Copyright

by

Cesar A. Mantilla

2010

The Dissertation Committee for Cesar A. Mantilla Certifies that this is the approved version of the following dissertation:

Feedback Control of Polymer Flooding Process Considering Geologic Uncertainty

Committee:

Sanjay Srinivasan, Supervisor

Chun Huh

Jairam Kamath

Quoc P. Nguyen

Gary A. Pope

Feedback Control of Polymer Flooding Process Considering Geologic Uncertainty

by

Cesar A. Mantilla, B. Sc., M.S.E.

Dissertation

Presented to the Faculty of the Graduate School of
The University of Texas at Austin
in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

The University of Texas at Austin
December 2010

Dedication

This dissertation is dedicated to Adriana, my ever-loving wife, who throughout these years always provided me with the love and endurance necessary to culminate this goal.

Acknowledgements

I would like to express my deepest gratitude to Dr. Sanjay Srinivasan, who supervised my work during these 5 years at UT. I am very thankful for his excellent technical guidance and his sincere personal advice that helped me mature as a researcher and a person. The conversations with him were always full of innovative and critical thinking and branded an indelible footprint on my personality as a researcher. He fostered in me the desire to be an excellent and stellar researcher. I feel very fortunate to have him as a mentor, an educator and a friend.

I am also thankful with the members of my committee for their questions and suggestions that made me think more critically about this project. My special thanks to Dr. Quoc Nguyen, who gave me advice in model updating and pushed me to publish my accomplishments. I thank Dr. Chun Huh for his expert advice on polymer modeling; it is difficult to find a more knowledgeable person in this area. I thank Dr. Gary Pope for his lessons on EOR and his unforgettable messages about maintaining a critical and skeptical attitude in research. I also thank Dr. Jairam Kamath, for his time and willingness to serve in this committee, and his advice and support during my internships.

I also want to thank the people who shared with me time at the office: Selin, Sayantan, Alvaro, Kiomars, Donovan, John, Brandon, Pradeep, Harpreet, Nitish and others whose names I do not recall now. It was a pleasure sharing time with them. Special thanks to my research buddy Dr. Juliana Leung for her sincere friendship and profound technical discussions. I sincerely admire her intelligence and brightness.

Thanks to my family in Colombia, my mom, Lucho, Maria Camila, my aunts Leonor, Amparo and Carmen for their infinite support that made me a better person. Finally, thanks to God, all this would not be possible without His grace.

Feedback Control of Polymer Flooding Process Considering Geologic Uncertainty

Publication No._____________

Cesar A. Mantilla, Ph. D.
The University of Texas at Austin, 2010

Supervisor: Sanjay Srinivasan

Polymer flooding is economically successful in reservoirs where the water flood mobility ratio is high, and/or the reservoir heterogeneity is adverse, because of the improved sweep resulting from the mobility-controlled oil displacement. The performance of a polymer flood can be further improved if the process is dynamically controlled using updated reservoir models and a closed-loop production optimization scheme is implemented. However, the formulation of an optimal production strategy is based on uncertain production forecasts resulting from uncertainty in spatial representation of reservoir heterogeneity, geologic scenarios, inaccurate modeling, scaling, just to cite a few factors. Assessing the uncertainty in reservoir modeling and transferring it to uncertainty in production forecasts is crucial for efficiently controlling the process. This dissertation presents a feedback control framework that (1) assesses uncertainty in reservoir modeling and production forecasts, (2) updates the prior uncertainty in reservoir models by integrating continuously monitored production data, and (3) formulates optimal injection/production rates for the updated reservoir models.

This approach focuses on assessing uncertainty in reservoir modeling and production forecasts originated mainly by uncertain geologic scenarios and spatial variations of reservoir properties (heterogeneity). This uncertainty is mapped in a metric space created by comparing multiple reservoir models and measuring differences in effective heterogeneity related to well connectivity and well responses characteristic of polymer flooding.

Continuously monitored production data is used to refine the uncertainty map using a Bayesian inversion algorithm. In contrast to classical approach of history matching by model perturbation, a model selection problem is implemented where highly probable reservoir models are selected to represent the posterior uncertainty in production forecasts. The model selection procedure yields the posterior uncertainty associated with the reservoir model. The production optimization problem is solved using the posterior models and a proxy model of polymer flooding to rapidly evaluate the objective function and response surfaces to represent the relationship between well controls and an economic objective function. The value of the feedback control framework is demonstrated with two examples of polymer flooding where the economic performance was maximized.

Table of Contents

List of Tables xiii

List of Figures xiv

Chapter 1: Introduction 1

1.1 Problem Description 1

1.2 Research Objectives 2

1.3 Methods of Solution 2

1.3.1 Method for Updating/Selection of Reservoir Models 3

1.3.2. Optimization of Well Conditions 4

1.3 Dissertation outline 5

Chapter 2: Background and Literature Review 6

2.1. Overview 6

2.2. Feedback Control of Enhanced Oil Recovery 6

2.3. Reservoir Model Updating and History Matching 8

2.3.1 Gradient-based Methods 9

2.3.2 Ensemble-Kalman filter 10

2.3.3 Probability Perturbation Method 12

2.4 Optimization of Well Controls for Enhanced Oil Recovery 14

2.4.1 Optimal Control Theory and Adjoint Models 15

2.4.2 Ensemble Based Optimization 18

2.4.3 Response Surface Methods 19

2.5 Polymer Flooding For Enhanced Oil Recovery 21

2.5.1 Modeling Polymer Flooding through Fractional Flow Theory...22

2.5.2 Modeling Polymer Flooding with UTCHEM 24

2.5.2.1 Polymer Retention in UTCHEM 25

2.5.2.2 Polymer Viscosity in UTCHEM 26

2.5.2.3 Permeability Reduction and Inaccessible Pore Volume28

2.6 Conclusion 28

Chapter 3: A Fast Proxy Model for Polymer Flooding..30
3.1. Purpose and Assumptions in the Proxy Model..................................30
  3.3.1 Assumptions..31
3.2. Random Walks..31
3.3. Particle Propagation Guided by Velocity Vectors..............................33
3.4. Calibration of Proxy Model..35
  3.3.2 Movement of Water Particles..38
    Estimation of Water Cut from Particle Count at Producers........38
  3.3.3 Polymer Particles..43
3.4 Inputs and Outputs from the Proxy Model...44
  3.4.1 Inputs..44
  3.4.2 Outputs...44
3.5 Correlation to Full Simulation Responses...46
3.6 Conclusion...50

Chapter 4: Organizing Multiple Reservoir Models using Measures of Dissimilarity 51
4.1. Overview...51
4.2. Motivation...51
4.3. Characterization of Heterogeneous Reservoir Models........................52
4.4. Generation of the Initial Ensemble of Reservoir Models....................55
4.5. Use of Proxy Model to Measure Well Connectivity Characteristics...56
  4.3.1 Particles Path Length Distribution...58
  4.3.2 Breakthrough Time and Recovery Factor..................................59
  4.3.3 Proxy Response for Well Pressure..62
  4.3.4 Summary of Proxy Responses..64
4.4 Measure of Dissimilarity between Reservoir Models..........................64
  4.4.1 Construction of a Metric Space using Multi-Dimensional Scale (MDS)..65
  4.4.2 Principal Component Analysis for Constructing a Metric Space67
  4.4.3 Correlation between Well Connectivity and Principal Components 69

Chapter 5: Selection of Reservoir Models using Production Data 72
- 5.1 Overview 72
- 5.2 Relationship between Production Data and Distance in Metric Space 72
- 5.3 Reservoir Model Selection Guided by Production Data 78
- 5.4 Computing and Updating Probability of Clusters of Reservoir Models 82
- 5.5 Hierarchical Selection of Reservoir Models 85
- 5.6 Reservoir Model Selection in a 5-Spot Inverted Injection Pattern 91
  - 5.5.1 Model Selection for a Well Pair 93
  - 5.5.2 New Reservoir Models Conditioned to Prior Selection 97
- 5.6 Conclusion 102

Chapter 6: Polymer Flooding Optimization using a Proxy Model and Response Surface 103
- 6.1 Overview 103
- 6.2 Production Optimization 103
- 6.3 Adaptive Response Surface for Production Optimization 104
- 6.4 Production Optimization of the Two-Well Example 110
- 6.5 Production Optimization of the Five-Well Example 114
- 6.6 Conclusion 117

Chapter 7: Feedback Control of Polymer Flooding 119
- 7.1 Feedback Control of Polymer Flooding 119
- 7.2 Implementation of the Feedback Control System to the Two-Well Reservoir Example 120
  - 7.2.1 Formulation of Optimal Control without Previous Production Data 120
  - 7.2.2 First Updating and Optimization of the Second Period 122
  - 7.2.3 Second Updating of Model Selection 127
  - 7.2.4 Feedback Control with Updated Selection of Models 133
- 7.3 Implementation of the Feedback Control System to the Five-Well Reservoir Example 136
  - 7.3.1 Optimizing the Second Period 136
  - 7.3.2 Updating the Model Selection after Water Breakthrough 139

7.4 Conclusion 147

Chapter 8: Conclusions and Future Work..149

8.1 Summary and General Conclusions..149

8.2 Specific Conclusions Based on the Research 150

8.3 Recommendations for Future Work..152

Appendix 1 – UTCHEM Deck Files for Cases Studied 155

A1-1. Two-Well Reservoir Model 155

A1-2. Five-Well Reservoir Model 161

Appendix 2 – Library of MATLAB® Functions to Interact with UTCHEM 169

A2-1. Reading Well and Production Reports from UTCHEM 169

A2-2. Writing UTCHEM Input File...170

A2-3. Reading a Grid Property 182

Appendix 3- Model Selection Algorithm 183

A3-1. Matlab Code for Model Selection Algorithm 183

A3-2. Matlab Code for Optimization using Response Surface...........190

Appendix -4 Computer Code of Proxy Model..193

A4-1. Program for Obtaining the Particle Length Distribution 193

A4-1.1. Parameter File 193

A4-1.2. Computer Code 193

A4-2. Computer Code for Estimation of Recovery Factor and Breakthrough Time 200

A4-3. Program for Economic Parameters 206

A4-3.1. Parameter File 206

A4-3.2. Computer Code 206

References...221

Vita 226

List of Tables

Table 3-1 Rate schedule for multi-well test case 41
Table 6-1 Results of a significance test on regression coefficients for a response surface with 4 rate controls. Rows in gray are significant at 0.1 level.105
Table 6-2 Input Parameters for Economic Analysis 111
Table 6-3. Optimal control settings for the five wells corresponding to the reservoir model shown in Figure 6-4. Well rates in ft^3/day 115
Table 7-1 Results of control strategies for a random model, feedback control system and the reference reservoir...136
Table 7-2 Expected value of optimal control settings for the five-well reservoir, starting at 250 days 138
Table 7-3 – Criteria used to screen reservoir models for the new ensemble 140
Table 7-4 Expected value of optimal control settings for the five-well reservoir, starting at 500 days 145

List of Figures

Figure 2-1 Generic components of a feedback control system. Adapted from Burns (2001) 7

Figure 2-2 Fractional flow curves used for determination of saturation and water cut at water and polymer fronts 24

Figure 2-3 Polymer retention as a function of aqueous phase effective permeability. Adapted from Huh, *et al.* (1990b).......26

Figure 2-4 Polymer viscosity as a function of polymer concentration and brine salinity. Data used was taken from Dakhlia (1995).......27

Figure 2-5 Polymer viscosity-shear rate relationship inputted in UTCHEM for the polymer flooding simulations in this dissertation. Data used was taken from Dakhlia (1995) 28

Figure 3-1 Illustration of the movement of a particle between grid-block centers in two steps. a) the movement initiates in x-direction; b) in the first time step, the particle travels a distance less than the separation between centers; and c) during the second time step the particles arrives to the center of the adjacent cell and the new transitioning probability is sampled for the next movement.......34

Figure 3-2 Reservoir model used for comparison of production profiles from proxy model and UTCHEM. Polymer is injected at the well located at the bottom 36

Figure 3-3 Reservoir model with an inverted 5-spot pattern injection used to calibrate the proxy model. 36

Figure 3-4 Relative permeability curves for the calibration cases.......37

Figure 3-5 The velocity of the shock front in a unit mobility ratio, water-oil and polymer-oil systems is proportional to the slope of the tangent lines drawn from the initial saturation to the respective fractional flow curve. 37

Figure 3-6 Water cut profiles obtained from proxy model and UTCHEM for the reservoir model shown in Figure 3-2 40

Figure 3-7 Random walkers representing the water phase are assigned to producers according to their proximity to the wells. Position of water particles at 500 days. 42

Figure 3-8 Field water cut profiles obtained from proxy model and UTCHEM for the reservoir model shown in Figure 3-3 (5-spot inverted injection pattern) 42

Figure 3-9 Polymer saturation fronts from the proxy model and UTCHEM. Position of polymer particles represents the spread of the polymer front in the proxy model. 44

Figure 3-10 Variance of proxy responses as a function of the number of particles used in the proxy model. The ensemble size over which the variance stabilizes is a function of the transition probability used to regulate the movement of particles and that in turn is controlled by the nature of heterogeneity exhibited by the geology. 45

Figure 3-11 Histogram of the net present value obtained from the proxy model using 10000 particles. 46

Figure 3-12 Correlation between recovery factor estimated by the proxy model and UTCHEM for 100 different injection/production rate schedules. 47

Figure 3-13 Correlation between the net present value at 2000 days estimated by the proxy model and UTCHEM for 100 different rate schedules..........48

Figure 3-14 Samples of geologic models used to test the correlation of the proxy model and UTCHEM...49

Figure 3-15 Correlation between recovery factor estimated by the proxy model and UTCHEM for 100 geologic models..49

Figure 3-16 Correlation between the net present value at 2000 days estimated by the proxy model and UTCHEM for 100 geologic models.....................50

Figure 4-1 Recovery curves for polymer flooding simulated with UTCHEM for 50 reservoir models that reflect the same variogram model.................53

Figure 4-2 Recovery curves for polymer flooding simulated using UTCHEM for 50 reservoir models generated constrained to the statistics retrieved from the same training image. 54

Figure 4-3 Examples of realizations generated using *sisim* and *snesim*. A total of 600 reservoir models were generated. The injector is located at the bottom center and the producer at the top center. 57

Figure 4-4 a) Histogram of path length for a homogeneous medium. b) Probable particle paths. 1000 particles were sent from the injector. Color scale indicates the number of particles that passed through each location on their way to the producer. 58

Figure 4-5 a) Histogram of path length for a medium with a high permeability streak. b) Probable particle path is preferentially through the high permeability streak. 59

Figure 4-6 Correlation between water breakthrough time from UTCHEM and the breakthrough of particles from the proxy model. 60

Figure 4-7 Correlation between initial pressure drawdown and the breakthrough time of particles from the proxy model..61

Figure 4-8 Correlation between recovery factors at 0.27 pore volume injected from UTCHEM and recovery factor after injecting 0.5 pore volume from the proxy model. 61

Figure 4-9 Correlation between injection pressure at early time and effective permeability around well calculated using the proxy. 63

Figure 4-10 Correlation between production pressure close to breakthrough and the effective permeability around the well calculated using the proxy. .63

Figure 4-11 Projection of 600 reservoir models on top 2 eigenvectors. Each dot represents one reservoir model. 69

Figure 4-12 Clustering of reservoir models based on Euclidean distance in the top 2 eigenvectors. Average permeability maps from each cluster are displayed. The eight clusters obtained using k-means are color coded. 71

Figure 5-1 Simulated water cut for 600 reservoir models after polymer injection for 250 days. 73

Figure 5-2 Simulated injection pressure for 600 reservoir models after polymer injection for 250 days..74

Figure 5-3 Simulated producer pressure for 600 reservoir models after polymer injection for 250 days..74

Figure 5-4 Projection of 600 reservoir models on top 3 eigen vectors.................76

Figure 5-5 Reservoir models labeled in red and blue were selected for flow simulation using UTCHEM because they belong to the two farthest clusters. ..77

Figure 5-6 Production data for reservoir models belonging to the two distant clusters indicated in Figure 5-5. 78
Figure 5-7 Facies map of reference reservoir with two wells...............................80
Figure 5-8 Comparison of water saturation corresponding to water (left) and polymer (right) flooding after 0.3 pore volumes of injection simulated using UTCHEM 80
Figure 5-9 Comparison of net present value for water and polymer flooding.......81
Figure 5-10 Production data from reference reservoir after the initial 250 days of polymer injection. 82
Figure 5-11Uncertainty envelopes around reference data. The production data of one reservoir model falls within the 66% probability envelope. 84
Figure 5-12 Projection of 600 reservoir models on space defined by the two top eigen vectors. Initial ensemble of reservoir models is partitioned in three large clusters: cluster 1 (blue), cluster 2 (green) and cluster 3 (gray)........85
Figure 5-13 Production data from samples of cluster 1 (blue), cluster 2 (green) and cluster 3 (gray) are compared to the reference production data (red).86
Figure 5-14 First and second level of selection of reservoir models 87
Figure 5-15 Variation in injection and producer pressure over the models after the second level of selection. 87
Figure 5-16 Cluster of reservoir models retained at the second and third levels of selection 88
Figure 5-17 Production data of reservoir models retained at the third level of model selection. 88
Figure 5-18 Production data computed on the final suite of reservoir models. The reference response is shown in red. 89

Figure 5-19 Facies map of four reservoir models from the final set. 90
Figure 5-20 Facies map of reference reservoir with four producers at the corners and one injector at the center 91
Figure 5-21 Well pressure at 4 producers and 1 injector in the reference reservoir (Figure 5-20) for the first 250 days..92
Figure 5-22 Illustration of the process for updating the model selection using data from multiple wells. 93
Figure 5-23 Examples of reservoir models included in the initial ensemble.........94
Figure 5-24. First two levels of selection in example of inverted 5-spot injection pattern 95
Figure 5-25 Final two levels of selection in example of inverted 5-spot injection pattern 95
Figure 5-26 Well pressure at Inj#1 and Prod#1 (First well pair) for initial ensemble compared to the final suite. 96
Figure 5-27 Maps of 6 reservoir models in the final set obtained by using production data from Inj#1 and Prod#1 (labeled in the maps) for model selection. 97
Figure 5-28 Four reservoir models of the new ensemble generated with the combined local probabilities for each well pairs. The new ensemble is composed of 100 reservoir models. This is the starting data set for the refinement of the selection. 98
Figure 5-29 Well pressures for the reference, and reservoir models of the initial and new ensembles. 99
Figure 5-30 Well pressures for the reference and final selection of reservoir models after the refinement step..100

Figure 5-31 Selected realizations from the final set after model selection using the data from the entire well pattern. 101

Figure 6-1 Response surface fitted for optimization using initial points (left) and adapted after further function evaluations around the optimal (right). The blue dots are the initial function evaluations and the red dots are the new function evaluations after each iteration. 109

Figure 6-2 Reference reservoir model with 1 injector and 4 producers reproduced from Figure 5-7 110

Figure 6-3 Expected value of the optimal rate sequence for the two-well case ..112

Figure 6-4 Net present value for optimal rate solutions from multiple runs and for the expected optimal rate sequence. 113

Figure 6-5 Reference reservoir model with 1 injector and 4 producers reproduced from Figure 5-20 114

Figure 6-6 a) Net present value curves and b) recovery curves for 10 optimization runs (gray). Curves in black correspond to the expected value of the 10 optimization runs. Curves in blue are for an equal production rates schedule. Curves in red correspond to a simulation operated at the expected value of the optimal rates...116

Figure 6-7 Water saturation map at 2000 days for a) optimal control rates and b) evenly distributed production rates. Regions with major difference in polymer saturations are indicated in the maps. 117

Figure 7-1 Flowchart of the feedback control system for polymer flooding.......120

Figure 7-2 Reservoir models randomly selected to be submitted to the optimization algorithm 121

Figure 7-3 Optimal control rates calculated for the reservoir models randomly selected before data integration 121

Figure 7-4 Production data for reference reservoir and 4 randomly selected realizations during the initial control period..................................122

Figure 7-5 Production data for the “true” reservoir and 4 reservoir models from the reduced set at the end of the model selection process 123

Figure 7-6. Assessment of water saturation profile in the selected reservoir models. The figure on the left shows the average water saturation over the 4 posterior reservoir models retained. The figure on the right shows the variances over the same set of reservoir models............................124

Figure 7-7 Optimal control settings for the reduced set of reservoir models after using production data for 250 days. The optimization starts from 250 days to 2000 days 125

Figure 7-8 Net present value profile corresponding to the optimal control vector of each of the four selected realizations (blue) compared to the NPV profile corresponding to the expected value of the control vector (red) calculated using Equation 7.1. 126

Figure 7-9 Production response of the reference reservoir compared to the simulated response for the 4 reservoir models over the first two control periods (500 days)..127

Figure 7-10 Water saturation in the reference model after 500 days of operation128

Figure 7-11 Fifty reservoir models in the augmented ensemble honor the spatial statistics of the above training image..129

Figure 7-12 Map of probability of having a high permeability rock type from the selection of models at 250 days 130

Figure 7-13 Comparison of production data of prior and posterior ensembles of reservoir models compared to the reference. These results were obtained after the prior ensemble was augmented so as to include additional models with channels exhibiting complex connectivity. 131

Figure 7-14 Production data for the final set of reservoir models 132

Figure 7-15 Four reservoir models selected from the final set after incorporating production data of 500 days 133

Figure 7-16 Optimal control settings for the selected four realizations after model selection using data gathered over 500 days. 134

Figure 7-17 Production data after 8 control periods (2000 days) for the selected reservoir models and the reference. 135

Figure 7-18 Optimal control settings for 4 realizations selected after updating the model selection with production data of 250 days. Control settings of Prod#4 are deduced from equilibrium between injection and production rates. 137

Figure 7-19 Production data for reference and set of reservoir models selected after the first control period (250 days)...139

Figure 7-20 Illustration of the model selection process. Projection of reservoir models on top-3 eigenvectors (right) and water cut evaluated for reservoir models selected from the clusters in the right. Production data and clusters are color coded...141

Figure 7-21 Field production data from selected models after 500 days of production is compared to the monitored water cut and cumulative oil production in the true reservoir 142

Figure 7-22 Production data simulated by the selected reservoir models selected (blue) is compared to the pressure and water cut observed from the reference (red)..143
Figure 7-23 Six reservoir models from the selected set after updating with 500 days of production data..144
Figure 7-24 Production data simulated by the selected reservoir models selected (blue) is compared to the pressure and water cut observed from the reference (red)..146
Figure 7-25 Field water cut and cumulative oil produced for selected reservoir models (blue) and reference (red)..147
Figure 7-26 Comparison of net present value curves obtained producing 100 ft3/day at all producers and the optimal control settings 147

Chapter 1: Introduction

1.1 PROBLEM DESCRIPTION

Closed-loop reservoir management is a decision making process in which the operation of the reservoir is controlled to optimize an economic objective function. The formulation of the control strategy is highly dependent on production forecasts, which are uncertain due to the uncertainty in the reservoir model. The main sources of uncertainty in reservoir modeling are heterogeneity, spatial resolution, the mathematical model itself, and numerical inaccuracies. Among these sources, reservoir heterogeneity is the major contributor, which is highly dependent on the geological environment of the reservoir. The main problem is how to characterize the uncertainty in production forecasting by integrating uncertainty in reservoir heterogeneity and all information available to us (including production history). Once this problem is solved, the production optimization process can be undertaken so as to yield better decisions for reservoir development. Solving this problem for polymer flooding is important because heterogeneity has an important influence on the displacement of the polymer front and also because polymers are important for ensuring increased efficiency of chemical enhanced oil recovery techniques.

The main problem can be divided into three sub-problems: (1) how to represent the uncertainty in reservoir heterogeneity while considering ambiguity in geologic interpretations and data scarcity; (2) how to update the uncertainty in geologic models such that well measurements are integrated and production forecasts are more accurate, and, (3) how to optimize well operations so as to maximize a particular economic performance indicator given the updated information about the reservoir?

1.2 RESEARCH OBJECTIVES

The primary objective of this research is to formulate a framework to maximize the performance of polymer flooding that utilizes dynamic data from wells to update prior reservoir models, and subsequently uses an optimization algorithm to find the optimum well control strategy that maximizes the project economics. To accomplish this objective, the following steps are formulated:

- To characterize uncertainty in reservoir modeling and forecasting considering multiple geologic scenarios.
- To update the prior uncertainty using past production data such that the uncertainty in production forecasts is refined.
- To optimize well operating conditions based on the updated production forecasts in order to maximize the economic profit.

1.3 METHODS OF SOLUTION

Within the scope of this research, the two major focal points are reservoir model updating and the optimization of well controls. The method for model updating addresses the first two objectives because production data is integrated into a prior set of reservoir models, such that the uncertainty in production forecasting is represented by the updated set of models. Subsequently, the third objective is achieved by the production optimization method, so that the closed-loop framework is completed. . Polymer flooding was selected to demonstrate the closed-loop optimization framework because reservoir heterogeneity has a subtle and important influence on the displacement of the polymer front and also because polymers are important for ensuring increased efficiency of chemical enhanced oil recovery techniques.

1.3.1 Method for Updating/Selection of Reservoir Models

Geological models of reservoirs constructed using static data (well logs, seismic, etc.) can vary greatly due to the subjectivity of the data interpretation, resulting in different reservoir architectures, depositional environments, etc. Conceptual models of reservoir geology are expressed through geostatistical parameters such as a variogram model for two-point statistics or a training image for multiple point statistics. However, the geological interpretation of the reservoir depositional model itself is a major source of uncertainty. A key research objective is to develop a method that considers a large ensemble of initial reservoir models, which are representative of the prior uncertainty. From this suite of prior models, a subset of posterior models are selected that best reproduce the production history. The final set of models, selected from the initial ensemble, may come from different interpretations of the geological environment, but share the same connectivity characteristics that influence flow performance and are anchored to the same well log and core data as well as any available seismic information.

There are two important aspects of the presented method that distinguish it from methods traditionally used for history matching. First, there is shift in paradigm from a model perturbation problem (conventional methods) to a model selection problem (this approach). Conventional methods for reservoir history matching perturb an initial reservoir model until a match with the observed production data is obtained. In contrast, the method developed in this dissertation works with the model taken as a whole, and evaluates whether the connectivity characteristics exhibited by it are similar to other models within a cluster. The observed production history is then used to select the cluster that exhibits production performance closest to the observation. The second aspect is that in the traditional algorithms, the history matching scheme is too entwined with the process of generation of reservoir models. In other words, the scheme for integrating production data in the reservoir model is specific to the algorithm used for generating the spatial variations in reservoir attributes. In the method presented, the process of generation of reservoir models is completely dissociated from the task of selecting an optimum cluster of models that exhibit the production characteristics observed.

The following is a brief outline of the presented method. The first step is to group realizations according to their similarities in terms of some connectivity measure(s) such as well communication. It is presumed that reservoir models with similar characteristics in connectivity/well communication share similar production data. Conversely, it is expected that reservoir models with dissimilar characteristics in well communication exhibit dissimilar production data. Based on the above assumption, the production data evaluated through reservoir simulation of one of the models is a good representation of the production data from all the members of that group. The deviation of the production response of a representative member of a cluster from the observed production history is used to compute the posterior probability for selecting that cluster. A Bayesian scheme is presented for accomplishing this. A cluster is sampled on the basis of the posterior probability and the process of cluster refinement is continued until the calculated posterior probabilities make it impossible to distinguish between clusters.

1.3.2. Optimization of Well Conditions

The third research objective is to develop a procedure that determines optimal well controls which maximize the economic net present value of a polymer flooding process. A naïve approach would be to define an objective function that reflects the objective of the polymer flooding project, and to identify the controllable well conditions that have the greatest impact on the objective function. The problem is highly complex and difficult to manage because the number of control variables can be large and repeated execution of a reservoir simulator for polymer flooding, can quickly make the task infeasible.

For the above reasons, the application of more rigorous methods based on optimal control theory or adjoint models is difficult. This difficulty underlines the necessity of adopting tools like proxy models and response surfaces in order to explore the combinatorials of control parameters. Thus, a proxy model for polymer flooding that estimates the objective function as a function of available well controls was developed for rapid evaluation. The developed workflow is as follows: first, the proxy model of polymer flooding is used for function evaluation, then a response surface is constructed from the responses of the proxy model, and finally the optimization problem is solved over the response surface. The developed methodology results in a more efficient implementation of optimum control schemes.

1.3 DISSERTATION OUTLINE

Chapter 2 discusses relevant works on reservoir history matching, model updating and production optimization, and provides a background on polymer flooding. Chapter 3 presents the algorithm developed for rapid polymer flooding simulation, which is extensively used throughout the rest of the dissertation. Chapter 4 covers the method for uncertainty characterization using a distance-metric approach. Chapter 5 presents the method for selection and updating of reservoir models using production data, which is illustrated by an example based on synthetic reservoir cases. Chapter 6 presents the production optimization algorithm, which is applied on the example in Chapter 5. In Chapter 7, the model selection and production optimization algorithms are coupled, closing the feedback control loop. Finally, this dissertation is concluded in Chapter 8 with the main findings of this study and the proposed future work.

Chapter 2: Background and Literature Review

2.1. OVERVIEW

This chapter starts with a review of the main components of a feedback control system and its the applicability to enhanced oil recovery. This is followed by a critical review of traditional and modern methods for reservoir history matching that also summarizes some of the main issues addressed in this dissertation. After that, a revision of recent works on closed-loop reservoir management demonstrates the need for efficient methods for production optimization of enhanced oil recovery processes. The final section of this chapter provides background on polymer flooding, including the application of fractional flow theory for polymer flooding and simulation with UTCHEM.

2.2. FEEDBACK CONTROL OF ENHANCED OIL RECOVERY

Control engineering is applicable in many industrial settings like in chemical plants, aircraft navigation or financial systems to name the most common fields of application. The objective of a control system is to manipulate the input of a system such that the corresponding output matches a desired value. In 1961, Kalman (1961) introduced the dual concept of controllability and observability. Kalman demonstrated that when the system dynamic equations are linear and the objective function is quadratic, then the mathematical problem has an explicit solution, making possible the concept of feedback control or closed-loop systems. Burns (2001) presents the main components of a closed-loop control system from a control engineering viewpoint as illustrated in Figure 2-1. A system is a process that produces an output as a response to a controllable input. In this dissertation, the system to be controlled is the reservoir and the controllable input is the operation of wells. A controller is the set of control instruments that takes the desired value and modifies the dynamic state of the system through controllable inputs. With the advent of intelligent completion tools, a controller in an enhanced oil recovery process can be down-hole chokes for flow rate regulation. Permanent down-hole gauges or reservoir surveillance tools are sensors that take the response of the reservoir and provide the operator with actual measurements. Knowledge of the system input together with the mathematical model allows the outputs to be calculated and compared against the actual measurements.

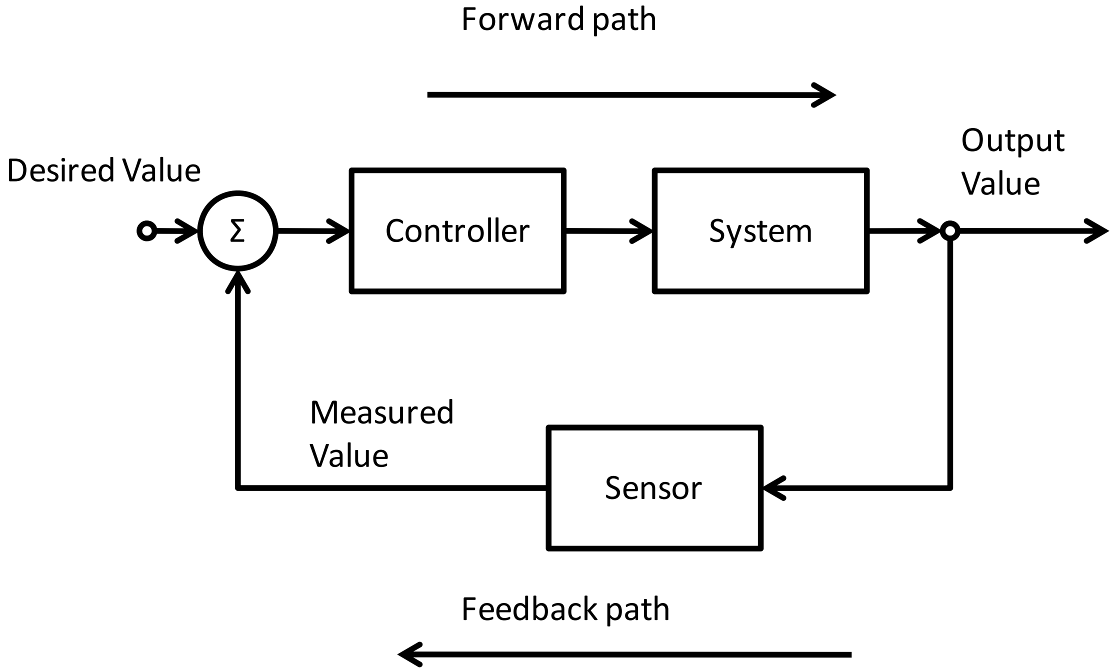

Figure 2-1 Generic components of a feedback control system. Adapted from Burns (2001)

According to Franklin *et al.* (2010) the basic requirements for a feedback control system are: (1) stability, (2) controllability and (3) observability. Feedback control cannot be applied to a system that becomes unstable after a control has been implemented or if the control is turned off. In most cases, a reservoir is a naturally stable system where the effect of well interventions takes place slowly. Controllability refers to the possibility of forcing changes in the dynamic state of the system using a control signal. A reservoir is a controllable system because its dynamic state is directly affected by well operations which provide the control signals. Observability refers to the ability to monitor the response of the system to a given control input. A reservoir is observable through pressure sensors, fluid samples, etc. Hence, it is possible to conceive of a feedback control scheme to oil recovery processes.

However, the following complications arise when a feedback control scheme is implemented for enhanced oil recovery: (1) the mathematical model is a computationally expensive reservoir simulator, (2) the physical process does not occur at the same scale as the resolution of a reservoir simulator, (3) the uncertainty in the reservoir simulation model due to reservoir heterogeneity can turn the calculated outputs highly uncertain, and (4) the time elapsed from the implementation of a control input to the observation of a response can be long, thereby rendering the process of calibration of mathematical model difficult.

Closed-loop systems for reservoir management have to deal with these complications. Brouwer *et al*. (2004), Sarma *et al*. (2005) and Sarma *et at*. (2006) have implemented feedback control procedures for water flooding management by coupling model updating and optimization methods. Despite these applications in reservoir engineering, a practical feedback control workflow for chemical flooding is still lacking.

2.3. RESERVOIR MODEL UPDATING AND HISTORY MATCHING

In order to implement feedback control it is crucial to have a mathematical model that delivers reliable predictions of reservoir response as a function of the control variables. In a feedback control process, the output measured by the sensors reflects the dynamic state of the system. The measurement from sensors is fed back into the mathematical model to improve the quality of the predictions. For this reason it is important to have a procedure to update the mathematical model using the observed responses, and this is termed as production history matching in reservoir engineering.

History matching methods aim to condition prior reservoir model to dynamic data (also called production data) observed in the field. The variables in reservoir flow simulation can be broadly classified in static variables, i.e. porosity, permeability, relative permeability curves, etc., and dynamic variables, i.e. pressure, saturation, water cut, well pressure, etc. Initial reservoir models are constructed based on scarce direct measurements of static variables such as well logs and indirect data such as seismic interpretation. Nevertheless, dynamic well variables such as well water cuts, flowing pressures obtained from flow simulation often differ from observations at wells.

Traditional methods for history matching generally have a model perturbation step. That is, they perturb prior reservoir models until the mismatch between the simulated production data and the actual production data is minimized below certain tolerance. The intricate relation between static and dynamic variables, and the large number of equations involved make this task difficult and the resultant solution non-unique. History matching methods can be broadly classified into gradient based methods, ensemble-based methods and stochastic algorithms.

