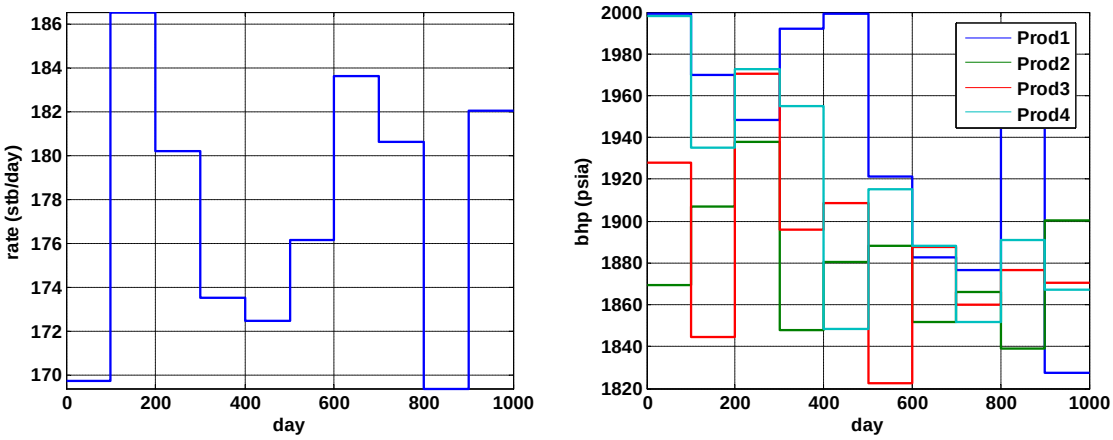
4.7 Chapter Summary

In this chapter the Carleman bilinear approximation of a single phase and two phase flow were discussed. The result for single phase flow was acceptable and the Krylov based model reduction without any training could successfully reduce the dimension of the model. However, this approach could not approximate fractional



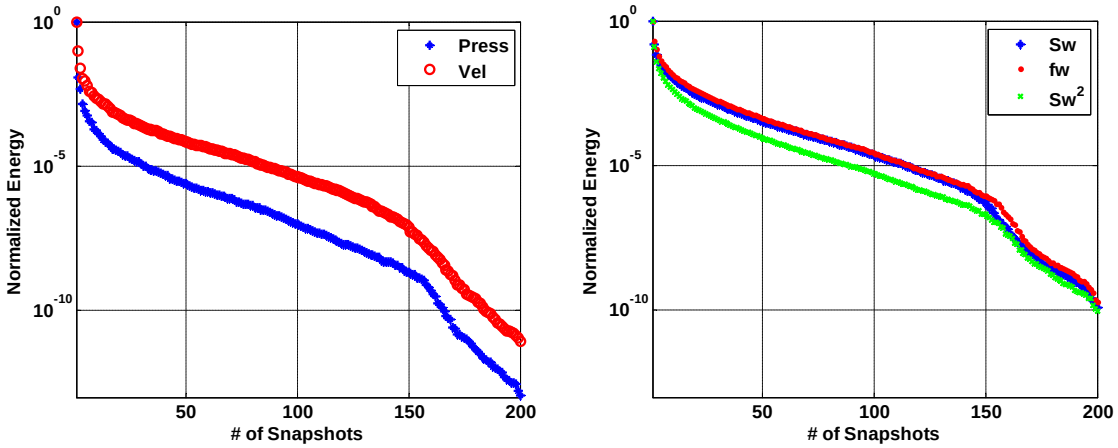
(a) permeability (b) relative perm

Figure 4.8: Layer 10th of SPE10



(a) injection rate (b) producers BHP

Figure 4.9: Training schedule for applying POD on QBDAE



(a) pressure and velocity (b) saturation, its squared and fractional flow

Figure 4.10: Singular values of snapshot matrices of QBDAE's states

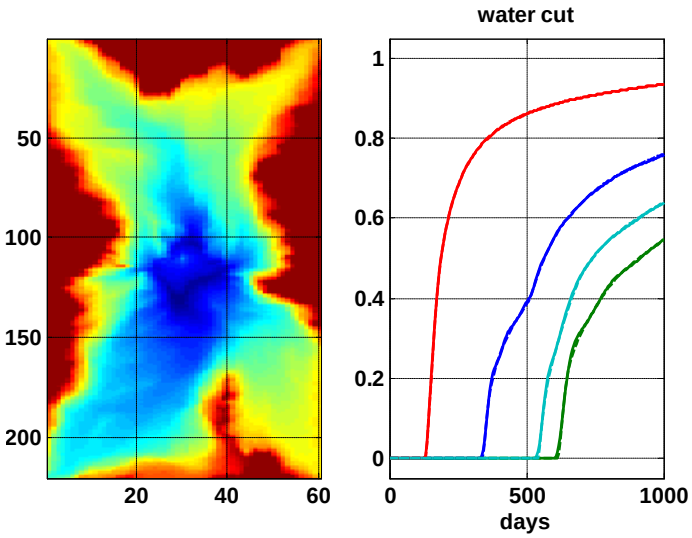
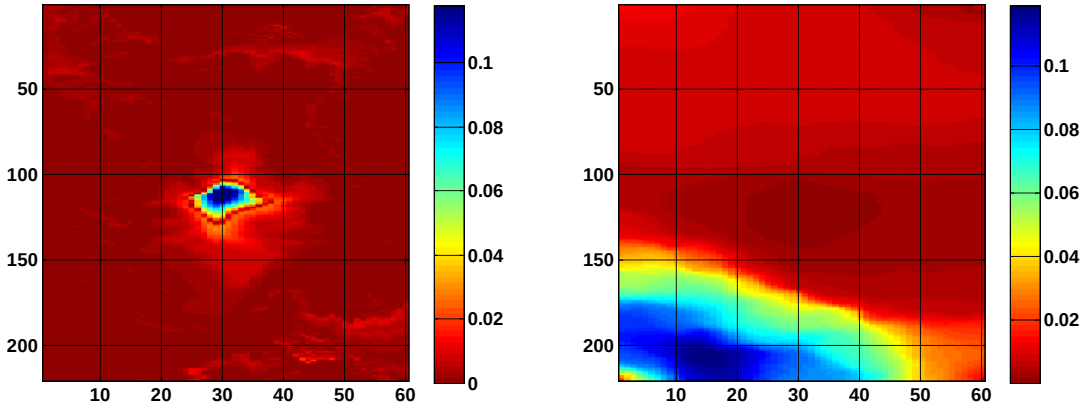


Figure 4.11: Final saturation and water cut at producer resulted from high fidelity model (dashed line) and reduced model (solid line). Both models have very similar results for training schedule



(a) Final saturation error

(b) Final pressure error

Figure 4.12: Final relative error of applying POD on QBDAE for training schedule

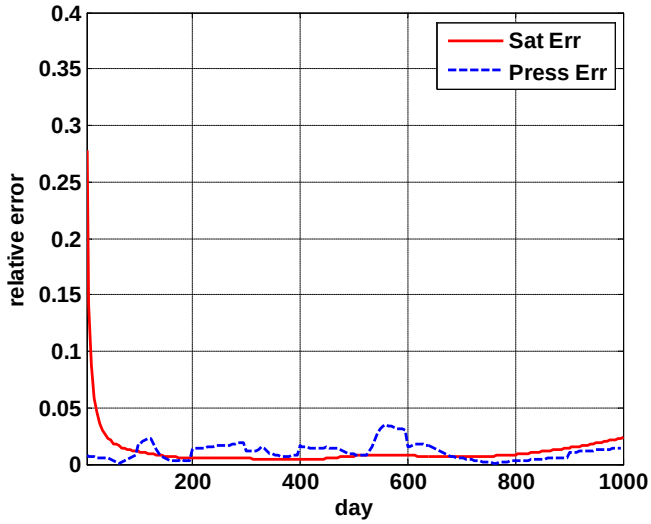
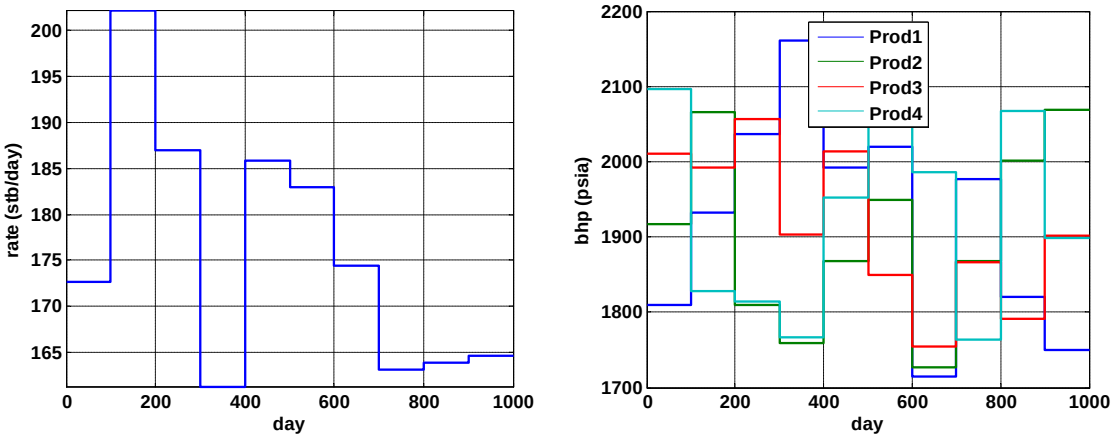


Figure 4.13: Temporal variation of saturation and pressure error of applying POD on QBDAE for training schedule



(a) injection rate (b) producers BHP

Figure 4.14: Test schedule, obtained from ±10% random variation in the training schedule

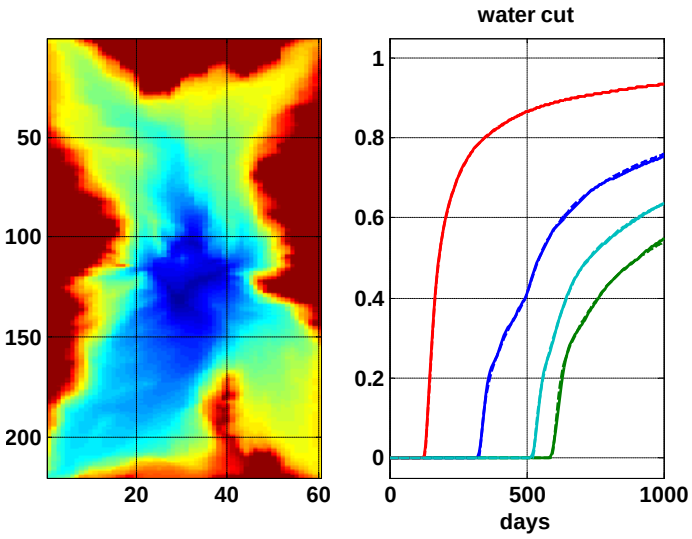


Figure 4.15: Final saturation and water cut at producers for both high fidelity model (dashed line) and reduced model (solid line). Both models have very similar results for test schedule

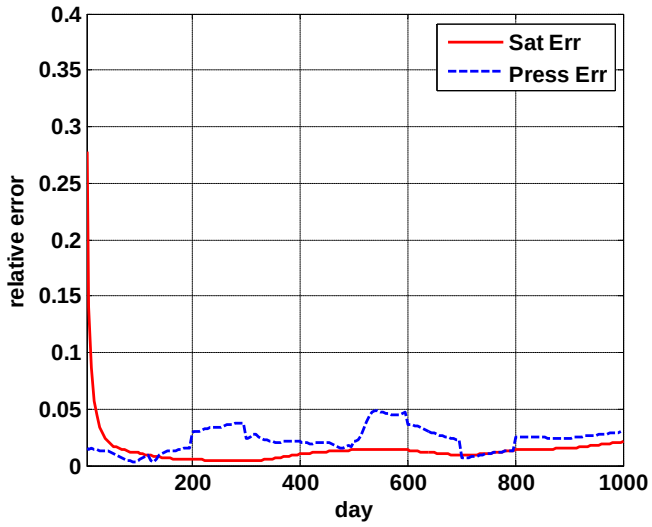
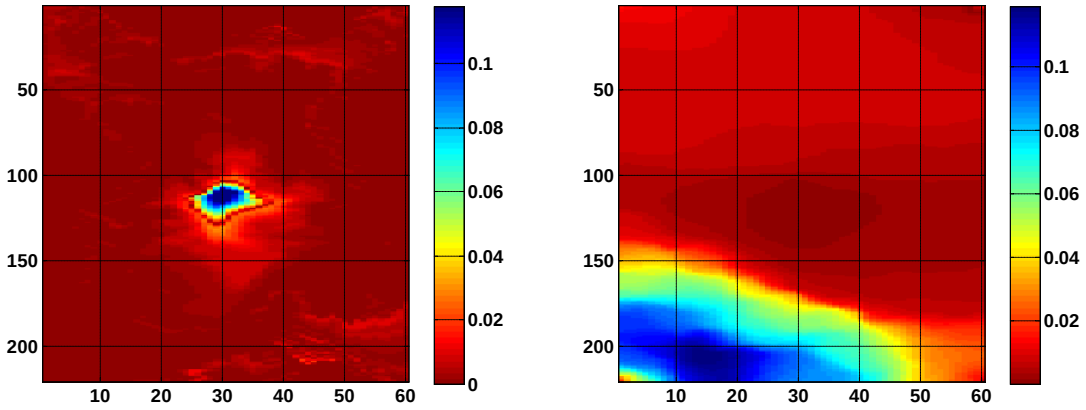


Figure 4.16: Temporal variation of saturation and pressure error of applying POD on QBDAE for the test schedule



(a) Final saturation error

(b) Final pressure error

Figure 4.17: Final relative error of applying POD on QBDAE for test schedule

flow with Taylor expansion, and two phase flow results were wrong.

Thus, new approach based on quadratic bilinear differential algebraic equation (QBDAE) formulation were applied to two phase flow to transform it into quadratic bilinear system. This method introduces new variable for some nonlinear terms and there is no approximation in this transformation. POD model order reduction was applied to reduce the dimension of this new system and the results were acceptable.