2.3.1 Gradient-based Methods

Gradient-based methods present the history matching problem as an optimization problem. The objective function to be minimized is the squared error between the simulated data and the observed data. History matching is achieved by perturbing the initial model using the gradient of the objective function with respect to each uncertain variable. The following shortcomings make gradient-based methods unsuitable for rapid updating of reservoir models:

- There is no guarantee that the posterior model preserves the prior geologic information unless some measurement of geologic consistency is introduced in the objective function.
- Only one reservoir model is obtained at the end, and that model depends entirely on the prior model. It is desirable to obtain multiple models for uncertainty assessment.
- The calculation of gradients or sensitivity coefficients is generally based on a linearized form of the flow equations. This poses problems in reservoirs exhibiting complex geologic features that causes the response to be highly non-linearly related to the reservoir heterogeneity.

2.3.2 Ensemble-Kalman filter

Ensemble Kalman filter (Evensen, 1994) is an evolution from the classical Kalman filter (Kalman, 1960), one of the pioneering works in closed-loop control systems. Kalman filter considers the system as a random process that can be modeled as a Markov Chain process. The state variables are all the variables that define the state of the system at any given time and these may include dynamic and static variables. Kalman filter utilizes the Kalman gain that captures the relationship between the state variables and the observations to update the prior state variables. The state variables are updated so as to reflect better the observations and the Kalman gain is updated by assimilating the observations. However, the Kalman gain in the traditional form of the filter is formulated for a forward model that is approximately linear. In order to generalize the filter to non-linear transfer functions, the Ensemble Kalman Filter (EnKF) was devised by Evensen (1994) in which the traditional Kalman gain is replaced by an experimental covariance matrix obtained from an ensemble of realizations of the state space. The term "realization" is used throughout this dissertation to refer to a reservoir model, which is an observation from a random variable. The complete derivation of Ensemble Kalman filter equations can be found in Evensen (2007). The updating equation (Equation 2.1) has been derived to guarantee that the posterior density function is maximized if the prior density function is Gaussian.

$$\hat{\mathbf{x}}_k^{+} = \hat{\mathbf{x}}_k^{-} + \mathbf{C}_{x,k}^{-}\mathbf{H}^T\left(\mathbf{H}\mathbf{C}_{x,k}^{-}\mathbf{H}^T + \mathbf{R}\right)^{-1}\left(\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_k^{-}\right) \tag{2.1}$$

In the above equation $\mathbf{x}$ is the state vector that contains all variables that define a realization of the ensemble. The superscripts + and – indicates posterior (or updated) and prior estimates for the state variables respectively. $\mathbf{C}$ is the covariance matrix linking the state variable $\mathbf{x}$ to the observed data $\mathbf{z}_k$, $\mathbf{H}$ is a matrix that when it is multiplied by the state vector $\hat{\mathbf{x}}_{\bar{k}}$ returns the simulated observations $\hat{\mathbf{z}}_k$. $\mathbf{R}$ is the noise covariance matrix, which is assumed to be unbiased and Gaussian. $\mathbf{z}$ is a vector that contains the observations from the actual system (reservoir and wells). The subscript k denotes the kth realization from the ensemble.

EnKF is attractive for reservoir model updating because the implementation is easy: it uses the reservoir simulator as a black-box that outputs production data (simulated observations) as a function of the state of the system (static and dynamic variables). In that way, the complexity of flow equations is by-passed. Another attractive property is the simplicity of the linear equation used to update the state of the system. Advocates of EnKF also claim that it allows refining the uncertainty in reservoir performance using responses from multiple realizations calibrated against measured data. Another advantage of EnKF is that unlike traditional history matching methods, EnKF integrates new data incrementally rather than cumulatively from the start of production.

In reservoir simulation this is a great advantage because the update can be done through restart files.

Despite the above attractive properties of EnKF, many authors have pointed out some pitfalls that limit its application to reservoir model updating. First, the prior probability density function is assumed to be Gaussian in the derivation of the updating equation. Although the distribution of many reservoir properties can be assumed to be Gaussian, some rock properties like permeability do not follow a Gaussian distribution; for instance, the facies distribution is multimodal. The consequence of evolving a non-Gaussian distribution through a linear update equation is that the distribution of the updated ensemble progressively tends to be Gaussian as shown by Zhou *et al.* (2010).

A second drawback of EnKF is that the state variables are updated with data forward in time without validating prior matches. This means that it looks to minimize the error for the next time step regardless of the results from previous time steps. Thus, geologic consistency of the final models can be compromised. Mantilla *et al.* (2007) showed a particular application of EnKF for water coning where the experimental variogram of the final model is inconsistent with the prior geologic model. This might also lead to inconsistency between the updated state variables and static variables because the update proceeds without explicitly account for the interdependencies between the static and dynamic state variables.

2.3.3 Probability Perturbation Method

Since multiple reservoir models can match the observed production history, it is worthwhile to consider a stochastic approach to history matching that can explicitly calibrate the information in dynamic data and use that information to update the prior uncertainty in reservoir model to the posterior uncertainty conditioned to the available data. This method considers $X(\mathbf{u})$ as a random variable where X is the reservoir property and $\mathbf{u}$ the location vector. In probability perturbation methods, the objective is to integrate two conditional probabilities: (1) the conditional probability to the geology $P\{X(\mathbf{u})|Geology\}$, and (2) the conditional probability to the observed dynamic data $P\{X(\mathbf{u})|Dynamic\ data\}$. The integrated conditional probability $P\{X(\mathbf{u})|Geology, Dynamic\ data\}$ is the updated posterior distribution. Sequential simulation methods like *sgsim*, *sisim* or *snesim* can generate multiple realizations of reservoir models conditioned to the geological data and assess the conditional probability $P\{X(\mathbf{u})|Geology\}$.

For obtaining the probability conditioned to dynamic data, the probability density function at each location is expressed as a function of a deformation parameter. Therefore transitions from one realization to other are obtained upon perturbation of that probability using the deformation parameter. Subsequently, a 1D minimization procedure such as Dekker-Brent algorithm is implemented to perturb the deformation parameter so as to minimize the mismatch between the observed and simulated dynamic data. Upon convergence, the conditional probability $P\{X(\mathbf{u})|Dynamic\ data\}$ can be assessed. The two conditional probabilities are merged using the permanence of ratio hypothesis formulated by Journel (2002). As a result, the joint conditional probability distribution $P\{X(\mathbf{u})|Geology, Dynamic\ data\}$ is obtained and realizations sampled from that joint probability distribution honor both the prior geology and the dynamic data.

This probabilistic approach estimate the probability distribution of a variable node by node, and identify regions where it is highly probable to find high permeability streaks, low-permeability barriers, etc. Hofman and Caers (2004), Yadav *et al.* (2005), Mantilla *et al.* (2010) are applications of this methodology that have been successfully implemented.

Although the final models obtained by probability perturbation are geologically consistent, they are constrained only to one geologic scenario (the prior geologic model). In reality, the scarce hard data available is not enough to identify a unique prior model. Multiple geologic interpretations can emerge, introducing additional uncertainty in the final history matched model. Probability perturbation methods neglect this source of uncertainty.

2.4 OPTIMIZATION OF WELL CONTROLS FOR ENHANCED OIL RECOVERY

The efficiency of a polymer flooding process can be improved by implementing optimum operating conditions at the wells. Economic performance indices like net present value (NPV) are usually chosen as cost function in enhanced oil recovery processes (Ramirez *et. al.* (1986), Sarma *et. al.* (2005) and Kraijevanger *et. al.* (2007)). Vaskas (1996) developed a comprehensive chemical flooding economic model (CFEM) to evaluate the economics of chemical flooding processes. CFEM include chemical oil recovery, capital costs, operating costs, taxation rates, oil price, etc. Vaskas (1996) recommended investment efficiency for evaluating and optimizing chemical flooding projects. Regardless of the explicit formula, any objective function with fixed final time related to the cumulative recovery is convex because the original oil in place is finite. Thus, gradient-based methods for optimization are suitable for this problem.

The search for optimal control settings starts from an initial guess of the well control settings, and continues guided by the gradient of the objective function towards the optimal control settings where the gradient is zero or the feasibility bounds are encountered. The question is how to obtain that gradient from a complex mathematical model like a flow simulator, where several variables influence the solution. The following section reviews applications of optimization algorithms to enhanced oil recovery processes, classified according to the method employed to obtain the required gradient.

2.4.1 Optimal Control Theory and Adjoint Models

In optimal control theory a performance index needs to be minimized as a function of a set of control variables implemented at discrete time intervals subject to equality and inequality constraints. In polymer flooding optimization, the performance index (*J*) can be an economic index such as the net present value (2.2); the control variables (**u**) are the production and injection controls at the wells such as injection and production rate and/or pressure, polymer concentration and slug size. The constraints are the equations such as the continuity and pressure equations (such as in Eq. 2.3) that the mathematical model should satisfy, and the operational limits of the wells or production facilities (such as in Eq. 2.4). Usually the problem is formally presented as:

Minimize

$$J = \int_0^T \frac{\left(\omega_w q_w(t) + \omega_o q_o(t) + \omega_w{}^{inj} q_{w,inj}(t)\right)}{(1+i)^t} dt = \sum_{n=1}^{N} \frac{\left(\omega_w q_w{}^n + \omega_w q_o{}^n + \omega_w{}^{inj} q_{w,inj}{}^n\right)}{(1+i)^{n\Delta t}} \Delta t \tag{2.2}$$

Subject to

$$\mathbf{f}^n\left(\mathbf{x}^n, \mathbf{x}^{n-1}, \mathbf{u}^n\right) = \mathbf{0} \tag{2.3}$$

$$\mathbf{LB}^n \le \mathbf{u}^n \le \mathbf{UB}^n \tag{2.4}$$

ω_w, ω_o and $\omega_w{}^{inj}$ are weights assigned to water production (q_w), oil production (q_o) and injection ($q_{w,inj}$) rates respectively. These weights are function of the revenue and costs associated with the production, injection and processing of a particular fluid phase. N is the total number of time steps in which the process has been discretized, while n is the time step index. **f** is the set of flow equations that the reservoir simulator solves. $\mathbf{x}^n$ is the vector that contains the state variables that describe the system in the simulation, for example the pressure and saturation at all grid-blocks. The control vector (**u**) has lower (**LB**) and upper (**UB**) bounds to ensure that the solution is physically feasible.

Implementation of the method of Lagrange multipliers and Karush–Kuhn–Tucker (Hull, 2003) necessary conditions for optimality leads to a set of co-state or adjoint equations (2.5) and (2.6).

$$\boldsymbol{\lambda}^{TN} = -\left[\frac{\partial J^{N-1}}{\partial \mathbf{x}^{N}}\right]\left[\frac{\partial \mathbf{f}^{N-1}}{\partial \mathbf{x}^{N}}\right]^{-1} \tag{2.5}$$

$$\boldsymbol{\lambda}^{Tn} = -\left[\frac{\partial J^{n-1}}{\partial \mathbf{x}^{n}} + \boldsymbol{\lambda}^{T(n+1)}\frac{\partial \mathbf{f}^{n}}{\partial \mathbf{x}^{n}}\right]\left[\frac{\partial \mathbf{f}^{n-1}}{\partial \mathbf{x}^{n}}\right]^{-1} \tag{2.6}$$

The superscript T denotes transpose. First, the Lagrange multipliers corresponding to the final time step ($\boldsymbol{\lambda}^{TN}$) are solved using Equation 2.5 with the gradient of the objective function and the flow equations at the final state. Then, Equation 2.6 is used sequentially to calculate the Lagrange multipliers corresponding to the previous time steps ($\boldsymbol{\lambda}^{Tn}$) using the gradient of the objective function and the flow equations along with the previously calculated Lagrange multiplier ($\boldsymbol{\lambda}^{T(n+1)}$). A major drawback of this backward solution scheme is that it is required to store all the state variables for each time step calculated during the forward simulation. In addition, the calculation of the partial derivatives ($\partial\mathbf{f}/\partial\mathbf{x}$) requires the explicit knowledge of the flow equation and its derivatives.

In enhanced oil recovery, optimal control theory was first applied for optimization of surfactant (Porzucek, 1990), micellar/polymer (Fathi and Ramirez, 1984), and CO_2 flooding (Mehos, 1989), in relatively simple homogeneous porous media. For a micellar/polymer one-dimensional displacement, Fathi and Ramirez (1984) computed the gradient numerically, although it was a complex and time consuming process. Ramirez (1987) and Porzucek (1987) extended the optimization of micellar/polymer flooding to a 2D homogeneous reservoir using streamline simulation. The computational procedure for the Jacobians was the same as in Fathi and Ramirez (1984). Since the permeability of the porous medium is assumed to be known, no model updating was necessary in the above studies. Sudaryanto and Yorstos (2001) combined optimal control theory with a "bang-bang" injection policy for dynamic optimization of water flooding at laboratory scale.

More recently, Brouwer *et al*. (2004) and Sarma *et al*. (2005) pioneered the application of optimal control theory for water flooding optimization in the context of closed-loop control. Peters *et al.* (2009) set an SPE comparative test case to benchmark techniques on water flooding optimization using optimal control theory or ensemble-based methods. The general procedure in all these works is common: first the simulator is run forward to the final time; during the run, the state variables (pressure and saturation) along with the two Jacobians of the simulation equation for each time step are stored; and finally the adjoint equations are solved backwards in time for the solution of the Lagrange multipliers; then, the Lagrange multipliers are used to calculate the gradient of the objective function with respect to the control variables ($\partial J/\partial \mathbf{u}$); and used within a gradient-based method to iteratively reach the optimal control settings.

The purpose of the adjoint equations is to find the partial derivative of the objective function with respect to the control vector. To solve the adjoint equation it requires the calculation of the gradients $\partial \mathbf{f}/\partial \mathbf{x}$ and $\partial \mathbf{f}/\partial \mathbf{u}$ where $\mathbf{x}$ are the spatial reservoir variables, $\mathbf{u}$ are the control parameters and $\mathbf{f}$ is the set of flow equations that the reservoir simulator solves. This is the main difficulty of the method because $\partial \mathbf{f}/\partial \mathbf{x}$ and $\partial \mathbf{f}/\partial \mathbf{u}$ should be extracted from the reservoir simulator at each time step and it requires the explicit knowledge of the flow equations (Lorentzen *et. al.,* 2006). Nevertheless, the advantage of optimal control theory with adjoint models is that only one forward simulation is required to find the gradient ($\partial J/\partial \mathbf{u}$) for each iteration, thus the number of runs of the forward model is equal to the number of iterations.

This method does not account for uncertainty in the description of the reservoir model. The optimal solution is sought for a given reservoir model, so the solution is guaranteed to be optimal only for that specific reservoir model. Van Essen *et. al.* (2009) raises the problem that the optimal solution found for a single reservoir model might not be realistically optimal if more than one reservoir model matches the production history. In order to account for uncertainty and to have a more robust solution, Van Essen *et. al.* (2009) proposed to reformulate the objective function such that the control vector optimizes the net present value of all the plausible reservoir models simultaneously. If multiple reservoir models can match the production history, the optimal control is not simply the expected value of each optimal control vector because the transfer function is highly non-linear. The optimization of well control when the reservoir description is uncertain is still very much a research issue.

2.4.2 Ensemble Based Optimization

Chen *et. al.* (2008), Chen and Oliver (2009), and Chaudhri *et. al.* (2009) have implemented closed-loop systems for optimizing waterflooding using an ensemble-based optimization approach frequently called En-Opt. Ensemble Kalman Filter can be seen as a minimization method where the objective function is the mismatch of the response of realizations from the reference production data. This perspective of EnKF originated the so called ensemble based optimization methods. In these works, the objective function is the actual cost function such as the net present value or recovery efficiency, and the control vector is cast as an uncertain variable. This approach correlates each element of the control vector to the objective function using the experimental covariance computed over an ensemble of equi-probable realizations. The advantage of this method over adjoint models is that the flow equations are not required to be explicitly known nor is there any need to extract the Jacobian matrix from the simulator to calculate the gradient of the objective function. Instead, the numerical simulator is considered as a black-box that relates the control variables to the performance index. The method is then combined with an iterative gradient based algorithm such as steepest ascent (or descent) to search for the optimum control vector. The main disadvantage of this approach is its computational inefficiency: each time step after updating the model, the simulator is run to the final time to evaluate the objective function for each realization of the ensemble, and the process is repeated at each iteration of the gradient-based algorithm. Additionally, the assumption of a Markov Chain implies that the covariance between a control variable at time *t* and the objective function is independent of the control setting taken before *t*. In reality, it is the entire combination of control variables (**u**) at all times prior to the current time that impacts the resultant objective function.

2.4.3 Response Surface Methods

Response surface methods could be also considered as ensemble-based methods because both perturb the control variables to evaluate the variation on the objective function, and use the ensemble to compute the correlation between the objective function and the control variables. In the case of response surfaces the relationship between the control variables and the objective function is calibrated in the form of a regression where the independent variable is the set of well controls. Once calibrated, response surfaces are a useful proxy for reservoir simulators (Friedmann *et. al*. (2001) and Sangvaree (2008)). The response surface is created by running full simulations and evaluating the objective function corresponding to different realizations of the control vector. Experimental Design is a common technique used to sample the control vector (Ramirez (2009), Prasanphanich (2009)). Experimental Design investigates the sensitivity of the response to the independent variables by taking combinations of two extreme (2^k technique) values of the independent variables, or two extremes and one middle point (3^k technique) in order to account for non-linearity. The number of function evaluations is 2^k or 3^k respectively, where k is the number of independent variables. It is obvious that for large number of independent variables the methodology is impractical; hence, Experimental Design is useful when the independent variable is static, it does not change over time.

In general, a relatively simple polynomial equation ($J^*=f(\mathbf{u})$) is fitted to the response surface and used as a proxy model to correlate the design variables to the objective function. In this method, the gradient ($\partial J/\partial \mathbf{u}$) is replaced by the gradient of the proxy model ($\partial J^*/\partial \mathbf{u}$), so that the search of the optimal settings is driven by the surrogate gradient. Although the search for the optimum is fast, the construction of the surface response is expensive; the number of simulation runs increases exponentially as the number of control variables increases.

Anderson *et. al.* (2006) performed a sensitivity analysis of the design of chemical flooding in a mixed-wet dolomite reservoir. The sensitivity of the net present value with respect to a set of design variables such as surfactant concentration, slug size, salinity, mass of polymer, and a set of uncertain variables such as surfactant adsorption, polymer adsorption, and average permeability was examined. A set of curves of net present value versus each design variable was presented to show the impact of each variable. These curves could be used to find the optimal design variables.

Yamali, *et. al.* (2007) developed a proxy model using Experimental Design and response surfaces to control unwanted water production in gas reservoirs. The optimal control scheme was sought in the response surface using a hybrid optimization technique involving genetic algorithm to search for global optimum, and steepest ascent or descent method to search for local optimum. This combination of optimization methods saved computational time by using less simulation runs.

Sangvaree (2008) used experimental design and response surface methodology to optimize the design of polymer and surfactant/polymer flooding. In Sangvaree's work the control variables were not time dependent. Rather, the control variables were 5 design parameters related to the chemical formulation of the slug and a geologic variable that is the ratio of vertical to horizontal permeability. The objective function was the net present value. Although the reservoir model used was heterogeneous, sensitivity to heterogeneity was not studied.

Application of response surface methodology to time-dependent control variables in enhanced oil recovery has not been reported yet. Perhaps because the number of control variables directly depends on the number of discrete-time control steps and the number of wells, which is a large number of variables. As consequence, the task can be computationally unfeasible.

2.5 POLYMER FLOODING FOR ENHANCED OIL RECOVERY

Polymer flooding is a mobility-controlled enhanced oil recovery process. Polymer flooding is opted when the water-oil mobility ratio is high for water-flooding and/or the reservoir geology is highly heterogeneous. Reservoirs with evidence of geological heterogeneity or extensive stratification and high permeability contrast between channels and the rest of the formation ($k_{hi}/k_{av}>4$) are potential candidates for polymer flooding by delaying water breakthrough and providing more uniform sweep of the reservoir (Sorbie, 1991).

High-viscosity solutions of large polymeric molecules dissolved in water at a small concentration are injected into the reservoir for oil displacement. The water-oil mobility ratio is lowered by the high-viscosity of the aqueous solution, resulting in lower oil saturation behind the polymer front and promoting cross-flow between swept and upswept areas. As a result, the overall oil recovery is improved with greater areal coverage.

Partially hydrolyzed polyacrylamide (HPAM) is most commonly used for enhanced oil recovery. HPAM is produced in large quantities with different molecular weights and degree of hydrolysis (Kim, *et al.* 2010). Two decades ago, good reservoirs for polymer flooding with HPAM were limited to temperature less than 80 °C (max 95 °C), low salinity/hardness due to possible chemical or thermal degradation of the polymer molecules (Sorbie, 1991). However, Levitt and Pope (2008) and Lee *et al.* (2010) reported new polymers with higher resistance to thermal degradation (>100°C) and harsh salinity/hardness conditions, expanding the application of polymer flooding to more reservoirs.

2.5.1 Modeling Polymer Flooding through Fractional Flow Theory

The fractional flow theory for enhanced oil recovery generalized by Pope (1981) explains the fundamentals of polymer flooding, and it is employed in the development of the proxy model for polymer flooding presented in Chapter 3. Although the theory was originally developed for one-dimensional displacement, it has been extended to 2D and 3D through application of streamlines (Patton (1971)). The effect of inaccessible pore volume, polymer adsorption, and the formation of two shock fronts are important considerations in the application of fractional flow theory to polymer flooding. A portion of the total pore volume is inaccessible to polymer molecules because the size of the polymer molecules is comparable to the size of the entrance to small pores and throats.

Consider a one-dimensional pore volume with initial connate water saturation equal to the irreducible water saturation. Polymer solution is injected on one end and fluids (oil and water) are produced on the other end. As the polymer solution is injected, the connate water that is contacted is pushed downstream. Higher water saturation corresponds to faster velocity, resulting in the formation of a shock water front. Behind the water front, polymer solution moves slower but also with velocity that is higher corresponding to higher saturations, thereby forming a polymer front. The velocity of the polymer front is, on one hand, retarded by adsorption onto the rock, but on the other hand it is speeded up due to the inaccessible pore volume. The saturation levels at the fronts can determined graphically as in Figure 2.2. The water saturation at the polymer front corresponds to the point of departure of the tangent line (red straight in Figure 2-2) from the polymer-oil fractional flow curve (red curve in Figure 2-2) that intercepts the x-axis at $-D_p$. D_p is called the retardation factor (Lake, 1989) and it accounts for the effect of adsorption and inaccessible pore volume. The water saturation at the water front corresponds to the intersection of the same tangent line and the fractional flow curve of water (blue curve in Figure 2-2). The water cuts at the polymer and water fronts are read from their corresponding fractional flow curves at the front saturation.

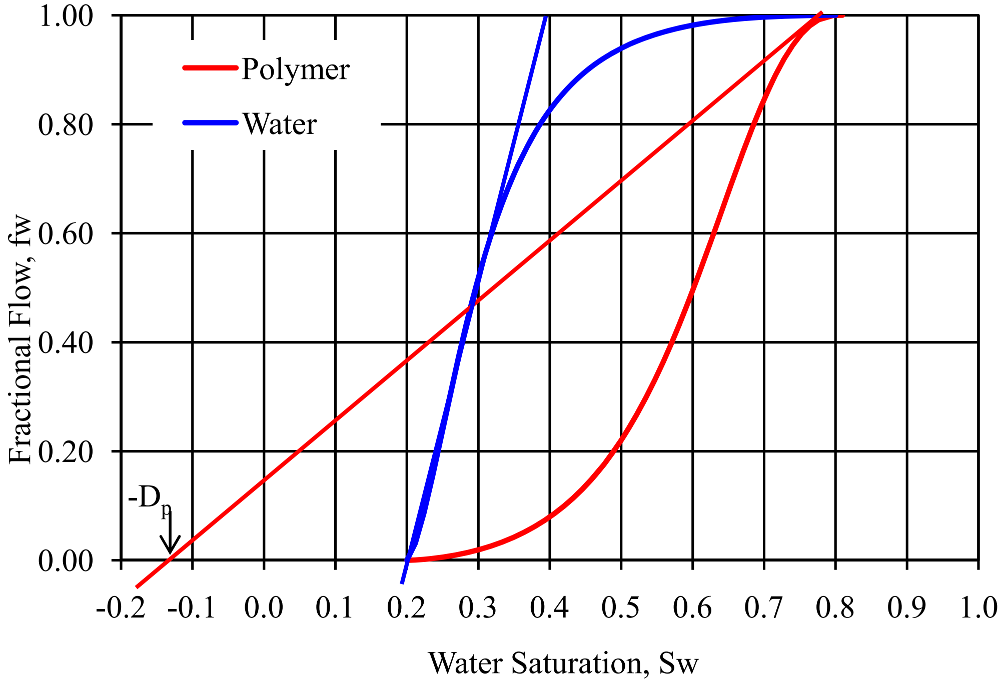

Figure 2-2 Fractional flow curves used for determination of saturation and water cut at water and polymer fronts

Before water breakthrough, the recovery factor is equal to the injected pore volume. From water breakthrough to polymer breakthrough, the produced water cut is the fractional flow for the water front. After the arrival of the polymer front, the produced water cut is practically the fractional flow for the polymer front. The arrival of the polymer front denotes the economical end of the process because after that, oil production is negligible.

2.5.2 Modeling Polymer Flooding with UTCHEM

The numerical simulator used throughout this study is UTCHEM, a compositional chemical flood simulator developed at the University of Texas at Austin (Pope and Nelson, 1978). UTCHEM is a state-of-the-art numerical simulator for polymer flooding, which takes into consideration polymer adsorption, polymer rheology, permeability reduction and inaccessible pore volume.

2.5.2.1 Polymer Retention in UTCHEM

Polymer flow in porous media is affected by adsorption of polymer molecules onto the rock surface in small pores and throats, resulting in a reduction of the polymer concentration at the leading front. Polymer retention depends on many factors such as polymer type, molecular weight, rock composition, brine salinity, flow rate, etc. (Lake (1989)). Polymer retention is termed as adsorption in UTCHEM, and it is modeled as a Langmuir-type isotherm with adsorption coefficient dependent on permeability, salinity and some empirical coefficients (UTCHEM Technical Documentation, 2000). According to Huh, et al. (1990a) polymer retention depends on effective permeability to the aqueous phase as reproduced in Figure 2-3. Equation 2.7 was proposed by Mantilla (2010) to include dependence on relative permeability for the calculation of the retention coefficient in order to match experimental observations by Green and Willhite (1998) and Huh, *et* al. (1990b).

$$a = \left(a_{41} + a_{42}C_{SEP}\right)\left(\frac{k_{ref}}{k_{eff,w}}\right)^{1/2} = \left(a_{41} + a_{42}C_{SEP}\right)\left(\frac{k_{ref}}{k_{rw}k}\right)^{1/2} \qquad (2.7)$$

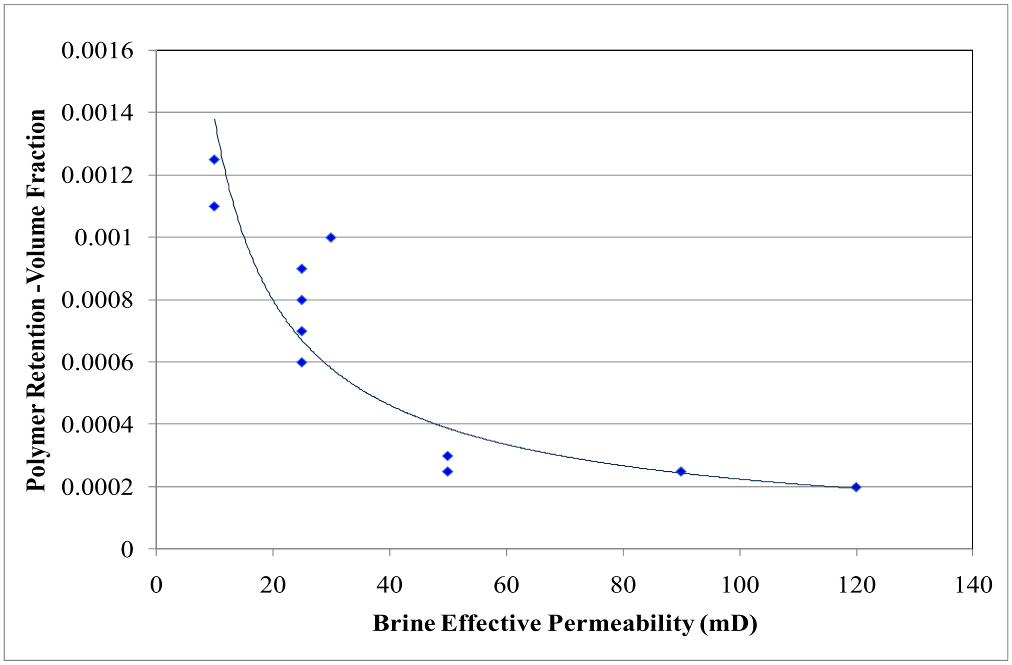

Figure 2-3 Polymer retention as a function of aqueous phase effective permeability. Adapted from Huh, *et al.* (1990b).

2.5.2.2 Polymer Viscosity in UTCHEM

The rheology of polymer solutions used in enhanced oil recovery depends on the effective concentration of polymer molecules, salinity and effective shear rate. UTCHEM models the aqueous phase viscosity as a function of effective salinity (CSEP) and polymer concentration increases as illustrated in Figure 2-4, which shows the curve (CSEP=0.08) with the actual parameters used in the simulations carried out in this dissertation.

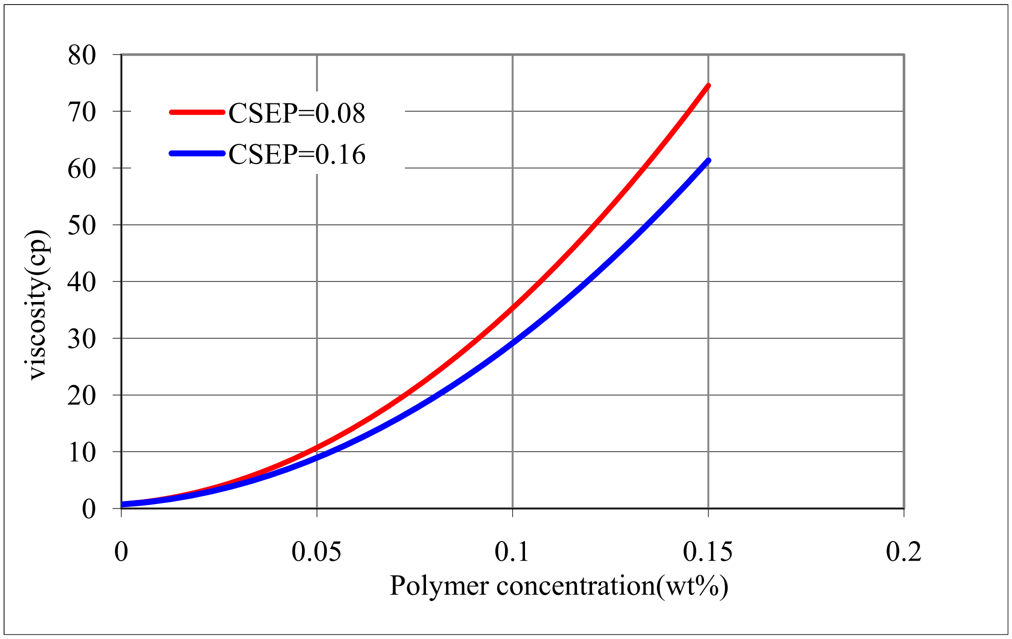

Figure 2-4 Polymer viscosity as a function of polymer concentration and brine salinity. Data used was taken from Dakhlia (1995)

Polymer solutions used in enhanced oil recovery are shear-thinning fluids (Canella *et al.*, 1988). that exhibit low viscosity at high shear rate (near the wellbore area) and high viscosity at low shear rate (away from the wells) Figure 2-5 shows the polymer viscosity as a function of effective shear rate as used in the simulations throughout this dissertation.

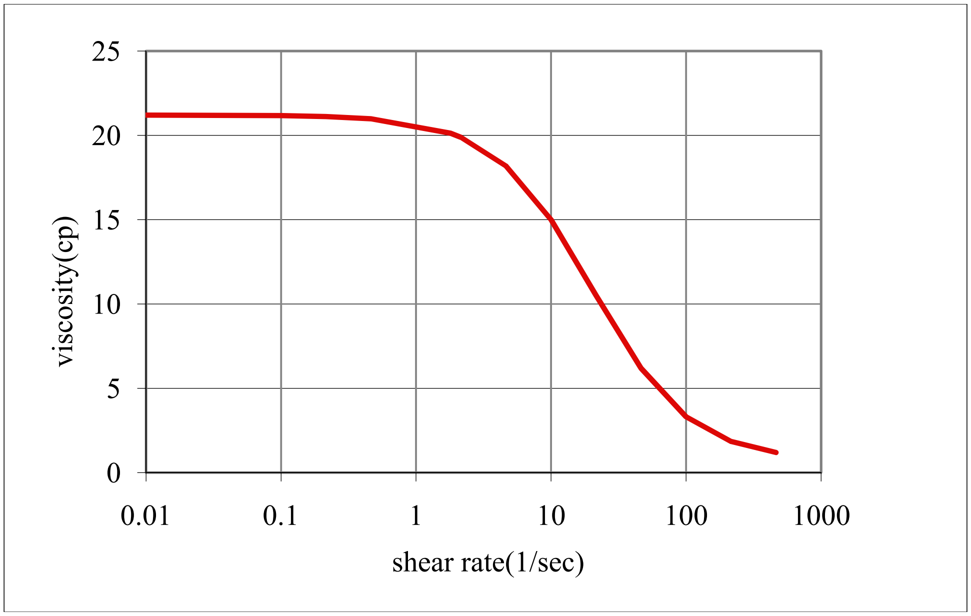

Figure 2-5 Polymer viscosity-shear rate relationship inputted in UTCHEM for the polymer flooding simulations in this dissertation. Data used was taken from Dakhlia (1995)

2.5.2.3 Permeability Reduction and Inaccessible Pore Volume

Polymer solutions reduce the absolute permeability of the rock permanently as the molecules plug the small pore throats (Smith, 1970). UTCHEM uses a permeability reduction factor to model this phenomenon (UTCHEM Technical Documentation, 2000). Additionally, the inaccessible pore volume effect is modeled in UTCHEM through an inaccessible pore volume (IPV) factor that reduces the pore volume occupied by the polymeric phase.

2.6 CONCLUSION

In conclusion, a feedback control system is feasible for polymer flooding processes by combining a method for model updating that considers uncertainty in geology and integrates observations from the system, with a production optimization algorithm able to solve a complex mathematical problem efficiently. UTCHEM is a state-of-the-art simulator polymer flooding with capability to model the physics of polymer flooding accurately, but the high computational cost involved motivates the development of a proxy model that can be rapidly evaluated. Chapter 3 presents the development of a proxy model for polymer flooding that it is employed in the model updating and optimization parts of the framework.

Chapter 3: A Fast Proxy Model for Polymer Flooding

In order to implement model selection and update the prior uncertainty in reservoir model, it is required to rapidly assess characteristics of the heterogeneity of a large number of reservoir models. Later, during the process of optimizing well controls to maximize project economics, a large number of evaluations of the objective function for different reservoir models under different well constraints will be required. These two needs motivated the development of a fast analog of flow simulation sensitive to reservoir heterogeneity and well controls. Thus, the role of the proxy model for polymer flooding is crucial in the feedback control framework because: (1) it captures flow characteristics related to heterogeneity in a given reservoir model, and (2) it provides a rapid estimate of recovery-based functions given a geologic model and operating conditions at the wells. The proxy model developed here employs a random walk algorithm to quickly propagate random particles from injectors to producers mimicking the displacement of polymer solution from the injector to the producer. It is important to emphasize that the same calibrated proxy model is employed both for the classification of reservoir models and optimization of well controls.

3.1. PURPOSE AND ASSUMPTIONS IN THE PROXY MODEL

The purpose of the proxy model is to obtain differences in heterogeneous characteristics that affect the performance of a polymer flood between any two reservoir models. The proxy model developed in this work does not aim to replace a full-physics simulator; it can only provide estimates of recovery or more importantly the impact of reservoir heterogeneity on recovery. This proxy model does not provide physical variables like pressure, polymer concentration, retention, shear rate effects, etc. A more accurate model for polymer flooding is the work of Naiki (1979), who developed a finite-difference numerical model using "moving points" to trace the concentration of polymer in 2D heterogeneous reservoirs. However, the proxy model developed in this dissertation is sufficient to accomplish the purpose of distinguishing the production characteristics between any two heterogeneous reservoir models.

3.1.1 Assumptions

The proxy model is a simplified model of polymer flow with the following assumptions:

- The pressure gradient corresponding to a steady state does not evolve over time. Therefore, the velocity vectors are static as well.
- Shear rate and visco-elastic effects in polymer rheology are neglected.
- Polymer viscosity is constant and equal to the viscosity corresponding to the concentration at the injection point.
- The above assumption implies that polymer concentration at the polymer front and behind it is equal to the injected polymer concentration.
- Polymer retention and inaccessible pore volume are modeled through fractional flow theory. Local variations in retention and inaccessible pore volume due to heterogeneity are neglected.

3.2. RANDOM WALKS

The term random walk was first introduced by Pearson (1905). Random walk is a stochastic method for transitioning the state of random entities from an initial state to a final state through successive random steps. Random walk analysis has been applied in computer science, physics, ecology, economics, psychology and other fields (Lawler and Limic, 2010). Random walk algorithm assumes a Markov chain process, in which the probability of transitioning to the next state depends only on the current state. In transport problems, random walks are useful for assessing the ability of the medium to transport concentrations taking into consideration the heterogeneous properties of the medium. The word “particles” is employed in this dissertation in the context of random walk algorithm. It should not be confused with a physical particle of water or polymer. The position of physical water or polymer particles differs from the position of the random “particles” used in the proxy model which track the location of the fronts.

In the present application the paths of random particles placed in the reservoir are traced from their initial state (injection points) to their final state (production points). The probability rules for particles to transition from the current position to the next position are governed by the velocity field obtained as a solution of the pressure equation corresponding to prescribed injection and production rates, and considering reservoir heterogeneity. In this way, particles follow the pressure diffusion path while accounting for heterogeneity on the reservoir. The objective of tracing the position of random particles is merely to estimate the position of the two shock fronts formed in polymer flooding.

The main advantage of the algorithm is the computational speed; the pressure equation is solved only once, and the propagation of particles in time is simulated as a random process that only requires successive sampling of the transitioning probability. As a result, random walk simulation of a polymer flooding process takes few seconds on a desktop computer, while a full simulation run would take several minutes. In general, this proxy model can be compared to streamline simulation, where the pressure equation is solved few times and the saturation profile is propagated using an analytical solution.

3.3. PARTICLE PROPAGATION GUIDED BY VELOCITY VECTORS

The pressure equation is solved for single phase flow, which is equivalent to a two-phase unit-mobility-ratio system. In a two-well system, this operation is performed only once because the velocity vector corresponding to any other rate status is proportional to the initial one. In other words, the flow direction does not change, only the magnitude changes. In a multi-well system, the pressure equation needs to be solved every time the rate allocation is redefined. Subsequently, the velocity vector exiting each grid-block is calculated from the pressure solution using Darcy's law at each location (considering the distance between centers of grid-blocks. Then, a large number of random particles are initially placed in the reservoir at the grid-block center of the injection locations. Particles are propagated throughout the reservoir in a sequence of time steps.

The probability that a particle located at the center of the grid-block will move to each of the exiting faces is calculated proportional to the magnitude of the velocity vector crossing that face. Only exiting faces are taken into account. Thus, the transitioning probability densities are discrete probability values assigned to each exiting face proportional to the magnitude of the velocity vector in this direction as expressed in Equation 3.1:

$$\Pr\left\{\mathbf{u}_p = \mathbf{u}^{l+1} \mid \mathbf{u}_p = \mathbf{u}^l\right\} \sim \left\|\vec{v}\left(\mathbf{u}^{l+1}, \mathbf{u}^l\right)\right\| \tag{3.1}$$

Where $\mathbf{u}_\mathrm{p}$ is a vector with the particle p, $\mathbf{u}^{l+1}$ and $\mathbf{u}^l$ are grid-block coordinates, and $\vec{v}\left(\mathbf{u}^{l+1}, \mathbf{u}^l\right)$ is the velocity vector between grid-blocks located at $\mathbf{u}^{l+1}$ and $\mathbf{u}^l$.

The next movement of the random walker is determined by randomly sampling an exit face based on the transitioning probabilities. The movement of the particles from block to block is illustrated in Figure 3-1. Once the direction of movement of a particle has been determined, the step length is calculated from the velocity and the time step size using Equation 3.1. If the distance calculated in Equation 3.2 exceeds the distance between two cell centers, the particle is stopped at the arriving block center, and a new move is calculated. The time spent by the particle for that new movement is calculated using Equation 3.2. In the event that the distance calculated in Equation 3.1 is less that the distance between grid-block centers, the particle only travels that distance and stays in that position until next time step. The particle movement is resumed at the next time step.

$$dx = \min\left(v\Delta t, \Delta x\right) \tag{3.2}$$

$$dt = \min\left(\Delta t, \frac{\left|x_p - x_{c2}\right|}{v_x}\right) \tag{3.3}$$

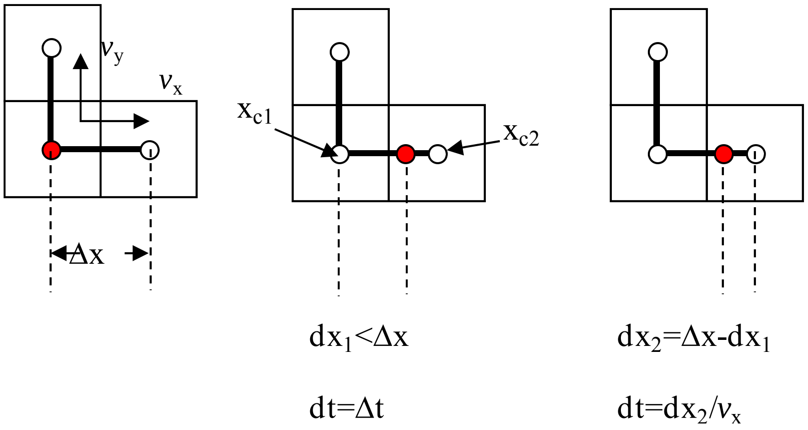

Figure 3-1 Illustration of the movement of a particle between grid-block centers in two steps. a) the movement initiates in x-direction; b) in the first time step, the particle travels a distance less than the separation between centers; and c) during the second time step the particles arrives to the center of the adjacent cell and the new transitioning probability is sampled for the next movement.

The probability of particle movements depend on velocity vectors, which are function of well constraints. For example if a well has been shut in, the velocity calculated for the particles around the well location is zero so the particles stay stagnant. Particle movement is also sensitive to the duration of the injection or production conditions because the length of each movement is conditioned to the current time step size. Hence, the results from the proxy model are sensitive to well controls and can be used for production optimization.

3.4. CALIBRATION OF PROXY MODEL

In order to demonstrate the proxy model, the movement of water and polymer particles was simulated first on a 2D reservoir model with two wells shown in Figure 3-2. Then, in order to account for well interactions the proxy model was demonstrated on a larger 2D reservoir model with an inverted 5-spot injection well pattern (shown in Figure 3-3). For both models, the mean of the high permeability rock type is 1100 mD whereas for the low permeability rock is 100 mD. The porosity is constant and equal to 0.1. The oil, polymer and water properties used for these test cases were taken from Dakhlia (1995), who used data from an actual polymer flooding project implemented on Chatenaraud field and presented by Putz, *et al.* (1988). Oil viscosity is 40 cP and water viscosity is 0.73 cP at reservoir conditions. The end-point mobility ratio is 10.96, which is unfavorable for immiscible displacement. The relative permeability curves are plotted in Figure 3-4. Polymer viscosity at the injected concentration is 20 cP. The fractional flow curves plotted in Figure 3-5 define the water and polymer saturation at the fronts, and the frontal velocities. The water saturation at the front is 0.28 and the polymer saturation at the front is 0.76. The fractional flow value at water front is 0.48 and at the polymer front is 0.97.

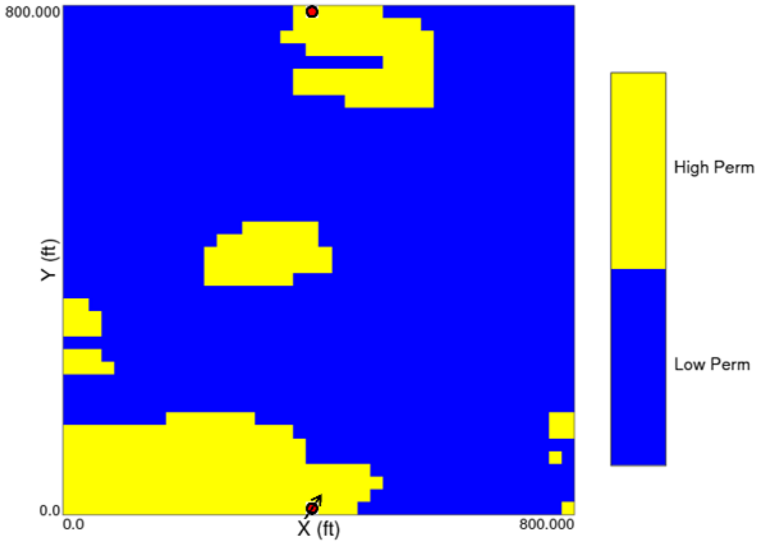

Figure 3-2 Reservoir model used for comparison of production profiles from proxy model and UTCHEM. Polymer is injected at the well located at the bottom

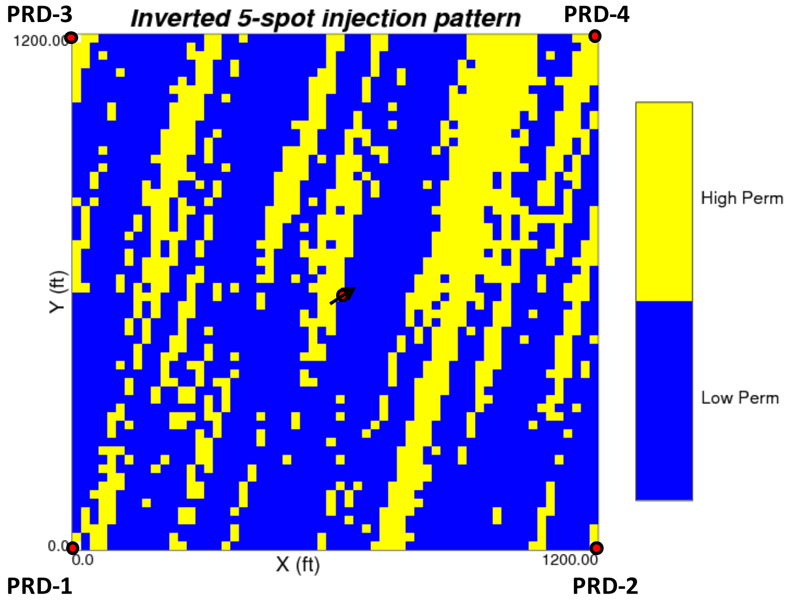

Figure 3-3 Reservoir model with an inverted 5-spot pattern injection used to calibrate the proxy model.

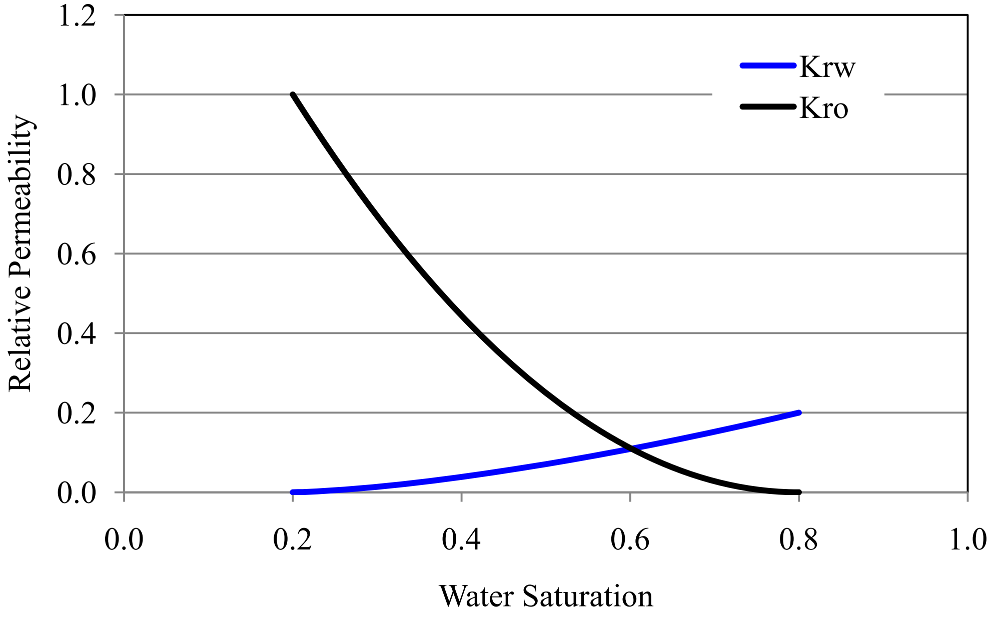

Figure 3-4 Relative permeability curves for the calibration cases.

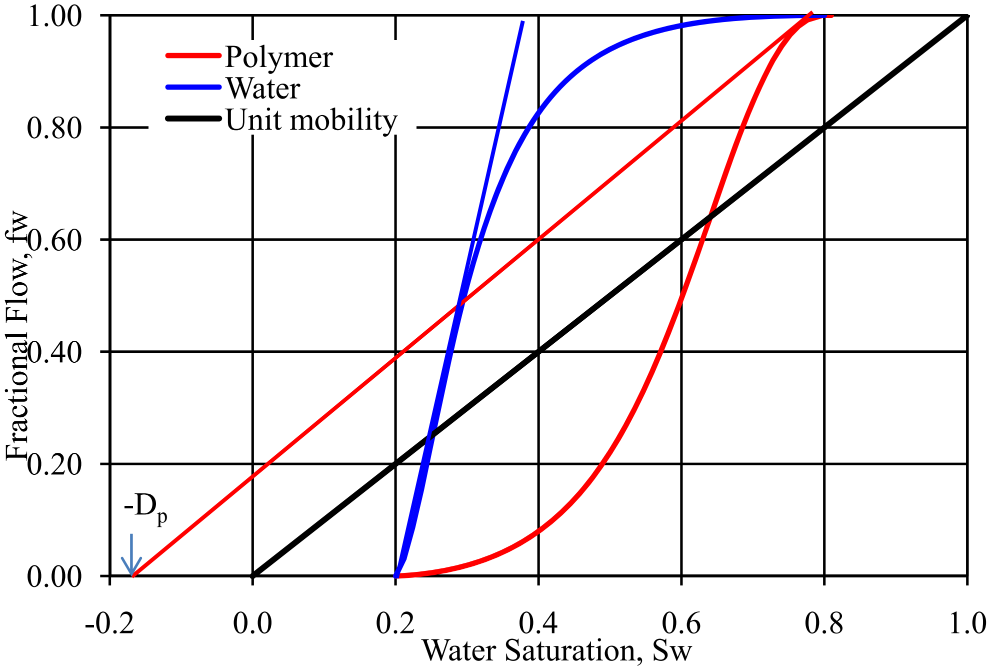

Figure 3-5 The velocity of the shock front in a unit mobility ratio, water-oil and polymer-oil systems is proportional to the slope of the tangent lines drawn from the initial saturation to the respective fractional flow curve.

3.1.2 Movement of Water Particles

In a one-dimensional displacement, initial connate water is pushed by the injected polymer solution, creating a shock front as explained by Pope (1981). In the proxy model, water particles that exist as part of the initial water saturation are pushed by polymer particles along streamlines from injection to production points. The velocity of water particles is proportional to the slope of the tangent line to the fractional flow curve of a water-oil system drawn from the initial water saturation (See Figure 3-5).

Since the pressure equation (and the corresponding velocity vector) was solved for a single phase system (unit mobility ratio), the velocity of water particles needs to be updated to reflect the unequal velocities of the water and polymer fluid phases. Equation 3.4 assumes that the ratio of velocities (of the single phase and the polymer/water/oil system) is equal to the ratio of slope of the tangent lines to the single phase and the water-oil systems. Since the slope in a unit-mobility-ratio displacement (single phase system) is equal to 1, the velocity of water particles is the product of the velocity of the single phase system and the slope of the tangent line in the water-oil system. Consequently, water breaks into producers earlier than polymer or for a unit-mobility-ratio system.

$$V_w = V_{um}\left(\frac{df}{dS}\right)_{ow} \tag{3.4}$$

V_w is the velocity of water particles, V_{um} is the velocity calculated from the single phase system (unit mobility ratio), (df/dS) is the derivative of water cut with respect to water saturation, and the subscripts *ow* means oil-water system and the subscript *um* means unit mobility ratio.

Estimation of Water Cut from Particle Count at Producers.

The position of water particles represents the position of the water saturation front along a streamline. In 1D, when the water front arrives to a producer, water saturation goes from the initial water saturation to the water saturation at the front, and the corresponding water cut is read from the fractional flow curve. By analogy, in our proxy model when all the displaced water particles arrive to a producer, the water saturation is equal to the saturation at the water front. In contrast to 1D, in 2D and 3D water particles arrive progressively according to the flow path followed. Some particles follow a short path and arrive quickly while others follow tortuous paths. Therefore, in the proxy model each water particle that arrives at a producer increases the water saturation by an amount proportional to the total number of water particles expected to arrive over all paths. Hence, water production increases gradually from the breakthrough of the first particle until it reaches a plateau (corresponding to the total number of particles arriving at the well). The equivalent water saturation is calculated as in Equation 3.5.

$$S_w = S_{wi} + \left(S_{wf} - S_{wi}\right)\frac{N_w}{N_{wT}} \tag{3.5}$$

N_w is the number of water particles that have arrived to the producer and N_{wT} is the total number of water particles expected to arrive at that producer. If all water particles have arrived, the water saturation is equal to the water saturation at the front. The random walk simulation is carried out in time to ascertain the total number of particles arriving at a producer and that is used to compute the water cut at all previous times using Equation 3.5 above.

Figure 3-6 shows the gradual increase in water cut from the proxy model as well as from the simulation with UTCHEM for the two-well reservoir model. No sharp water front was formed. In fact, not all water is mobilized; some amount remains stagnant for long time in areas disconnected from producers. Also, some water was contacted and mobilized but it did not make its way to the producer. The simulation with UTCHEM indicates that an oil bank was formed as can be inferred from the slight reduction of water cut about 1600 days (See Figure 3-6). A limitation of the proxy model is the inability to predict the oil bank, since a separate species of random walkers reflecting the oil phase is not considered.

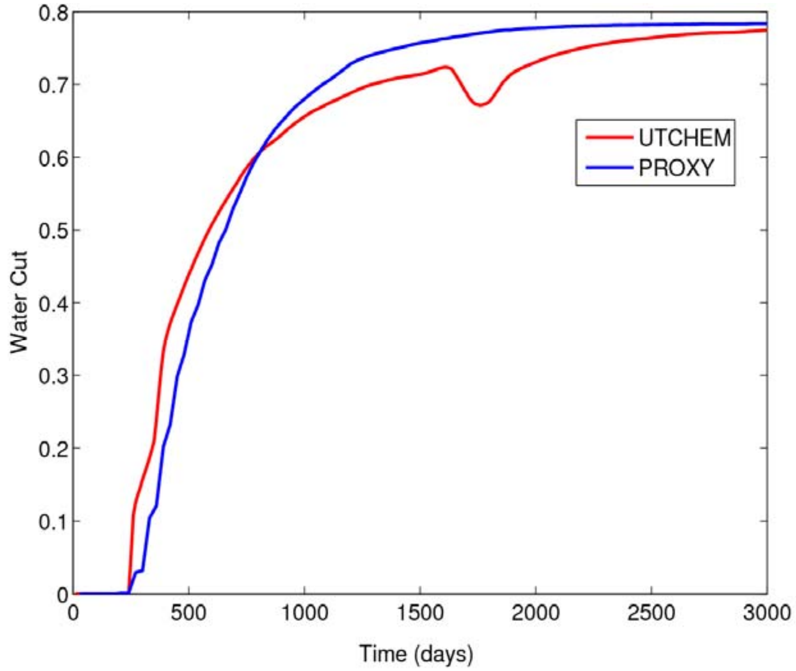

Figure 3-6 Water cut profiles obtained from proxy model and UTCHEM for the reservoir model shown in Figure 3-2

In a multi-well injection pattern such as the well configuration shown in Figure 3-3, if all producers are open at equal flow rates, all producers are expected to receive equal number of particles. When a well has produced the total number of water particles assigned to it, the water saturation is the saturation at the water front. However, if the production rates are uneven among the producers and the individual flow rates changes over time, the calculation of water production is more elaborated as explained in the next paragraph.

To illustrate this case, the reservoir model shown in Figure 3-3 was subjected to production and injection rates schedule shown in Table 3-1. Note that for the first 100 days the two producers on the bottom produce fluids at high rates while the wells at the top are practically shut-in. Subsequently, the producers at the bottom are shut-in and the production is diverted to the wells at the top of the reservoir. Water and polymer particles are injected at the center of the reservoir and start moving towards the corners. As particles move towards the corners, the particles are distributed among the producers according to the distance from the particle position to the producer location as shown in Figure 3-7. Even though in the first production period 99% of the production comes from the producers at the top of the reservoir, it is not expected that all the injected water and polymer particles reach the producers at the top, only the particles within the drainage area of the respective wells will be produced. If all the water particles assigned to a producer arrive there, the corresponding water saturation at that producer is equal to the saturation at the water front. Figure 3-8 compares the water cut obtained from UTCHEM and from the proxy model for a reservoir with four producers and one injector.

Table 3-1 Rate schedule for multi-well test case

Well Name	**Rate (0-1000 days)**	**Rate (1000-2000 days)**
PRD-1	99 ft^3/day	1 ft^3/day
PRD-2	99 ft^3/day	1 ft^3/day
PRD-3	1 ft^3/day	99 ft^3/day
PRD-4	1 ft^3/day	99 ft^3/day
Total Production	200 ft^3/day	200 ft^3/day

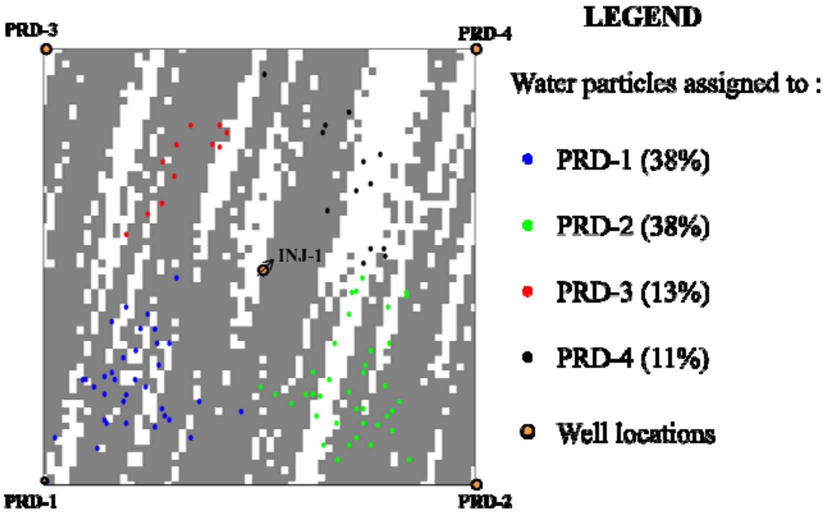

Figure 3-7 Random walkers representing the water phase are assigned to producers according to their proximity to the wells. Position of water particles at 500 days.

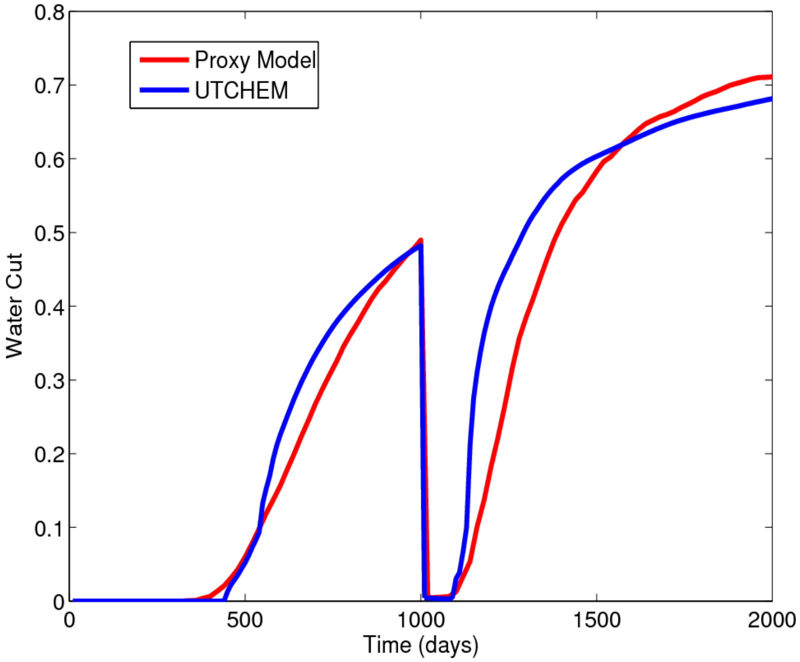

Figure 3-8 Field water cut profiles obtained from proxy model and UTCHEM for the reservoir model shown in Figure 3-3 (5-spot inverted injection pattern)

3.1.3 Polymer Particles

The position of polymer particles represents the position of the polymer front within a streamline. The velocity of water particles is different from the polymer particles. Fractional flow theory predicts that polymer particles should move significantly slower than water particles. The same rule for transition probability controlling the movement of water particles governs the movement of polymer particles also. Polymer particles velocity is calculated as in Equation 3.6.

$$V_p = V_{um}\left(\frac{df}{dS}\right)_{op} \tag{3.6}$$

V_p is the velocity of polymer particles and the subscript op means oil-polymer system. Similar to the water front, when polymer front breaks into a producer, the water saturation increases instantly from the saturation at the water front to the saturation at the polymer front. Hence, water saturation is re-calculated when polymer particles arrive at the producer using Equation 3.7.

$$Sw = S_{wi} + \left(S_{wf} - S_{wi}\right)\frac{N_w}{N_{wT}} + \left(S_{wp} - S_{wf}\right)\frac{N_p}{N_{pT}} \tag{3.7}$$

N_p is the number of polymer particles that have arrived to the producer and N_{pT} is the total number of polymer particles assigned to that producer. At the end of the simulation that particular producer receives N_{pT} particles. Figure 3-9 compares the polymer concentration maps obtained from UTCHEM to the position of polymer particles simulated with the proxy model. The fact that the pressure gradient used for particle propagation is obtained by scaling the single phase velocity makes the particle migration (proxy) more spread than the more compact polymer bank (UTCHEM).

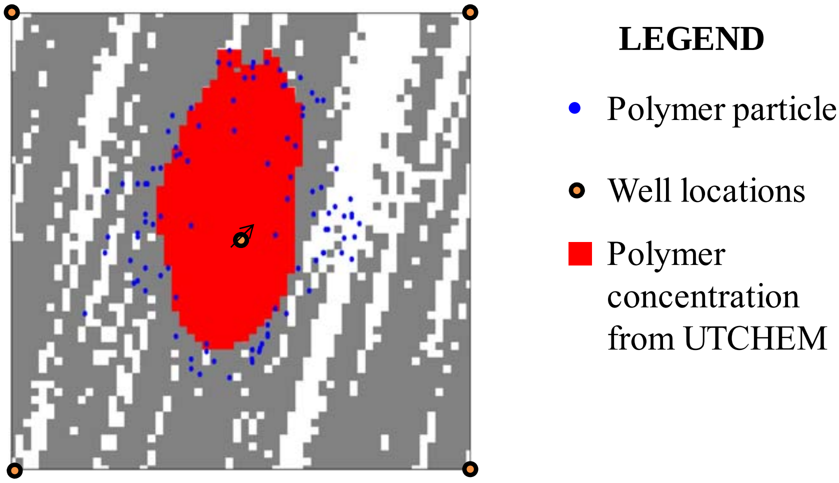

Figure 3-9 Polymer saturation fronts from the proxy model and UTCHEM. Position of polymer particles represents the spread of the polymer front in the proxy model.

3.2 INPUTS AND OUTPUTS FROM THE PROXY MODEL

3.2.1 Inputs

The proxy model requires the permeability field and the rate schedule for all wells as input. The combination of heterogeneity and local velocity determines the transition probabilities at that location. At any time, the particle position for the next time step depends on the current position and the velocity vectors. Additional inputs such as fluid viscosities and relative permeability curve parameters are necessary to calculate water and polymer velocities. Additionally, economic parameters such as oil price, oil production costs, water production costs, polymer price, discount rate, taxation rates and fixed costs are required for the calculation of the net present value using Vaskas' (1997) economic model that is used later in the control optimization calculations.

3.2.2 Outputs

The main outputs are the recovery curves and the economic objective function for a given rate schedule. Recovery curves are obtained from material balance using production rates and initial oil-in-place. Water production profiles are also an important output used for the calibration of the proxy. Additionally, the net present value curve is also given as an output by the proxy model. The outputs are stochastic because they are the outcome of a random function. Every time the proxy model is run, the output varies as a result of the random sampling that takes place at every step of the displacement. As more particles are introduced in the simulation *i.e.* more samples from the transitioning probability distributions are taken, the variance of the recovery factor and final net present value decreases as shown in Figure 3-10. About 5000 particles are sufficient to stabilize the variability in recovery statistics and yield reliable estimates.

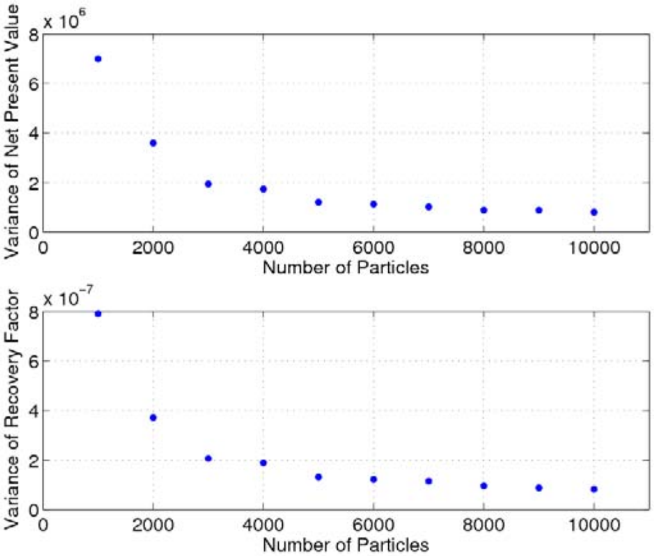

Figure 3-10 Variance of proxy responses as a function of the number of particles used in the proxy model. The ensemble size over which the variance stabilizes is a function of the transition probability used to regulate the movement of particles and that in turn is controlled by the nature of heterogeneity exhibited by the geology.

The distribution of the net present value estimated by this means is approximately Gaussian as shown in Figure 3-11. In order to obtain a reliable estimate of the mean of the net present value, multiple samples should be drawn by executing the proxy model multiple times. A significance test for Gaussian distributions, known as p-test, is used to calculate the minimum number of samples needed to have a reliable estimate of the net present value. As a result, five runs of the proxy model are necessary to obtain a reliable estimate of the mean with 95% confidence and a 5000 dollars margin of error. This assessment of the adequacy of the size of the particle ensemble for reliably representing the arrival statistics should be performed for each specific reservoir setting and is important for ensuring that the subsequent well control optimization scheme is robust.

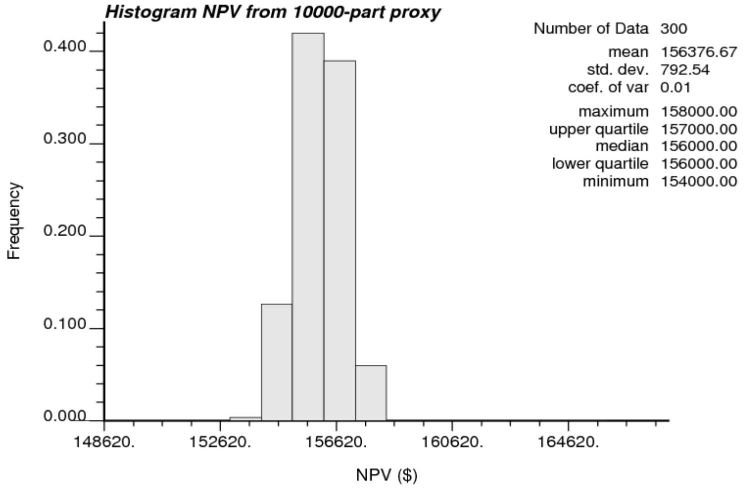

Figure 3-11 Histogram of the net present value obtained from the proxy model using 10000 particles.

3.3 CORRELATION TO FULL SIMULATION RESPONSES

Polymer flooding was simulated for 100 randomly selected injection/production schedules within the operational limits using the proxy model and UTCHEM for the two-well reservoir model shown in Figure 3-2. The approximated recovery factor (from proxy model) is estimated as the mean of five runs and compared to the recovery factor from UTCHEM in Figure 3-12.

Another useful performance indicator is the net present value. We have calculated the net present value of each injection/production schedule in this reservoir model using the chemical flooding economic model (Vaskas, 1996) and the parameters listed in Table 6.2. The final net present value obtained from both methods is compared in Figure 3-13. The correlation coefficient for both the recovery factor (ρ=0.997) and the final net present value (ρ=0.996) is high indicating that the proxy model is an effective surrogate for evaluation of these responses as a function of injection/production rates.

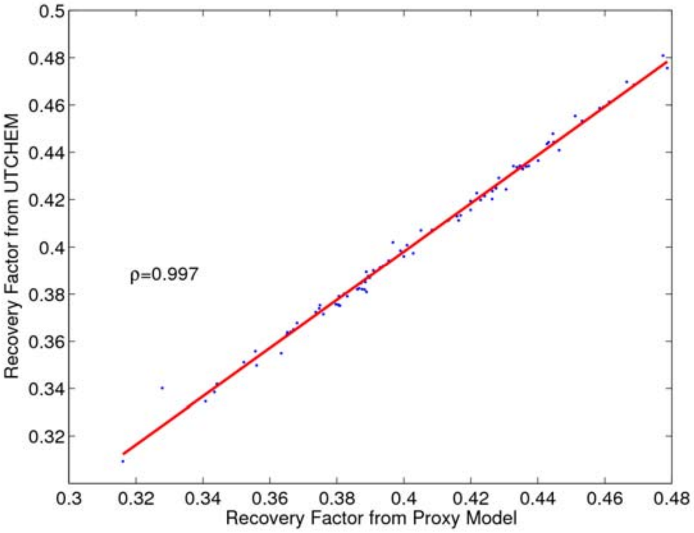

Figure 3-12 Correlation between recovery factor estimated by the proxy model and UTCHEM for 100 different injection/production rate schedules.

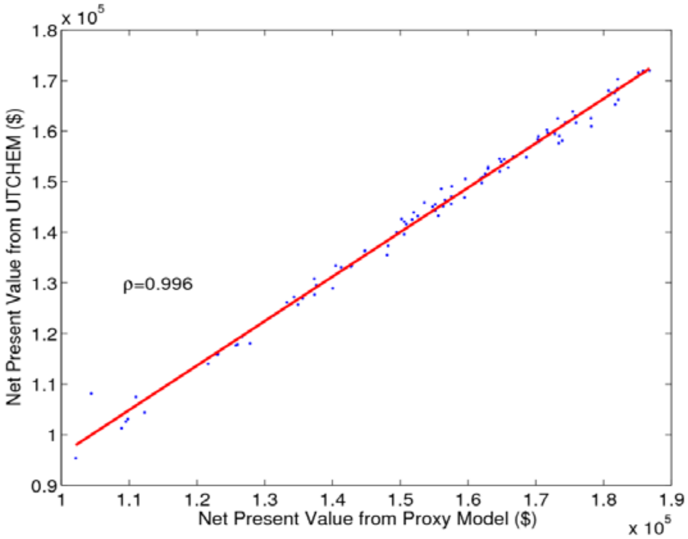

Figure 3-13 Correlation between the net present value at 2000 days estimated by the proxy model and UTCHEM for 100 different rate schedules

The responses from the proxy model were also compared to the results of UTCHEM for 100 different geologic models generated using various variogram models and training images (Figure 3-14). Figure 3-15 shows good correlation (ρ=0.87) between the recovery factor with the proxy and UTHCEM. The correlation for the net present value is also good (ρ=0.891), as shown in Figure 3-16. These high correlation coefficients indicate that the proxy model will be effective in capturing the essential differences between the flow characteristic of different geologic models.