In the next chapter, the gradient based methods are explained in the context of production optimization. Specifically, it will be discussed how to find the gradient efficiently based on the adjoint formulation. Finally, we demonstrate the applicability of POD in a production optimization problem by integrating it into the optimization workflow.

5. OPTIMIZING RESERVOIR PERFORMANCE USING REDUCED ORDER MODELS

The development of an oil field involves many challenges, e.g. maximizing profit and oil recovery factor while reducing risks and environmental footprints. Although the weight put into some of these issues depend upon the life stage of an oil field, and can be non-existent in many circumstances, the objective of any project development is certainly related to profits and risks.

The management of an oil field can be tackled by finding the optimal production strategy (or strategies during the reservoir life cycle) by means of dynamically adjusting the production flow rates or bottom-hole pressures over the reservoir production time. This optimization is tied to some form of an economical objective function, very often indicated by the net present value (NPV) of the operations. As it is known, realistic field production is not set as the operator will, but it is driven by operations constraints defined by the capacity of the platform and the surface facilities required to separate, process and store (or drain) all production (oil, water and gas). A major difficulty occurs in optimizing production of all wells, according to these constraints along the production of a field.

For this reason, designing optimal well control strategies taking into account realistic production constraints and well allocation rates is of paramount for the increase in reservoir recovery factor and, in turn, gaining the maximum NPV of the project.

The authors in [79] described the E-field production optimization system (EFOS) and its application to the Prudhoe Bay oil field in Alaska. EFOS facilitated the short term field optimization. For long term operation, well inflow performances were used

instead of using full reservoir model. However, one can formulate the problem as a classical optimal control and use calculus of variation to solve the problem. The authors in [20] formulate the optimization problem by considering the reservoir model as a forward simulation model. They also integrated POD model order reduction into the optimization for reducing the computational cost.

The hierarchal optimization that deal with different time scale were suggested in [102]. In this workflow the proper realtime system identification, optimization, and feedback control at various levels of oil field were discussed. Many studies published in the literature [110, 53] emphasize the need and importance of performing closed-loop optimization using optimal control theory.

One of the benchmarks developed for the optimization under the uncertain paradigm is the Norne field model [100]. Different optimization were applied on this benchmark and the results are improved in terms of the economic gains and reducing uncertainties in the field.

This chapter addresses the optimization of well schedules under bound constraints. This workflow is combined with model order reduction for reducing the computational costs associated with running reservoir models [101]. Finally, this process is utilized on a new reservoir benchmark model called UNISIM-I-D [45].

5.1 Problem Statement

As it was explained in Chapter 2, the pressure and saturation at each time step are found sequentially, meaning that the pressure is solved via mixed finite element method formulation of 2.14 and the saturation is derived from Newton Raphson iterations of 2.16. These equations can be written as an implicit function,

$$g(x^n, x^{n-1}, u^n) = 0 \tag{5.1}$$

where x^n is the dynamic states such as pressure, saturation, compositional, etc. u^n is the input control such as bottom hole pressure in producers and water or gas injection rate in injectors.

The objective function is to maximize net present value (NPV) in the water flooding process of a reservoir. It generally comprises cost of water injection and production, and revenues of oil production. The objective function is defined as,

$$\text{NPV} = \sum_{n=1}^{N} \left\{ \sum_{i=1}^{N_{\text{prod}}} [r_o(q_o)_i^n + r_w(q_w)_i^n] + \sum_{i=1}^{N_{\text{inj}}} \hat{r}_w(q_w)_i^n \right\} \frac{\Delta t^n}{(1+d)^{t_n}} \tag{5.2}$$

where N_{inj} and N_{prod} are the total number of injectors and producers, respectively; q_w and q_o are the water and oil production rate, respectively; r_o is the revenue of produced oil, r_w and $\hat{r}_w$ are the cost of water production and injection, respectively; d is the discount factor.

For the wells that are BHP controlled, the NPV becomes a function of states (pressure and saturation) and inputs (bottom hole pressure) of the system. Furthermore, for the rate controlled wells, the rate is assumed to be independent of pressure and saturation of the reservoir. The cost function can be written in general form as,

$$J = J_F(x^N) + \sum_{n=1}^{N} L^n(x^n, u^n). \tag{5.3}$$

where J is the cost function which consists of two major terms. The first term, $J_F(x^N)$ that can be the abandonment cost and the second term, that is the objective function, e.g. NPV, over all control steps.

The production optimization problem addresses finding a sequences of control inputs u^n to get the maximum cumulative oil production, NPV, recovery factor, under different constraints. Note that the reservoir simulation model, namely forward

model as in Eq. 5.1, can also be considered as an equality constraint. Thus, the optimization problem can be described in general form as,

$$\max_{u^n} J = J_F(x^N) + \sum_{n=1}^{N} L\left(x^n, u^n\right) \tag{5.4}$$

$$g^n\left(x^n, x^{n-1}, u^n\right) = 0, \quad \forall n \in \{0, ..., N\} \tag{5.5}$$

$$x^0 = x_0, \quad \text{initial condition} \tag{5.6}$$

$$c\left(x^n, u^n\right) \leq 0, \quad \text{inequality constraints} \tag{5.7}$$

In this optimization problem, the equalities and inequalities implicate the limitations on each well. There are also physical and practical constraints in each production optimization problem that should be considered to get a reasonable results. For example, the production rate might be limited to avoid sand production and keep wellbore stability, or the injection rate might be limited due to surface facilities or avoiding large seismic events. These constraints will be considered as inequality or equality in each problem.

There are different techniques to solve each optimization problem, e.g. gradient based methods, direct search, and heuristic approaches. Here we review them briefly and choose the one that is more appropriate for our purpose.

Direct search methods are best known as optimizations techniques that do not explicitly use gradient of the objective function to find the optimal solution. These methods can be usually integrated with parallel computing with small modifications. A direct search algorithm searches a set of points at each iteration, looking for a point where the value of the objective function is lower than the current value. This method can be used for optimizing objective functions that are not differentiable. Different direct search methods for unconstrained optimization are discussed in [77].

Heuristic optimization techniques are designed for solving a problem faster when classical methods are too slow, or finding the approximate solution where the other classical optimization methods fail to find the exact solution. Often times these methods can be combined with classical optimization techniques to improve their performances. One can refer to [92, 62] for comparing some of these methods.

In this thesis, the gradient based optimization is considered because the objective function is smooth and also this method showed faster convergence. An efficient method to find the gradient is to solve the adjoint model [20, 103]. This model is run to find the Lagrange multipliers and consequently the gradient. Next, this method will be discussed more.

5.1.1 Gradient with Adjoint Method

In gradient based optimization, adjoint methods are widely used for calculating the gradient in a problems with large number of variable parameters. The first step is to solve the forward model, e.g. reservoir simulator, for all time steps with given initial conditions and control inputs. At this stage all the dynamic states, e.g. pressure and saturation should be stored for all the time steps. The cost function should be evaluated with results of the forward model simulation as for example NPV in equation 5.2.

The necessary optimality conditions are derived from classical theory of calculus of variation. One can construct the augmented cost function by multiplying the constraints with a set of coefficient known as Lagrange multipliers and add them to the main objective function as follows (for simplicity the inequality constraints are not considered),

$$J_A = J_F(x^N) + \sum_{n=1}^{N} L^n\left(x^n, u^n\right) + \left(\lambda^0\right)^T\left(x^0 - x_0\right) + \sum_{n=1}^{N}\left(\lambda^n\right)^T g^n\left(x^n, x^{n-1}, u^n\right) \quad (5.8)$$

where the inner product of the Lagrange multipliers and the forward model can be expanded as follow (see [74]),

$$\begin{aligned}
(\lambda^n)^T g^n\left(x^n, x^{n-1}, u^n\right) &= (\lambda_u{}^n)^T\left(B^n u^n - C p^n\right) && (5.9)\\
&+ (\lambda_p{}^n)^T\left(C^n u^n\right) && (5.10)\\
&+ (\lambda_s{}^n)^T\left(s^n - s^{n-1} - F^n\right) && (5.11)
\end{aligned}$$

where $F^n = \Delta t^n {D_{pv}}^{-1}\left(A(u^n) f_w(s^n) + q^+\right)$. The first optimality condition imposes $\nabla_x J_A = 0$, which results in following linear equations,

$$\begin{aligned}
\left(\lambda^N\right)^T &= -\left(\frac{\partial J_F}{\partial x^N} + \frac{\partial L^N}{\partial x^N}\right)\left(\frac{\partial g^N}{\partial x^N}\right)^{-1} && (5.12)\\
\left(\lambda^{n-1}\right)^T &= -\left((\lambda^n)^T \frac{\partial g^n}{\partial x^{n-1}} + \frac{\partial L^{n-1}}{\partial x^{n-1}}\right)\left(\frac{\partial g^{n-1}}{\partial x^{n-1}}\right)^{-1} && (5.13)\\
\left(\lambda^0\right)^T &= -\left(\lambda^1\right)^T \frac{\partial g^1}{\partial x^0} && (5.14)
\end{aligned}$$

The final condition for Lagrange coefficients are found based on Eq. 5.12. The adjoint model can be solved using the stored states to calculated the Lagrange multipliers as Eq. 5.13. Equation 5.13 is linear and is solved backward in time for obtaining Lagrange multipliers. The initial Lagrange multiplier can also be found as Eq. 5.14

The coefficients corresponding to pressure equations are found at each time step $n = N, \cdots, 1$ as follows,

$$\begin{pmatrix} B(s^{n-1}) & -C^T \\ C & 0 \end{pmatrix}\begin{pmatrix} \lambda_v{}^n \\ \lambda_p{}^n \end{pmatrix} = \begin{pmatrix} \left(\frac{\partial g^n}{\partial v^n}\right)^T \lambda_s{}^n - \left(\frac{\partial L^n}{\partial v^n}\right)^T \\ 0 \end{pmatrix} \qquad (5.15)$$

and the Lagrange coefficients corresponding to saturation equation is calculated as

follow,

$$\left(I - \left(\frac{\partial g^n}{\partial s^n}\right)^T\right) {\lambda_s}^n = {\lambda_s}^{n+1} - \left(\frac{\partial L^n}{\partial s^n}\right)^T \tag{5.16}$$

$$- \left(\frac{\partial}{\partial s^n}\left(B^{n+1}u^{n+1}\right)\right)^T {\lambda_v}^{n+1} \tag{5.17}$$

One can refer to [74] for the derivation and more discussion on adjoint formulation in a sequential mixed finite volume problem.

Finally, the gradient of the cost function with respect to input parameters is,

$$\frac{\partial J}{\partial u^n} = \frac{\partial J_A}{\partial u^n} = \frac{\partial L^n}{\partial u^n} + \left(\lambda^n\right)^T \frac{\partial g^n}{\partial u^n} \tag{5.18}$$

Thus, the gradients will be calculated using the Lagrange multipliers and forward model results as in Eq. 5.18. If the gradient is sufficiently close to zero (smaller than a tolerance), the optimality has been satisfied, otherwise this gradient can be used in an iterative method to find optimal direction to update the control inputs.

5.1.2 Optimization Process

Optimizing the well placement and production schedule are one of the main problems in the development of a new field. These problems should be addressed based on engineering analysis and mathematical optimization algorithm to achieve a good solution. In this thesis, I focus on the production optimization, meaning that the location of the wells are fixed and the question is how to gain the most profit by finding the optimal schedule for each well.

The optimization workflow that is used in this thesis is based on line search. In each iteration of the line search method, a descent direction is computed and then it

is determined how far to move along that direction. The iteration is,

$$u_{k+1} = u_k + \alpha_k \Delta_k \tag{5.19}$$

where the positive scalar α is called the step length. The effective choice of both the direction and the step length guarantees successful convergence in the line-search. The search direction usually has the form,

$$\Delta^k = -{B_k}^{-1} \nabla_u J^k \tag{5.20}$$

This forms guarantees that we have a descent direction if matrix B_k is positive definite. In the steepest descent method the direction is along the gradient of the cost function, and B_k is simply identity matrix, whereas in the Newton's method it is the exact Hessian $B_k = \nabla^2 J$. In quasi-Newton methods, B_k is an approximation to the Hessian that is updated at every iteration by means of a low-rank formula such as BFGS [85].

In this thesis I only consider the steepest descent method for optimization. The forward model combined with adjoint one are used to find the gradient, followed by aggressive line-search to find the steepest growth/decay of the cost function in the direction of the gradient. This workflow is depicted in figure 5.1. The bottleneck of this workflow is that running the reservoir model is computationally expensive and it needs to be run many times in the optimization process. The solution used here is to replace the original large scale model in the inner iteration loop with the reduced model. This inner iteration is used to find the step size that gives the steepest growth/decay in the cost function.

In this approach, after running the forward one can use the snapshots of the

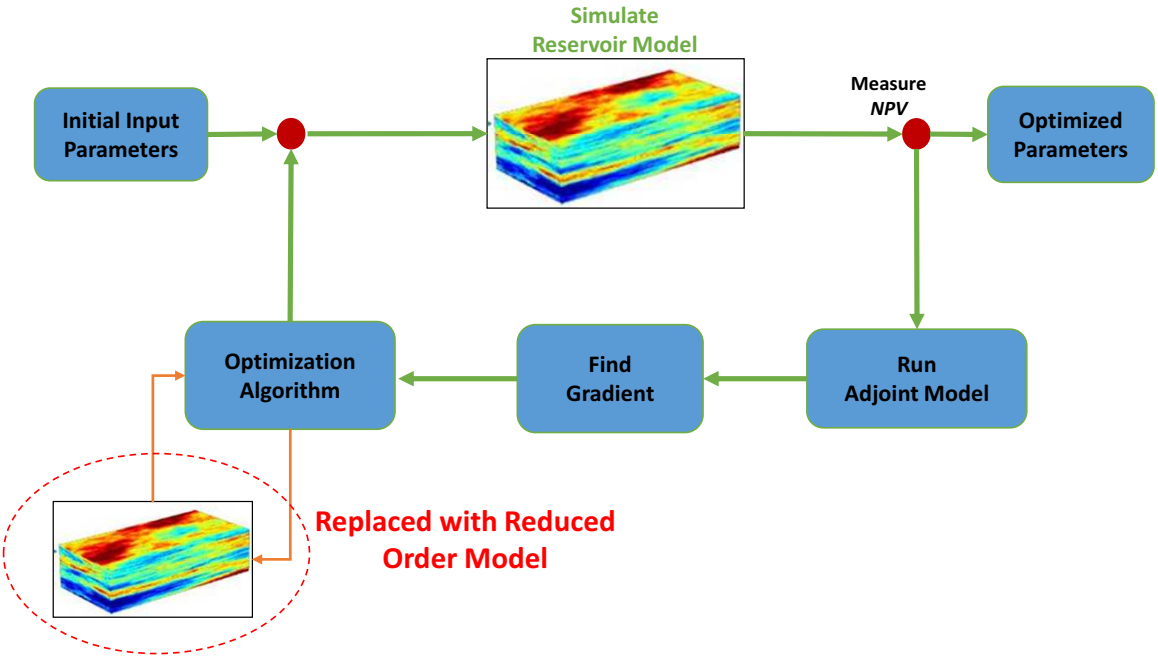
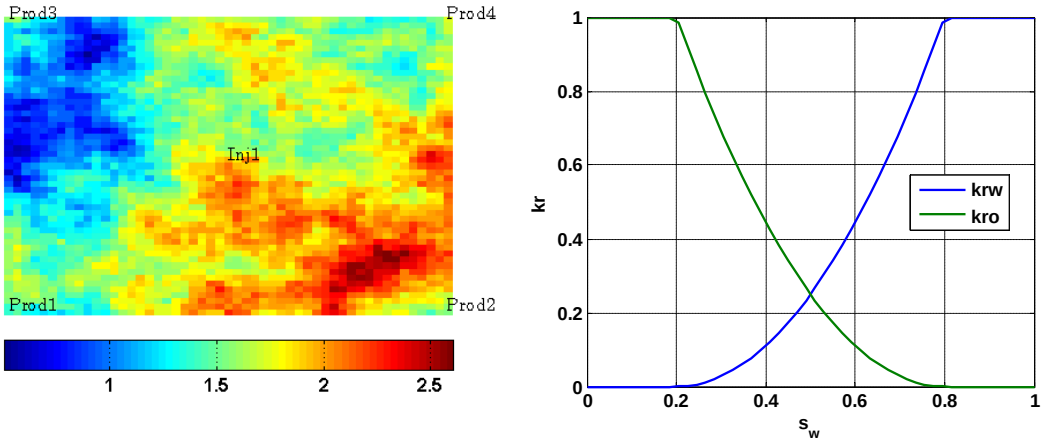


Figure 5.1: Gradient based optimization workflow combined with reduced order modeling

solution to compute the projection basis in order to construct the reduced model. Also, the solution snapshots of the forward model is used in the adjoint model to calculate the gradient.

In this approach the reduced model will be updated at every outer iteration loop. This will guarantee that the reduced model is trained throughout the optimization. Also in this study, it is assumed that the reduced model is valid within 10% of variation in the original schedule, used to obtain snapshots. However one can apply trust region method to change this range at each iteration based on the results of the reduced model and comparing it with high fidelity results [108].

In the next section, we apply the proposed optimization workflow on waterflooding the reservoir. All the numerical examples were run in MATLAB Reservoir Simulation Toolbox (MRST, see [1]). Note that the code was modified for running reduced



(a) permeability (log(mD)) (b) relative permeability

Figure 5.2: Heterogeneous model with 5-spot pattern

order models.

5.2 Numerical Example

The reservoir model in this example consists of 51x51x1 gridblocks, where the dimension of each grid is 10ft x 10ft x 10ft. The residual saturation is 0.2 for both water and oil. The porosity is constant and is set to 0.2, whereas the permeability is heterogeneous as shown in Figure 5.2a. The viscosity ratio is set to 5 ($\mu_o/\mu_w = 5$), and the initial water saturation is equal to the residual water saturation ($s_w(t = 0) = 0.2$). The well configuration is a 5-spot pattern with an injector in the middle and four producers at the corners.

It is assumed that the reservoir is dead black oil above bubble point pressure and it is saturated with oil and water. The Corey's model of the relative permeability curves as depicted in Figure 5.2b is used for this example.

This problem is solved based on adjoint formulation. At each optimization iteration, the forward model is run and then the adjoint model will be run to get the gradient of the NPV with respect to optimization parameters (bottom-hole pressure

and injection rate). This gradient is used to update the initial guess for parameters, until the optimal values are found, meaning that the gradient is close to zero or the relative change in NPV is smaller than a tolerance.

The optimization is done for time horizon of 1000 days. It is assumed that all the wells are BHP controlled and for each well it can be changed every 10 days. Thus, the number of control variables are $5 \times 100 = 500$. The BHP for the injector is constrained between 2000 psia to 3000 psia and for the producers between 1000 psia to 2000 psia. The initial BHP for the injector is set to 2500 psia and for all the producers is set to 1500 psia. The revenue of oil production is considered to be \$70 per barrel and the cost of injection or production of water is \$20 per barrel for each.

Moreover, the snapshots of the solution, pressure, saturation, and velocity are saved and used to find the basis for the reduced order model. Overall there are 100 snapshots for each state solution. These basis are updated at each outer loop iteration and are used to construct the reduced model for inner loop iterations. These iterations are to find the step size that results in the steepest increase in the NPV function.

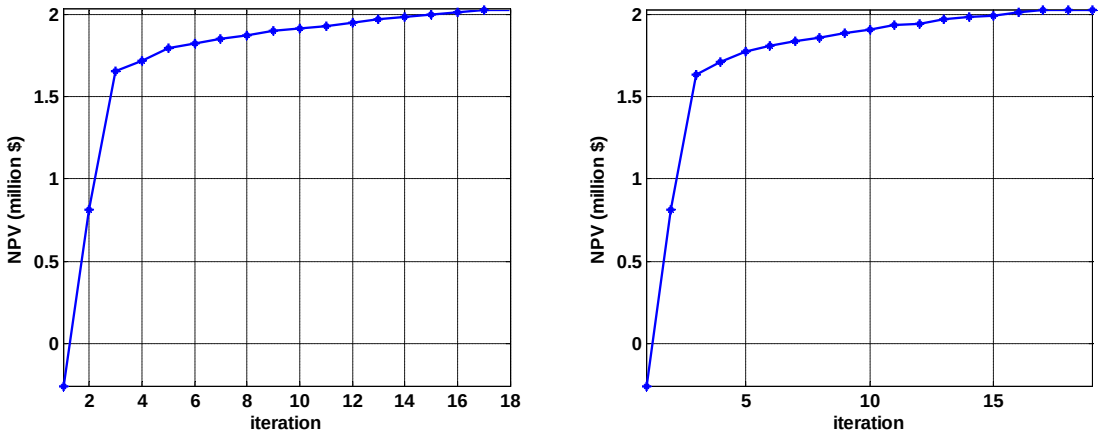
Since this is a relatively small model, the simulation time for fine scale model and the reduced one is close to each other. However, the goal in this example was to show that very similar results can be obtained with using the reduced model as a proxy model for the forward simulations. The optimization results are also compared in Table 5.1.

The optimization results are compared between two workflows, using the high fidelity model in the inner loop iterations or using the reduced order model instead. The NPV as a function of outer iteration is shown in Figure 5.3 for both models, revealing that using the reduced model generates very similar NPV results.

The similarity between the optimal solution resulted from these two workflows

	Single Sim. Time	# of Inner Iter.	# of Outer Iter.	Total Opt. Time	Final NPV
Fine Model	5.9 (s)	47	18	446 (s)	2.03 (million)
Reduced Model	4.7 (s)	56	19	431 (s)	2.03 (million)

Table 5.1: Comparing optimization results of high fidelity and reduced order model, for a 51x51 reservoir model with 5-spot pattern

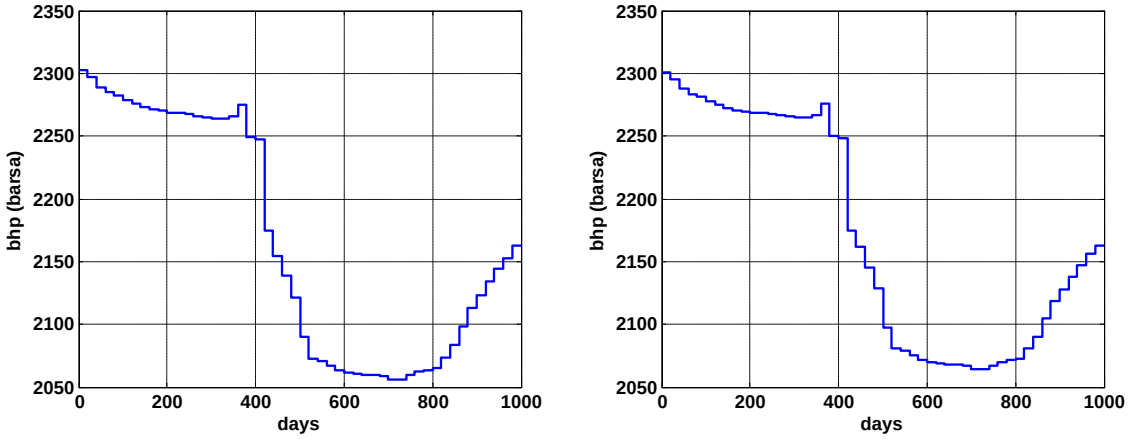


(a) high fidelity model (b) reduced model

Figure 5.3: NPV values through outer iterations

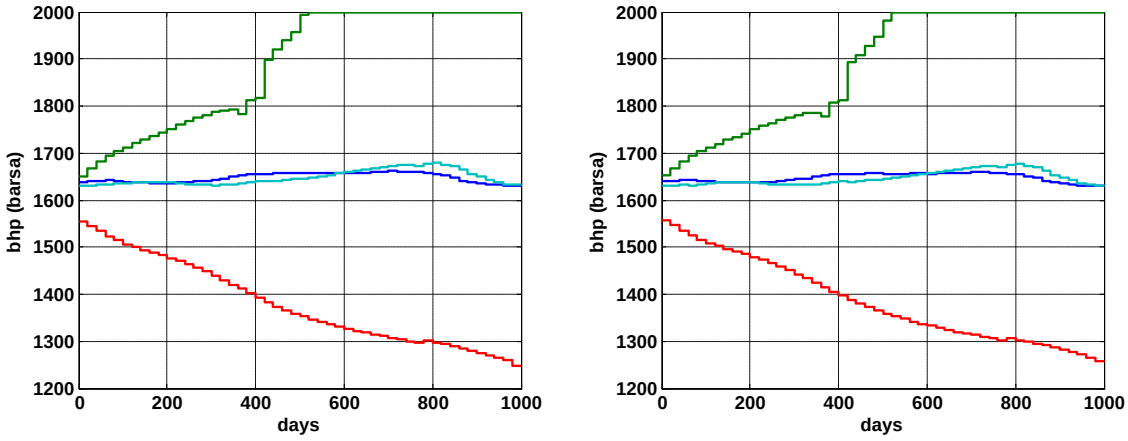
are more vivid by comparing the optimal BHP of the injector and producers in Figures 5.4 and 5.5, respectively.

Also the final oil saturation after optimal water flooding is compared for both workflows in Figure 5.6. The Optimal water cut at the producers is also shown for both workflows in Figure 5.7. All of these results verified the applicability of the proposed workflow and reveals that one can get very similar optimization results by using the reduced order model in the optimization workflow.



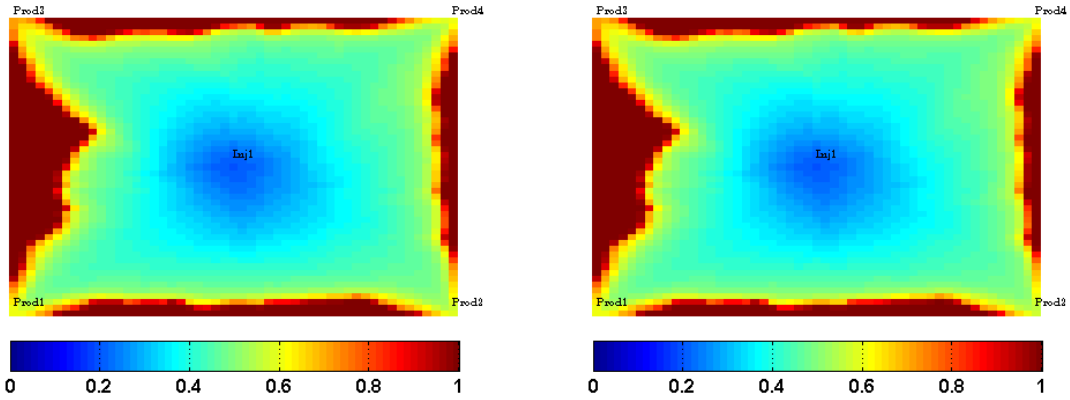
(a) high fidelity model (b) reduced model

Figure 5.4: Optimal BHP of the injector



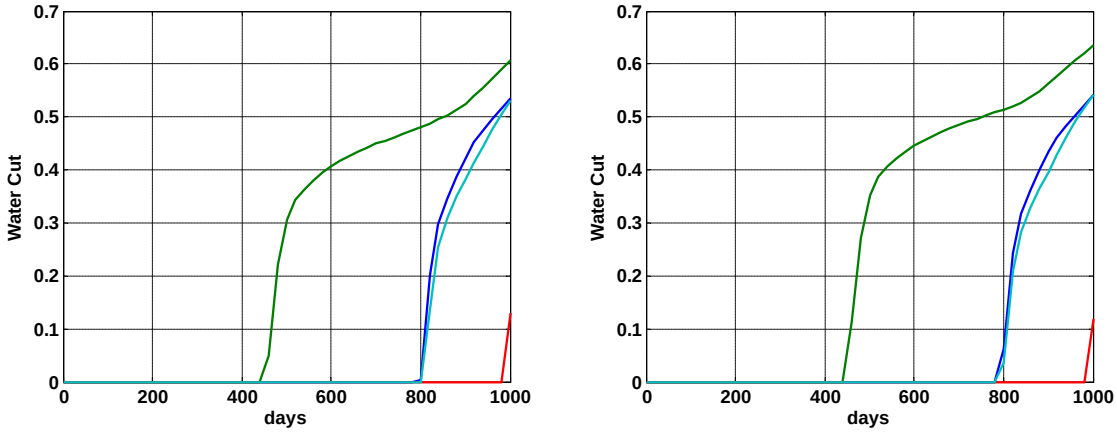
(a) high fidelity model (b) reduced model

Figure 5.5: Optimal BHP of the producers



(a) high fidelity model (b) reduced model

Figure 5.6: Final oil saturation after optimal water-flooding



(a) high fidelity model (b) reduced model

Figure 5.7: Water cut at producers after optimal water flooding

5.3 UNISIM-I-D Reservoir Model Benchmark

The UNISIM-I-D was developed in response to the Brazilian campaigns in dealing with carbonate reservoirs. The reference model is based on (after some modifications) the structural, facies and petrophysical characteristics of the Namorado Field, located in Campos Basin, Brazil. The original model as described in [45], contains approximately 3.5 million active grid blocks. Here I use a modified version of the original reservoir, such that the dimensions of the grid are 81x58x20 blocks with approximately 37,000 active cells. The Corner-point griding is used in this model.