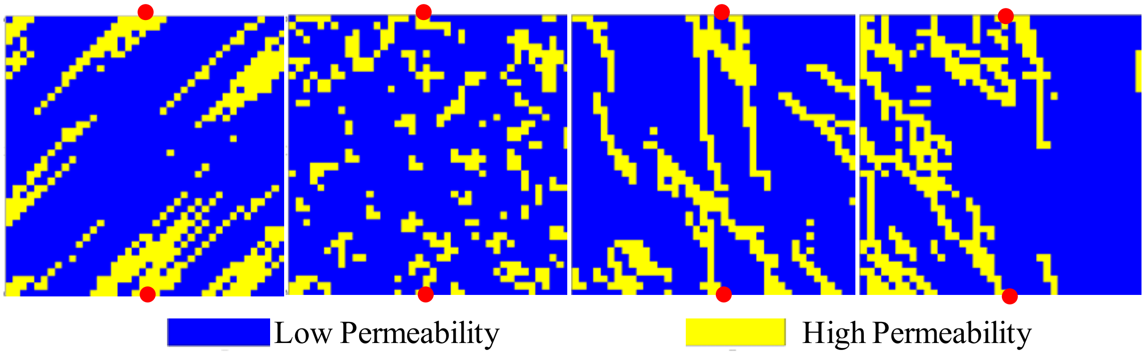

Figure 3-14 Samples of geologic models used to test the correlation of the proxy model and UTCHEM

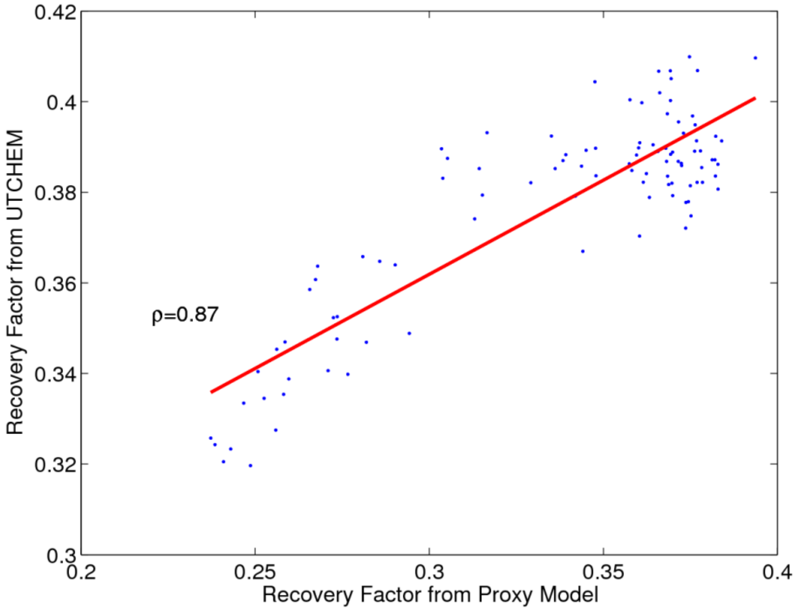

Figure 3-15 Correlation between recovery factor estimated by the proxy model and UTCHEM for 100 geologic models

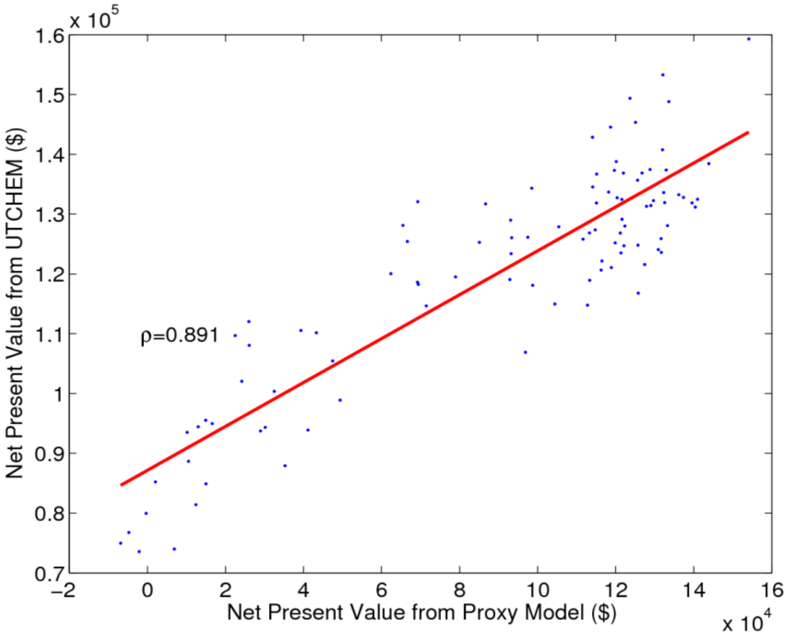

Figure 3-16 Correlation between the net present value at 2000 days estimated by the proxy model and UTCHEM for 100 geologic models

3.4 CONCLUSION

A proxy model simulates the flow of particles in a polymer flooding process has been put forward. The physics of polymer flooding have been incorporated in the model by applying concepts of fractional flow theory, and using the solution of the pressure equation for guiding the path of particles. This proxy model is a fast analog that is used in the feed control framework to assess heterogeneity in reservoir models and to estimate recovery based responses useful for optimization.

Chapter 4: Organizing Multiple Reservoir Models using Measures of Dissimilarity

4.1. OVERVIEW

The ultimate objective of this research work is to develop an optimum strategy for injecting polymer into the reservoir in order to maximize economic recovery while accounting for the uncertainty in predicting the flow performance of the reservoir. The starting premise is that there is significant prior geologic uncertainty that has to be resolved using the observed production data and subsequently the optimum control strategy is developed based on the reduced set of models. This chapter presents a novel approach for grouping reservoir models using measures of dissimilarity based on heterogeneity. The impact of heterogeneity on polymer flooding is assessed through the proxy model described in Chapter 3. At the end, the construction of a metric space for mapping reservoir models is illustrated through an example.

4.2. MOTIVATION

A major objective is to develop a general method for assessing the uncertainty in reservoir modeling, regardless of the source. Although the main focus of the following discussion is uncertainty due to reservoir heterogeneity and geology, the application of the method is not limited to that source of uncertainty.

Geostatistics is used to model uncertainty associated with spatial variations of reservoir properties. In fact, several measures of reservoir heterogeneity have been postulated and these include statistics of reservoir properties constrained to specific spatial templates. For example, sequential simulation algorithms based on variogram models describe the variations of reservoir properties between two points in space. This has been extended to multiple point templates and multiple realizations of reservoir models based on the variations of reservoir properties at multiple locations simultaneously are developed. In reality when a reservoir model is being constructed, there is no sufficient evidence to tie the geology of a reservoir to a specific spatial template and the residual uncertainty in the spatial description has to be discerned after additional data is integrated. Developing a method to update prior models of reservoir heterogeneity to a posterior set constrained to the observed production data is the primary motivation for the work presented in this chapter.

4.3. CHARACTERIZATION OF HETEROGENEOUS RESERVOIR MODELS

We want to characterize the heterogeneity of multiple reservoir models in order to establish differences between them. The obvious definition of a reservoir model is the collection of all its variables, but the large number of variables involved, usually thousands (or millions) according to the spatial resolution, make this definition too unwieldy to work with. Rather, we look for a more integrated measure for heterogeneity of reservoir models that reflects the role of heterogeneity on the performance of polymer flooding.

Some basic measures of heterogeneity are the mean, variance, Dykstra-Parsons coefficient, Lorenz coefficient and heterogeneity factor (Koval, 1963), which are by some means related to displacement efficiency. In general, sweep efficiency decreases as the heterogeneity factor increases (Lake, 1989). However, these basic measurements do not account for spatial variations of reservoir properties, especially for geologic models generated constrained to multiple-points statistics. It is likely that heterogeneity represented by these higher order statistics has an impact on the performance of polymer flooding.

Variogram is a common measurement of heterogeneity that accounts for variations of rock properties over a spatial lag. However, variogram only capture variability over two locations separated by a linear template. Moreover, the residual uncertainty of variogram-constrained models can result in wide variation in production response as can be observed in Figure 4-1. This figure depicts the variation in recovery factor corresponding to a polymer flood observed over a suite of models constrained to reproduce the semi-variogram. In fact, probabilistic history matching methods (Srinivasan and Bryant, 2004) work on reducing that residual uncertainty through dynamic data integration.

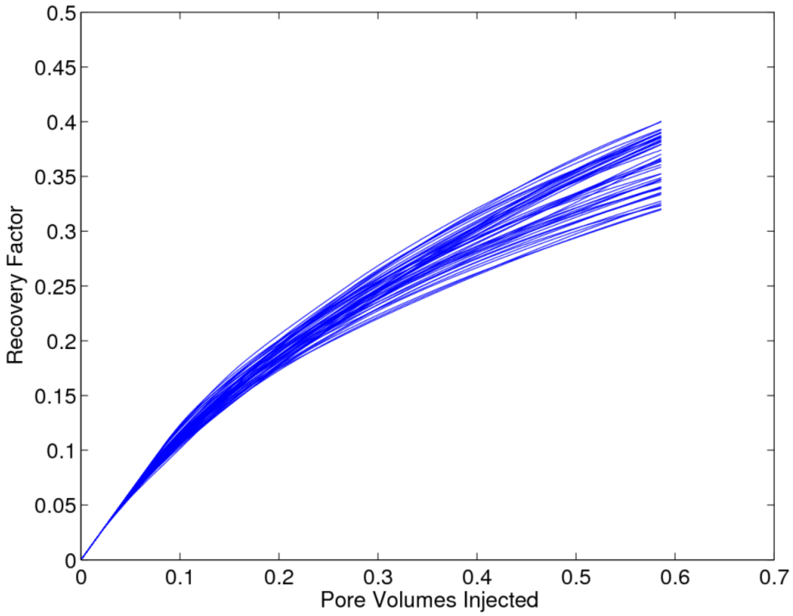

Figure 4-1 Recovery curves for polymer flooding simulated with UTCHEM for 50 reservoir models that reflect the same variogram model

Multiple-points statistics are higher order spatial statistics that capture more complex heterogeneity exhibited by reservoir properties. Multiple-point statistics algorithms such as *snesim* (Strebelle, 2001) have been developed over the last decade to generate multiple realizations of reservoir models while honoring statistics extracted from training images that depict complex sinusoidal or curvilinear shapes of geologic features. Although major geologic structures are important for delineating flow paths, the particular location of geologic bodies plays an important role on the well responses as well. Figure 4-2 shows that realizations constrained to statistics borrowed from training images yield a tighter spread in recovery factor compared to realizations from variogram models. But issues still remain such as the selection of training image most representative of the underlying geology and the choice of spatial template to retrieve the relevant statistics.

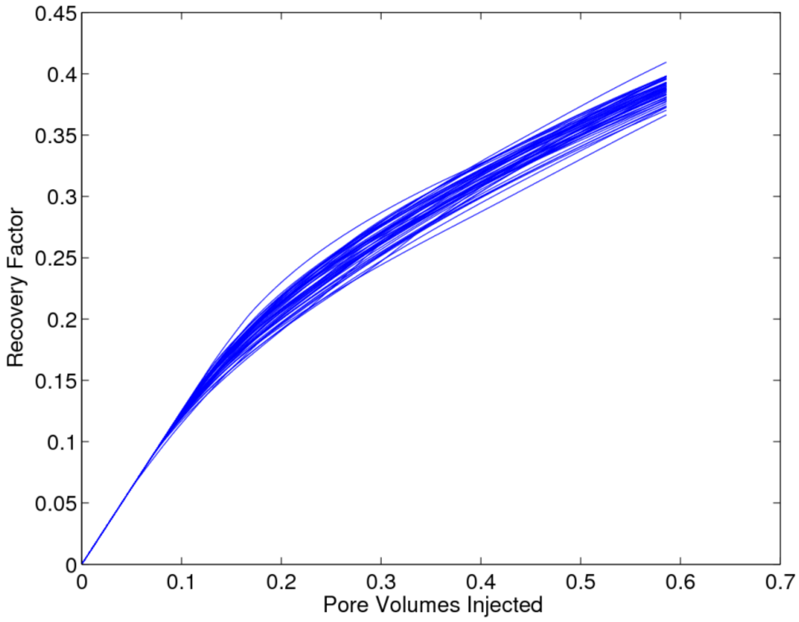

Figure 4-2 Recovery curves for polymer flooding simulated using UTCHEM for 50 reservoir models generated constrained to the statistics retrieved from the same training image.

Stochastic simulation algorithms such as sequential simulation reproducing a variogram model (two-point statistics) or multiple-point statistics inferred from a training image, are capable of representing spatial variations in reservoir properties. However, significant uncertainty still persists at the end of the modeling process and these models each impart subtle differences in production data. Our objective is to differentiate reservoir models based on the production response to polymer injection. It is our conjecture that the grouping based on dynamic characteristics may lump together models constrained to two and multiple point statistics based on the similarity of response over a local scale (even though the models may exhibit differences over a more global scale)

Shook and Mitchel (2009) used the term *dynamic heterogeneity* to describe spatial variations in reservoir attributes that influence the flow between wells in a reservoir. Characterization of dynamic heterogeneity focus on capturing the main flow paths and areal spread of displacement fronts. This is useful to identify reservoir models that yield similar characteristics of breakthrough and recovery. Polymer flooding, as displacement process, is sensitive to flow path lengths and connected structures within the reservoir from injectors to producers. For example, short flow paths manifest in early breakthrough of fluids and therefore reduced recovery. An appropriate measure of heterogeneity should be sensitive to these phenomena and allow differentiating one reservoir model from another even if they share two-point or multiple-point statistics.

4.4. GENERATION OF THE INITIAL ENSEMBLE OF RESERVOIR MODELS

The initial ensemble of reservoir models represents the uncertainty in geology and heterogeneity that is to be analyzed. Therefore, that ensemble should be wide enough to accommodate different geologic scenarios, usually expressed through variogram models and training images, and variations in heterogeneity within each geologic scenario. In addition, all reservoir models in the initial ensemble should honor the conditional data available, i.e. reservoir properties at sampled locations (well logs) and indirect information (seismic). Geostatistical algorithms like *sisim* and *snesim* are used to generate multiple reservoir models as stochastic realizations of a sequential simulation process. The initial ensemble should be large enough to include variations in well connectivity regardless of the spatial template used to generate the models because the uncertainty space is constructed based on differences in well connectivity assessed with the proxy model as explained in the next section.

An ensemble of 600 reservoir models was generated to demonstrate the method. The 600 reservoir models are realizations of stochastic algorithms for sequential simulation. 400 realizations were generated using sequential indicator simulation (*sisim* – Deutsch, 1998), which used 8 different variogram models to generate 50 realizations of each type. The other 200 realizations were generated using single normal equation simulation (*snesim*), which used 4 different training images (ellipsoids and sinusoids) to create 50 realizations of each type. The examples of reservoir models shown in Figure 4-3 give an idea about the variety of geologic scenarios used.

4.5. USE OF PROXY MODEL TO MEASURE WELL CONNECTIVITY CHARACTERISTICS

The proxy model for polymer flooding described in Chapter 3 is employed to extract characteristics of each reservoir model that are related to well connectivity. Since the responses from the proxy model are tailored to polymer flooding and its computations are fast, a large number of reservoir models can be processed within a reasonable timeframe. In this application, each reservoir model is characterized by the following four parameters: mean and variance of the flow path length, recovery factor and breakthrough time. Other suitable parameters are also proposed according to the response of interest.

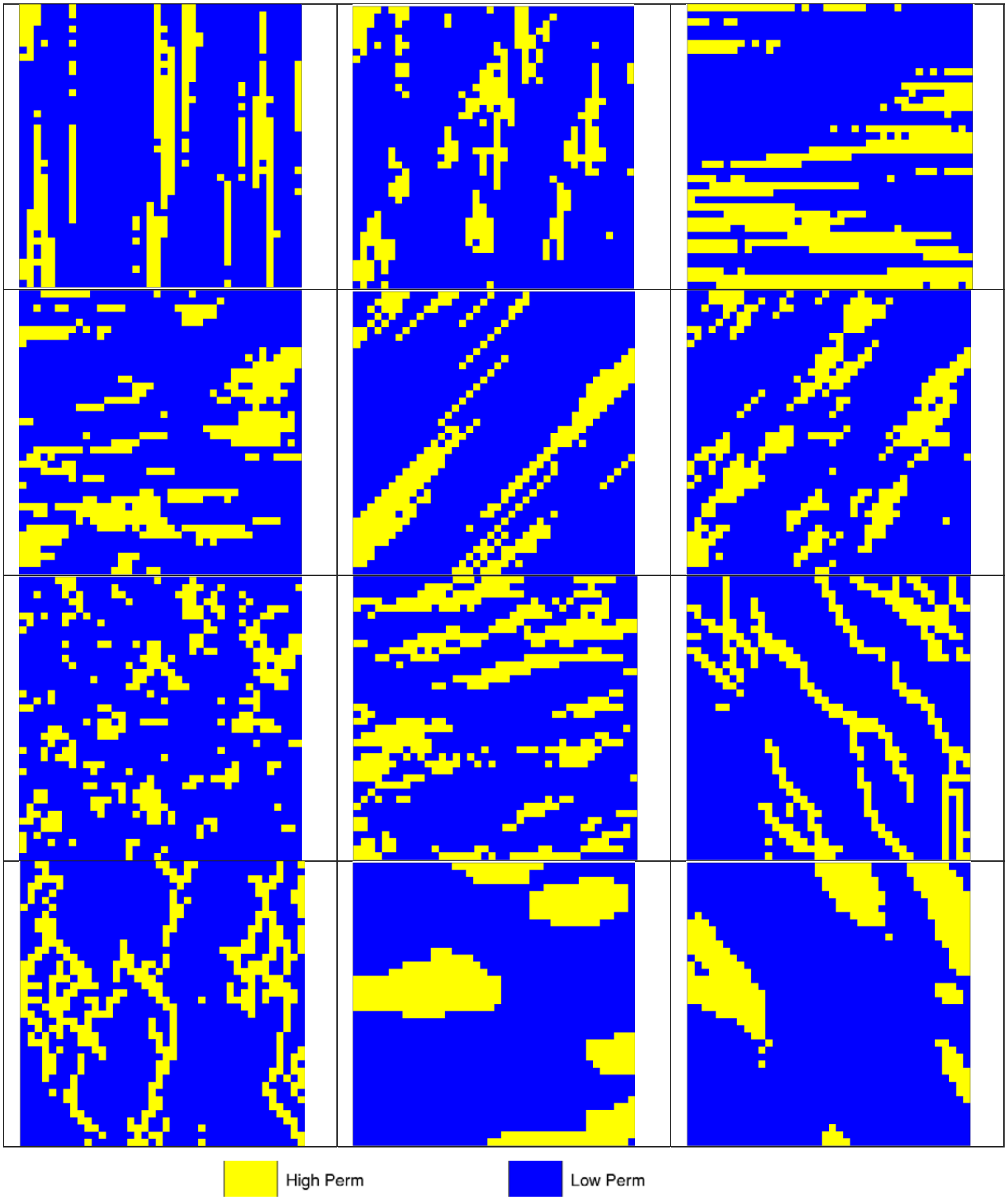

Figure 4-3 Examples of realizations generated using *sisim* and *snesim*. A total of 600 reservoir models were generated. The injector is located at the bottom center and the producer at the top center.

4.5.1 Particles Path Length Distribution

The successive steps taken by a random particle throughout its travel are traced internally within the proxy model. The distance traveled by a particle is an indicator of how tortuous its path is. Short flow paths are sign of early breakthrough, while long paths mean late breakthrough. The path length of all particles is recorded in order to analyze its distribution. The units of the path length are arbitrary, but it measures the deviation from the shortest path.

The distribution of path lengths varies from a uniform distribution for a homogeneous medium to a peaked distribution for a medium with a permeability streak as shown in Figure 4-4 and Figure 4-5 respectively. The mean of the distribution is an indicator of how fast the particles traveled. The variance is an indicator of the areal coverage of the displacement; better areal coverage is reached if the variance of the distribution is high. Each reservoir model has its characteristic distribution according to the level of heterogeneity and in most cases exhibits positive skewness.

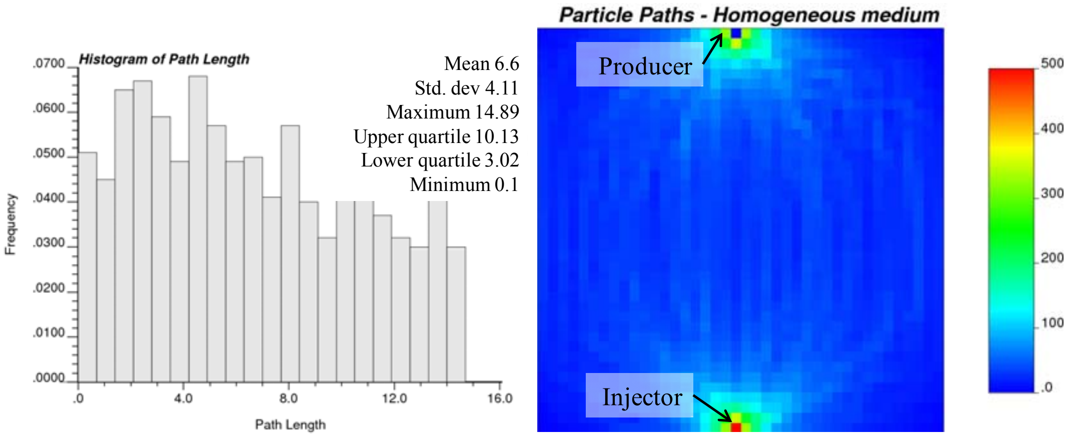

Figure 4-4 a) Histogram of path length for a homogeneous medium. b) Probable particle paths. 1000 particles were sent from the injector. Color scale indicates the number of particles that passed through each location on their way to the producer.

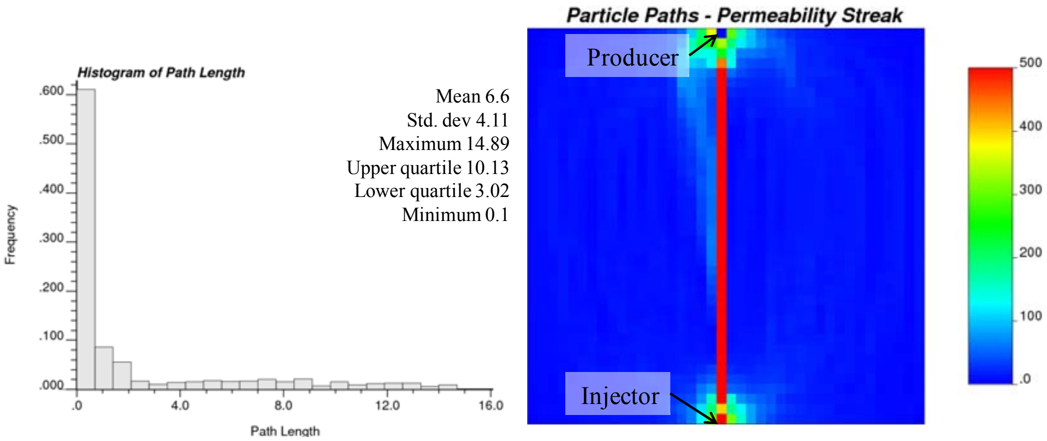

Figure 4-5 a) Histogram of path length for a medium with a high permeability streak. b) Probable particle path is preferentially through the high permeability streak.

4.5.2 Breakthrough Time and Recovery Factor

In order to test the correlation between the responses obtained with the proxy model and actual responses from flow simulation (UTCHEM), the 600 reservoir models in the initial ensemble were processed with the proxy model. The breakthrough time from the proxy model (arrival time of first particle) is well correlated to the production data relevant in polymer flooding. For example, the onset of water production in UTCHEM is well correlated to the particle breakthrough from the proxy model as shown in Figure 4-6.

The highest variability in recovery is exhibited after breakthrough because before breakthrough there is no variation regardless of the heterogeneity, while at late time most of recovery factor curves flatten. In this particular study the recovery factor at 0.27 pore volumes showed the highest variability among all the realizations.

Despite the good correlation between particle first arrival and breakthrough from UTCHEM, the correlation between the pressure drawdown and the breakthrough time from the proxy is weak as indicated in Figure 4-7. This is because well pressure is affected more by local permeability variations than well communication or large-scale heterogeneities.

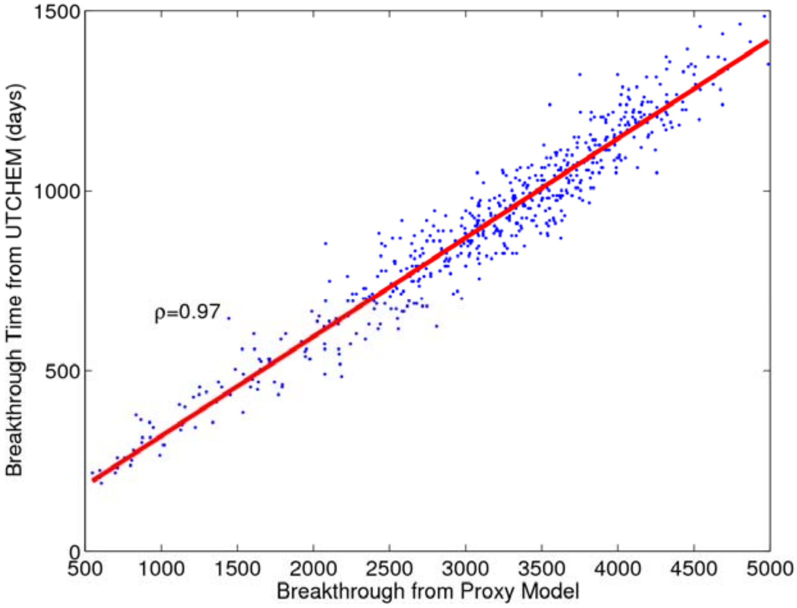

Figure 4-6 Correlation between water breakthrough time from UTCHEM and the breakthrough of particles from the proxy model.

The proxy model also yields an approximate recovery factor for a given dimensionless time (for example after 0.5 pore volumes injected). The recovery factor in the proxy model is calculated by performing a material balance considering the particles injected, the particles initially in the systems and the particles recovered. Figure 4-8 shows high variability on the recovery factor from the proxy for reservoir models with poor sweep efficiency, while the variability on the proxy response is much less for models with better sweep efficiency.

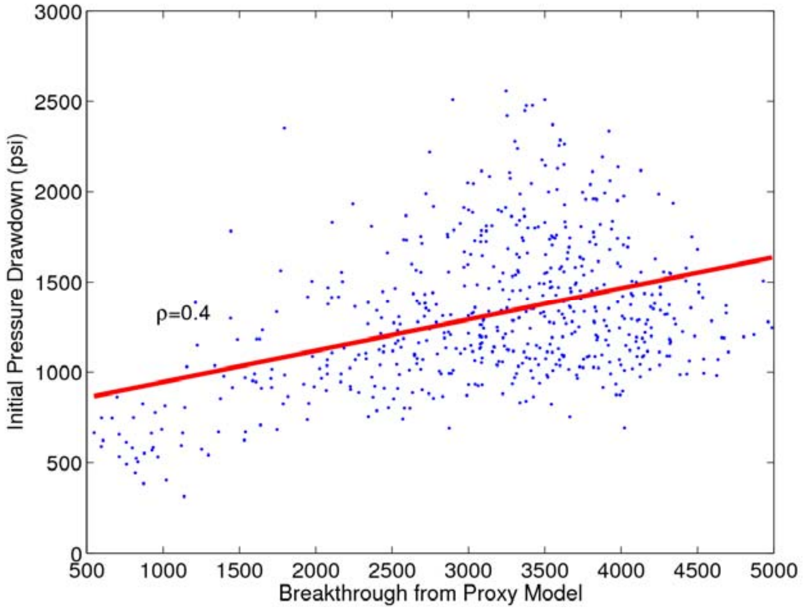

Figure 4-7 Correlation between initial pressure drawdown and the breakthrough time of particles from the proxy model

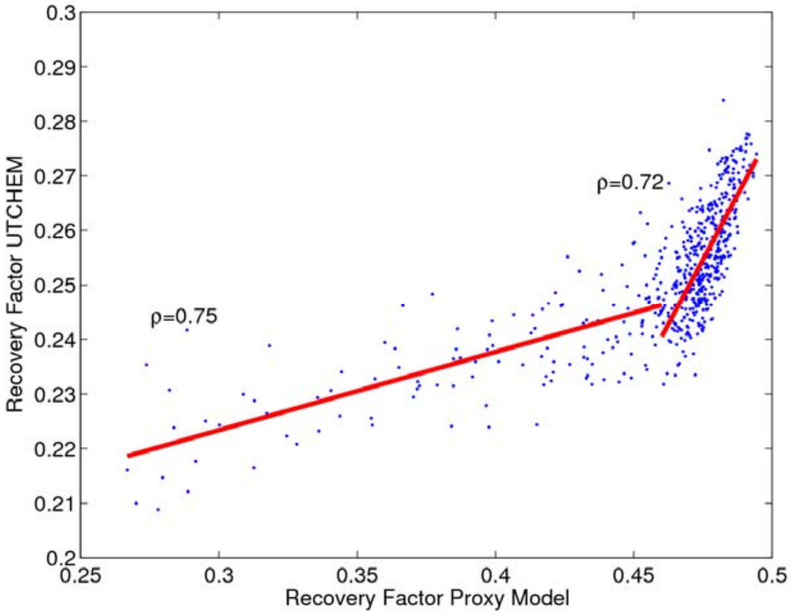

Figure 4-8 Correlation between recovery factors at 0.27 pore volume injected from UTCHEM and recovery factor after injecting 0.5 pore volume from the proxy model.

4.5.3 Proxy Response for Well Pressure

It is important to have a proxy response related to well pressure because before water breakthrough, only well pressure is available for model updating. As stated before, well pressure is mainly affected by well injectivity or productivity, which is function of the permeability surrounding the well. Pressure is dissipated along the least resistant flow path from the wells. We can infer the probable paths for pressure dissipation around wells by following the preferred particle paths. We retain the average permeability of the first 10 locations visited by particles departing from injectors and the last 10 locations visited by particles in their way to producers as expressed in Equation 4.1. The calculation is repeated for all particles in order to get the most probable effective permeability around the wells (injector and producer).

$$\bar{k} = \frac{1}{10NP} \sum_{np=1}^{NP} \sum_{i=1}^{10} k\left(\mathbf{u}_{i,np}\right) \qquad (4.1)$$

k is the permeability at location $\mathbf{u}_{\mathrm{i,np}}$. The sequence $\mathbf{u}_{1,\mathrm{np}}$ to $\mathbf{u}_{10,\mathrm{np}}$ denotes the first 10 (for injectors) or the last 10 locations (for producers) visited by particle np.

Figure 4-9 and Figure 4-10 indicate good correlation between well pressures and the effective permeability of the near well region calculated using the particle path information. For the producer pressure, it was observed that the proxy response is better correlated to the pressure close to the time of water breakthrough. This might be because the simultaneous flow of two fluids at breakthrough is more sensitive to reservoir heterogeneity and this strong dependence on geology of both the proxy and simulator response manifests itself in the form of higher correlation.

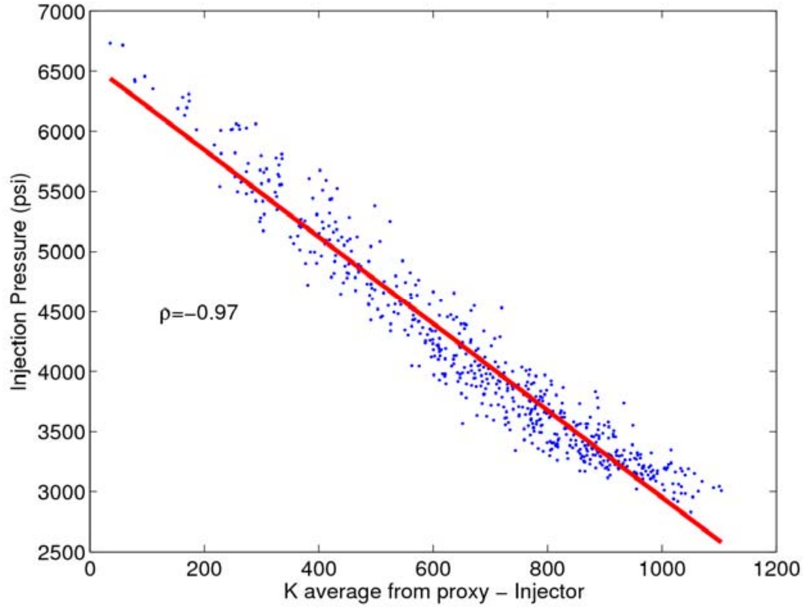

Figure 4-9 Correlation between injection pressure at early time and effective permeability around well calculated using the proxy.

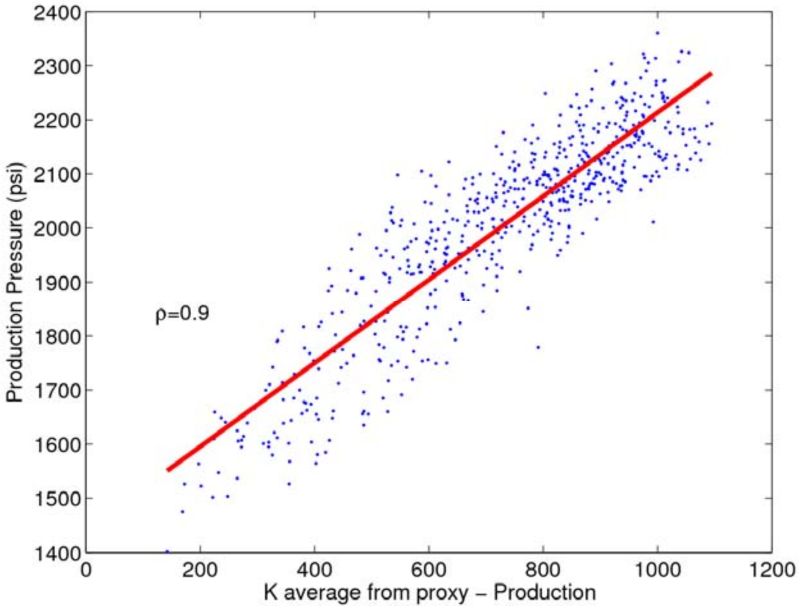

Figure 4-10 Correlation between production pressure close to breakthrough and the effective permeability around the well calculated using the proxy.

4.5.4 Summary of Proxy Responses

In summary, a few responses from the proxy model for polymer flooding that capture the impact of heterogeneity will be retained for facilitating the model selection process. These responses can be calculated for any given reservoir model with any given well configuration. Particle path distribution, breakthrough time, recovery factor and average permeability around wells are characteristic of each reservoir model and will be the basis for defining differences between any two of them as explained in the next sections.

4.6 MEASURE OF DISSIMILARITY BETWEEN RESERVOIR MODELS

Once we have an ensemble of reservoir models representative of the prior geologic uncertainty, the next step is to group them according to their similarity. The word similarity implies that we need a protocol for putting multiple reservoir models side by side and establish a metric for similarity or dissimilarity.