This reservoir is under development and recently assisted optimization workflow was applied to this reservoir model to optimize the number and location of wells as well as wells opening schedule and wells shut-in time [44]. The optimal solution obtained by utilizing genetic algorithm and an optimizer tool in a commercial simulator software.

The permeability of this benchmark is heterogeneous as shown in Figure 5.8. This figure also includes the location of the 25 wells that are used in the schedule optimization. The porosity is also heterogeneous, which results in a pore volume distribution as depicted in Figure 5.9. There are multiple natural faults in this model. All of the characteristics of this benchmark model can be found in [45].

5.3.1 Model Order Reduction on UNISIM-I-D

It is assumed that the reservoir is a black oil model above bubble point pressure and it is saturated with oil and water. the initial oil saturation is shown in Figure 5.10. The Corey's model of the relative permeability curve as depicted in Figure 5.2b is used.

Here only POD model order reduction is applied to the UNISIM benchmark, because DEIM did not result in a stable and plausible solutions in particular after

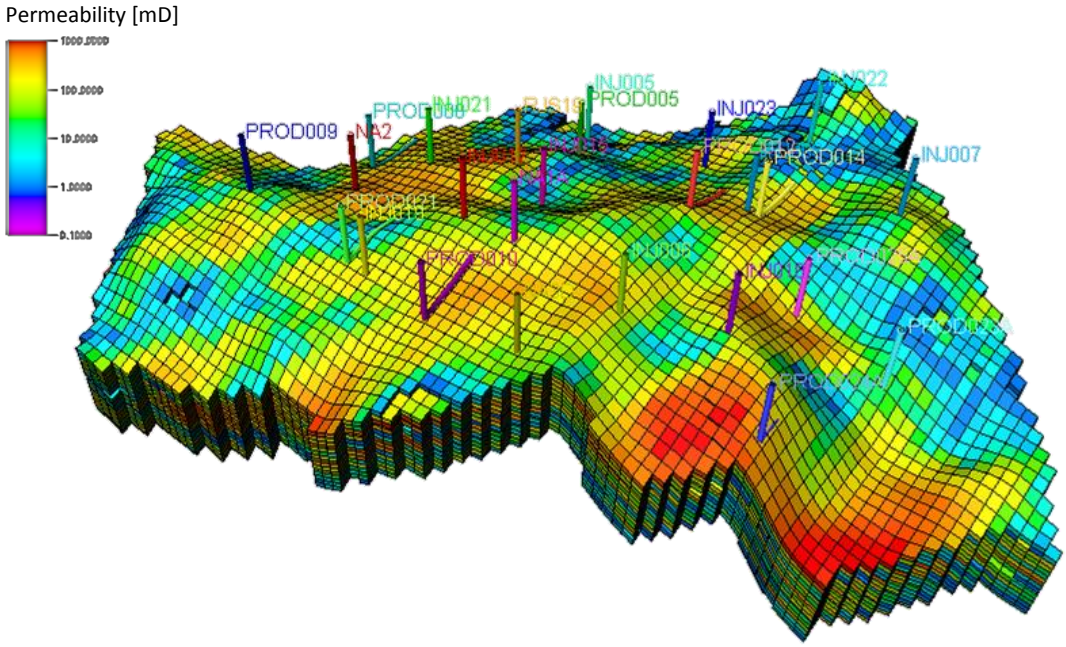


Figure 5.8: Permeability in x-direction of UNISIM-I-D reservoir model benchmark, including all 25 wells

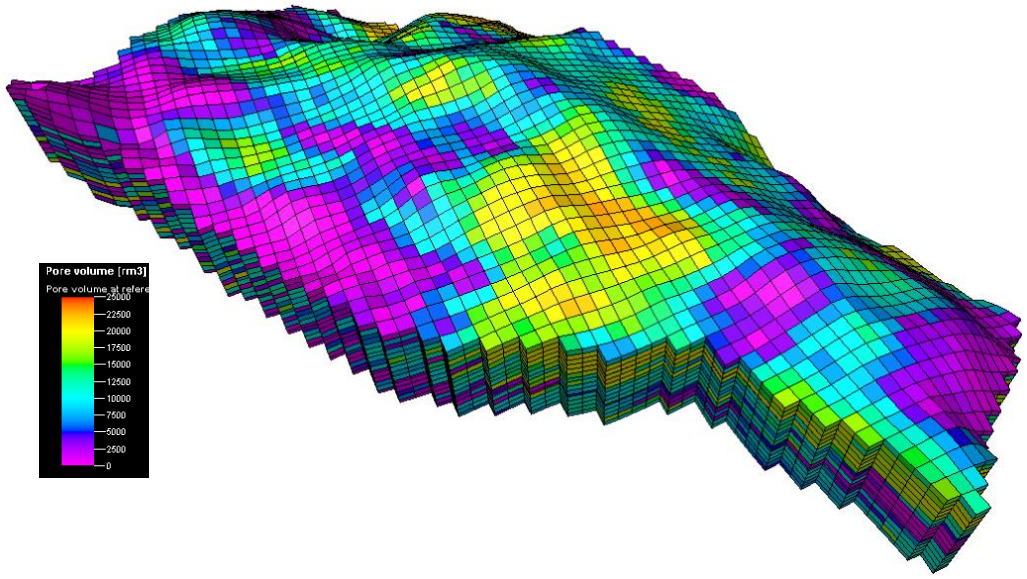


Figure 5.9: Pore volume of UNISIM-I-D reservoir model benchmark

integrating it with optimization. The main reason can be in the complex structure of the reservoir and also few number of basis due to the limitation in MRST. In the current implementation of the code, the control schedule and the training schedule should both have the same number of time steps. This limitation can yield a few snapshots if time steps are large or a very fine control schedule if the simulation time steps are small. One should separate these two schedules so that one can save the solution results in smaller time steps in order to improve the snapshots and consequently the basis for the reduced model.

The model was run for 1500 days based on schedules shown in Figure 5.11 for bottom-hole pressure of the producers and injectors. The pressure (Po), velocity (Vel), water saturation (Sw) and fractional flow of water (fw) are saved every 30 days. Thus there are 50 snapshots for each state. After applying *svd* to each snapshot

	# Pressure	# Velocity	# Saturation	Elapsed Time
Fine Model	38466	120897	38466	144 (sec)
Reduced Model	5	23	18	50 (sec)

Table 5.2: Comparing simulation of high fidelity model and reduced order UNISIM-I-D model

matrix, the singular values are obtained as are shown in Figure 5.12. There is a faster decay in the singular values for the pressure compared to velocity and water saturation. Thus, more basis for saturation and velocity are needed to have a reliable reduced order model.

The selection criteria here was to capture at least 99.5% of the energy of snapshots. The number of basis is compared between the reduced model and the original fine scale one in Table 5.2. It is obvious that the size of the model is reduced several orders of magnitude. Also, the simulation runtime reduced 3 time.

The final water cut for all the producers after 1500 days of simulation is shown in Figure 5.13 and is compared with fine scale model. The relative error of saturation is calculated as,

$$s_{err}(t_i) = \|s_{red}(t_i) - s_{ref}(t_i)\|_2 / \|s_{ref}(t_i)\|_2, \tag{5.21}$$

where $s_{red}(t_i)$ is the solution from the reduced order model and $s_{ref}(t_i)$ is the reference solution from high fidelity model. This error is less than 1% for the entire simulation time as shown in Figure 5.14. This indicates that the reduced model is a valid and reliable approximation.

To test the quality of basis and also robustness of the reduced model to input variations, it was run with a new test schedule as shown in Figure 5.15. This schedule is obtained by $\pm 5\%$ random perturbation of the training schedule. Note that the basis of the reduced model are not updated and we used the same basis obtained

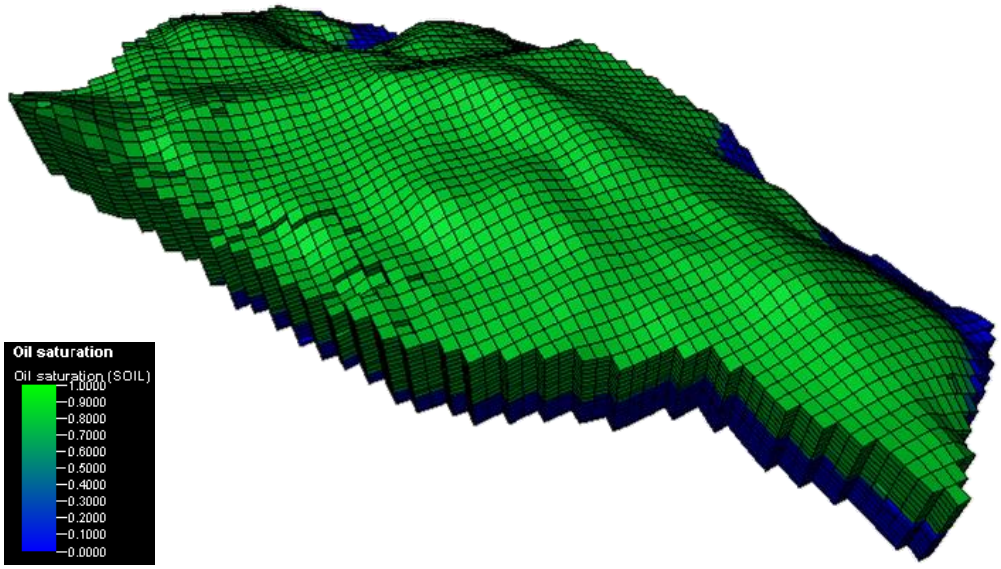
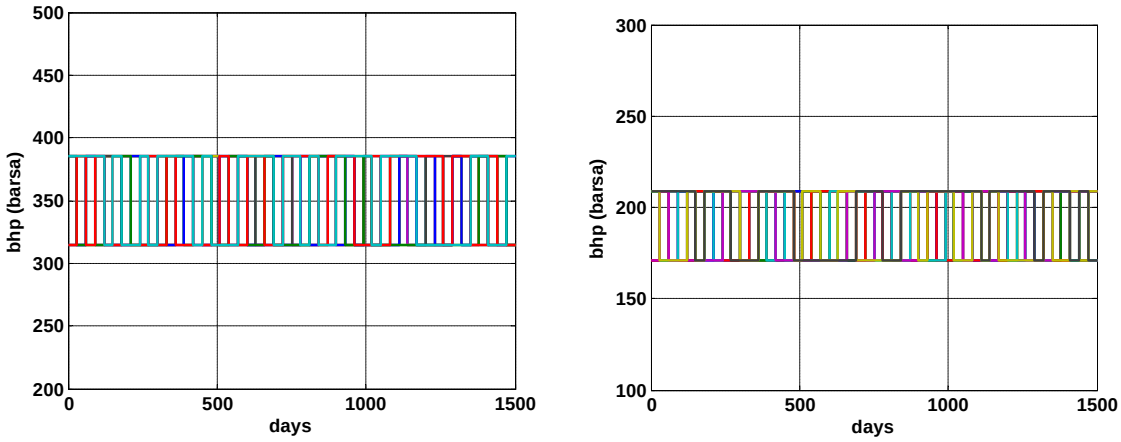


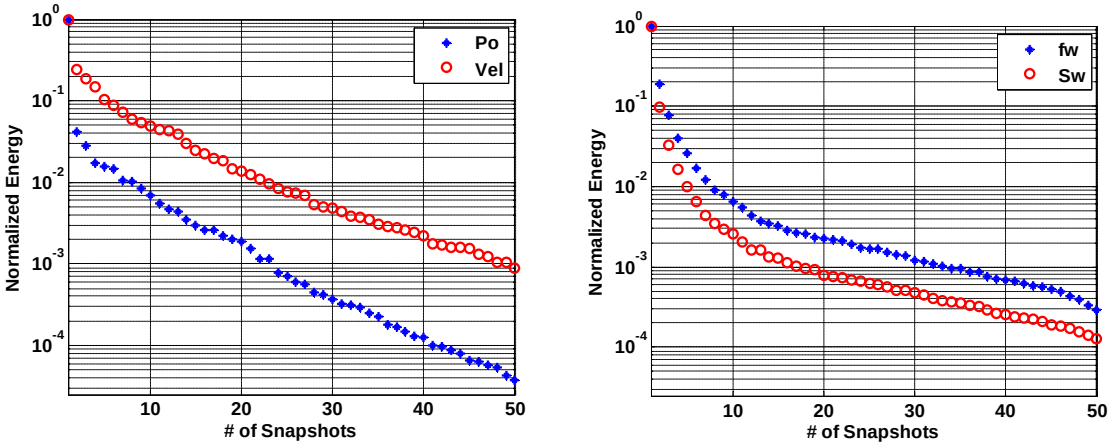
Figure 5.10: Initial oil saturation of UNISIM-I-D reservoir model