In mathematics, metric is a function that defines a distance (d) between the elements of two sets and satisfies conditions posed in Equations 4.2 through 4.5. In these equations, $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$ etcetera denote points in an n-dimensional space. Although Euclidean distance is the most commonly used definition of distance, other definitions of distances such as Haussdorf distance and Mahalanobis distance also satisfy these minimum conditions.

$$d(\mathbf{x},\mathbf{y})>=0 \text{ (non-negativity)} \tag{4.2}$$

$$d(\mathbf{x},\mathbf{y})=0 \text{ if and only if } \mathbf{x} = \mathbf{y} \text{ (identity of indiscernibles)} \tag{4.3}$$

$$d(\mathbf{x},\mathbf{y})=\mathrm{d}(\mathbf{y},\mathbf{x}) \text{ (symmetry)} \tag{4.4}$$

$$d(\mathbf{x},\mathbf{z}) \leq \mathrm{d}(\mathbf{x},\mathbf{y}) + \mathrm{d}(\mathbf{y},\mathbf{z}) \tag{4.5}$$

Arpat (2005) introduced the term *distance* between reservoir models, referring to a measure of dissimilarity between any two different reservoir models. Consider a vector

x comprising all reservoir properties within a reservoir model as in Equation 4.6, *e.g.* permeability and porosity at all locations. Equation 4.7 defines a measure of dissimilarity *d* between two reservoir models **x** and **y**, as the Euclidean norm the difference vector.

$$\mathbf{x} = \begin{bmatrix} x_1 & \dots & x_N \end{bmatrix} \tag{4.6}$$

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})^T (\mathbf{x} - \mathbf{y})} = \sqrt{\sum_{n=1}^{N} (x_n - y_n)^2} \tag{4.7}$$

If a reservoir model is defined as in Equation 4.6, differences in variables that do not affect the flow contribute to exaggerate the measure of dissimilarity. For instance, differences in permeability in areas of the reservoir disconnected from the main flow paths are added without significance on well responses.

Scheit and Caers (2009) defined distance between reservoir models on the basis of well connectivity using streamline simulation. This approach made the differences between reservoir models more sensitive to actual well responses in water flooding. Similarly, the presented method works with the measures of dynamic heterogeneity obtained from the proxy model and described in Section 4.3. We compare reservoir models by looking at differences in the distribution of the flow paths, breakthrough time, recovery factor and the proxy average permeability around wells. In this way, we can discriminate between reservoir models that will exhibit different production profiles without running a full flow simulation.

4.6.1 Construction of a Metric Space using Multi-Dimensional Scale (MDS)

A metric space is a mathematical set where the distance between elements is defined. The most intuitive metric space is the Euclidean 3D space, where the distance between points is the length of straight line that connects them. The objective is to create a space where multiple reservoir models can be mapped according to the metric of dissimilarity. An ample set of reservoir models that cover the uncertainty is the basis for the construction of the metric space. This metric space is sometimes referred to as the uncertainty space throughout this dissertation.

An illustrative analogy of a space constructed from distances is a topographic map. Suppose we know the distances between three cities. Then, based on those distances, it is possible to find a unique 2D map of the region between these cities, which constitutes the metric space. In that map the absolute coordinates are not important, in fact the latitude and longitude are relative to some reference point in the earth. A new point can be located in that map by referring it to its distance to the already located cities. Also note that although the metric space is two-dimensional, the distance function itself is one-dimensional.

Caers *et al.* (2009) used multi-dimensional scaling (MDS) to construct maps of reservoir models in a two-dimensional space. MDS is a traditional statistical technique used to translate the dissimilarity matrix **D** (Equation 4.8) in an nD Euclidean space. Mode details about MDS are provided in Cox and Cox (1994).

$$\mathbf{D}=\begin{bmatrix} 0 & d\left(\mathbf{x}_1,\mathbf{x}_2\right) & \ldots & d\left(\mathbf{x}_1,\mathbf{x}_{N-1}\right) & d\left(\mathbf{x}_1,\mathbf{x}_N\right) \\ d\left(\mathbf{x}_2,\mathbf{x}_1\right) & 0 & \ldots & \ldots & d\left(\mathbf{x}_2,\mathbf{x}_N\right) \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ d\left(\mathbf{x}_{N-1},\mathbf{x}_1\right) & \ldots & \ldots & 0 & d\left(\mathbf{x}_{N-1},\mathbf{x}_{NN}\right) \\ d\left(\mathbf{x}_N,\mathbf{x}_1\right) & d\left(\mathbf{x}_N,\mathbf{x}_2\right) & \ldots & d\left(\mathbf{x}_N,\mathbf{x}_{N-1}\right) & 0 \end{bmatrix} \tag{4.8}$$

$\mathbf{x}_i$ is a vector that represents one reservoir model as expressed in Equation 4.6. After applying MDS, Caers and Scheidt (2009) used eigen decomposition is to reduce the effect of linearly correlated variables, and retain the effect of independent variables. For example, if a correlation between porosity and permeability exist, the addition of differences in porosity plus differences in permeability would result in unwanted redundancy. Since we are dealing with variables spatially distributed, the reservoir properties separated by a short physical distance should be correlated. For instance, if a geologic model is constrained to the permeability at the well location, then the permeability in regions close to the well should be correlated. Eigen decomposition aims to retain variables that are statistically independent, making one reservoir model distinct from another.

As a result, Caers and Scheidt (2009) visualized a large number of reservoir models by projecting them to a 2D space, where a distance-based association of similar reservoir models is possible.

4.6.2 Principal Component Analysis for Constructing a Metric Space

We opted for using Principal Component Analysis (PCA) approach to construct the metric space because it retains only the measures of heterogeneity that exhibit more variability directly during the construction of the space. If we choose not to reduce the dimensions, the result is mathematically equivalent to the results with MDS. Principal component analysis (PCA) is a mathematical procedure that transforms a number of correlated variables into a smaller number of uncorrelated variables. The variability of the data is maximized on the first component, and the remaining variability is accounted for in the subsequent components. The variables retained from processing each reservoir model through the proxy model are correlated. The objective of PCA is to eliminate this redundancy by analyzing the covariance between variables. Let $\mathbf{x}_i$ be an n x 1 vector that contains the characteristic variables of the i^{th} reservoir model (Equation 4.9). The elements of the covariance matrix **C** (n x n matrix) are the covariance between the random variables.

$$\mathbf{x}_i \equiv \begin{bmatrix} \mu & \sigma^2 & t_{bt} & R \end{bmatrix}^T \equiv \begin{bmatrix} x_{i1} & x_{i2} & x_{i3} & x_{i4} \end{bmatrix}^T \in \mathbb{R}^n \tag{4.9}$$

$$\mathbf{C} = E\left[\left(\mathbf{x} - E[\mathbf{x}]\right)\left(\mathbf{x} - E[\mathbf{x}]\right)^T\right] \tag{4.10}$$

E[**x**] denotes the expected value vector, which contains the expected value of the random variables over all reservoir models. The covariance matrix is decomposed into its eigenvector and eigenvalues as in Equation 4.11.

$$\mathbf{C} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{U}^{T} \tag{4.11}$$

U is an n x n matrix that contains the eigenvectors of **C**, and **Σ** is a diagonal matrix that contains the eigenvalues of **C**. Finally, each vector $\mathbf{x}_i$ is multiplied by the eigenvectors of the covariance matrix (**U**) (See Equation 4.12). The result of this multiplication is called principal component (**PC**). Each reservoir model has its own n principal components. The top *k* principal components corresponding to the *k* highest eigenvalues are chosen such that these components retain most of the original variance. Finally, a plot of the top *k* principal components of each reservoir model defines the metric or uncertainty space.

$$\mathbf{PC}_i = \mathbf{U}^T \mathbf{x}_i \tag{4.12}$$

To illustrate this procedure, the 600 reservoir models described in Figure 4-3 were processed through the proxy model and the four responses were stored. Subsequently, the covariance between those responses was computed. The covariance matrix was decomposed into its eigenvector and eigenvalues. Finally, the top 2 principal components were plotted in Figure 4-11. It can be observed that the data has maximum variability over the first eigenvector. Furthermore, there is no systematic trend observable in the data, which implies independence between axes. This map helps visualize all realizations on a single plane. Also, it is useful for uncertainty analysis because this space can be easily sampled (to yield entire realizations). This map is at the core of the method for model selection presented subsequently.

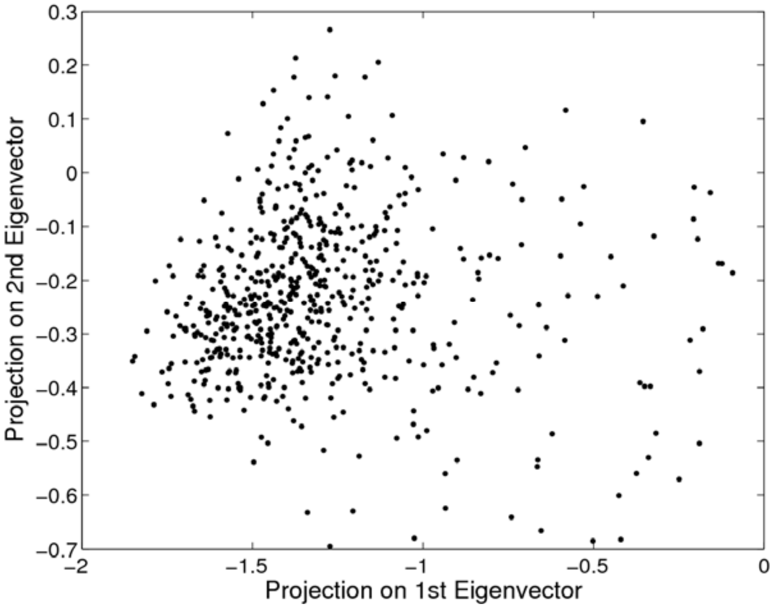

Figure 4-11 Projection of 600 reservoir models on top 2 eigenvectors. Each dot represents one reservoir model.

4.6.3 Correlation between Well Connectivity and Principal Components

The position of each reservoir model in the metric space is correlated with well connectivity characteristics. Reservoir models can be associated within clusters or groups using the k-means clustering algorithm - a data mining procedure for cluster analysis that partitions an entire set of points into *m* clusters. Points are associated to the cluster whose centroid is nearest to that point. k-means is not the only option used for clustering reservoir models: Scheidt and Caers (2009) used kernel-k-means (KKM) for clustering in which a non-linear transformation of the points is performed such that the cluster after the transformation exhibits linearity in the metric space. However, the choice of the kernel is arbitrary (e.g. radial basis functions). In contrast, the objective of eigen transformation is to project reservoir models into orthogonal vectors using the eigenvectors as the transformation basis.

Eight clusters of reservoir models were formed using simple k-means as indicated in Figure 4-12. The weighted average properties of models in a cluster such as the average permeability of each cluster are calculated using Equation 4.13.

$$\bar{k}(\mathbf{u})_{nc} = \sum_{nr=1}^{NR} \omega_{nr} k(\mathbf{u}) \quad \forall \text{realizations} \in \text{ cluster } nc \tag{4.13}$$

$$\omega_{nr} = \left[\left(\mathbf{PC}_{nr} - \overline{\mathbf{PC}} \right)^T \left(\mathbf{PC}_{nr} - \overline{\mathbf{PC}} \right) \right]^{-1} \tag{4.14}$$

Equation 4.16 implies that the weights for each realization (nr) within a cluster (nc) is inversely proportional to the distance to the center of the cluster. Reservoir models closer to the center of their cluster are more representative of the group than models at the boundaries. The ensemble of reservoir models was divided into eight clusters in order to form small groups at distant positions in the space, and to show transitions from regions in that space. Reservoir models at the right of Figure 4-12 (clusters 3 and 7) show a pronounced high permeability streak that imparts high well connectivity. In contrast, reservoir models at the left of Figure 4-12 (cluster 4) show poor well connectivity because communication between wells passes through low permeability zones. Reservoir models smoothly transition from high to low connectivity through this map.

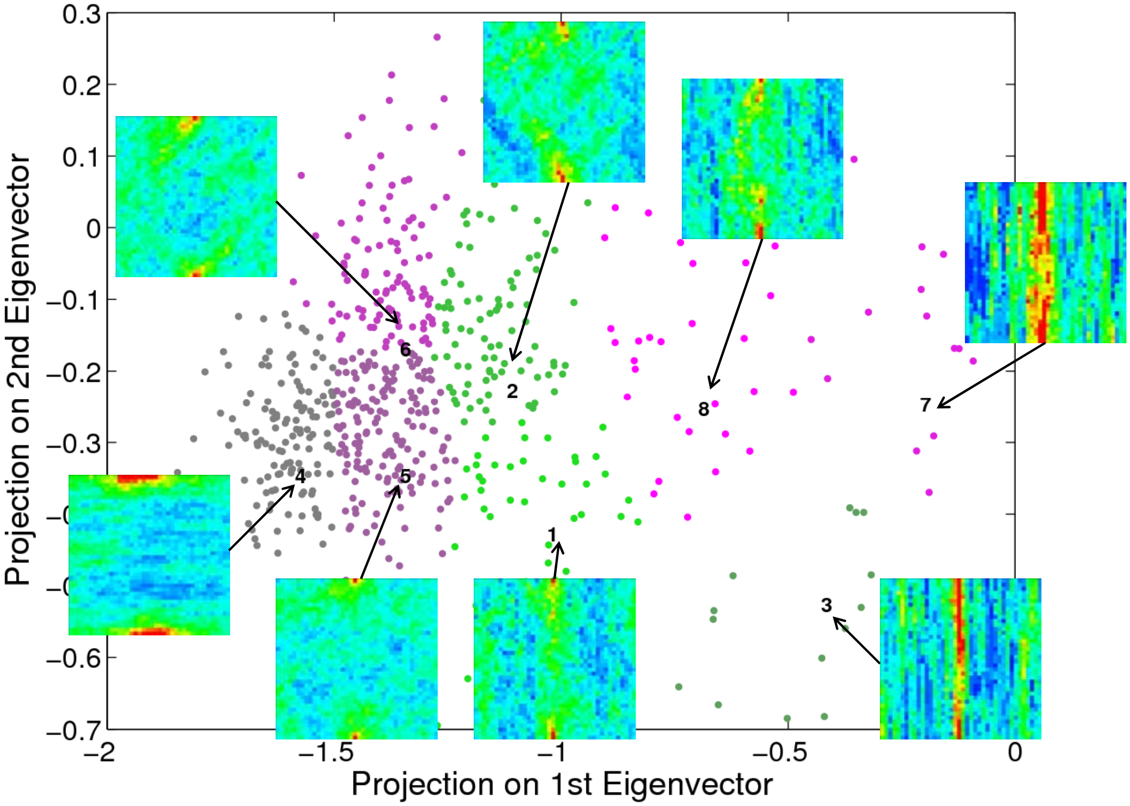

Figure 4-12 Clustering of reservoir models based on Euclidean distance in the top 2 eigenvectors. Average permeability maps from each cluster are displayed. The eight clusters obtained using k-means are color coded.

Intuitively, the organization in well connectivity should be reflected in production data since polymer flooding is a displacement process where fluids migrate according to well connectivity. It is hypothesized that the production data (water breakthrough, recovery factor) is correlated to the position of reservoir models in this map and consequently can be used to refine the selection of models in the prior set so as to render uncertainty quantification and production optimization using optimal control feasible. These issues and the practical implementation of the model selection process are discussed in the next and subsequent chapters.

Chapter 5: Selection of Reservoir Models using Production Data

5.1 OVERVIEW

This chapter describes a novel approach for selection of reservoir models based on the observed production data using a measure of dissimilarity and Bayesian inversion. Production data is used to guide the process of selection of clusters of reservoir models formed in the metric space described in Chapter 4. First, the relationship between distance in the metric space and production data is validated by analyzing production data from two clusters distant in the metric space. Then the method for reservoir model selection is described and demonstrated through two examples of polymer flooding. The selection of a small group of reservoir models with high probability of representing the actual reservoir production prepares the stage for production optimization discussed in Chapter 6.

5.2 RELATIONSHIP BETWEEN PRODUCTION DATA AND DISTANCE IN METRIC SPACE

The idea of selecting reservoir models based on their position in metric space relies on the hypothesis that distances in that space and difference in terms of production data are correlated. In a polymer flooding operation, wells are monitored with pressure sensors either at well head or down hole; samples of water production and polymer concentration are also routinely acquired to follow the progress of the flooding. Injection and production pressure depend mainly on rock properties in the near wellbore area and the viscosity of the injected polymer. On the other hand, the breakthrough of water and polymer depend on the length of the flow path along which the displacement takes place. Hence, these production data should be informative of effective heterogeneity and well connectivity.

Prior to fluid breakthrough the only production data available is injection and production pressure. Therefore, it is important for this application to establish differences among reservoir models based on a variable correlated with well pressure. It was shown in Section 4.3.3 that permeability at the locations most visited by random particles near the wells exhibit a good correlation with well pressure. The distribution of path lengths is used as secondary information.

Polymer injection was simulated with UTCHEM on the same 600 reservoir models described on Section 4.3.2. Water production, injection pressure and production pressure were monitored during the initial 250 days of simulation. The maximum injection pressure allowed is 9250 psi and the minimum production pressure is 200 psi. Figure 5-1, Figure 5-2 and Figure 5-3 show high variability of water production and well pressure observed over the prior geologic models.

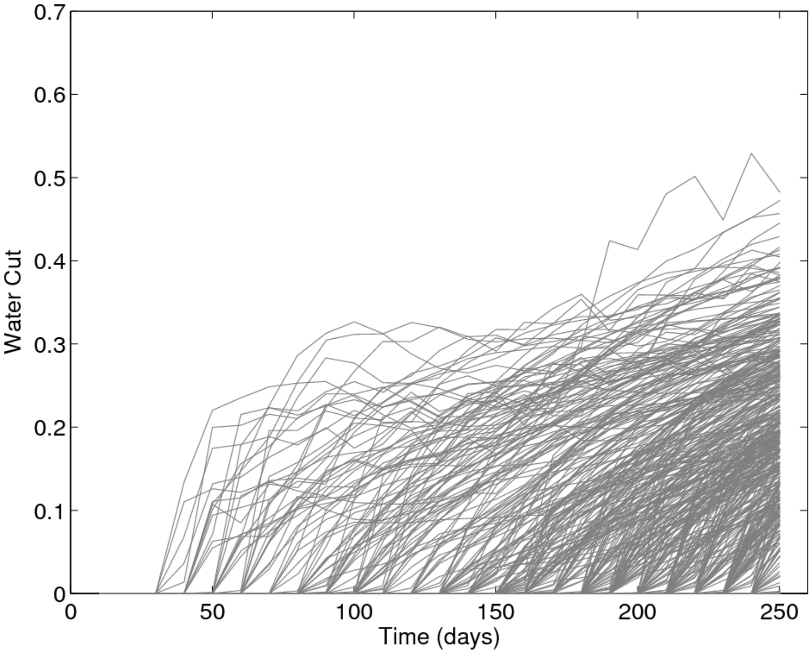

Figure 5-1 Simulated water cut for 600 reservoir models after polymer injection for 250 days.

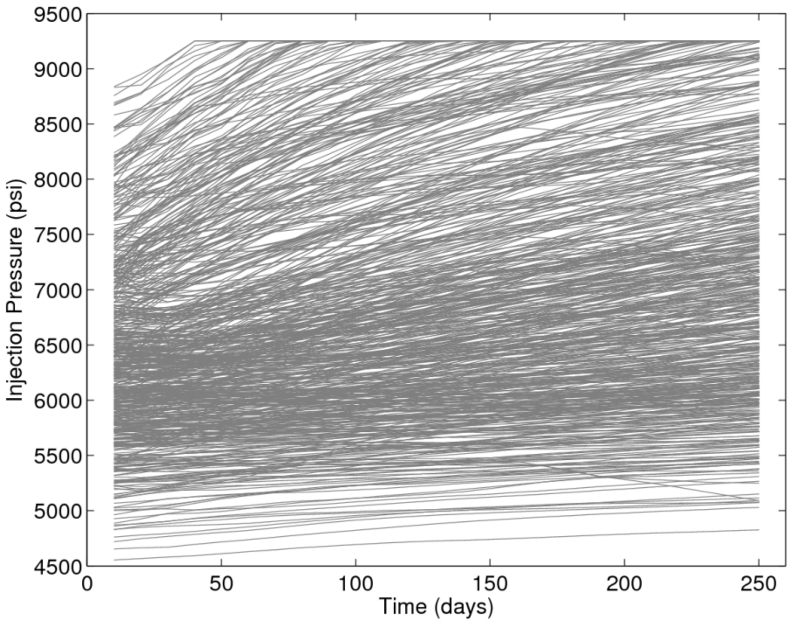

Figure 5-2 Simulated injection pressure for 600 reservoir models after polymer injection for 250 days.

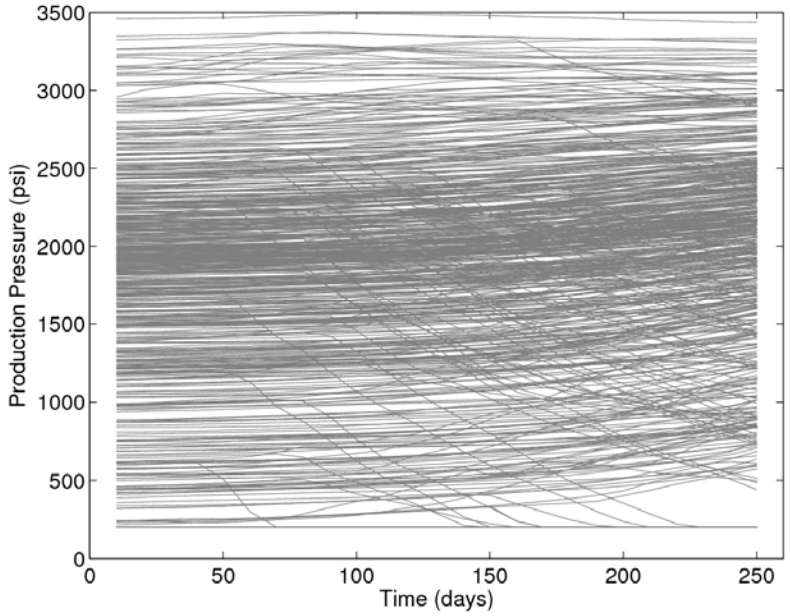

Figure 5-3 Simulated producer pressure for 600 reservoir models after polymer injection for 250 days.

The ensemble of realizations was processed through the proxy model to obtain (1) the mean and variance of the path length and (2) the averaged permeability around each well. The vector $\mathbf{x_i}$ in Equation 5.1 includes these four random variables for the i^{th} reservoir model. The correlation coefficient between the average permeability around injector ($\bar{k}_{inj}$) and the average permeability around producer ($\bar{k}_{prod}$) is 0.1135 as can be seen on the correlation coefficient matrix ($\boldsymbol{\rho}$) expressed in Equation 5.2. This poor correlation between $\bar{k}_{inj}$ and $\bar{k}_{prod}$ is because those variables depend on the permeability near well locations that are far apart. It can be noticed also that $\bar{k}_{inj}$ and $\bar{k}_{prod}$ are weakly correlated to the breakthrough and recovery factor from the proxy model, which implies that local permeability does not have a great impact on the length of the main flow paths.

$$\mathbf{x}_i = \begin{bmatrix} R & tbt & \bar{k}_{inj} & \bar{k}_{prod} \end{bmatrix}^T \tag{5.1}$$

$$\boldsymbol{\rho} = \begin{bmatrix} 1 & 0.8597 & -0.3006 & -0.1986 \\ 0.98597 & 1 & -0.2242 & -0.0432 \\ -0.3006 & -0.2242 & 1 & 0.1135 \\ -0.1986 & -0.0432 & 0.1135 & 1 \end{bmatrix} \tag{5.2}$$

The eigen decomposition of the covariance matrix is:

$$\mathbf{C} = \left(\mathbf{X} - \overline{\mathbf{X}}\right)\left(\mathbf{X} - \overline{\mathbf{X}}\right)^T = \mathbf{U\Sigma U}^T \tag{5.3a}$$

where

$$\boldsymbol{\Sigma} = \begin{bmatrix} 0.0798 & 0 & 0 & 0 \\ 0 & 0.0483 & 0 & 0 \\ 0 & 0 & 0.0376 & 0 \\ 0 & 0 & 0 & 0.0046 \end{bmatrix} \tag{5.3b}$$

$$\mathbf{U} = \begin{bmatrix} -0.5862 & 0.1337 & 0.2119 & 0.7705 \\ -0.6407 & 0.3238 & 0.3026 & -0.6269 \\ 0.4149 & 0.0783 & 0.9049 & 0.0532 \\ 0.2715 & 0.9333 & -0.2112 & 0.1027 \end{bmatrix} \tag{5.3c}$$

The sum of the first 3 eigenvalues is 95% of the total sum of the eigenvalues. Therefore, three eigenvectors are necessary to explain the variability of the data. This is a consequence of the low correlation coefficients between the average permeabilities and the other variables such as breakthrough time or recovery factor. Figure 5-4 shows the projection of the 600 models in a 3D space with axes defined by the three principal components.

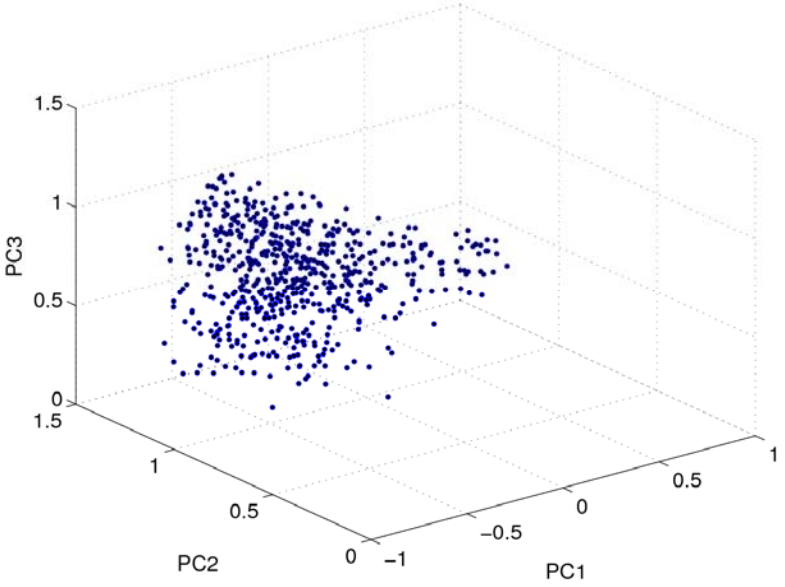

Figure 5-4 Projection of 600 reservoir models on top 3 eigen vectors

To verify if the position of a reservoir model in this eigenspace is reflective of the actual well pressure differences, the cloud of points was divided into eight clusters, and the reservoir models belonging to the two more distant clusters (illustrated in Figure 5-5) were submitted to flow simulation using UTCHEM.

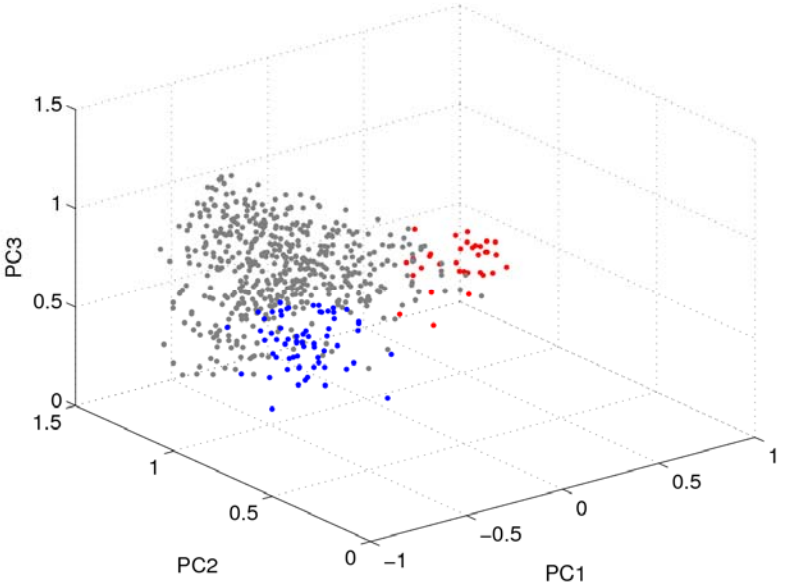

Figure 5-5 Reservoir models labeled in red and blue were selected for flow simulation using UTCHEM because they belong to the two farthest clusters.

Figure 5-6 demonstrates that the distance-based association of reservoir models results in a separation of models based on the similarity of production characteristics. Reservoir models from the red cluster have in common higher producer pressure, low injection pressure and early water breakthrough, as a sign of high well connectivity. On the other hand, reservoir models from the blue cluster exhibit lower well connectivity that results in high injection pressure, low producer pressure and late breakthrough, which implies that fluids are forced to pass through low permeability regions. It is expected that reservoir models from the blue cluster exhibit better areal sweep efficiency during the displacement. Consequently, a small number of realizations can be taken as representative of a group of reservoir models belonging to a cluster.

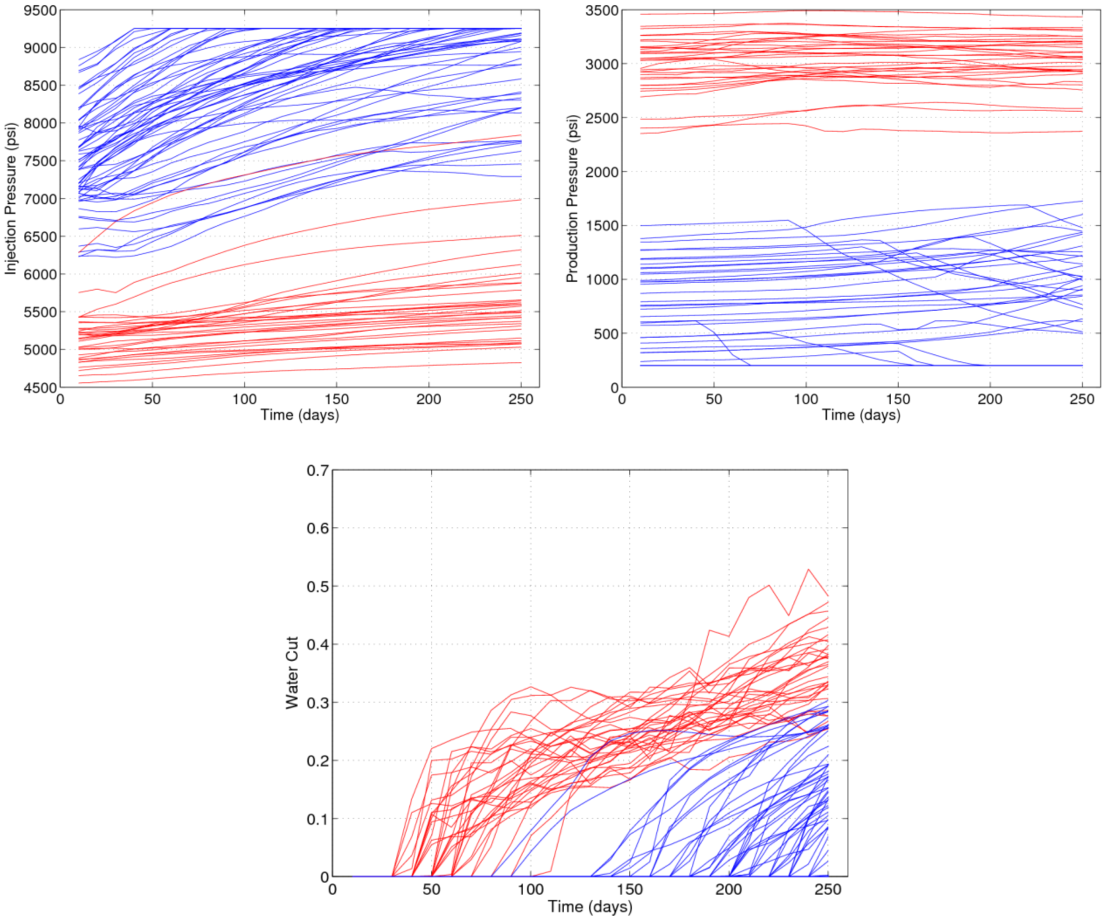

Figure 5-6 Production data for reservoir models belonging to the two distant clusters indicated in Figure 5-5.

5.3 RESERVOIR MODEL SELECTION GUIDED BY PRODUCTION DATA

The method for selection of reservoir models is illustrated through an example with a synthetic reservoir that has one polymer injector and one producer. The geology of the reference reservoir consists of high permeability channels embedded in a low permeability background as shown in Figure 5-7. The permeability contrast between channel and non-channel facies is 1000:10 approximately. Injector and producer are completed in the high permeability regions and communicate through a high permeability streak that causes flow channeling. The reservoir is horizontal and flat at a depth of 3500 ft. The reservoir has a width of 800 ft, a length of 800 ft long and a thickness of 8 ft. The porosity is uniform and equal to 0.1. The initial water saturation is equal to the irreducible water saturation (0.2). The reservoir was discretized into 160x160 grid-blocks of 5x5x8 ft^3 each one. Note that the resolution of the reference reservoir is not necessarily the same as the resolution of the reservoir models used to represent it (40x40 grid-blocks). The initial reservoir pressure is 2500 psi.

The water-oil mobility ratio at the end points (=10.9) and channel heterogeneity are unfavorable for oil recovery, making this reservoir a good candidate for polymer flooding. As a preliminary evaluation, polymer and water flooding were simulated with UTCHEM for a period of 2000 days. The water injection rate is 300 ft^3/day, whereas the polymer injection rate is limited to 50 ft^3/day due to the low injectivity of polymer. Therefore at the final time the injected volume of water is 6 times higher than the injected volume of polymer.

Partially hydrolyzed polyacrylamide (HPAM) polymer is injected at a concentration of 750 ppm. The polymer viscosity at the injected concentration is 20 cP at zero shear rate. The polymer solution injected is a shear thinning fluid whose viscosity is reduced to 10 cP at a shear rate of 280 sec^{-1} as shown in Figure 2-5. The polymer retention is 60 μgr/gr of pore volume at the injected concentration. The salinity of the aqueous phase is 0.006096 meq/ml. The modeling parameters of this polymer solution were taken from Dakhlia (1995). Specific parameters input of the rheologic model and adsorption can be found in Appendix-1, in the UTCHEM deck file.