(a) BHP injectors

(b) BHP producers

Figure 5.11: Training schedule for applying POD on the UNISIM-I-D benchmark



(a) pressure (Pres) and velocity (Vel) (b) saturation (Sw) and fractional function (fw)

Figure 5.12: Singular values of snapshot matrices after running UNISIM-I-D benchmark with training schedule

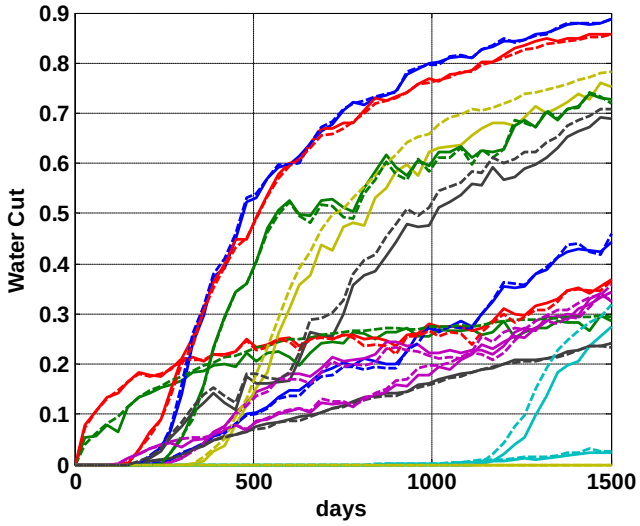


Figure 5.13: Comparison of water cut at producers between reduced (solid line) and high fidelity (dashed line) UNISIM-I-D model

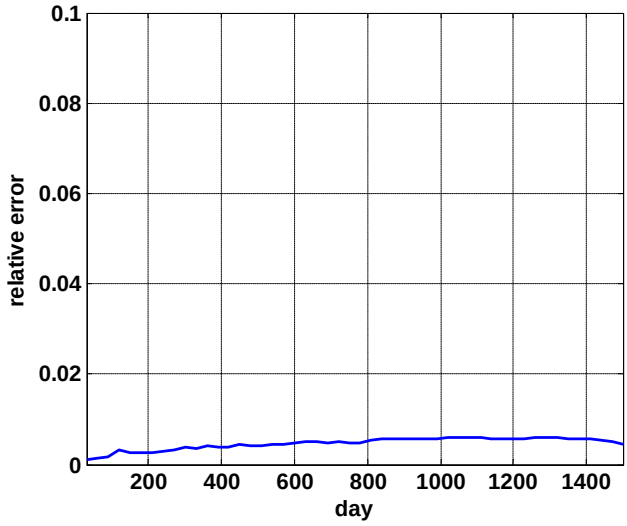


Figure 5.14: Temporal saturation error between reduced and high fidelity UNISIM-I-D model

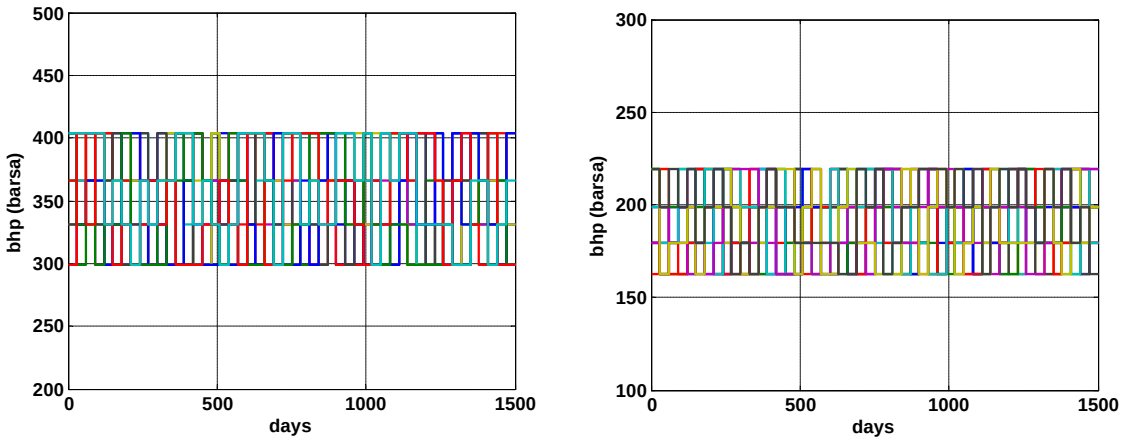
from training snapshots.

The final water cut for all the producers after 1500 days of simulation with the test schedule are shown in Figure 5.16 and compared with fine scale model. The temporal saturation error is smaller than 1% for the entire simulation time as shown in Figure 5.17. These two figures verify the applicability of the reduced order model in the workflows such as production optimization whereby the input schedule is varied.

5.3.2 Production Optimization of UNISIM-I-D

In this part, we apply the proposed optimization workflow with and without model order reduction on the UNISIM-I-D benchmark to find the optimal water flooding in the horizon of 1500 days.

The residual saturation is 0.2 for both water and oil. The viscosity ratio is set ot 5 ($\mu_o/\mu_w = 5$), and the initial water saturation is equal to the residual water saturation ($s_w(t = 0) = 0.2$). The revenue of oil production is considered to be \$70



(a) BHP injectors (b) BHP producers

Figure 5.15: Test schedule: ±5% random variation in the training schedule

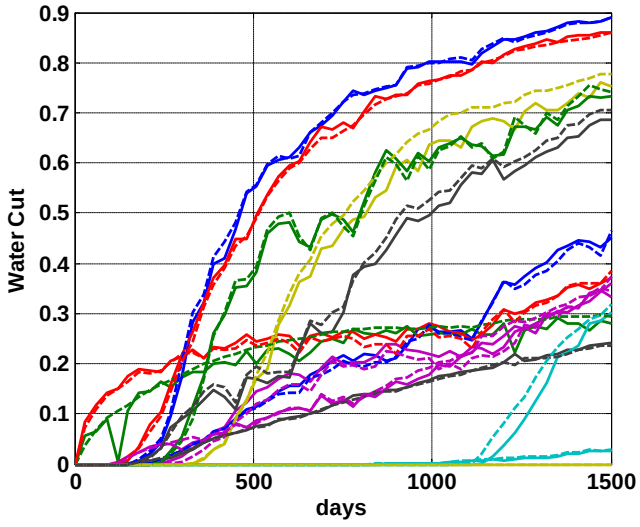


Figure 5.16: Comparison of water cut at producers between reduced (solid line) and high fidelity (dashed line) UNISIM-I-D model under test schedule

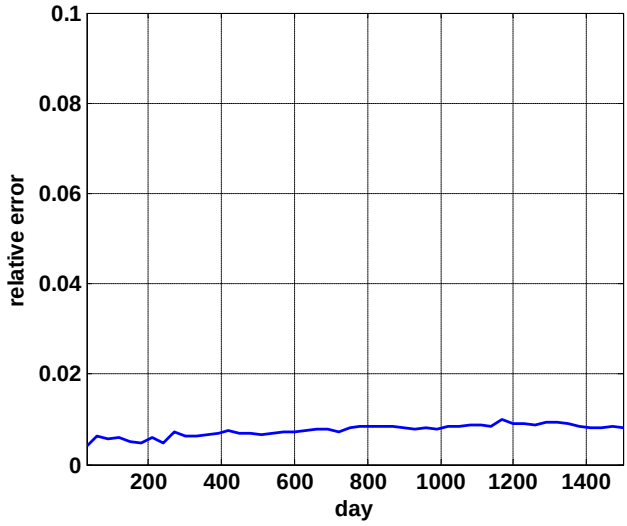


Figure 5.17: Temporal saturation error between reduced and high fidelity UNISIM-I-D model under test schedule

per barrel and the cost of injection or production of water is \$20 per barrel for each.

It is assumed that all the wells are BHP controlled and for each well it can be changed every 30 days. Thus, the number of control variables are $25 \times 50 = 1250$. The BHP for the injector is constrained between 300 bars to 700 bars and for the producers between 100 bars to 300 bars. The initial BHP for the injector is set to 500 bars and for all the producers is set to 250 bars.

Moreover, the snapshots of the solution, pressure, saturation, velocity are saved and used to find the basis for the reduced order model. Overall there are 50 snapshots for each state solution. These basis are updated at each outer loop iteration and are used to construct the reduced order model for inner loop iterations. These iterations are to find the step size that results in the maximum increase in the NPV function along the steepest decent (gradient) direction.

The optimization results are compared in Table 5.3 between two workflows. If one uses the high fidelity model in the inner loop iterations the NPV results is slightly

	Single Sim. Time	# of Inner Iter.	# of Outer Iter.	Total Opt. Time	Final NPV
Fine Model	154.2 (s)	64	22	15976 (s)	9309.8 (million)
Reduced Model	73.7 (s)	49	17	6964 (s)	9308.0 (million)

Table 5.3: Comparing optimization results using high fidelity and reduced UNISIM-I-D model in the inner loop

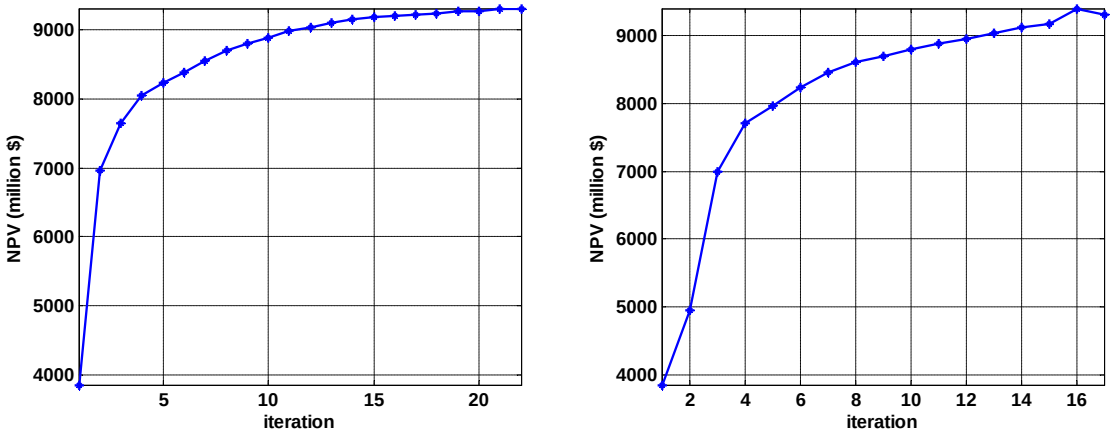
higher, however it takes more than 4.5 hour to get the optimal solution, whereas using the reduced model instead in the inner loop cut this computational time to less than 2 hours. The NPV as a function of outer iteration is shown in Figure 5.18 for both models, revealing that using the reduced model generates very similar NPV results.

The similarity between the optimal solution resulted from these two workflows are more vivid by comparing the optimal BHP of the injector and producers in Figures 5.19 and 5.20, respectively. The water cut at the producers after optimal water flooding sweeps the reservoir are also shown for both workflows in Figure 5.21. Also the final oil saturation is compared using the high fidelity model and reduced order model in Figure 5.22 and Figure 5.23, respectively. All of these results verified the applicability of the proposed workflow and reveals that one can get very similar optimization results by using the reduced model in the optimization workflow.

Note that to improve the results one can use the reduced order models to obtain the suboptimal solution in less computational time, and then use these results and the high fidelity model to get the optimal solution.

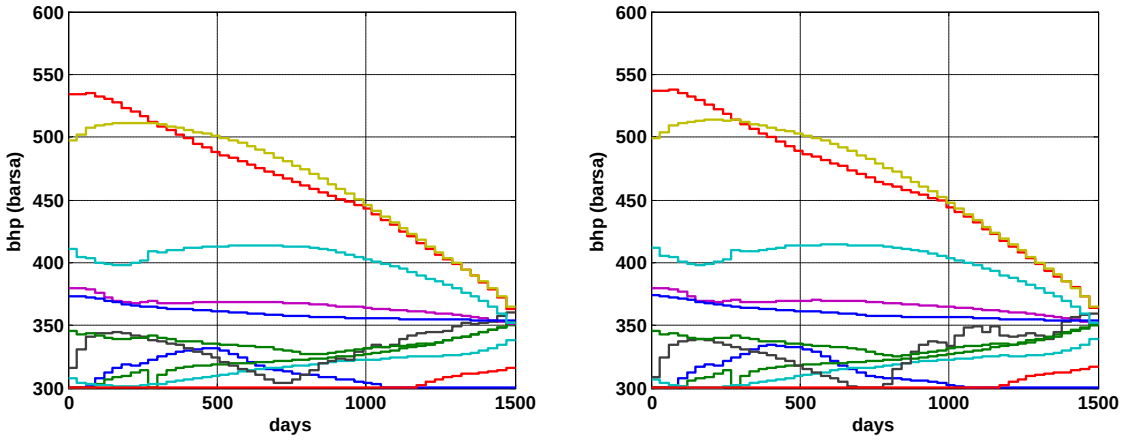
5.4 Summary

In this chapter, the gradient based optimization was discussed and applied to well scheduling. The gradient of the objective function with respect to input parameters were calculated efficiently based on the adjoint model. Thus, after running reservoir