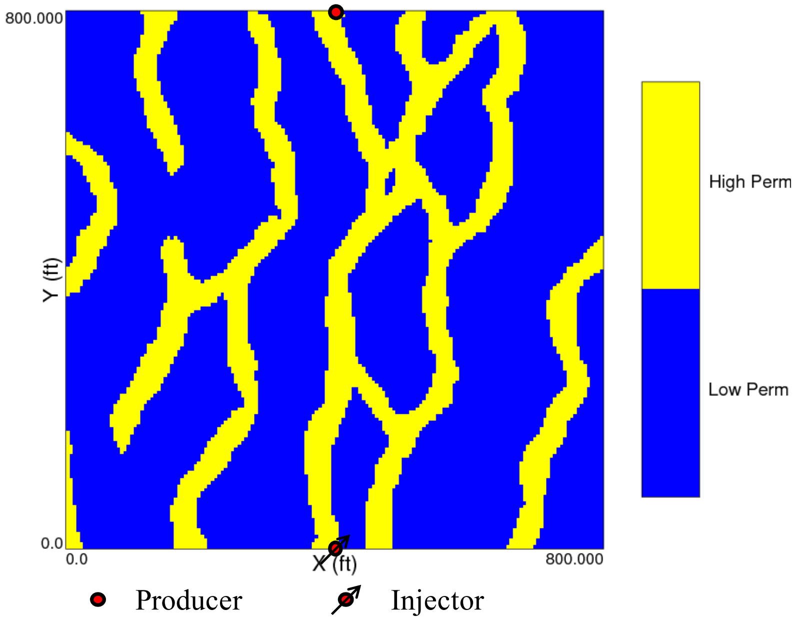

Figure 5-7 Facies map of reference reservoir with two wells

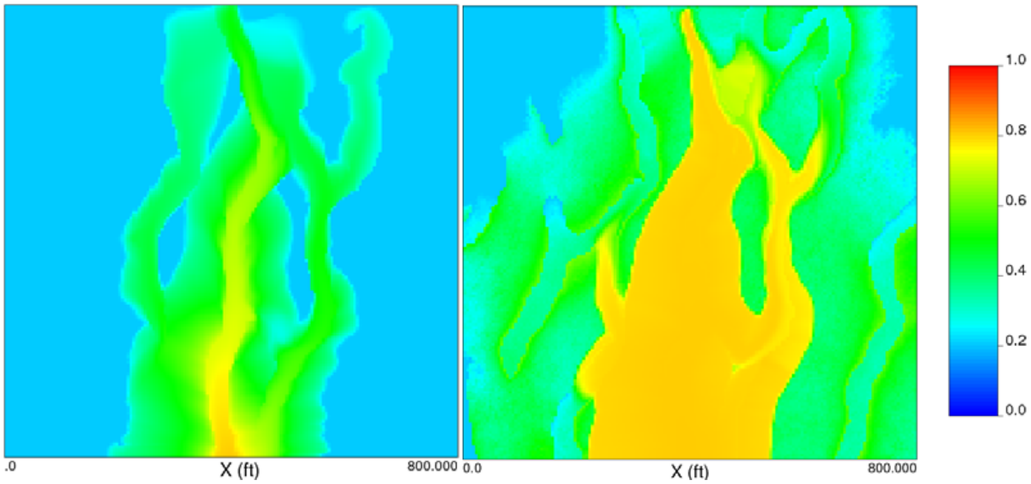

Figure 5-8 Comparison of water saturation corresponding to water (left) and polymer (right) flooding after 0.3 pore volumes of injection simulated using UTCHEM

Figure 5-8 shows that water flooding results in poor sweep due to the presence of flow channels, whereas the effect of heterogeneity is reduced in polymer flooding resulting in better areal sweep. For the water flooding case, the water cut at 0.3 pore volumes injection is already high (0.78); most of the injected water is channeled directly to the producer without mobilizing any additional oil. On the other hand, in the polymer flooding case although water has already broken through (water cut = 0.6) the polymer plume has swept a considerably larger area. Consequently, the economics of polymer flooding is better than that of water flooding as shown in Figure 5-9. Even though at the beginning of the injection process the revenue from water flooding is higher due to higher initial oil production, this effect is offset significantly by the incremental oil recovered later during polymer flooding.

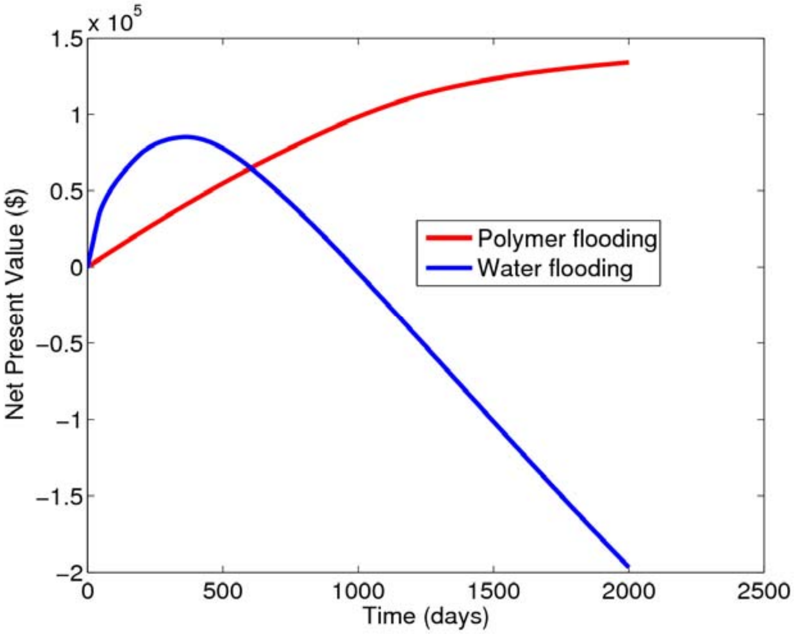

Figure 5-9 Comparison of net present value for water and polymer flooding

The production data that used as reference to guide the model selection process are the water cut, the injection pressure and the producer pressure observed during polymer flooding. This data was monitored every 10 days during the first 250 days as plotted in Figure 5-10. Injection pressure increases as the low-mobility polymer bank grows. The producer pressure also increases as water bank (high mobility) approaches to the well location. Water production initiates after about 50 days of production, and continues increasing as the water channels through the high permeability channel between the wells.

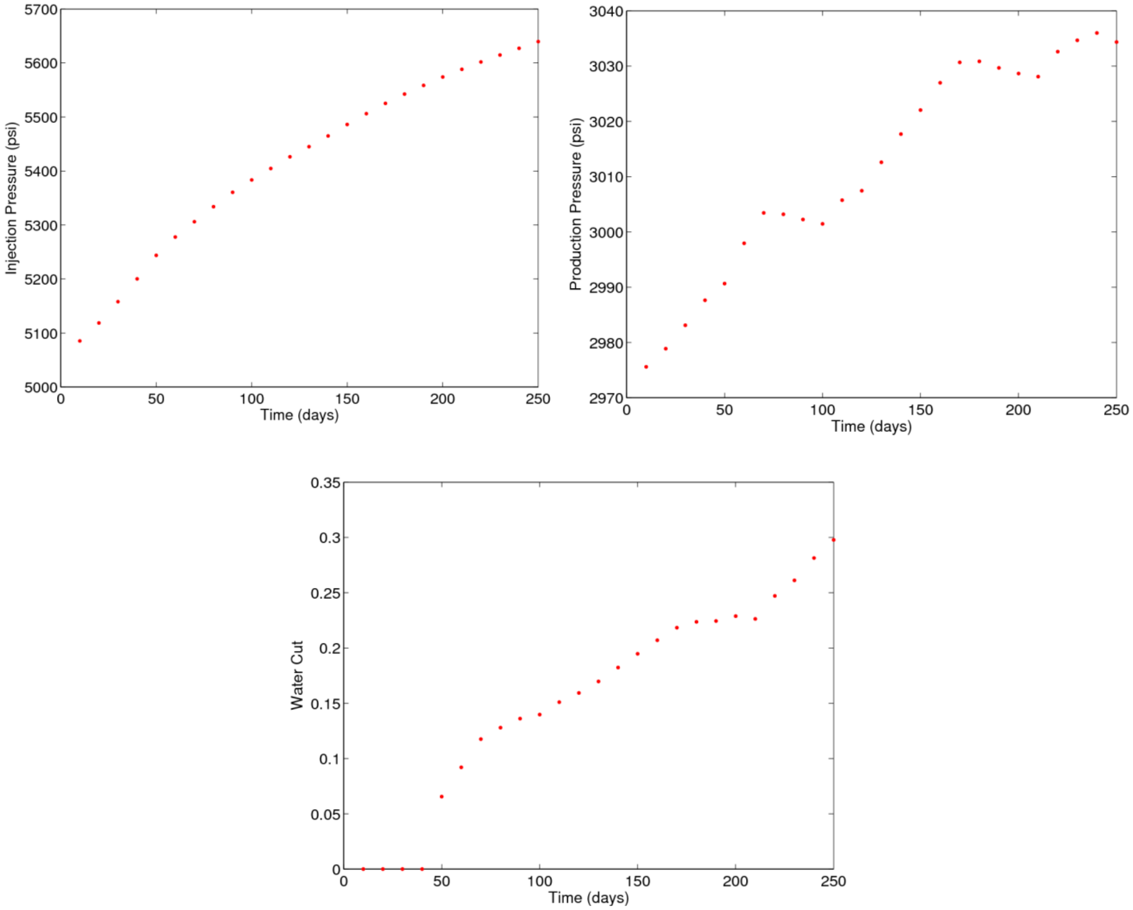

Figure 5-10 Production data from reference reservoir after the initial 250 days of polymer injection.

5.4 COMPUTING AND UPDATING PROBABILITY OF CLUSTERS OF RESERVOIR MODELS

Once the production data history is available, it is used to guide the selection of reservoir models that have similar production characteristics. The ensemble of reservoir models mapped in the metric space is the starting point of the selection process.

Before production data is presented, all reservoir models are equally probable because they all honor the static conditioning data and the prior geologic interpretations presented in the form of variogram models, training image or any other geological description. Therefore, the prior probability of sampling a realization from cluster *i* is equal to the fraction of realizations contained in cluster *i* over the total number of realizations in the ensemble (NR) as expressed in Equation 5.4.

$$\Pr\left\{\mathbf{x} \in \text{cluster } i\right\} = \frac{\text{Number of realizations in cluster } i}{\text{Total number of realizations}} \tag{5.4}$$

The objective of this step is to assess the conditional probability for a reservoir model given the measured production history. The posterior probability is obtained by application of Bayes' rule (Equation 5.5) and comparing representative production data of each cluster to the reference production data.

$$\Pr\left\{\mathbf{x} \in \text{cluster } i \mid \mathbf{D}_{ref}\right\} = \frac{\Pr\left\{\mathbf{D}_{ref} \mid \mathbf{x} \in \text{cluster } i\right\} \Pr\left\{\mathbf{x} \in \text{cluster } i\right\}}{\sum_{j} \Pr\left\{\mathbf{D}_{ref} \mid \mathbf{x} \in \text{cluster } j\right\}} \tag{5.5}$$

$\mathbf{D}_{\text{ref}}$ refers to the production data of the reference reservoir. Now, the question is how to obtain $\Pr\left\{\mathbf{D}_{ref} \mid \mathbf{x} \in \text{cluster } i\right\}$? The scheme is illustrated with an example. Suppose, the well pressure has been acquired for a time period as in Figure 5-11, an uncertainty envelope can be delineated assuming a Gaussian distribution with the measured pressure history as mean. The assumption of Gaussianity is reasonable since a Gaussian measurement error model is commonly assumed. The choice of the variance for that distribution can be based on the statistics of the measurement error.

Knowing the mean and the variance, various confidence intervals can be retrieved around the measured data. A few representative samples of each cluster are selected for flow simulation with UTCHEM.

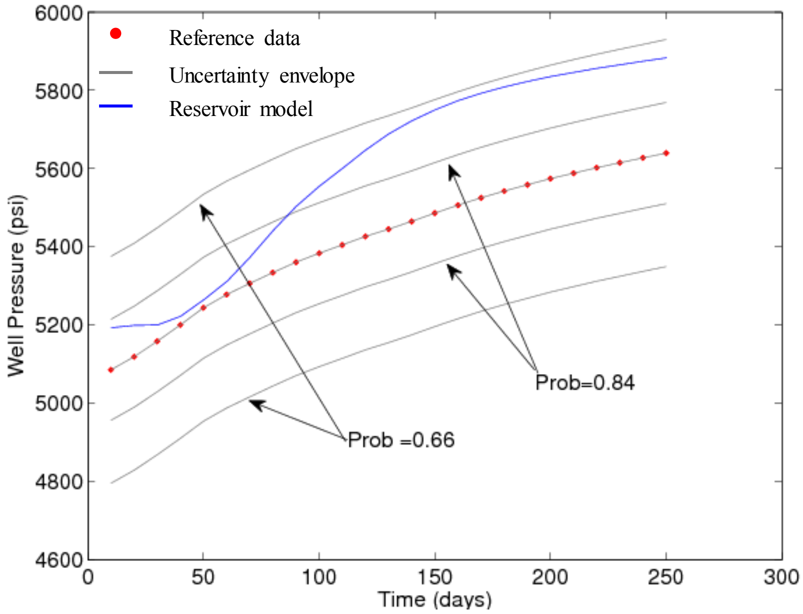

Figure 5-11Uncertainty envelopes around reference data. The production data of one reservoir model falls within the 66% probability envelope.

If the production data of a realization selected from cluster i fall within the 84% envelope, then that probability is assigned to cluster i. This yields the likelihood Pr{$\mathbf{D}_{ref}|\mathbf{x} \in$ cluster i}. The denominator in Equation 5.6 (Pr{$\mathbf{D}_{ref}$}) can be computed by applying the law of total probability. After that, Bayes' rule is used to calculate the posterior probability of the cluster.

$$\Pr\left\{\text{cluster } nc \mid \mathbf{D}_{ref}^{t}\right\} = \Pr\left\{\text{cluster } nc\right\} \frac{\Pr\left\{\mathbf{D}_{ref}^{t} \mid \text{cluster } nc\right\}}{\Pr\left\{\mathbf{D}_{ref}^{t}\right\}} \tag{5.6}$$

At any stage in the selection process, if the variance of the production data after selection does not decrease, it means that either (1) the measure of dissimilarity is not capable of discriminate among reservoir models appropriately, (2) the reservoir models in the group are truly similar such that the differences are intangible, or (3) the number of reservoir models is so limited that assessing the uncertainty is practically impossible.

5.5 HIERARCHICAL SELECTION OF RESERVOIR MODELS

The demonstration continues with the two-well reservoir example. The initial ensemble of reservoir models is partitioned into three clusters as illustrated in Figure 5-12. The prior probability of cluster 1 (blue) is 0.3583, 0.5033 for cluster 2 (green) and 0.1383 for cluster (gray) according to the number of reservoir models contained by each cluster. One out of thirty realizations was selected from each cluster for flow simulation using UTCHEM in order to simulate injection and production pressure during 250 days. A few reservoir models were selected from each cluster and processed using UTCHEM to obtain the injection pressure, producer pressure and water cut as shown in Figure 5-13. Then, the probability of each cluster is updated with the likelihood of the reference production data.

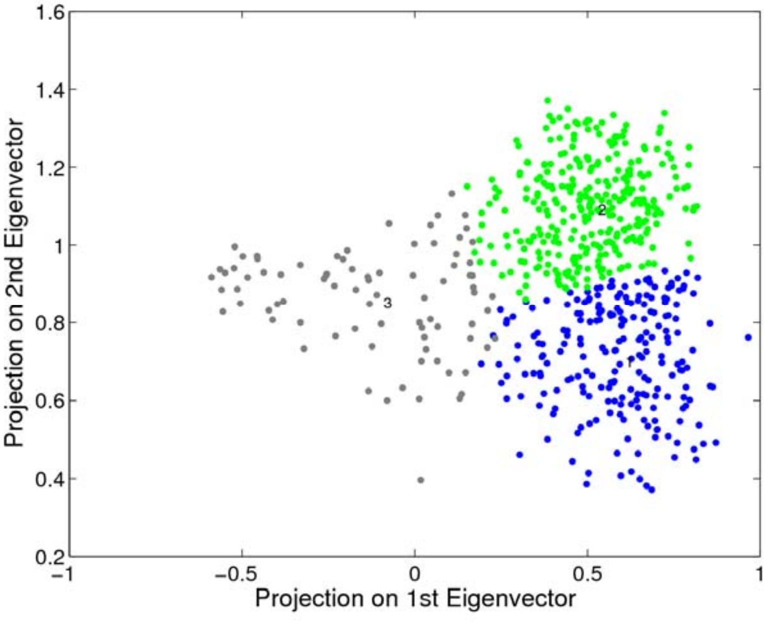

Figure 5-12 Projection of 600 reservoir models on space defined by the two top eigen vectors. Initial ensemble of reservoir models is partitioned in three large clusters: cluster 1 (blue), cluster 2 (green) and cluster 3 (gray)

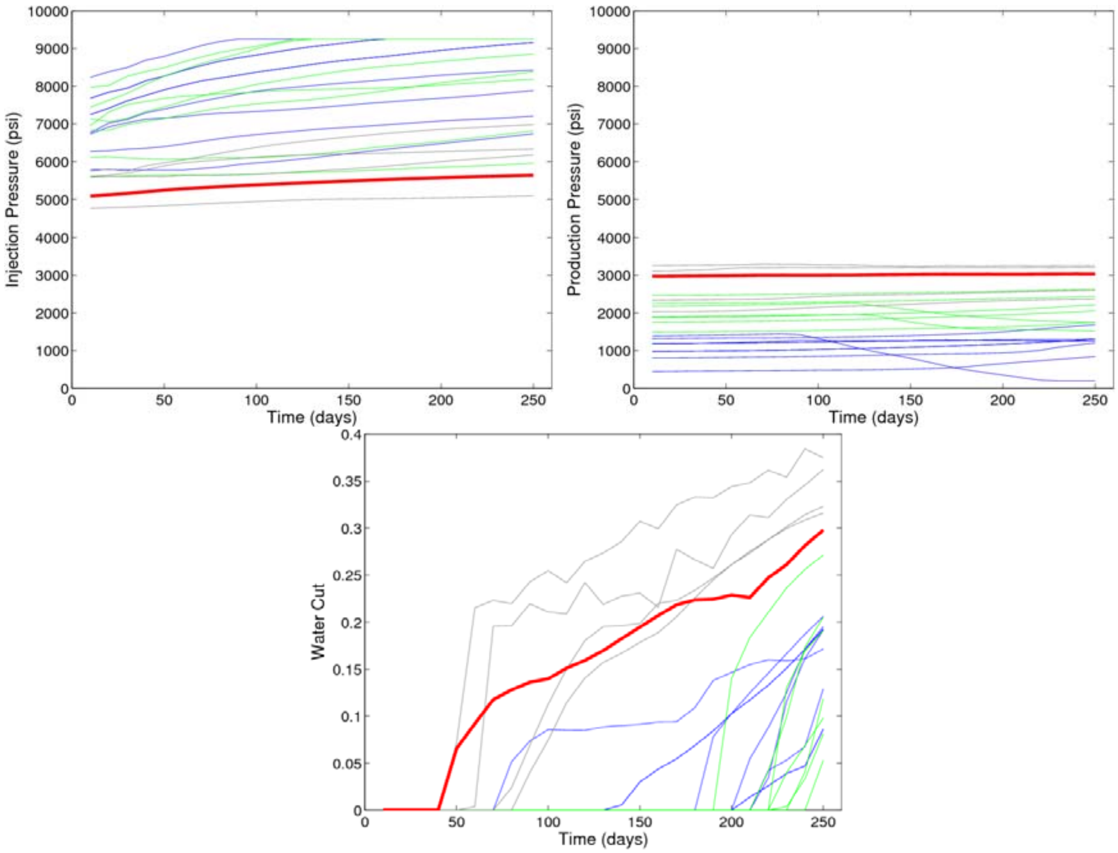

Figure 5-13 Production data from samples of cluster 1 (blue), cluster 2 (green) and cluster 3 (gray) are compared to the reference production data (red).

The posterior probabilities of the clusters after incorporating the reference production data are 0.13, 0.20 and 0.67 respectively. The cluster with the highest probability (gray cluster) is selected for a second level of selection as illustrated in Figure 5-14. Production data starts converging towards the reference at the second level of selection as shown in Figure 5-15. At the third level, few reservoir models remain as illustrated in Figure 5-16. There are two natural stopping criteria: either (1) the minimum number of reservoir models remaining is reached, or (2) the posterior probability distribution does not show preference for any cluster. In this case the process is stopped when the minimum number of reservoir models in the final cluster is fifteen.

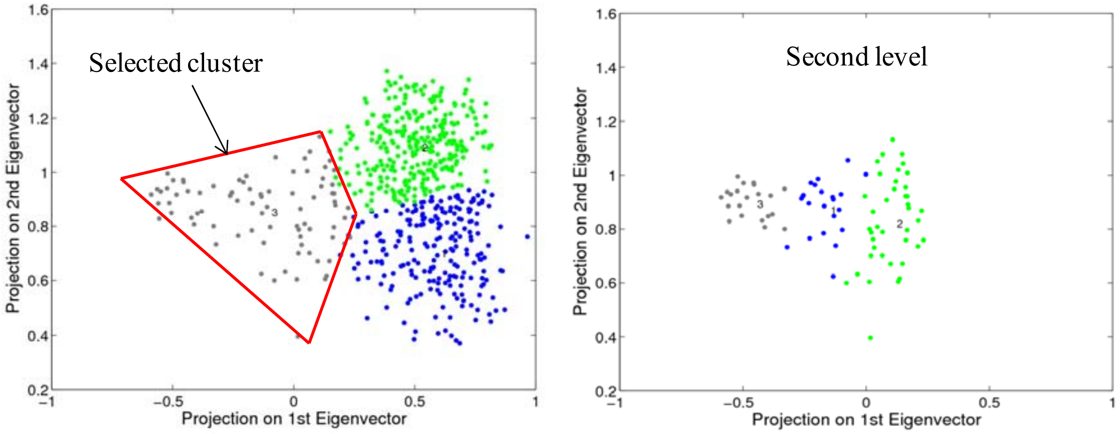

Figure 5-14 First and second level of selection of reservoir models

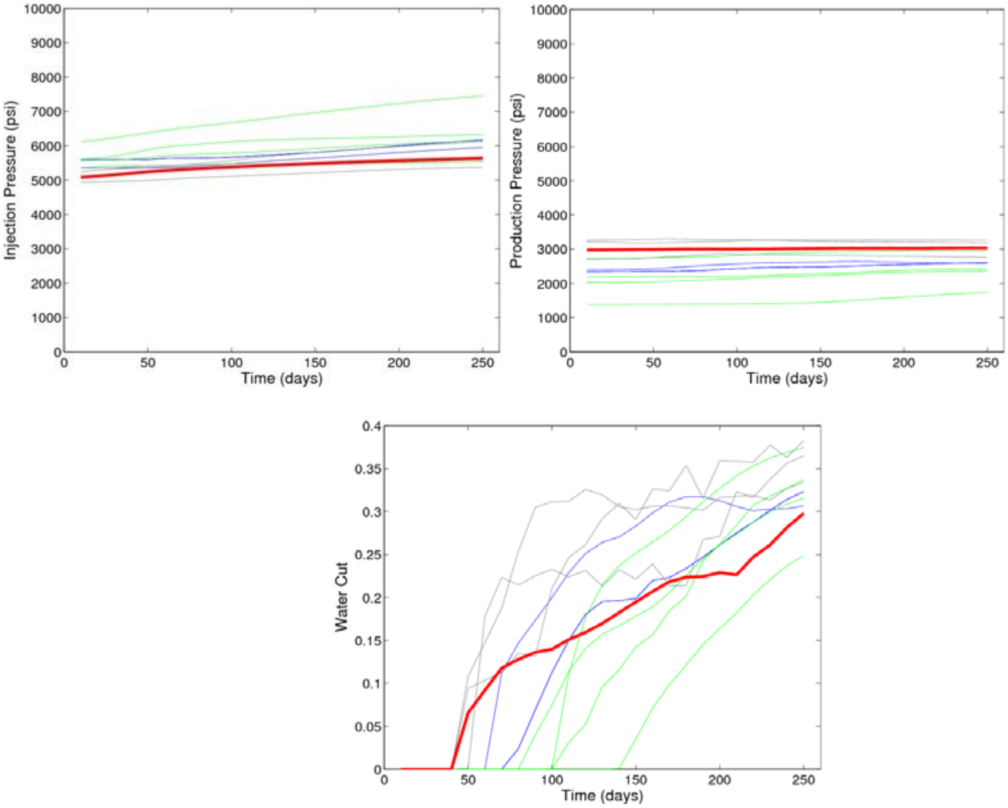

Figure 5-15 Variation in injection and producer pressure over the models after the second level of selection.

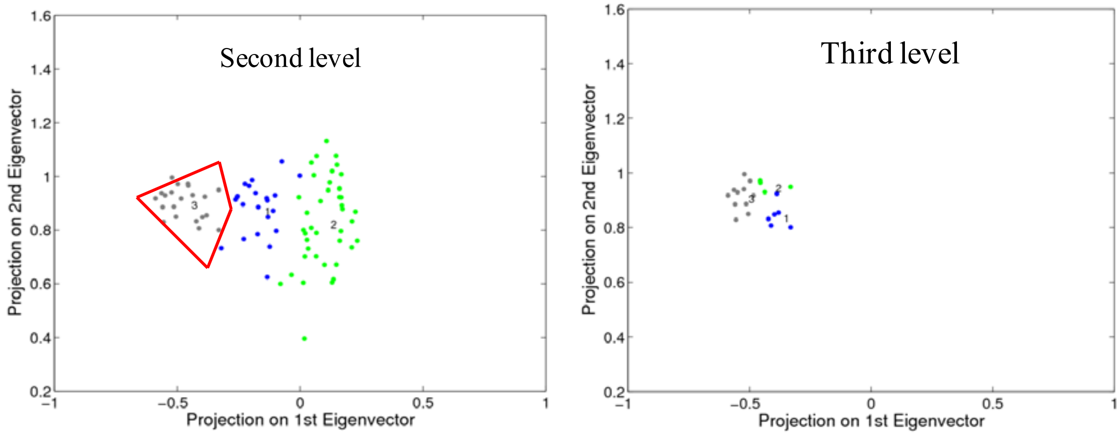

Figure 5-16 Cluster of reservoir models retained at the second and third levels of selection

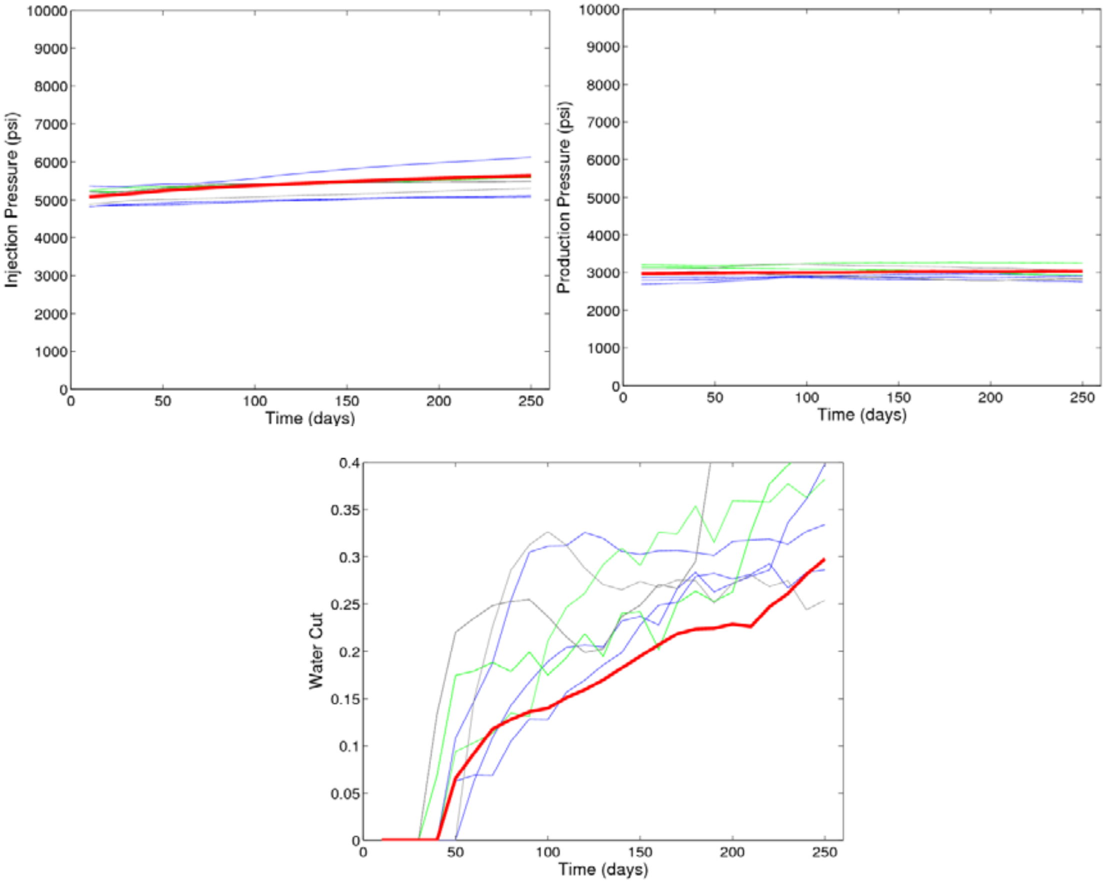

Figure 5-17 Production data of reservoir models retained at the third level of model selection.

Finally, each member of the final set was subject to flow simulation with UTCHEM. The injection and producer pressure from the final set is close to the reference as seen in Figure 5-18. Comparison of the initial sets of production data (Figure 5-1, Figure 5-2, and Figure 5-3) to the final set (Figure 5-18) demonstrates that the selection method was able to narrow the uncertainty in production data significantly using a minimum number of simulation runs. In this case 26 flow simulations with UTCHEM were carried out to calculate the posterior probability and for calibrating the random walk procedure.

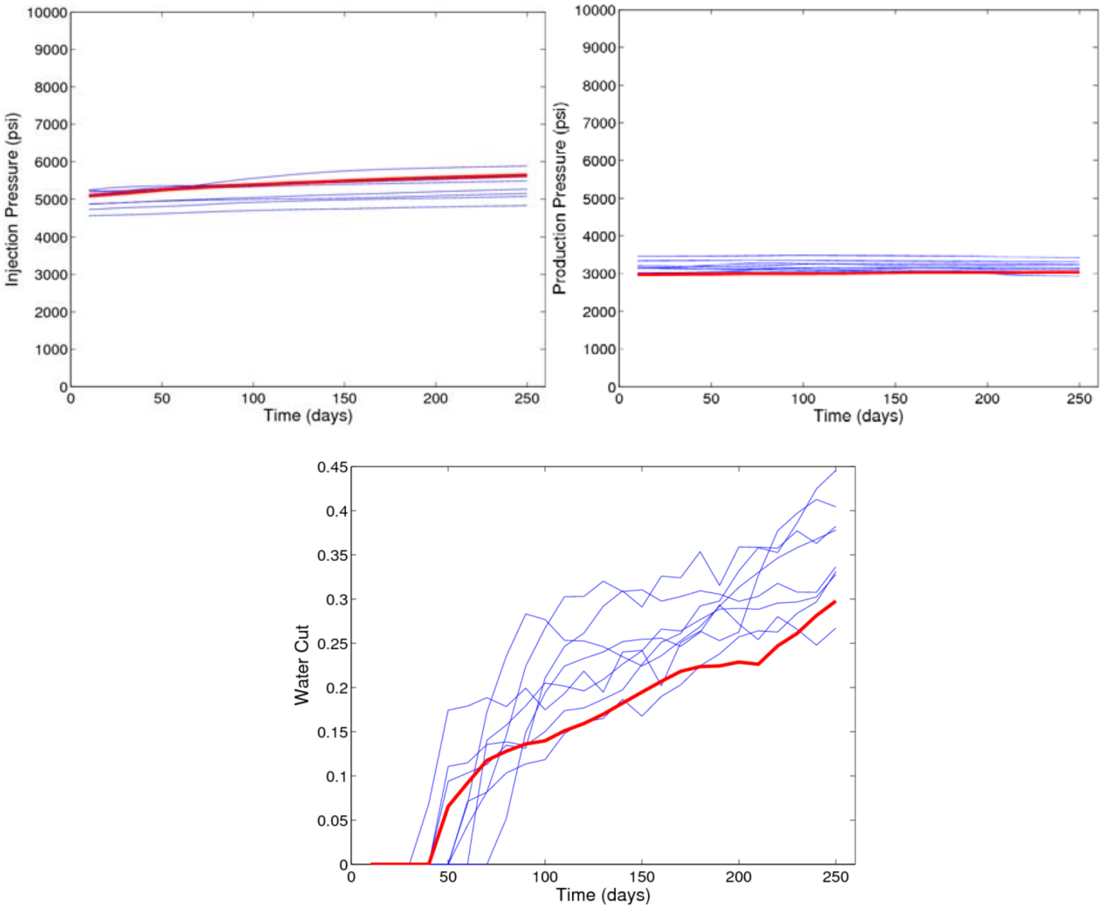

Figure 5-18 Production data computed on the final suite of reservoir models. The reference response is shown in red.

The permeability maps of the final set have in common a high permeability streak connecting the injector to the producer as shown in Figure 5-19. This is also a prominent characteristic of the reference model (Figure 5-7). In addition, the realizations in the final set share the same variogram model with a long range of continuity in y-direction. Therefore, in this case reservoir models conditioned to that variogram model are certainly preferred.

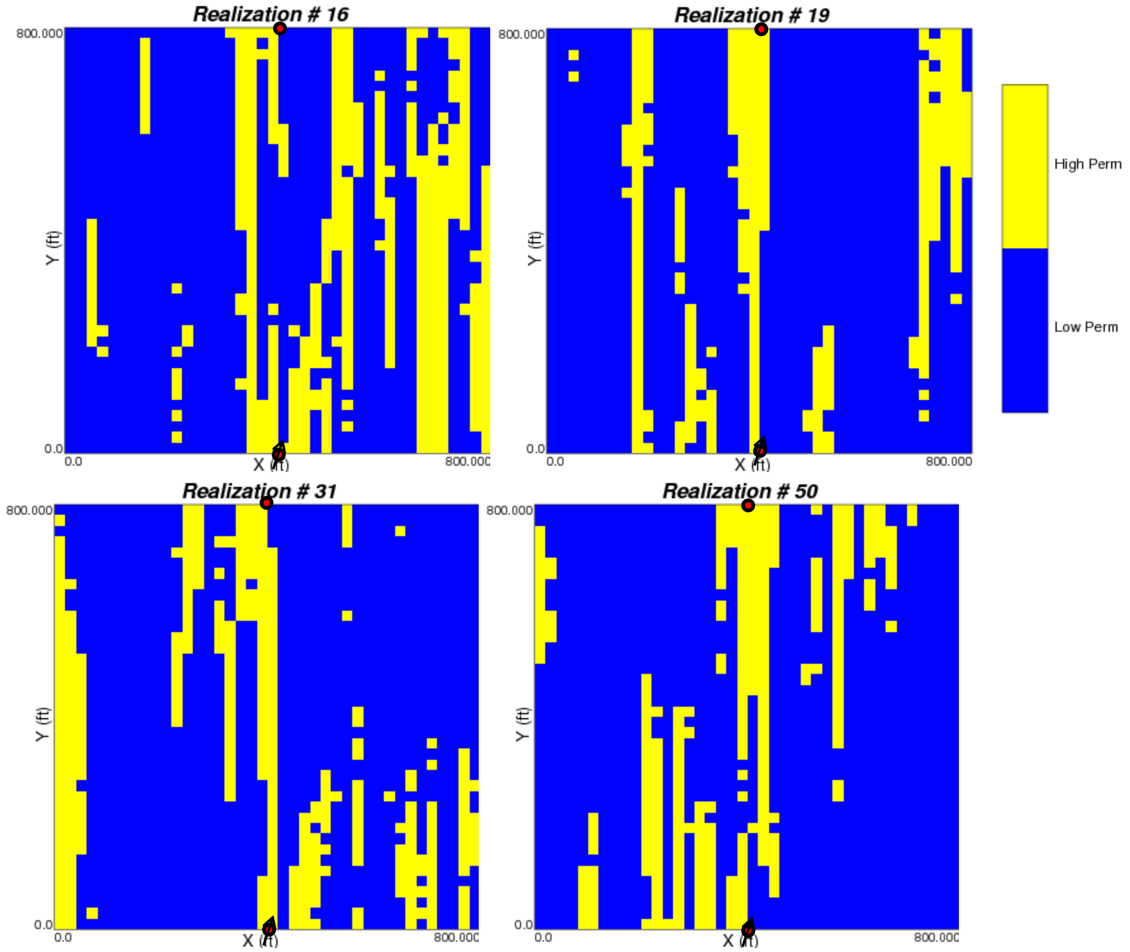

Figure 5-19 Facies map of four reservoir models from the final set.

5.6 RESERVOIR MODEL SELECTION IN A 5-SPOT INVERTED INJECTION PATTERN

Pilot tests of polymer flooding are commonly implemented either in inverted or regular 5-spot injection patterns. In a multi-well case, interactions of pressure gradients between producers dominate the expansion of polymer front around the injectors. Therefore, it is important to identify the preferential paths that polymer follows in order render the areal coverage uniform by rate control.

This has been studied through a synthetic example of a reservoir with sinusoidal streaks of high permeability that resemble fluvial channels as pictured in Figure 5-20. Polymer is injected at a rate of 400 ft^3/day and water and oil are produced at a total fluid rate of 100 ft^3/day per producer.