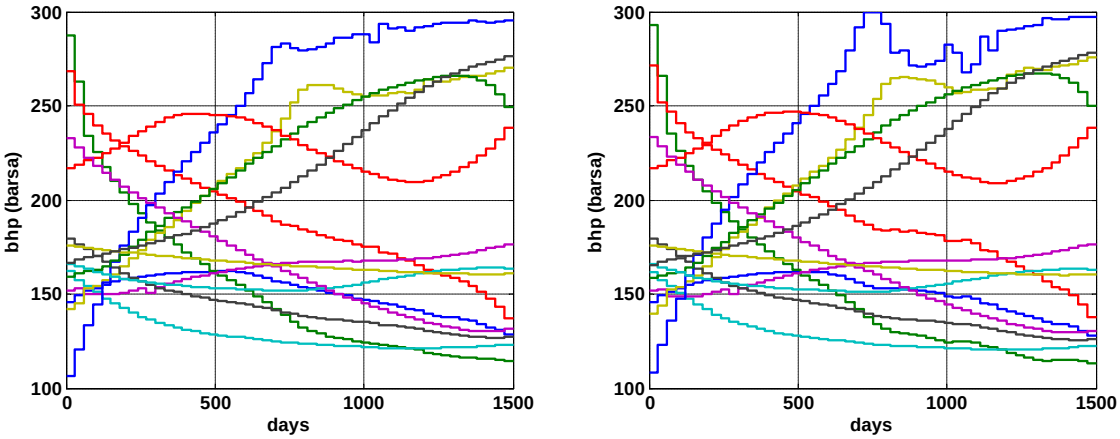
(a) high fidelity model (b) reduced model

Figure 5.18: NPV values through outer optimization iterations on UNISIM-I-D model



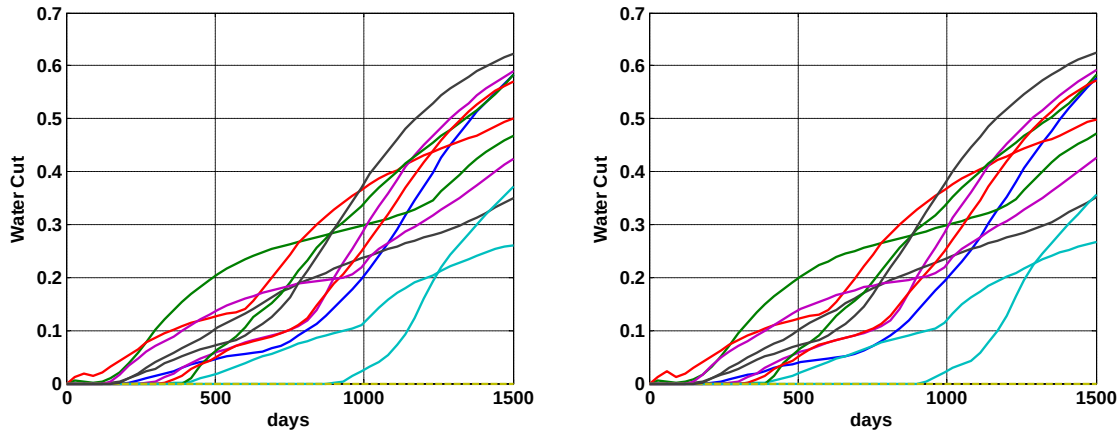
(a) high fidelity model (b) reduced model

Figure 5.19: Optimal BHP of the injectors obtained from optimization on UNISIM-I-D benchmark



(a) high fidelity model (b) reduced model

Figure 5.20: Optimal BHP of producers obtained from optimization on UNISIM-I-D benchmark



(a) high fidelity model (b) reduced model

Figure 5.21: Water cut at producers of UNISIM-I-D under optimal water flooding

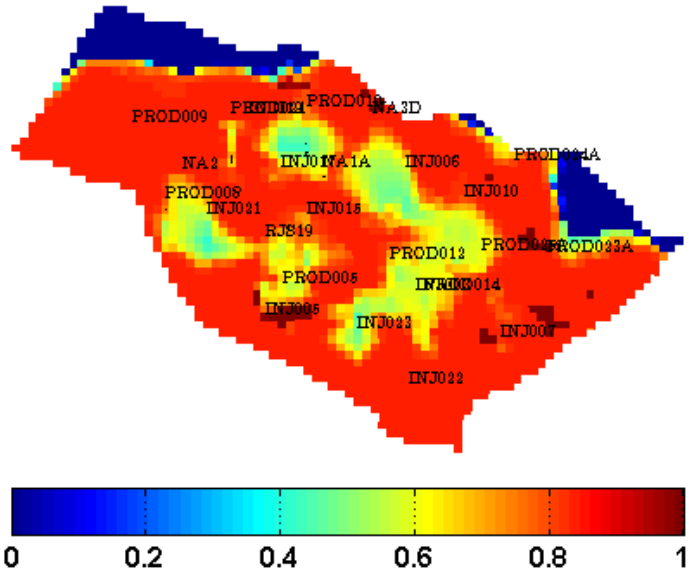


Figure 5.22: Final oil saturation after optimal water-flooding using high fidelity UNISIM-I-D model

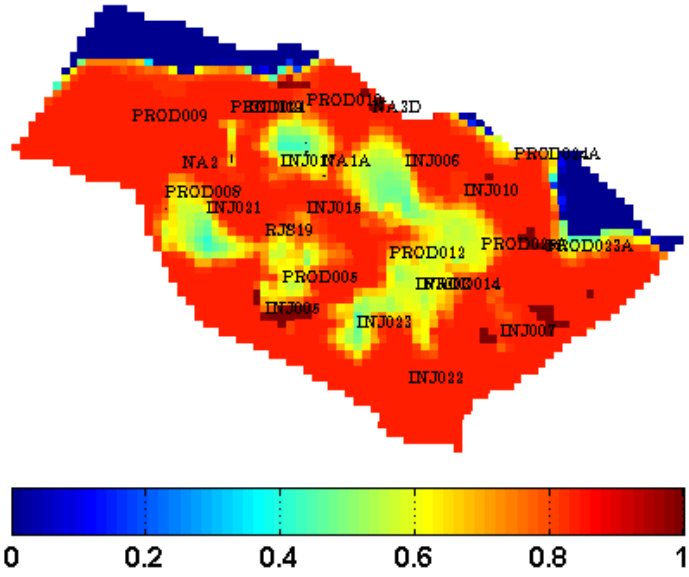


Figure 5.23: Final oil saturation after optimal water-flooding using reduced UNISIM-I-D model

simulation, the results were used to evaluate the objective function, and also to run the adjoint model backward in time. Line-search was used to find the proper step size for updating the input parameters in the direction of the gradient.

To accelerate this workflow, the large scale reservoir model in the line-search algorithm was replaced by a reduced order model that is computationally faster to run. Finally this workflow was successfully applied to a new benchmark based on a offshore reservoir, namely UNISIM-I-D, to solve the waterflooding optimization problem. The results between the reduced order model and the high fidelity model were similar in terms of final optimal NPV and the BHP of the producers and injectors.

In future work, this local optimization process can be combined with global optimization process like genetic algorithm or particle swarm optimization to find the global maximum.

6. CONCLUSION AND FUTURE WORK

This study addressed the application of model order reduction in the subsurface flow simulation. In this final chapter, we review the main contributions of this thesis and discuss some future research directions. This thesis can be summarized as follows.

6.1 POD-DEIM Model Order Reduction

The model order reduction (MOR) procedure based on proper orthogonal decomposition (POD) technique was extended, and incorporated into a MATLAB Reservoir Simulation Toolbox (MRST). POD provides a simple approach to extract a set of basis, which contains dynamical features of states and can be used to project a large scale model to a reduced dimensional model. This technique requires running a model with a training schedule and saving the solution results at each time step, namely snapshots of solution. These snapshots are used to find the POD basis by applying singular value decomposition (*svd*) to them.

As POD based techniques use Galerkin (or Petrov-Galerkin) projections, the resulting system may not be locally mass conservative. This issue may results in a large number of basis required to compensate the mass conservation error if only pressure basis is used as finite volume formulation. However, any number of velocity basis holds mass conservation. Thus, the mixed finite element formulation (cf. [2]) was used to solve for pressure and velocity simultaneously. Also, sequential time discretization were used to separate pressure and saturation equation [1]. The MOR was applied to reduce the pressure, velocity and saturation states.

In the case of nonlinear systems, such as porous media flow models, POD model order reduction scheme does not immediately yield a computationally efficient re-

duced system. The main difficulty arises in evaluating nonlinear terms on a reduced subspace. The first method that was applied to alleviate this issue was discrete empirical interpolation method (DEIM). This technique evaluates nonlinear terms in a few grid blocks that are selected based on greedy algorithm. Thus, the resulting reduced system is independent of the fine scale grid blocks.

POD and DEIM techniques are intrusive methods and were implemented in the MRST. In the offline (pre-processing) phase the solution of the high fidelity model is used to construct the required basis for the reduced order model. Thus, in the online phase the large scale high fidelity model is replaced with the reduced one that is computationally more efficient to run.

All of these MOR techniques were verified using a reservoir model benchmark, 5 layers of SPE10 [31], with 66,000 active grid blocks. This corresponds to 66,000 + 183,400 for pressure + velocity unknowns and 66,000 unknowns for saturation. The MOR was able to capture the dynamics of these models using only about 5+12 unknowns for pressure and velocity equation and 20 unknowns for saturation equation. Also, the nonlinear fractional flow function was evaluated at only 25 points instead of 66,000 grid points and were approximated in the reset of the grid blocks by DEIM. In terms of speed up, the reduced model was run approximately 70 times faster than the high fidelity model. Comparing the error between these models in terms of water cut, and final pressure and saturation error validated this MOR technique. The reduced model was also tested with a different schedule than the one used for saving snapshots and the error was still small (less than 5 % for most of the simulation time). This indicates that the reduced order model retain a reasonable level of robustness, which is important if they are to be used in optimization applications.

Furthermore, to improve the POD-DEIM in terms of required number of basis

and resulted error, new approach was developed in this study to update online. In the regular POD-DEIM workflow all the snapshots are used to find one single reduced subspace, whereas the Localized POD (LPOD) and Localized DEIM (LDEIM), compute several local subspaces. Each subspace characterize a period of time in the solution space. All together these local subspaces not only can approximate the high fidelity model better, but also reduce the computational cost of simulation. *k-means* clustering algorithm is used in LPOD and LDEIM to categorize similar snapshots at the offline (pre-processing) phase. Next, the basis of reduced subspaces are found for each cluster. In the online phase, at each time step, the reduced states is used in a classifier to find the most representative basis and to update the reduced subspace.

This new technique was used in the saturation equation and was tested on one layer of SPE-10 benchmark model with 13,200 active grid blocks. It was shown that with the same number of basis, both LPOD and LDEIM outperform regular POD and DEIM. These new techniques have some overhead in the online phase to pick the right set of basis. In order to reduce this overhead, an efficient classifier was developed that only uses solution in a few grid blocks to find the best representative cluster and the corresponding reduced basis. Thus, in terms of computational time, both LPOD and LDEIM are close to regular POD and DEIM. The speedup for all of these models were approximately 4 times. In comparison of water cut and final saturation error, the localized model reduction techniques yield less error in all the simulation time. For this example, the average temporal saturation error is 4% for POD and 2% for LPOD. It was 15% for DEIM whereas it was 5% for LDEIM.

6.2 MOR on Quadratic Bilinear Formulation

In this approach to overcome the issue of nonlinearity, the quadratic bilinear form (QBLF) of the original nonlinear porous media flow system is introduced, yielding a

system that is linear in the input and linear in the state, but not in both input and state jointly. Primarily, a new set of variables is used to change the problem into QBLF. Bilinear form of a nonlinear system can yield a better analysis by enabling us to use generalized transfer function.

In this thesis, the Carleman bilinear formulation based on Taylor's expansion is also discussed. This technique is developed for single phase flow and it was shown that the dimension of this model can be reduced by applying bilinear iterative rational Krylov algorithm (BIRKA). This MOR technique does not use training and solution snapshots to construct the reduced space, but rather uses the matrices in the bilinear form of the nonlinear system to find the projection matrices.

In a simple example it was shown that for a small model the Carleman bilinear form of the model outperforms the linear approximation. However, as this method is based on Taylor's expansion, the model is only valid close to the expansion point. It was also shown that this method failed in two phase flow because the fractional flow cannot be approximated well just by Taylor's polynomial expansion.

On the other hand, we developed a quadratic bilinear differential algebraic equation (QBDAE) procedure for the saturation equation in the subsurface two phase flow. This new technique involves representation of nonlinear terms as a new state. The author in [58] showed that this method is applicable to wide range of nonlinear systems. This approach will increase the dimension of the system linearly before we apply model order reduction. However, it can help to capture the dynamics of the nonlinear system better and to have more representative reduced order model.

This representation of the system has some advantageous in terms of MOR. First, applying POD to this system automatically yields a reduced system that can be solved without touching the fine scale model. Thus one does not need to use DEIM for further approximation. In addition, the reduced system has the same form of the

original system and thus it preserves the properties such as stability and passivity.

The proposed method was verified in numerical example. Primarily SPE10 benchmark was used as it is a very heterogeneous model and is used often times in testing different upscaling methods. The model reduction was able to reduce the complexity of the system significantly, and the errors in the solution were small even for different input. The error results from this method was smaller than using POD-DEIM. Still more efficient implementation of this formulation and the reduced model is required in order to do computational time analysis.

6.3 Production Optimization

The main focus in the last part of this thesis was to develop a robust workflow for optimization based on reduced order models. This procedure was developed and implemented in MRST and was tested on UNISIM-I-D [45], which is a reservoir benchmark model based on Namorado field. This model has around 38,000 active grid blocks, 11 injectors and 14 producers, which most of them are horizontal. However, in this study it was assumed that the reservoir is saturated with oil and water and there is no gas cap in the reservoir.

Primarily, POD based MOR was applied to this model to verify the applicably of MOR on this reservoir. It was shown that a reduced model with 5 + 23 unknowns for pressure + velocity and 18 unknowns for saturation could replicate a very similar results in water cut and oil saturation compared to high fidelity model with 38,466 + 120,897 unknowns for pressure + velocity and 38466 unknowns for saturation. The reduced model could be run three time faster. The error in saturation for all the simulation time is less than 1%. This reduced model was tested with a different input schedule from the training and the results were still reliable.

We considered the gradient based optimization workflow based on line-search. In

this procedure, there are two main loops. Primarily in the first loop, the reservoir model is run to obtain the production rates and to evaluate the NPV. Next, these results are integrated and used in an adjoint model to find the gradient of NPV with respect to input parameters, BHP of injectors and producers in this study. Finally, this gradient is used with a step size to be determined for updating the parameters values. We find the step size through an aggressive line-search, whereby the step size is increased/decrease until the NPV increases. Thus, one need another inner loop to run the model and decide the value of step size.