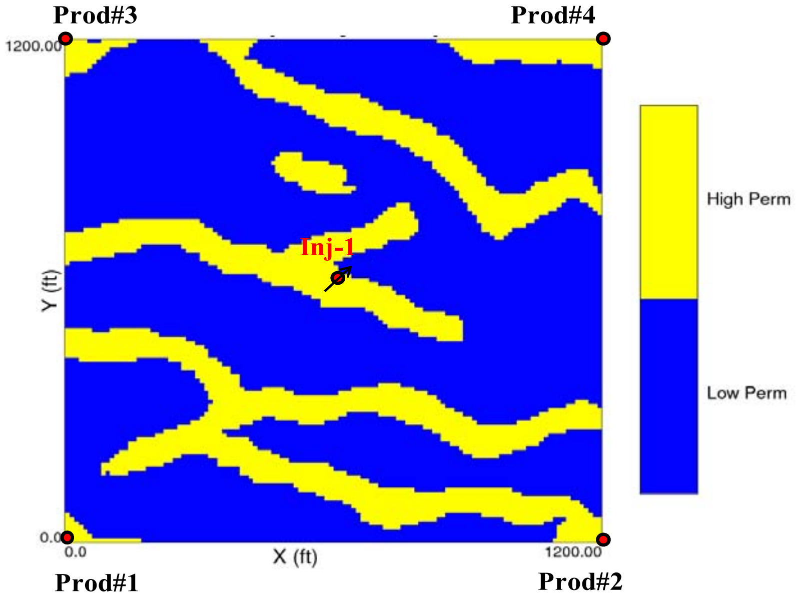

Figure 5-20 Facies map of reference reservoir with four producers at the corners and one injector at the center

The well pressure for the initial 250 days was recorded every 10 days and plotted in Figure 5-21. Water was not produced during this interval, therefore the production data available for model selection is just well pressure profiles. A careful examination of

Figure 5-20 and Figure 5-21 shows that producer pressure is correlated to the size of the high permeability area that the well is connected to. For instance, wells Prod#1 and Prod#3 show low production pressure because the high permeability zone around them is small. On the other hand, Prod#2 produce at a higher pressure because it is connected to a long channel.

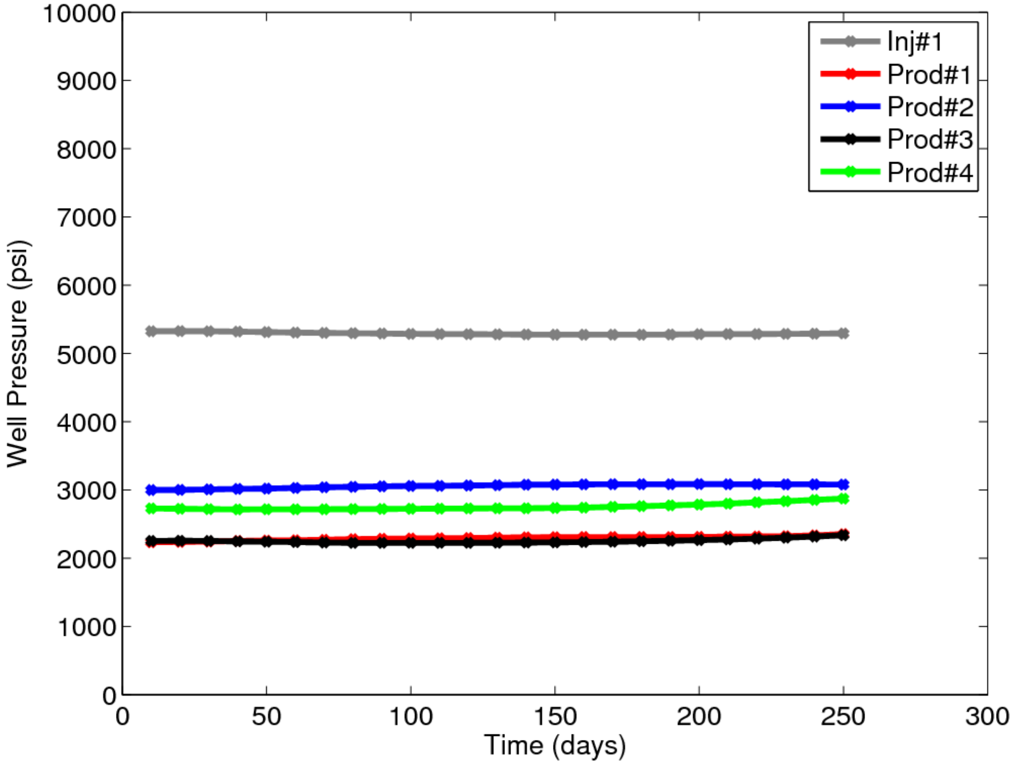

Figure 5-21 Well pressure at 4 producers and 1 injector in the reference reservoir (Figure 5-20) for the first 250 days

As production data from more wells are integrated within the model selection procedure, the selection of reservoir models becomes difficult because it is unlikely to find a group of reservoir models with production data similar to the observations at all wells simultaneously. The strategy for model selection takes four steps as illustrated in Figure 5-22: (1) the initial ensemble of reservoir models is used to make the model selection by well pairs independently, (2) the selected reservoir models for each well pair are used to infer the local probability of facies indicators around the well locations, (3) a new ensemble of reservoir models that reflect the local probabilities are generated, and (4) the selection process is repeated using all sets of pressure profiles for a final refinement. Step (3) is necessary for ensuring that there are sufficient models in the final cluster.

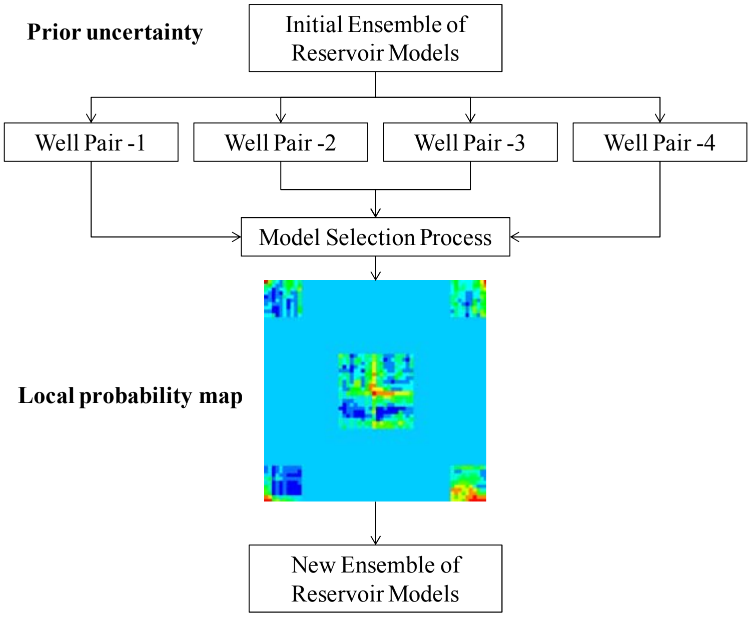

Figure 5-22 Illustration of the process for updating the model selection using data from multiple wells.

5.5.1 Model Selection for a Well Pair

In this step, only the outputs from the proxy models regarding the well pair that is being analyzed are considered. Even though particles are propagated throughout the entire reservoir, only the distribution of the path lengths that connect the well pair of interest is analyzed. Also, all wells are included in the flow simulation but we only look at the pressure profile of the well pair under consideration. This section illustrates the selection procedure for the first well pair. That procedure is repeated for the other three well pairs.

The initial ensemble of reservoir models has 6 types of reservoir models using 5 different variogram models and 1 sinusoidal training image as shown in Figure 5-23. The initial ensemble consists of 300 realizations generated constrained to the variogram models or training image information. All realizations are processed with the proxy model. The proxy responses analyzed are the average permeability around the analyzed wells, distribution of arrival times of the well pair analyzed and breakthrough time on the producer analyzed.

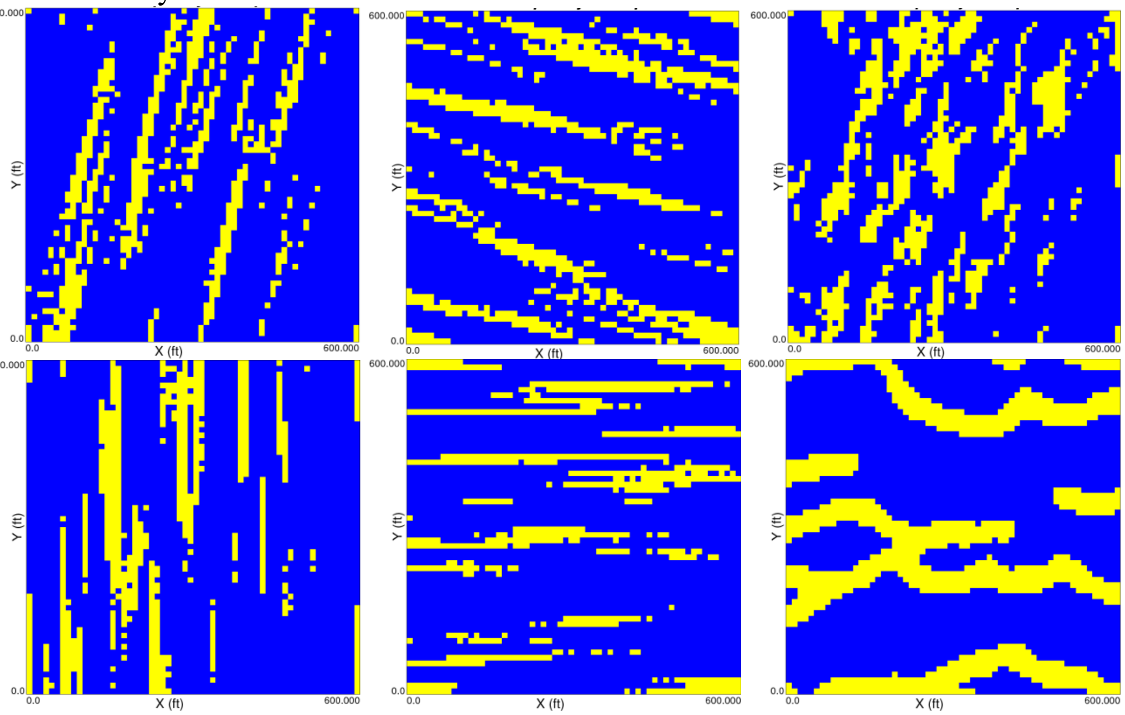

Figure 5-23 Examples of reservoir models included in the initial ensemble.

The proxy responses are projected on the space defined by the top 3 eigen-vectors as plotted in Figure 5-24. In the first clustering step, the initial cloud of reservoir models is initially divided into two clusters (left of Figure 5-24), from which the blue cluster was selected as its posterior probability is the highest. Subsequently, six more levels of selection took place until obtaining the final set of 9 reservoir models contained in the blue cluster (right of Figure 5-25).

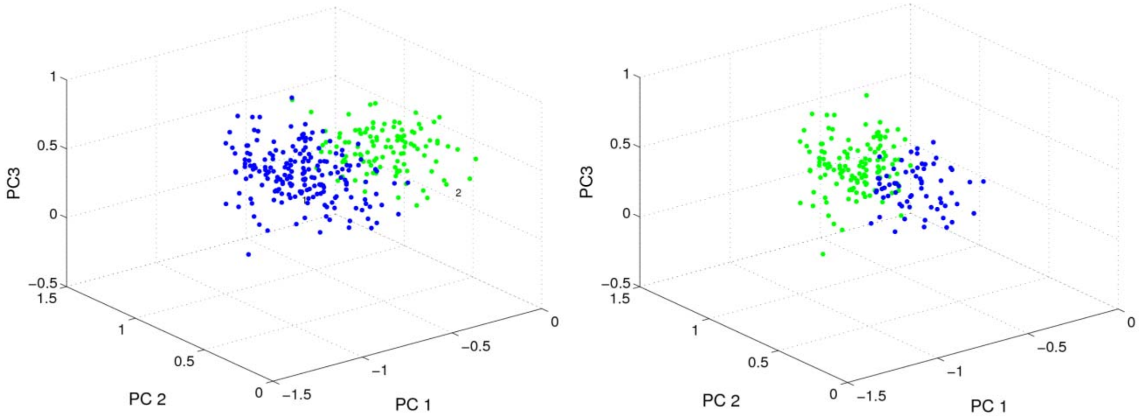

Figure 5-24. First two levels of selection in example of inverted 5-spot injection pattern

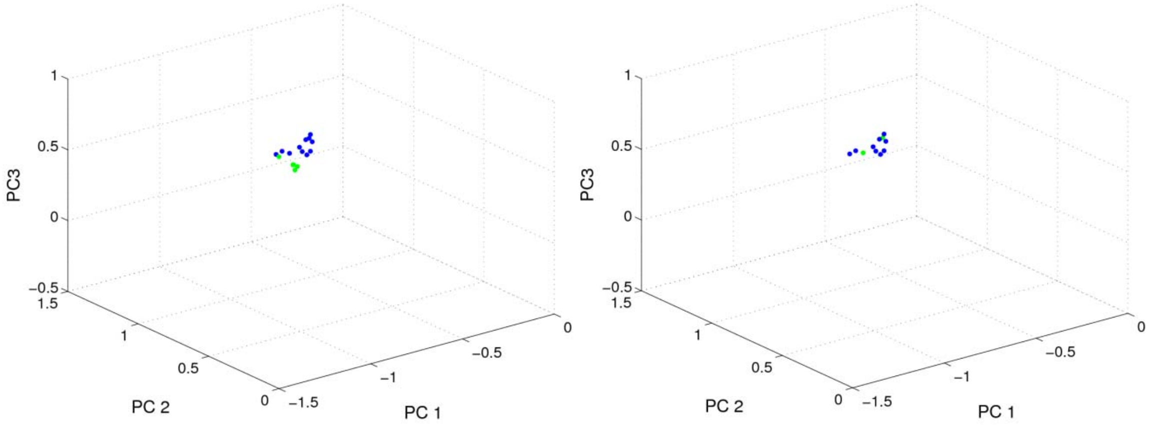

Figure 5-25 Final two levels of selection in example of inverted 5-spot injection pattern

The injection pressure of several reservoir models in the final set is close to the reference, but there are some outliers as observed in Figure 5-26. The distribution of pressure at Producer #1 is wider, but centered on the reference.

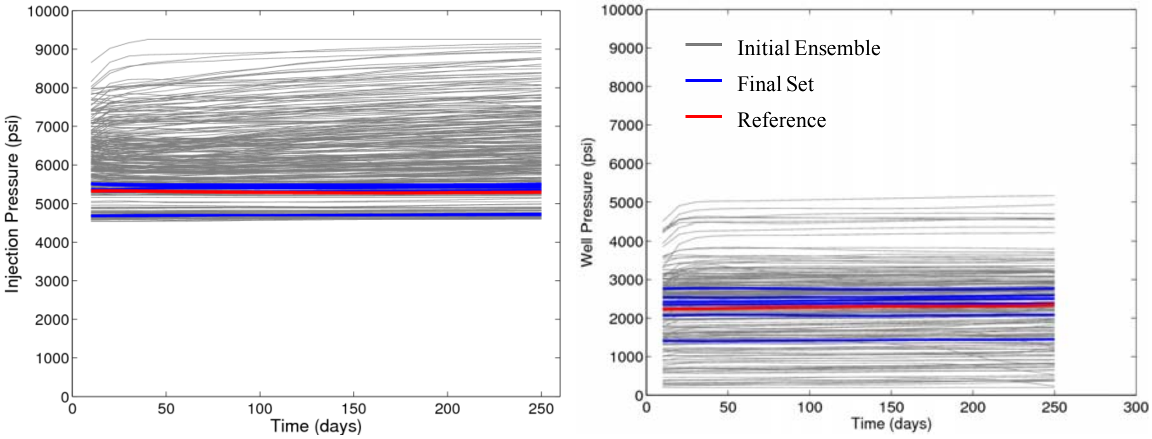

Figure 5-26 Well pressure at Inj#1 and Prod#1 (First well pair) for initial ensemble compared to the final suite.

The distribution of the production data for the prior ensemble (gray pressure profiles in Figure 5-26) and the final selection (blue pressure profiles in Figure 5-26) were analyzed. At 250 days the mean of the injection pressure in the prior ensemble is 6266 psi, the mean in the selected set is 5278 psi and the reference injection pressure is 5295 psi. In addition, the standard deviation of the injection pressure at 250 days for the prior ensemble is 1187 psi and the standard deviation in the selected set is 305 psi. Hence, the posterior set of reservoir models results in a refinement of uncertainty in production data for the well pair under study.

Figure 5-27 shows that although the reservoir models in the final set correspond to different variogram models/sinusoidal training image, there is a common small high-permeability area connected to the producer (bottom-left corner). The other common feature is that the high permeability area connected to the injector (center) extends over a wide region of the reservoir. At this early time after the start of injection, the main direction of growth of the polymer bank is not discernible. Therefore, at an early stage of injection it is difficult to identify the main pathways between wells.

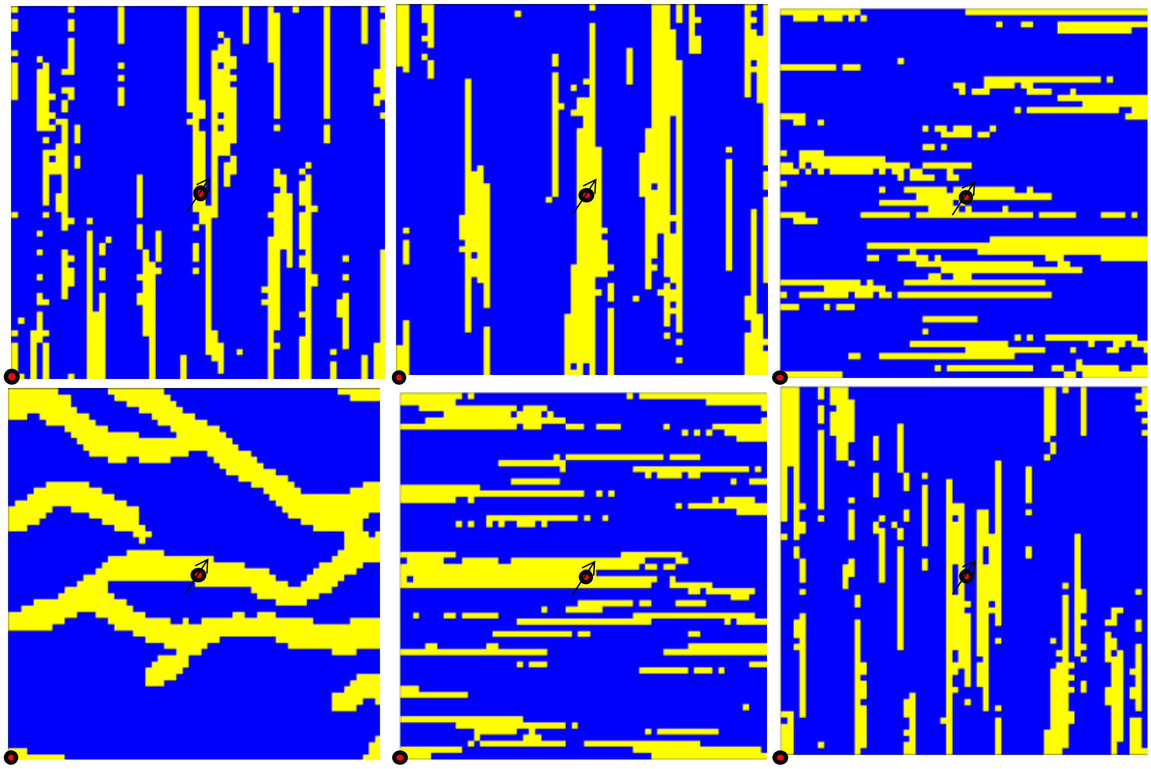

Figure 5-27 Maps of 6 reservoir models in the final set obtained by using production data from Inj#1 and Prod#1 (labeled in the maps) for model selection.

5.5.2 New Reservoir Models Conditioned to Prior Selection

After repeating the model selection process using the data for the other 3 well pairs, the realizations in the final cluster are used to infer both the local probability around the wells and the probability that each geologic description will jointly satisfy the requirements by injector-producer well pattern. A new ensemble of 100 reservoir models is generated using the local probability as soft data in *sisim* and *snesim*, so that the resulting realization share the features exhibited by the models in Figure 5-28.

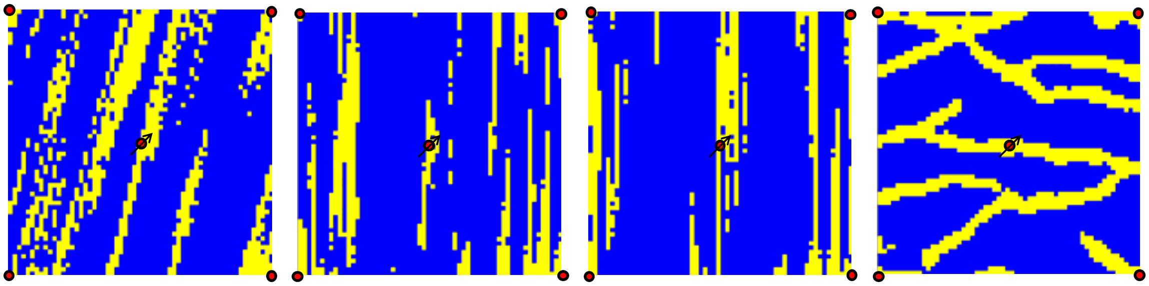

Figure 5-28 Four reservoir models of the new ensemble generated with the combined local probabilities for each well pairs. The new ensemble is composed of 100 reservoir models. This is the starting data set for the refinement of the selection.

The entire new ensemble was processed with UTCHEM to verify if the well pressures of the new models are consistent with the response of the original realizations used to originate them. The distributions shown in Figure 5-29 are fairly well centered on the reference, but there is still uncertainty that is refined in the next level of selection. Particularly, Prod#1 and Prod#3 show the largest variability in production data

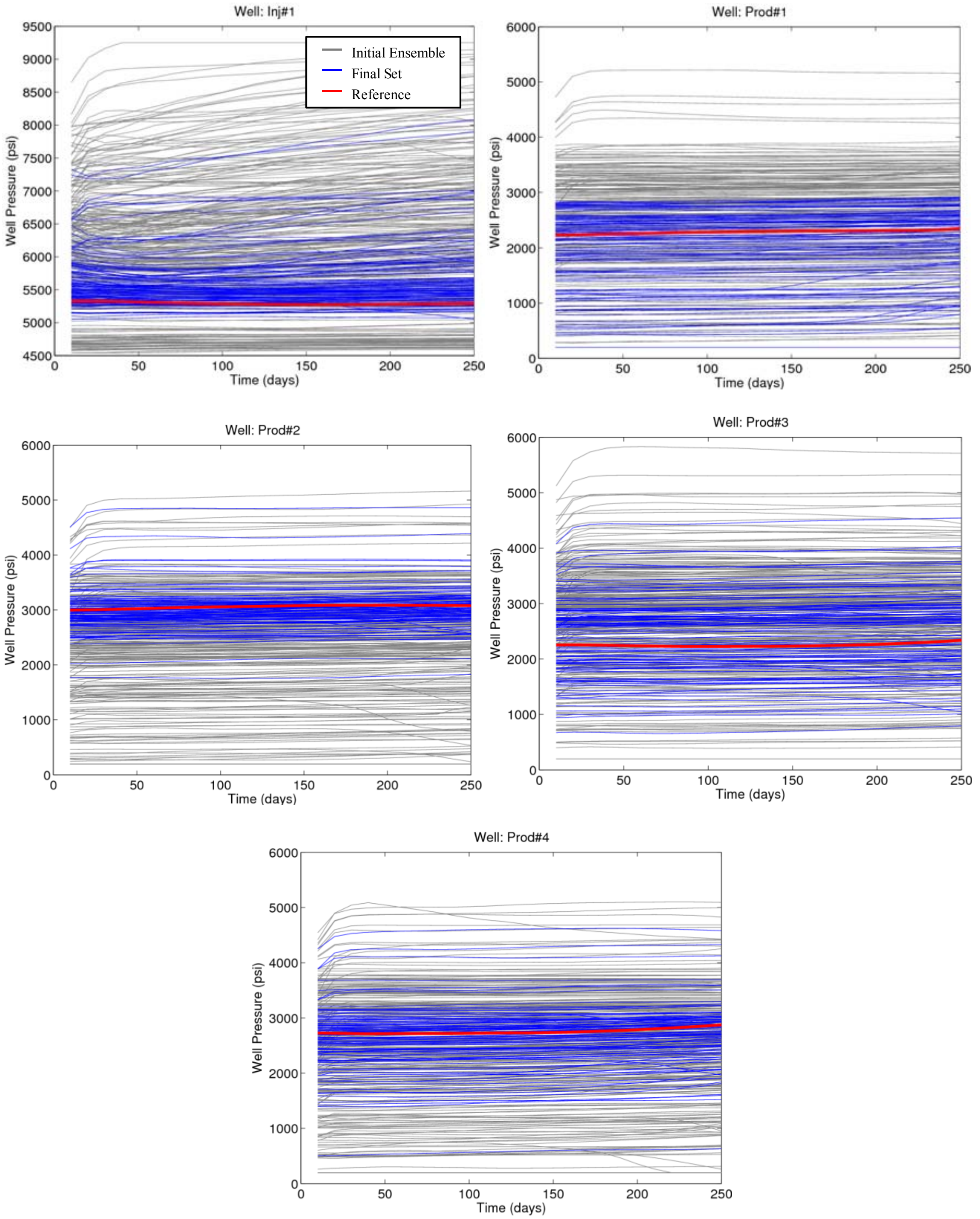

Figure 5-29 Well pressures for the reference, and reservoir models of the initial and new ensembles.

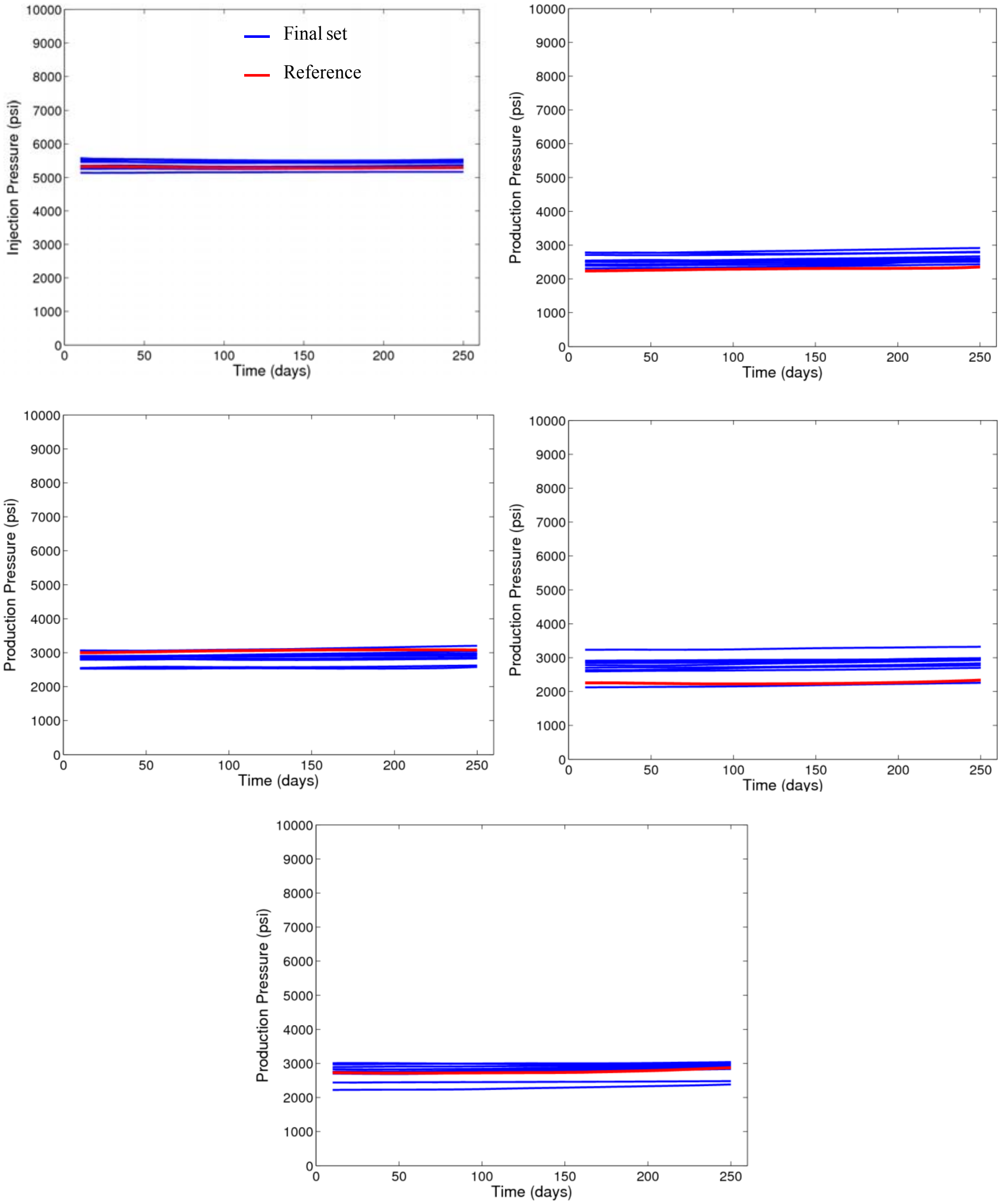

Figure 5-30 Well pressures for the reference and final selection of reservoir models after the refinement step.

The model selection process is then repeated with the new ensemble, but now using the proxy responses and production data of all wells jointly. The vector of responses has a dimension of 17 (= average permeability per well (5) + mean and variance of path lengths per well pair (8) + breakthrough time per well pair (4)). The dimension of the metric space is now 5, meaning that 5 eigen-vectors are necessary to account for the variability of the proxy responses. At the end of the selection process, the distribution of well pressure is narrower and centered to the reference as seen in Figure 5-30.

Despite the reduction in variability of production data in the final cluster, the uncertainty in the geology remains wide as different geologic descriptions are present in the final set as shown in Figure 5-31. Hence, it can be inferred that early-time pressure does not bring sufficient information to depict large scale heterogeneity features related to the inter-well communication. This supports the argument that the uncertainty in the geologic scenarios should be carried forward even if there is a good match in production history. Therefore, limiting the characterization of the reservoir to a single geologic scenario can lead to incomplete uncertainty assessment.

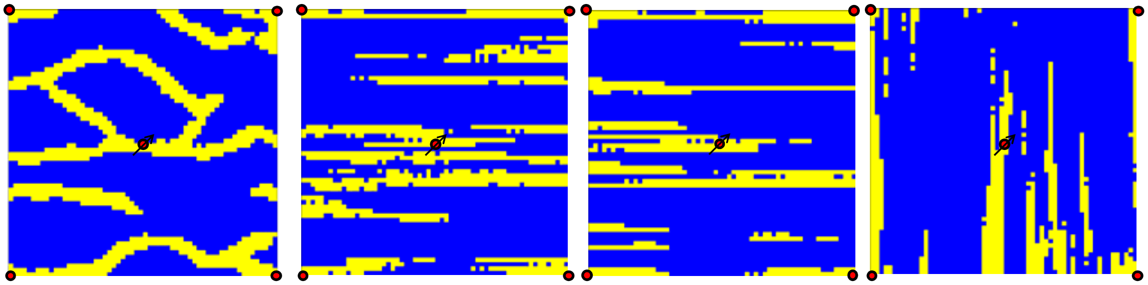

Figure 5-31 Selected realizations from the final set after model selection using the data from the entire well pattern.

5.6 CONCLUSION

In the examples shown in this chapter it was difficult to identify a single geologic depiction of a reservoir based only on integration of static data and production data. Even if a reservoir model conforms to the available static data and production history, there is no guarantee that the underlying geologic scenario has been certainly identified. In the 5-spot injection pattern example, the well pressure was correlated only to the size of the high permeability area connected to the well, regardless of the features that dominate the heterogeneity at larger scale. Hence, it is recognized that early production data is not sufficient to identify predominant well connectivity characteristics, which inflicts a limitation to the selection of reservoir models. Nevertheless, as shown in Chapter 7, fractional flow data such as water cut is more informative about well connectivity turning the feedback-control process more useful.

Chapter 6: Polymer Flooding Optimization using a Proxy Model and Response Surface

6.1 OVERVIEW

The research approach adopted to optimize well control for maximum economic NPV from a polymer flooding process is detailed in this chapter. The application of response surfaces and gradient-based optimization is initially illustrated through a simple example. Finally, the method is demonstrated for optimizing the injection and production rates for the two-well and the five-well pattern injection cases presented in previous chapters. Conclusions about the feasibility of implementing optimal control strategies for polymer injection are discussed at the end.

6.2 PRODUCTION OPTIMIZATION

Once a set of reservoir models have been identified that produce similar responses as observed in history, the next step is to find the optimal production strategy for maximizing the economics of the project. The production optimization problem is formalized as follows: find the optimal control vector **u**, composed by injection and production rates for *NT* time control periods, such that the net present value at final time (*J*) is maximized (Equation 6.1), subject to the following constraints: (1) flow equations for polymer flooding (Equation 6.2), (2) operational constraints (Equation 6.3) and (3) material balance between cumulative injection and cumulative production (Equation 6.4).

$$\max_{\mathbf{u}} J = \int_{0}^{t_f} f\left(\mathbf{x},\mathbf{u}\right)dt \qquad (6.1)$$

Subject to:

$$\mathbf{x}^{t} = g\left(\mathbf{x}^{t-1},\mathbf{u}^{t}\right) \qquad (6.2)$$

$$\mathbf{LB} \leq \mathbf{u} \leq \mathbf{UB} \qquad (6.3)$$

$$\sum_{inj} u_{inj}(t) = \sum_{prod} u_{prod}(t) \tag{6.4}$$

The number of control periods refers to the frequency of adjustments of well constraints during the operation of the wells. The number of control variables in this particular problem can be quite large: for example, the control vector **u** would have 40 elements in a 5-spot injection pattern with 10 control periods where only the rate is controlled, because Equation 6.4 is used to solve for the rate of the fifth well. The number of control variables (*n*) for this problem is expressed in Equation 6.5.

$$n = (NW - 1)NT \tag{6.5}$$

where, *NW* is the number of wells and *NT* is the number of control periods. A procedure that employs an adaptive response surface was developed to solve the optimization problem efficiently.

6.3 ADAPTIVE RESPONSE SURFACE FOR PRODUCTION OPTIMIZATION

Experimental design and response surfaces have been commonly used in optimization and uncertainty assessment when the relationship between control variables and objective function is difficult to express mathematically, particularly when the transfer function is numerical. A response surface is an empirical model that relates the model parameters (such as geologic variables and/or well controls) to corresponding responses. The generated empirical model is often represented as a linear regression model as shown in Equation 6.6, where *J* is the response, β_i's is a set of regression coefficients, u_i's are the independent variables and ε is an error term. (Myers and Montgomery, 2002)

$$J = \beta_0 + \beta_1 u_1 + \ldots + \beta_n u_n + \varepsilon \tag{6.6}$$

The *n* independent variables (u_i) are organized into an n x 1 vector (**u**) that contains the control variables to be optimized. Multiple evaluations of *J* are performed with independent realizations of **u** in order to obtain sufficient data to find the regression coefficients β_i that best fit the responses. At least *n+1* function evaluations are required to solve the linear regression problem of Equation 6.6. Fitting the regression coefficients is a simple linear regression whose solution is expressed in Equation 6.7.

$$\boldsymbol{\beta} = \left(\mathbf{U}^T\mathbf{U}\right)^{-1}\mathbf{U}^T\mathbf{J} \tag{6.7}$$

Where **β** is a vector that contains all regression coefficients, **U** is a matrix with all the **u** vectors evaluated and **J** is a vector that contains all the responses.