We implemented MOR to overcome the high computational cost associated with the gradient-based optimizations. After running the high fidelity model in the outer loop the reduced model was constructed based on the obtained solution and this model was used in the inner loop to find the value of step size.

We found out that when the production or injection well BHP schedule differed more than 10% from the training runs the accuracy of the reduced model degraded and the optimization results would be significantly different from high fidelity results. Thus, we enforced the search region at each iteration to be at most within 10% variation in the current schedule.

Finally, I tested the water flooding optimization based on POD reduced-order model for the UNISIM-I-D benchmark and compared it with the results from high fidelity model. The work flow based on the reduced order model results in the same NPV and very similar optimal BHP for injectors and producers. It took approximately 4.5 hours to obtain these results from high fidelity model, whereas it took less than 2 hours for the workflow with reduced model.

6.4 Suggestions for Future Work

In the POD-DEIM procedure, the upwind indices matrix A depends on the velocity. Thus, one needs to address the updating of upwinding matrix during simulation. Our approach was to project back to fine scale solution at each time step after the pressure equation is solved and update the upwinding matrix based on new velocity solution. Other alternative method is to update the upwinding matrix based on reduced velocity vector, i.e. finding regions in the reduced subspace where the velocity direction has changed using the reduced velocity and only updates those regions. These issues can be investigated in future research.

All of the proposed MOR methods in this thesis were applied to two phase flow incompressible or slightly compressible fluid without gravity or capillary pressure forces. One should investigate the application of these methods in multi-physics and multi-components reservoirs, which usually arise in practice. POD-DEIM procedure should be extended to three phase flow problems. It should also be verified for larger models containing $\mathcal{O}(10^6)$ grid blocks, which should magnify the superiority of POD-DEIM compared to regular POD method. Note that strong nonlinearity in relative permeability curves yields that large number of POD/DEIM basis would be required to have reliable reduced model, which may increase the computational time. These issues require further investigation.

In this study localized POD and localized DEIM were only applied to saturation equation. One can investigate their applicability in pressure equation as well. Also, these two methods are not combined in this thesis, meaning that in LDEIM, the saturation basis are still selected from regular POD and not LPOD. Thus, this integration should be considered in future research. More work still needs to be done to select the appropriate number of clusters and the specific clustering technique used.

After obtaining the QBDAE of the two phase flow equation, POD model order reduction was applied to the system to reduce the state dimension. However, one can apply Bilinear Iterative Rational Krylov Algorithm (BIRKA) to obtain the basis. This technique is independent of the input/output schedule and there will be no need for training. Thus, the reduced system will be robust to a wide range of inputs. However, as this model results in a multi input and output system more investigation is needed to apply training free model order reduction techniques.

Our strategy for integrating the reduced model in the optimization workflow was based on aggressive line search method in optimization. Thus, only the model in the inside loop is replaced with the reduced model to update the basis at each outer loop. This approach yielded very similar results between the workflow based on high fidelity model and the one with the reduced order model. However, it is still computationally demanding specially because we still run the fine scale model. One can consider integrating the reduced model with other workflow such as trust region. Moreover, applications of the LPOD and LDEIM as well as quadratic bilinear form in reservoir production optimization workflow should be addressed in future work.

Currently, there is a limitation in MRST such that one the control schedule and the training schedule should both have the same number of time steps. This limitation can yield a few snapshots if time steps are large or a very fine control schedule if time steps are small. One should separate the optimization schedule from the training schedule so that one can save the solution results in smaller time steps. This will improve the snapshots and consequently the basis for the reduced model.

Ultimately, the work that has been accomplished in this thesis can be applied to other areas of research such as subsurface shallow water management and geothermal reservoir simulations. It also enables us to change the reservoir management practices from a simple reactive solution to a more active methodology of designing real-time

intelligent production strategies to meet energy demands of the society.

REFERENCES

[1] J. E. Aarnes, T. Gimse, and K.-A. Lie. An introduction to the numerics of flow in porous media using matlab. In *Geometrical Modeling, Numerical Simulation, and Optimization: Industrial Mathematics at SINTEF, Eds., G. Hasle, K.-A. Lie, and E. Quak*, pages 265–306. Springer Verlag, 2007.

[2] Jorg E Aarnes, Stein Krogstad, and Knut-Andreas Lie. A hierarchical multiscale method for two-phase flow based upon mixed finite elements and nonuniform coarse grids. *Multiscale Modeling & Simulation*, 5(2):337–363, 2006.

[3] S. Afra, E. Gildin, and M. Tarrahi. Heterogeneous reservoir characterization using efficient parameterization through higher order svd (hosvd). In *American Control Conference*, Portland, Oregon, USA, June 2014. IEEE.

[4] M. Alfi, B. Yan, Y. Cao, C. An, Y. Wang, and J. E. Killough. How to improve our understanding of gas and oil production mechanisms in liquid-rich shale. In *SPE Annual Technical Conference and Exhibition*, Amsterdam, Netherlands, 2014. Society of Petroleum Engineers. SPE 170959-MS.

[5] M. Alfi, B. Yan, Y. Cao, C. An, Y. Wang, and J. E. Killough. Three-phase flow simulation in ultra-low permeability organic shale via a multiple permeability approach. In *Unconvenctional Resources Technology Conference*, Colorado, USA, 2014. Society of Petroleum Engineers.

[6] David Amsallem, Matthew J Zahr, and Charbel Farhat. Nonlinear model order reduction based on local reduced-order bases. *International Journal for Numerical Methods in Engineering*, 92(10):891–916, 2012.

[7] A.C. Antoulas. *Approximation of Large-Scale Dynamical Systems*. SIAM Press, Philadelphia, 2005.

[8] A.C. Antoulas, D.C. Sorensen, and S. Gugercin. A survey of model reduction methods for large-scale systems. *Contemporary Mathematics in Numerical Algorithms*, 2001.

[9] Athanasios C Antoulas and Dan C Sorensen. Approximation of large-scale dynamical systems: An overview. *International Journal of Applied Mathematics and Computer Science*, 11:1093–1121, 2001.

[10] P. Astrid, S. Weiland, K. Willcox, and T. Backx. Missing point estimation in models described by proper orthogonal decomposition. In *43rd IEEE Conference on Decision and Control (CDC)*, volume 2, pages 1767–1772, 2004.

[11] Patricia Astrid, Siep Weiland, Karen Willcox, and Ton Backx. Missing point estimation in models described by proper orthogonal decomposition. *Automatic Control, IEEE Transactions on*, 53(10):2237–2251, 2008.

[12] Christophe Audouze, Florian De Vuyst, and Prasanth B. Nair. Nonintrusive reduced-order modeling of parametrized time-dependent partial differential equations. *Numerical Methods for Partial Differential Equations*, 29(5):1587–1628, 2013.

[13] K. Aziz and A. Settari. *Petroleum Reservoir Simulation*. Elsevier Applied Science Publishers, 1986.

[14] Zhaojun Bai. Krylov subspace techniques for reduced-order modeling of large-scale dynamical systems. *Applied Numerical Mathematics*, 43(1):9–44, 2002.

[15] Zhaojun Bai and Daniel Skoogh. A projection method for model reduction of bilinear dynamical systems. *Linear algebra and its applications*, 415(2):406–

425, 2006.

[16] Maxime Barrault, Yvon Maday, Ngoc Cuong Nguyen, and Anthony T Patera. An empirical interpolation method: application to efficient reduced-basis discretization of partial differential equations. *Comptes Rendus Mathematique*, 339(9):667–672, 2004.

[17] P. Benner and T. Breiten. Krylov-subspace based model reduction of nonlinear circuit models using bilinear and quadratic-linear approximations. *Progress in Industrial Mathematics at ECMI 2010*, pages 153–159, 2012.

[18] Peter Benner and Tobias Breiten. Two-sided moment matching methods for nonlinear model reduction. Preprint MPIMD/12-12, Max Planck Institute Magdeburg, June 2012. Available from `http://www.mpi-magdeburg.mpg.de/preprints/`.

[19] Peter Benner and Tobias Damm. Lyapunov equations, energy functionals, and model order reduction of bilinear and stochastic systems. *SIAM journal on control and optimization*, 49(2):686–711, 2011.

[20] D.R. Brouwer and J. D. Jansen. Dynamic optimization of waterflooding with smart wells using optimal control theory. *SPE Journal*, 9(4), 2004.

[21] Marco Antonio Cardoso. *Development and Application of Reduced Order Modeling Procedures for Reservoir Simulation*. PhD thesis, Department of Energy Resources Engineeering at Stanford University, 2009.

[22] Marco Antonio Cardoso and Louis J Durlofsky. Use of reduced-order modeling procedures for production optimization. *SPE Journal*, 15(2):426–435, 2010.

[23] Kevin Carlberg, Charbel Bou-Mosleh, and Charbel Farhat. Efficient non-linear model reduction via a least-squares petrov–galerkin projection and compres-

sive tensor approximations. *International Journal for Numerical Methods in Engineering*, 86(2):155–181, 2011.

[24] Dimitrios Chaniotis and MA Pai. Model reduction in power systems using krylov subspace methods. *IEEE Transactions on Power Systems*, 20(2):888–894, 2005.

[25] Anindya Chatterjee. An introduction to the proper orthogonal decomposition. *Current science*, 78(7):808–817, 2000.

[26] S. Chaturantabut and D. C. Sorensen. Discrete empirical interpolation for nonlinear model reduction. *SIAM J. Sci. Comput.*, 32(5):2737–2764, 2010.

[27] S. Chaturantabut and D.C. Sorensen. Application of pod and deim to dimension reduction of nonlinear miscible viscous fingering in porous media. *Math. Comput. Model. Dyn. Syst, and Technical Report*, 2009.

[28] Saifon Chaturantabut and Danny C Sorensen. Nonlinear model reduction via discrete empirical interpolation. *SIAM Journal on Scientific Computing*, 32(5):2737–2764, 2010.

[29] Guy Chavent and Jérôme Jaffré. *Mathematical models and finite elements for reservoir simulation: single phase, multiphase and multicomponent flows through porous media*. Elsevier, 1986.

[30] Jinghong Chen and Sung-Mo Steve Kang. An algorithm for automatic model-order reduction of nonlinear mems devices. In *Circuits and Systems, 2000. Proceedings. ISCAS 2000 Geneva. The 2000 IEEE International Symposium on*, volume 2, pages 445–448. IEEE, 2000.

[31] M.A. Christie and M.J. Blunt. Tenth spe comparative solution project: a comparison of upscaling techniques. spe-72469. *SPEREE*, 4:308–317, 2001.

[32] DOE/EIA. Annual energy outlook 2014. Technical Report DOE/EIA-0383(2014), US Energy Information and Administration, 2014.

[33] JornF.M. Doren, Renato Markovinovi, and Jan-Dirk Jansen. Reduced-order optimal control of water flooding using proper orthogonal decomposition. *Computational Geosciences*, 10(1):137–158, 2006.

[34] Martin Drohmann, Bernard Haasdonk, and Mario Ohlberger. Adaptive reduced basis methods for nonlinear convection–diffusion equations. In *Finite Volumes for Complex Applications VI Problems & Perspectives*, pages 369–377. Springer, 2011.

[35] L.J. Durlofsky. Upscaling and gridding of fine scale geological models for flow simulation. In *8th International Forum on Reservoir Simulation*, Stresa, Italy, June 20-24 2005.

[36] Y. Efendiev and T.Y. Hou. *Multiscale finite element methods. Theory and applications. Surveys and Tutorials in the Applied Mathematical Sciences, 4.* Springer, New York, 2009.

[37] Yalchin Efendiev, Juan Galvis, and Eduardo Gildin. Local–global multiscale model reduction for flows in high-contrast heterogeneous media. *Journal of Computational Physics*, 231(24):8100–8113, 2012.

[38] Turgay Ertekin, J.H. Abou-Kassem, and G.R. King. *Basic Applied Reservoir Simulation*. Society of Petroleum Engineers, 2001.

[39] Richard Everson and Lawrence Sirovich. Karhunen–loeve procedure for gappy data. *JOSA A*, 12(8):1657–1664, 1995.

[40] ExxonMobil. The outlook for energy: A view to 2040. Technical report, Exxon Mobil Corporation, 2014.

[41] G. M. Flagg. *Interpolation Methods for the Model Reduction of Bilinear Systems*. PhD. thesis, Virginia Polytechnic Institute and State University, 2012.

[42] Garret M Flagg. *Interpolation Methods for the Model Reduction of Bilinear Systems*. PhD thesis, Virginia Polytechnic Institute and State University, 2012.

[43] Michel Fortin and F Brezzi. *Mixed and hybrid finite element methods*. Springer, 1991.

[44] A. T. Gaspar, C. E. Barreto, E. O. Munoz Mazo, and D. J. Schiozer. Application of assisted optimization to aid oil exploitation strategy selection for offshore fields. In *SPE Latin America and Caribbean Petroleum Engineering Conference*, Maracaibo, Venezuela, 2014.

[45] A. T. Gaspar, A. A. Santos, C. Maschio, G.D. Avansi, J. C. Von Hohendorff, and D.J. Schiozer. Study case for reservoir exploitation strategy selection based on unisim-i field. `http://www.unisim.cepetro.unicamp.br/unisim-i/`, retrieved: February 2015.

[46] Mohammadreza Ghasemi, Ibrahim Ashraf, and Eduardo Gildin. Reduced order modeling in reservoir simulation using the bilinear approximation techniques. In *SPE Latin American and Caribbean Petroleum Engineering Conference*, Maracaibo, Venezuela, May 2014. Society of Petroleum Engineers. SPE 169357-MS.

[47] Mohammadreza Ghasemi and Eduardo Gildin. Localized model reduction in porous media flow. In *2nd IFAC Workshop on Automatic Control in Offshore Oil and Gas Production*, Florianpolis, Brazil, May 2015.

[48] Mohammadreza Ghasemi, Yanfang Yang, Eduardo Gildin, Yalchin Efendiev, and Victor Calo. Fast multiscale reservoir simulations using pod-deim model

reduction. In *SPE Reservoir Simulation Symposium*, Houston, Texas, Feb 2015. Society of Petroleum Engineers. SPE 173271.

[49] Mohammadreza Ghasemi, S. Zhao, T. Insperger, and Tamas Kalmar-Nagy. Act-and-wait control of discrete systems with random delays. In *American Control Conference (ACC)*, pages 5440–5443, Montreal, Canada, June 2012. IEEE.

[50] M. Ghommem, V. M. Calo, Y. Efendiev, and E. Gildin. Complexity reduction of multi-phase flows in heterogeneous porous media. In *SPE Kuwait Oil and Gas Show and Conference*, Kuwait City, Kuwait, 2013. SPE. SPE 167295.

[51] Mehdi Ghommem, Eduardo Gildin, and Mohammadreza Ghasemi. Complexity reduction of multi-phase flows in heterogeneous porous media. *SPE Journal*, 2015. SPE 167295.

[52] E. Gildin, H. Klie, M. F. Wheeler, A. Rodriguez, and R.H. Bishop. Projection-based approximation methods for the optimal control of smart fields. In *10th. European Conference of the Mathematics of Oil Recovery*, Amsterdam, The Netherlands, September 2006.

[53] E. Gildin and T. J. Lopez. Closed-loop reservoir management: Do we need complex models ? In *SPE Digital Energy Conference and Exhibition*, The Woodlands, Texas, USA, April 2011.

[54] Eduardo Gildin and Mohammadreza Ghasemi. A new model reduction technique applied to reservoir simulation. In *ECMOR XIV-14th European conference on the mathematics of oil recovery*, Sicily, Italy, September 2014. European Association of Geoscientists and Engineers.

[55] Eduardo Gildin, Mohammadreza Ghasemi, A. Romanovskay, and Yalchin Efendiev. Nonlinear complexity reduction for fast simulation of flow in heterogeneous porous media. In *SPE Reservoir Simulation Symposium*, The Woodlands, Texas, Feb 2013. Society of Petroleum Engineers. SPE 163618-MS.

[56] David Gratton and Karen Willcox. *Reduced-order, trajectory piecewise-linear models for nonlinear computational fluid dynamics*. PhD thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, 2004.

[57] Eric James Grimme. *Krylov projection methods for model reduction*. PhD thesis, University of Illinois at Urbana-Champaign, 1997.

[58] Chenjie Gu. Qlmor: a projection-based nonlinear model order reduction approach using quadratic-linear representation of nonlinear systems. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 30(9):1307–1320, 2011.

[59] S Gugercin, C Beattie, and AC Antoulas. Rational krylov methods for optimal h2 model reduction. *submitted for publication*, 2006.

[60] Serkan Gugercin, Athanasios C Antoulas, and Cchristopher Beattie. H_2 model reduction for large-scale linear dynamical systems. *SIAM journal on matrix analysis and applications*, 30(2):609–638, 2008.

[61] John A Hartigan and Manchek A Wong. Algorithm as 136: A k-means clustering algorithm. *Applied statistics*, pages 100–108, 1979.

[62] Rania Hassan, Babak Cohanim, Olivier De Weck, and Gerhard Venter. A comparison of particle swarm optimization and the genetic algorithm. In *Proceedings of the 1st AIAA multidisciplinary design optimization specialist conference*, pages 18–21, 2005.

[63] Jincong He, J Sætrom, and Louis J Durlofsky. Enhanced linearized reduced-order models for subsurface flow simulation. *Journal of Computational Physics*, 230(23):8313–8341, 2011.

[64] T Heijn, R Markovinovic, J-D Jansen, et al. Generation of low-order reservoir models using system-theoretical concepts. *SPE Journal*, 9(02):202–218, 2004.

[65] T. Heijn, R. Markovinovic, and J.D. Jansen. Generation of low-order reservoir models using system-theoretical concepts. *SPE Journal*, 9(2), June 2004.

[66] L. Holden and B.F. Nelsen. Global upscaling of penneability in heterogeneous reservoir; the output least squares method. *Transport in Porous Media*, 40:115–143, 2000.

[67] B Jafarpour and Mohammadali Tarrahi. Assessing the performance of the ensemble kalman filter for subsurface flow data integration under variogram uncertainty. *Water Resources Research*, 47(5), 2011.

[68] Anil K Jain. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31(8):651–666, 2010.

[69] J.D. Jansen, O.H. Bosgra, and P.M.J. van den Hof. Model-based control of multiphase flow in subsurface oil reservoirs. *Journal of Process Control*, 18(9):846–855, 2008.

[70] Stephen C Johnson. Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254, 1967.

[71] Tapas Kanungo, David M Mount, Nathan S Netanyahu, Christine D Piatko, Ruth Silverman, and Angela Y Wu. An efficient k-means clustering algorithm: Analysis and implementation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(7):881–892, 2002.

[72] Arthur J Krener. Bilinear and nonlinear realizations of input-output maps. *SIAM Journal on Control*, 13(4):827–834, 1975.

[73] Hans-Peter Kriegel, Peer Kröger, Jörg Sander, and Arthur Zimek. Density-based clustering. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(3):231–240, 2011.

[74] Stein Krogstad, Vera Louise Hauge, Astrid Fossum Gulbransen, et al. Adjoint multiscale mixed finite elements. *SPE J*, 16(1):162–171, 2011.

[75] S. Lall, J. Marsden, and S. Glavaski. Empirical model reduction of controlled nonlinear systems. In *IFAC World Congress, Vol. F*, pages 473–478, Beijing, 1999.

[76] P. Lerlertpakdee, B. Jafarpour, and E. Gildin. Efficient production optimization with flow-network models. *SPE Journal*, 2014.

[77] Robert Michael Lewis, Virginia Torczon, and Michael W. Trosset. Direct search methods: then and now. *Journal of Computational and Applied Mathematics*, 124(12):191 – 207, 2000. Numerical Analysis 2000. Vol. IV: Optimization and Nonlinear Equations.

[78] YC Liang, HP Lee, SP Lim, WZ Lin, KH Lee, and CG Wu. Proper orthogonal decomposition and its applicationspart i: Theory. *Journal of Sound and Vibration*, 252(3):527–544, 2002.

[79] M. L. Litvak, L. A. Hutchins, R. C. Skinner, B. L. Darlow, R. C. Wood, and L. J. Kuest. Prudhoe bay e-field production optimization system based on integrated reservoir and facility simulation. In *SPE Annual Technical Conference and Exhibition*, San Antonio, Texas, 2002.

[80] David J Lucia, Philip S Beran, and Walter A Silva. Reduced-order modeling: new approaches for computational physics. *Progress in Aerospace Sciences*, 40(1):51–117, 2004.

[81] Hung V Ly and Hien T Tran. Modeling and control of physical processes using proper orthogonal decomposition. *Mathematical and computer modelling*, 33(1):223–236, 2001.

[82] MATLAB. *version 8.3.0 (R2014a)*. The MathWorks Inc., Natick, Massachusetts, 2014.

[83] S.D. Mohaghegh, J. Liu, R. Gaskari, and M.Maysami. Application of well-base surrogate reservoir models (srms) to two offshore fields in saudi arabia, case study. In *SPE Western Regional Meeting*, Bakersfield, California, USA, March 2012.

[84] R Mohler. *Bilinear control processes*. Academic Press New York, 1973.

[85] Larry Nazareth. A relationship between the bfgs and conjugate gradient algorithms and its implications for new algorithms. *SIAM Journal on Numerical Analysis*, 16(5):794–800, 1979.

[86] N. C. Nguyen, A. T. Patera, and J. Peraire. A best points interpolation method for efficient approximation of parametrized functions. *Int. J. Numer. Meth. Eng.*, 73:521–543, 2008.

[87] Dean S Oliver, Albert C Reynolds, and Ning Liu. *Inverse theory for petroleum reservoir characterization and history matching*, volume 1. 2008.

[88] D. W. Peaceman. Interpretation of well-block pressures in numerical reservoir simulation with nonsquare grid blocks and anisotropic permeability. *SPE Journal*, 1983.

[89] Benjamin Peherstorfer, Daniel Butnaru, Karen Willcox, and Hans-Joachim Bungartz. Localized discrete empirical interpolation method. *SIAM Journal on Scientific Computing*, 36(1):A168–A192, 2014.

[90] Joel R Phillips. Projection-based approaches for model reduction of weakly nonlinear, time-varying systems. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 22(2):171–187, 2003.

[91] Liyan Qu and P.L. Chapman. Extraction of dynamic low-order nonlinear inductor models based on steady state solutions. In *Power Electronics Specialists Conference, 2006. PESC '06. 37th IEEE*, pages 1–9, June 2006.

[92] Ronald L Rardin and Reha Uzsoy. Experimental evaluation of heuristic optimization algorithms: A tutorial. *Journal of Heuristics*, 7(3):261–304, 2001.

[93] Muruhan Rathinam and Linda R Petzold. A new look at proper orthogonal decomposition. *SIAM Journal on Numerical Analysis*, 41(5):1893–1925, 2003.

[94] M. Rewienski and J. White. A trajectory piecewise-linear approach to model order reduction and fast simulation of nonlinear circuits and micromachined devices. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 22(2):155–170, Feb 2003.

[95] Michal Rewienski and Jacob White. A trajectory piecewise-linear approach to model order reduction and fast simulation of nonlinear circuits and micromachined devices. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 22(2):155–170, 2003.

[96] Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20(0):53 – 65, 1987.

[97] C. W. Rowley. Model reduction for fluids, using balanced proper orthogonal decomposition. *International Journal of Bifurcation and Chaos*, 15(03):997–1013, 2005.

[98] W. J. Rugh. *Nonlinear System Theory: The Volterra/Wiener Approach*. Johns Jopkins University Press, Baltimore, 1981.

[99] Wilson J Rugh. *Nonlinear system theory*. Johns Hopkins University Press Baltimore, 1981.

[100] R. W. Rwechungura, E. Suwartadi, M. Dadashpour, J. Kleppe, and B. A. Foss. The norne field case - a unique comparative case study. In *SPE Intelligent Energy Conference and Exhibition*, Utrecht, The Netherlands, 2010.

[101] Marcio Augusto Sampaio Pinto, Mohammadreza Ghasemi, Nadav Sorek, and Eduardo Gildin. Hybrid optimization for closed-loop reservoir management. In *SPE Reservoir Simulation Symposium*, Houston, Texas, Feb 2015. Society of Petroleum Engineers. SPE 173278.

[102] L Saputelli, M Nikolaou, and MJ Economides. Real-time reservoir management: A multiscale adaptive optimization and control approach. *Computational Geosciences*, 10(1):61–96, 2006.

[103] P. Sarma, K. Aziz, and L. J. Durlofsky. Implementation of adjoint solution for optimal control of smart wells. In *SPE Reservoir Simulation Symposium, SPE 92864-MS*, The Woodlands, Texas, USA, 2005.

[104] Pallav Sarma, Wen H Chen, Louis J Durlofsky, Khalid Aziz, et al. Production optimization with adjoint models under nonlinear control-state path inequality constraints. *SPE Reservoir Evaluation & Engineering*, 11(02):326–339, 2008.

[105] Pallav Sarma, Louis J Durlofsky, Khalid Aziz, and Wen H Chen. Efficient real-time reservoir management using adjoint-based optimal control and model updating. *Computational Geosciences*, 10(1):3–36, 2006.

[106] Wilhelmus HA Schilders, Henk A Van der Vorst, and Joost Rommes. *Model order reduction: theory, research aspects and applications*, volume 13. Springer, 2008.

[107] Yun Q Shi and Huifang Sun. *Image and video compression for multimedia engineering: fundamentals, algorithms, and standards*. CRC press, 1999.

[108] Eka Suwartadi. *Gradient-based Methods for Production Optimization of Oil Reservoirs*. PhD thesis, Norwegian University of Science and Technology, 2012.

[109] Dmitry Vasilyev, Micha Rewieski, and Jacob White. Macromodel generation for biomems components using a stabilized balanced truncation plus trajectory piecewise linear approach. In Krishnendu Chakrabarty and Jun Zeng, editors, *Design Automation Methods and Tools for Microfluidics-Based Biochips*, pages 169–187. Springer Netherlands, 2006.

[110] Giorgio Viadana, Danilo Albani, Marco Nicola Distaso, Almatasem Sharon, et al. Integrated production optimization and surface facilities management through advanced optimization techniques. In *SPE International Production and Operations Conference & Exhibition*. Society of Petroleum Engineers, 2012.

[111] S. Volkwein and M. Hinze. Proper orthogonal decomposition surrogate models for nonlinear dynamical systems: error estimates and suboptimal control. In *Reduction of Large-Scale Systems, P. Benner, V. Mehrmann, D. C. Sorensen (eds.), Lecture Notes in Computational Science and Engineering*, volume 45, pages 261–306. Edited by: Dafermos and Pokorny, Elsevier, 2005.

[112] S. Volkwein and K. Kunisch. Proper orthogonal decomposition for optimality systems. *ESAIM: Mathememetical Modelling and Numerical Analysis*, 42:1–23, 2008.

[113] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

[114] D. Weber, F.T Edgar, L.W. Lake, L.S. Larson, and K. Sawas. Improvements of the capacitance resistive modeling and optmization of large scale reservoirs. In *SPE Western Regional Meeting*, San Jose, California, March 2009. SPE 121299.

[115] Xiaowei Xu, Martin Ester, H-P Kriegel, and Jörg Sander. A distribution-based clustering algorithm for mining in large spatial databases. In *Data Engineering, 1998. Proceedings., 14th International Conference on*, pages 324–331. IEEE, 1998.

[116] Dongxiao Zhang, Liyong Li, and H. Tchelepi. Stochastic formulation for uncertainty analysis of two-phase flow in heterogeneous reservoirs. *SPE Journal*, 5(01):60–70, 2000.