In order to capture better the non-linearity of the relationship between control variables and responses, Equation 6.6 is expanded to include second order and interaction terms as in Equation 6.7.

$$J = \beta_0 + \beta_1 u_1 + \ldots + \beta_n u_n + \sum_{i<j}\sum_{j=2}^{n}\beta_{ij}u_iu_j + \sum_{i=1}^{n}\beta_{ii}u_i^2 + \varepsilon \tag{6.8}$$

Denoting *nruns* as the minimum number of function evaluations required to solve for the regression coefficients in Equation 6.7, *nruns* is equal to $(n^2+3n)/2+1$. Although adding more terms to the equation of the response surface yields a better fit of the true responses, it is not necessarily convenient because overfitting problems may arise. To avoid that, a test of significance (Jablonowski and Strachan, 2008) can be performed to identify the statistically significant regression parameters, *i.e.* regression coefficients statistically correlated to the response. Fewer terms in Equation 6.7 remain after the test of significance and the effect of the variables eliminated is absorbed in the error term. The *nruns* number of is reduced to about 40% in this case. For example a significance test of the regression parameters for the example shown in Table 6-1 using four rate controls indicates that 6 out of the 15 regression coefficients at significant at 0.1 level.

Table 6-1 Results of a significance test on regression coefficients for a response surface with 4 rate controls. Rows in gray are significant at 0.1 level.

	β_i	*t Stat*	*P-value*
Intercept	98732.18156	26.87433	0.023678
u(1)	324.1191909	13.25313	0.047945
u(2)	95.23858161	2.317147	0.259371
u(3)	-12.36779119	-0.6824	0.61878
u(4)	114.1411162	3.504443	0.176958
$u(1)^2$	-0.220022502	-3.00847	0.204295
$u(2)^2$	-0.555307378	-12.9249	0.049157
$u(3)^2$	-0.551752287	-6.46356	0.097719
$u(4)^2$	-0.502352461	-5.88385	0.107174
u(1)*u(2)	-0.12175673	-1.0286	0.491027
u(1)*u(3)	-0.252022526	-4.32498	0.144654
u(1)*u(4)	0.2128789	2.336046	0.257494
u(2)*u(3)	0.325023825	3.248814	0.190096
u(2)*u(4)	-0.339633572	-5.31376	0.118421
u(3)*u(4)	-0.477039874	-2.56397	0.236742

Experimental Design is a technique often employed to choose combinations of independent variables **u** that will yield a response surface that is most representative of the phenomenon being modeled. In Experimental Design, two-level factorial designs are commonly used to investigate the effect of a single parameter and its interactions with other parameters. Two-level factorial design considers only two levels of the variables **u**: - a minimum and a maximum. For example, if we want to investigate the effect of the injection rate during the first 100 days, the function is evaluated corresponding to the maximum and minimum injection rates allowable. The number of response function evaluations required is 2^k corresponding to *k* variables, and that might be too large within a dynamic optimization framework. Even if the proxy model is used for a rapid function evaluation, the time taken to process 2^k runs would render it unsuitable within a feedback control framework.

An adaptive response surface procedure is formulated that yields a representative response surface using a limited number of function evaluations. The construction of the response surface starts by using the proxy model to evaluate the net present value for at least *nruns* control vectors **u**. The vectors **u** for which the function is evaluated are samples drawn from a symmetric distribution like multi-Gaussian or uniform in order to avoid bias. If we have a prior knowledge about the optimal region, that bias can be introduced by using multi-Gaussian pdf. In the absence of prior preference, we have chosen the distribution centered between the feasible bounds and truncated at the lower and upper bounds of the control variables as shown in Equation 6.9.

$$\mathbf{u} \sim N_n\left(\boldsymbol{\mu}, \boldsymbol{\Sigma}\right) \tag{6.9a}$$

$$\boldsymbol{\mu} = \left[\frac{u_1^{\max} + u_1^{\min}}{2}, \ldots, \frac{u_n^{\max} + u_n^{\min}}{2} \right]^T \tag{6.9b}$$

$$\boldsymbol{\Sigma} = diag\left(\frac{u_1^{\max} - u_1^{\min}}{2}, \ldots, \frac{u_n^{\max} - u_n^{\min}}{2} \right) \tag{6.9c}$$

The covariance matrix is diagonal because the control variables are assumed independent. For each vector **u**, the proxy is run 5 times in order to get the expected value of the net present value for that combination of injection and production rates. The minimum number of runs required depends on the variability of the response of the proxy model. As stated in Chapter 3, the outcome of the proxy model is stochastic and 5 samples are sufficient to have an estimate of the mean with 95% confidence.

Then, the regression coefficients are fitted and the approximated response surface is obtained. The equation of the surface is subject to a gradient-based optimization function that takes the negative of the net present value as objective function along with its gradient function, the linear constraints specified by the operational bounds and the condition that the cumulative injection and production be balanced at all times. This gradient based optimization is implemented in the MATLAB® function *fmincon* that returns the control vector **u*** that minimizes the objective function.

Then, the NPV is evaluated with the returned control values to check for optimality. The new estimation of NPV is incorporated to fit a new response surface and the process is repeated until convergence of the optimal control vector. At the end, the final response surface fits well the region where the optimal NPV is because it contains several regression points in that region.

This procedure is illustrated with the two-well reservoir model presented in Chapter 3. For simplicity, this example uses only 2 control periods of 1000 days each one, for a total of 2000 days with wells under rate control. Therefore, the control vector **u** (Equation 6.10) has 2 independent variables (n=2).

$$\mathbf{u} = \begin{bmatrix} q_{inj}^{1} & q_{inj}^{2} \end{bmatrix}^{T} = \begin{bmatrix} u_1 & u_2 \end{bmatrix}^{T} \tag{6.10}$$

u is initially sampled 6 times in order to obtain sufficient data to fit a response surface equation as shown in Figure 6-1. The surface regression equation is submitted to *fmincon* in order to obtain the optimal **u**. Then, the proxy evaluates the function at the "optimal" **u**, but the objective function turned out to be sub-optimal when the response is reevaluated at this point. This is due to the lack of regression points in the optimal region. Then, the new point is incorporated into the data set for fitting the response surface at the next iteration and new regression coefficients (**β**) are computed. The process is repeated until the change in optimal **u** over successive iterations is less than a tolerance. In this way the surface is progressively better fitted in the area of optimal control settings.

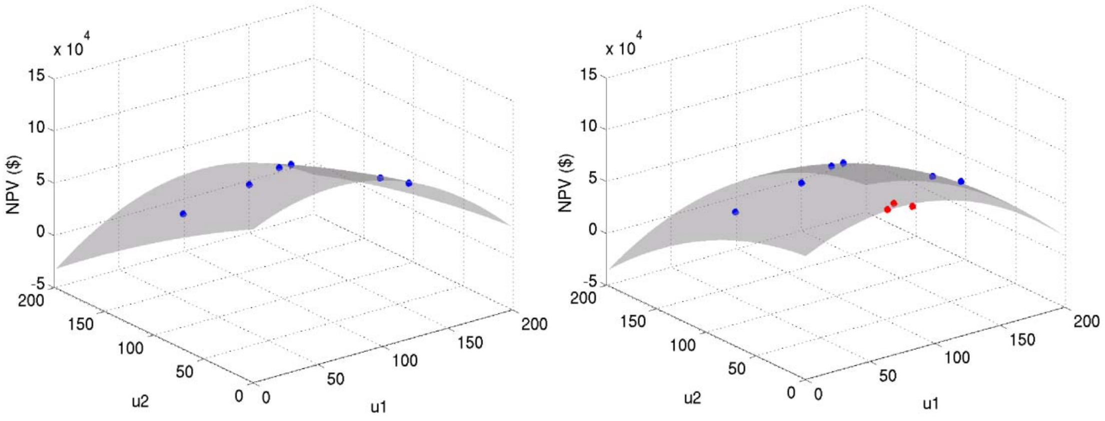

Figure 6-1 Response surface fitted for optimization using initial points (left) and adapted after further function evaluations around the optimal (right). The blue dots are the initial function evaluations and the red dots are the new function evaluations after each iteration.

The above procedure has been tested extensively, showing convergence to the same optimal region. If the concavity of the response surface is outside of the bounds of feasible control variables, the optimal solution is close to the bounds as indicated by the red dots in Figure 6-1, where the optimal solution requires u_2 to be a minimum, which means that the well is shut-in.

Note that the optimal solution found through this procedure is stochastic, first because the proxy model is stochastic and second because the surface fitting depends on the initial values used for regression, that are randomly sampled. Even if UTCHEM were used for the evaluation of the objective function, the optimal control rates are stochastic due to the random sampling of control variables. Therefore, the process is repeated a few times to obtain the expected value of the optimal control vector. This procedure is extensible to control vectors with more control periods and more wells.

6.4 PRODUCTION OPTIMIZATION OF THE TWO-WELL EXAMPLE

The injection and production rates for the reference reservoir shown in Figure 6-2 were optimized using the adaptive response surface method explained above. The total injection time is 2000 days divided into 10 control periods of 200 days. The objective function is the net present value using the economic parameters shown in Table 6-2. These economic parameters were chosen only for demonstration purposes and are not reflective of current economic conditions; the aim was to obtain a net present value curve that makes the trivial solution (maximum production rates) sub-optimal. For that reason, the assumed cost of water treatment penalizes excessive water production. The physical constraints are: the injection/production rates are bounded between 0 and 200 ft^3/day, and injection rate is equal to production rate at all times.

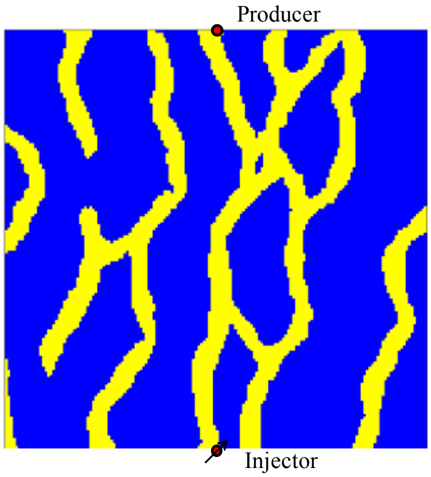

Figure 6-2 Reference reservoir model with 1 injector and 4 producers reproduced from Figure 5-7

The balance between the injection and production rates implies that only the injection rate needs to be optimized. The high permeability channel that causes communication between the wells causes early water breakthrough. Maintaining the production at maximum is an inefficient strategy because the high water production reduces the revenue severely and increases the production costs. A constant rate control is also inefficient because the early production is clean oil while water production needs to be controlled at late time. In summary, there is no trivial solution to maximize the net present value.

Table 6-2 Input Parameters for Economic Analysis

Economic Parameter	**Value**
Oil Price	$ 35
Discount Rate	10 % /year
Production Facility Cost	$ 10,000 /month
Water Injection Cost	$1/bbl
Water Production Cost	$10/bbl
Oil Production Cost	$4/bbl
Polymer Injection Cost	$1/lb
Royalty Tax	38.5%
Ad Valorem Tax	$0.046/bbl

Figure 6-3 shows that the optimal control sequence starts at high injection rates and it decreases as water production evolves. This strategy is consistent with the discount rate scheme in which early production has higher net present value than late production. The optimal control rate sequence indicates that for many control periods the well is operated as on or off, *i.e.* the optimal rate falls on either the upper or lower bound. This solution is common in convex optimization problems with upper and lower bounding constraints.

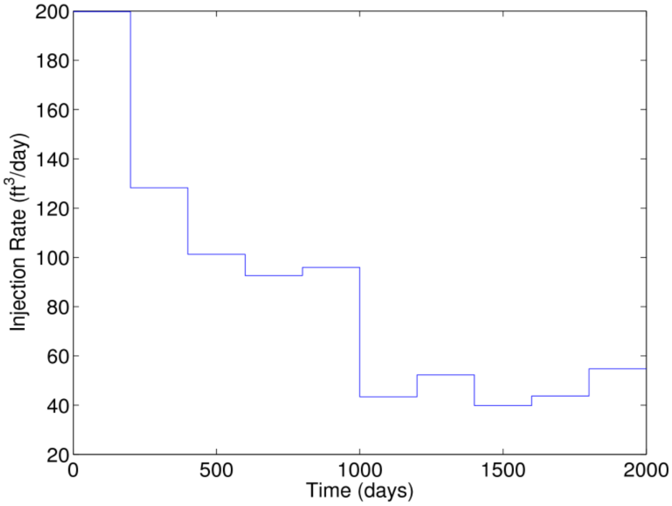

Figure 6-3 Expected value of the optimal rate sequence for the two-well case

Is the NPV of the optimal control vector better than the NPV of any control vector obtained from multiple runs? Since the objective function is a non-linear function of the control vector, the NPV corresponding to the expected optimal control vector (E{**u***}) is different than the expected value of the net present value as expressed in Equation 6.11.

$$E\left\{J\left(\mathbf{u}^*\right)\right\} \neq J\left(E\left\{\mathbf{u}^*\right\}\right) \tag{6.11}$$

To explore this issue, the response surface optimization procedure was repeated 10 times resulting in 10 different optimal control vectors. Polymer flooding was simulated using UTCHEM corresponding to these 10 control vectors and their corresponding net present value curves were computed. Figure 6-4 compares the net present value curves for each control vector found as optimal from individual runs to the net present value curve obtained for the expected control vector. The net present value curve of the expected control vector is superior to any individual run. Note that the net present value curve reaches a maximum at about 1200 days, and after that the injection and production rates are adjusted (post breakthrough) to maintain the production without economic loss.

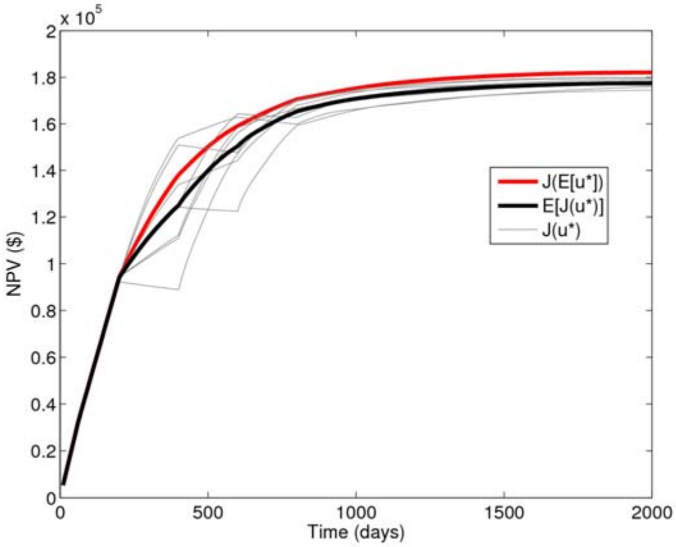

Figure 6-4 Net present value for optimal rate solutions from multiple runs and for the expected optimal rate sequence.

Figure 6-4 also indicates that $J(\mathrm{E}\{\mathbf{u}\})$ is greater or equal than $\mathrm{E}\{J(\mathbf{u})\}$. This result is consistent in many other cases evaluated. To explain this, consider that a minimum-rate operation yields a negative NPV, whereas a maximum-rate operation is not optimal because water breaks through early. Thus, $J(\mathbf{u})$ can be modeled as a quadratic function that is convex-up as in Equation 6.12, where the maximum of J occurs between the two boundaries.

$$J\left(u\right)=\beta_1 u^2+\beta_2 u \tag{6.12}$$

Where $\beta_1 \leq 0$. Then, the inequality stated above can be easily proved as follows. J(E{u}) and E{J(u)} can be computed by Equations 6.13 and 6.14.

$$E\left\{J\left(u\right)\right\}=\beta_1 E\left\{u^2\right\}+\beta_2 E\left\{u\right\} \tag{6.13}$$

$$J\left\{E\left(u\right)\right\}=\beta_1 E\left\{u\right\}^2+\beta_2 E\left\{u\right\} \tag{6.14}$$

Subtracting Equations 6.13 and 6.14, an using Cauchy-Schwarz inequality (Bishop, 2007), it can be easily shown that $J(\mathrm{E}\{\mathbf{u}\}) \geq \mathrm{E}\{J(\mathbf{u})\}$. Consequently, the optimization algorithm is set to first compute optimal control settings for multiple responses surfaces, and then to compute the expected value of the control vector.

6.5 PRODUCTION OPTIMIZATION OF THE FIVE-WELL EXAMPLE

To continue the demonstration, the optimization algorithm was used to find optimal injection and production rates for the reservoir models with a 5-spot injection pattern shown in Figure 6-5. The total production time (2000 days) was divided into 4 injection periods of 500 days. The maximum injection or production rates are 400 ft^3/day (71.2 STB/day). The injector and 3 producers are operated independently, Prod#4 is operated such that the total injection and production rates are in equilibrium at all times. Thus, the number of control variables is 16: 4 control periods for 4 wells.

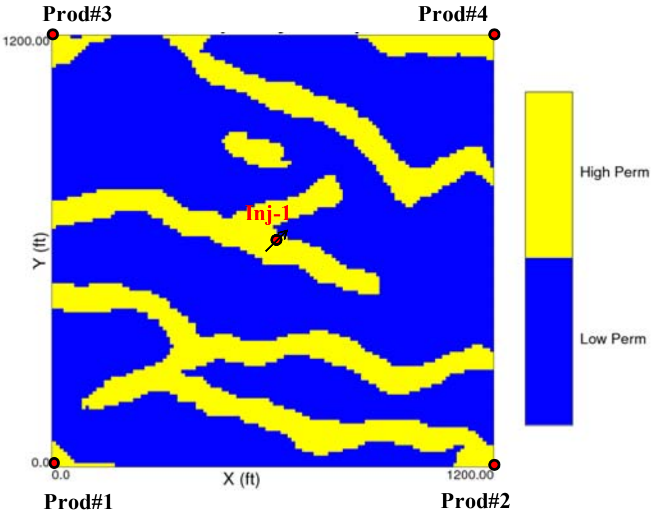

Figure 6-5 Reference reservoir model with 1 injector and 4 producers reproduced from Figure 5-20

The optimization algorithm returned the control settings presented in Table 6-3. The optimal control settings start with a high injection rate (395 ft^3/day), followed by a reduction in injection rate to control water production. Then, the injection rate is increased at the final period to maintain the production at Prod#1, which still had a low water cut.

Table 6-3. Optimal control settings for the five wells corresponding to the reservoir model shown in Figure 6-4. Well rates in ft^3/day

Time (days)	Inj#1	Prod#1	Prod#2	Prod#3	Prod#4
0-500	395	0	269	36	90
500-1000	365	60	24	140	141
1000-1500	201	114	5	53	29
1500-2000	300	132	67	24	77

In the first period, the main producer is Prod#2, in the second period Prod#3 and Prod#4 produce more than 75% of the fluids, and in the third and fourth periods the main producer is Prod#1. The alternation of production rates among producers controls the shape of the polymer bank by redirecting the flow when a flow channel is being developed.

The distribution of the cumulative oil production is: 17.8% from Prod#1, 36.75% from Prod#2, 19.8% from Prod#3 and 25.5% from Prod#4. Thus, 62.25% of the oil came from the wells at the right of the injector (Prod#2 and Prod#4), while 37.75% came from the wells at the left (Prod#1 and Prod#3). This production scheme is logical because the polymer and water banks move toward the left edge of the reservoir as shown in Figure 6-7 conforming to the high permeability channel next to Inj#1. The optimal control setting preferentially distributes fluids away from that flow channel.

Figure 6-6 confirms the result that the final net present value of the expected value of the control vector is higher than the mean of the other runs ($J(E\{\mathbf{u}\})>E\{J(\mathbf{u})\}$). Hence, the mean of the control vectors from multiple runs is considered as the optimal control vector. The optimized control strategy is compared to the strategy where the production is distributed evenly among the 4 producers and the injection rate is maintained at maximum. The optimized production strategy results in a slightly higher net present value after 2000 days of operation. It is important to note that the optimal control settings yielded higher NPV with less oil produced (Figure 6-6b), implying that the water-oil ratio was maintained low thereby reducing the produced water handling cost.

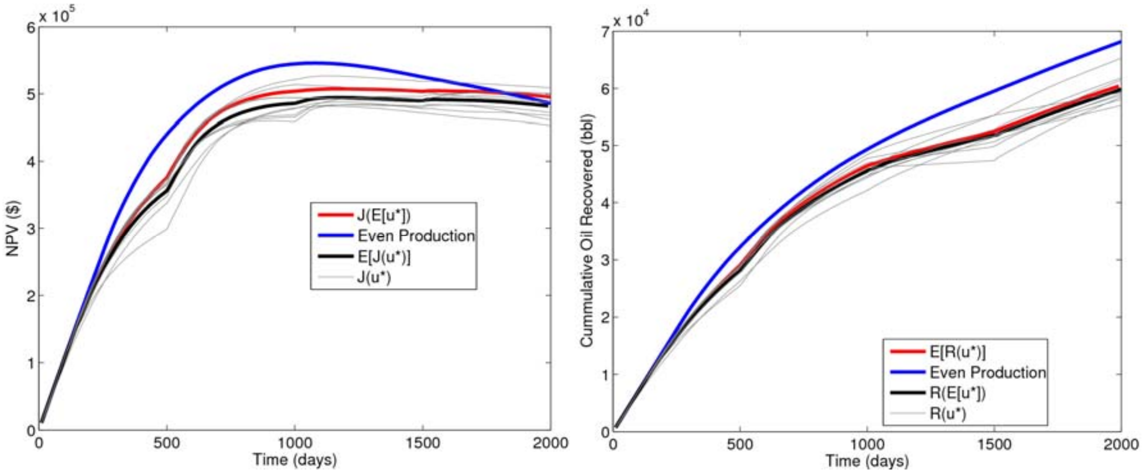

Figure 6-6 a) Net present value curves and b) recovery curves for 10 optimization runs (gray). Curves in black correspond to the expected value of the 10 optimization runs. Curves in blue are for an equal production rates schedule. Curves in red correspond to a simulation operated at the expected value of the optimal rates.

The total injected volume for the optimal control setting is less than that of the maximum injection setting, resulting in a smaller and more compact polymer bank as seen in Figure 6-7. Although the permeability contrast makes the reservoir highly heterogeneous, the position and extension of high permeability channels favors good sweep efficiency because there is no direct connection between the injector and any producer. Rather, the high permeability channels provide pathways for polymer to reach edges of the reservoir that otherwise would not have been contacted. Nevertheless, it is important to consider that the optimal control strategy depends on the geology of the reservoir. This issue is probed further in the next chapter and leads to the development of a feedback control scheme for maximizing oil recovery.

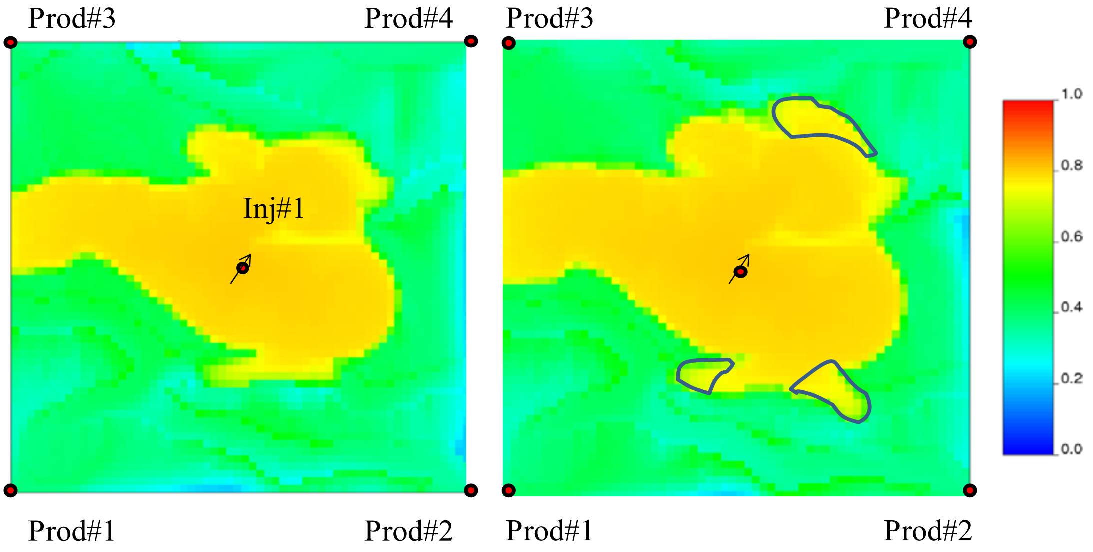

Figure 6-7 Water saturation map at 2000 days for a) optimal control rates and b) evenly distributed production rates. Regions with major difference in polymer saturations are indicated in the maps.

6.6 CONCLUSION

Two examples of production optimization demonstrated that a combination of the proxy model, response surface and a gradient based optimization is a feasible alternative for finding optimal control strategies. It is clear from the first example that the optimal control setting is a strong function of the effective permeability heterogeneity that results in communication between the injection and production well pairs. In the five-spot injection pattern example, alternation of peak production rates between producers allows controlling the growth of polymer bank, resulting in improved economics for the project. That example also showed that maximizing recovery does not imply maximizing NPV;

higher NPV can be achieved with less oil and water production. The optimization method presented in this chapter is essential to implement a feedback control scheme for polymer flooding as demonstrated in Chapter 7.

Chapter 7: Feedback Control of Polymer Flooding

7.1 FEEDBACK CONTROL OF POLYMER FLOODING

The feedback control framework integrates the model selection method presented in Chapter 5 and the production optimization algorithm presented in Chapter 6 as illustrated in Figure 7-1. The process starts with a large ensemble of reservoir models that reflects the prior uncertainty in reservoir description, including all plausible geologic scenarios conditioned to the available static data. In the absence of production data all prior reservoir models are equally probable, thus the model selection algorithm is applied to select a few reservoir models. The optimal control settings are computed over the retained subset of reservoir models by the optimization algorithm and then implemented in the field through flow control devices that regulate injection and production rates. The well responses are monitored with pressure sensors at injectors and producers, and fluid samples are analyzed to determine the water cut and polymer concentration at the producers. At the same time, a reservoir simulation model (UTCHEM) is constrained to the current well controls for simulating the production data during that control period. Then, the field production data is compared to the production data simulated with UTCHEM. If the mismatch between simulated and actual production data is tolerable, the operation continues with the already formulated control settings. On the contrary, if the mismatch is more than a tolerance threshold, the observed production data is utilized to refine the selected reservoir models, and the control settings are revised with the optimization algorithm. In this way the control system is a closed-loop operation that continuously updates the reservoir information and revises the production strategy.

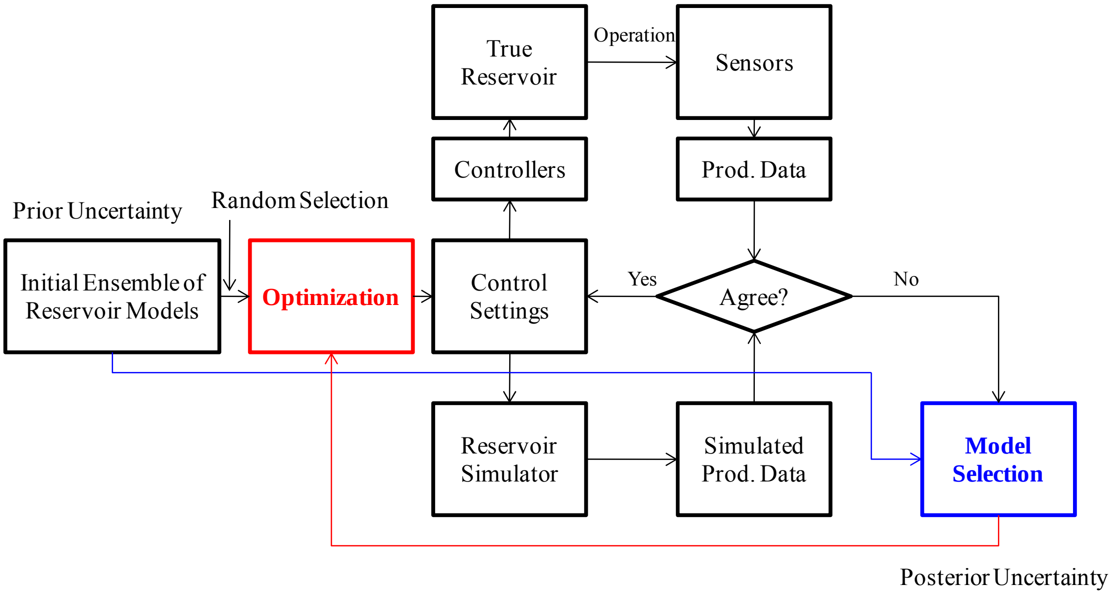

Figure 7-1 Flowchart of the feedback control system for polymer flooding

7.2 IMPLEMENTATION OF THE FEEDBACK CONTROL SYSTEM TO THE TWO-WELL RESERVOIR EXAMPLE

7.2.1 Formulation of Optimal Control without Previous Production Data

The two-well reservoir model described in previous chapters was used to demonstrate the feedback control system. The injection and production rates are to be optimized during 2000 days of operation such that the final net present value is maximized. The total operation time is divided into 8 control periods of 250 days. The control vector consists only of the injection rates because the production rates are equal to the injection rates at any time. The prior geologic uncertainty is represented by the ensemble of 600 reservoir models described back in Chapter 4, Figure 4-3. The four reservoir models shown in Figure 7-2 were randomly selected from that ensemble, and processed with the optimization algorithm to formulate the optimal injection and production rates for the first period. The control schedule for all 4 realizations coincides in the first control period when the injection rate is the maximum (200 ft^3/day=35.61

BTB/day). Therefore, the reservoir was operated at 200 ft3/day during the first 250 days. After that, the optimal injection rates vary as shown in Figure 7-3.

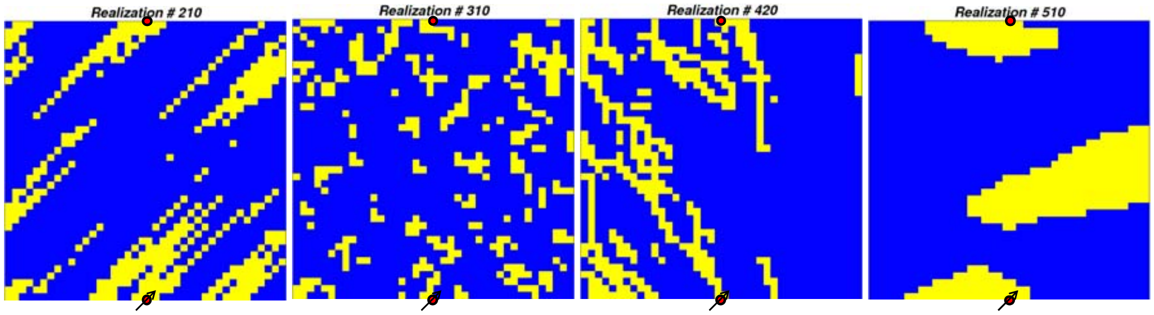

Figure 7-2 Reservoir models randomly selected to be submitted to the optimization algorithm

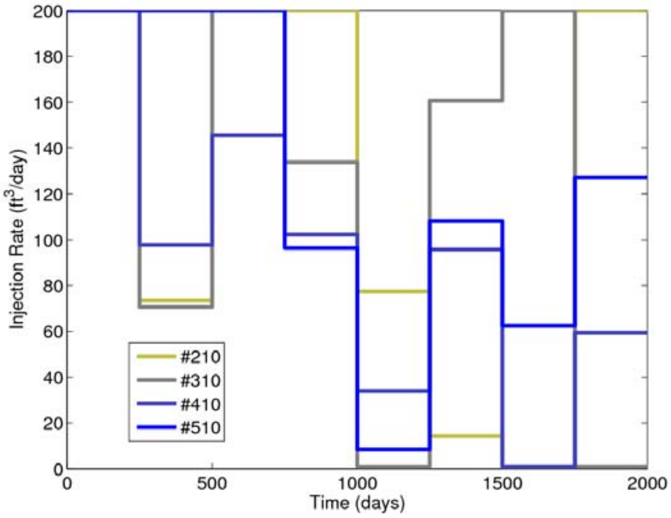

Figure 7-3 Optimal control rates calculated for the reservoir models randomly selected before data integration

The injection pressure, production pressure and water cut are monitored during 250 days in the reference reservoir and compared to the production data simulated for the 4 selected reservoir models. The huge uncertainty in the prior ensemble is reflected in the wide variance of well pressure profiles observed in Figure 7-4. Figure 7-4 also shows that the randomly selected models predicted the onset of water production to occur at a much later time than the actual measurements.

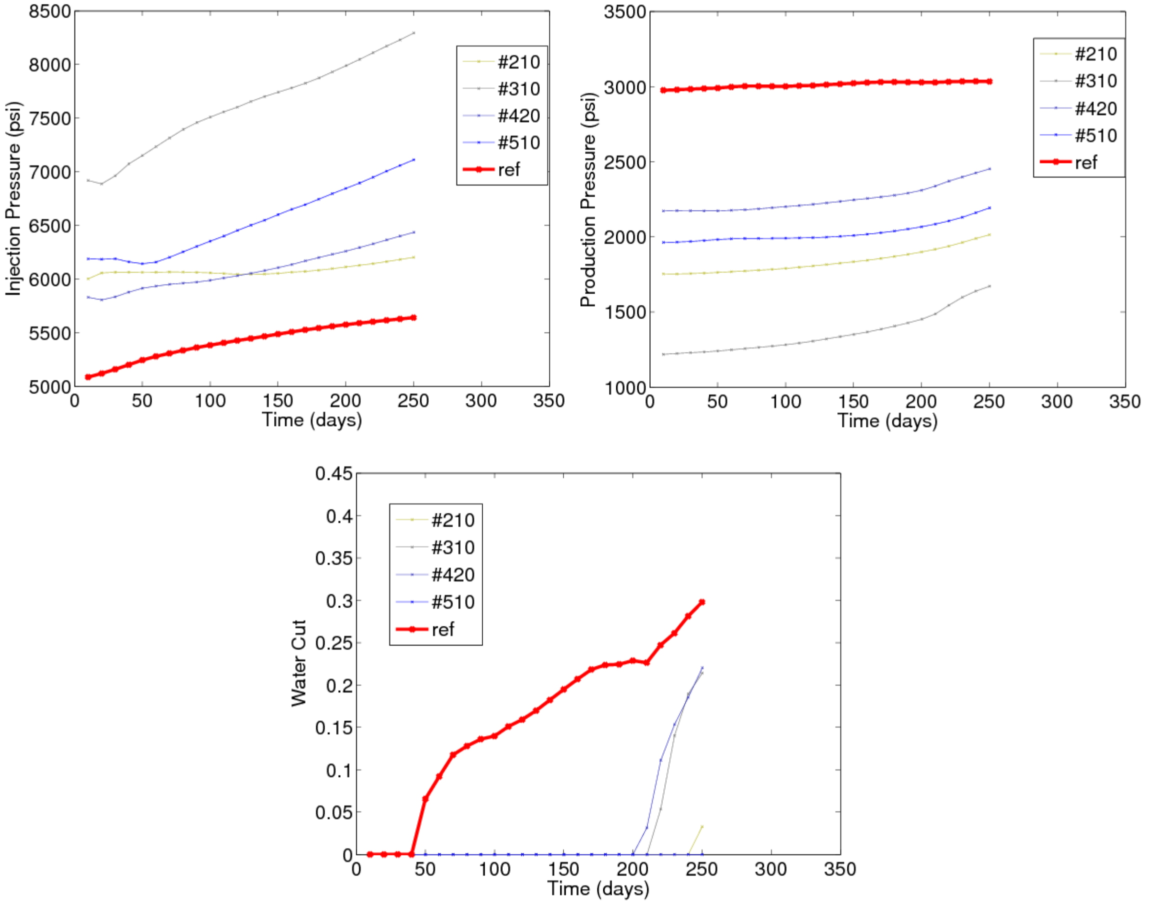

Figure 7-4 Production data for reference reservoir and 4 randomly selected realizations during the initial control period

7.2.2 First Updating and Optimization of the Second Period

Subsequently, the production response observed in the "true" reservoir over the first 250 days of injection, was used to select the posterior set using the distance-based model selection algorithm described in Section 5.4. Only four reservoir models from the reduced suite shown in Figure 5-19 are used to determine in the optimization in order to save time. These for reservoir models have in common a high connectivity streak between injector and producer and the same spatial correlation. These reservoir models exhibit similar injector pressure, producer pressure and water production profiles as the observations as shown in Figure 7-5. The selected reservoir models captured the onset of water production but the evolution of water cut is slightly overestimated.

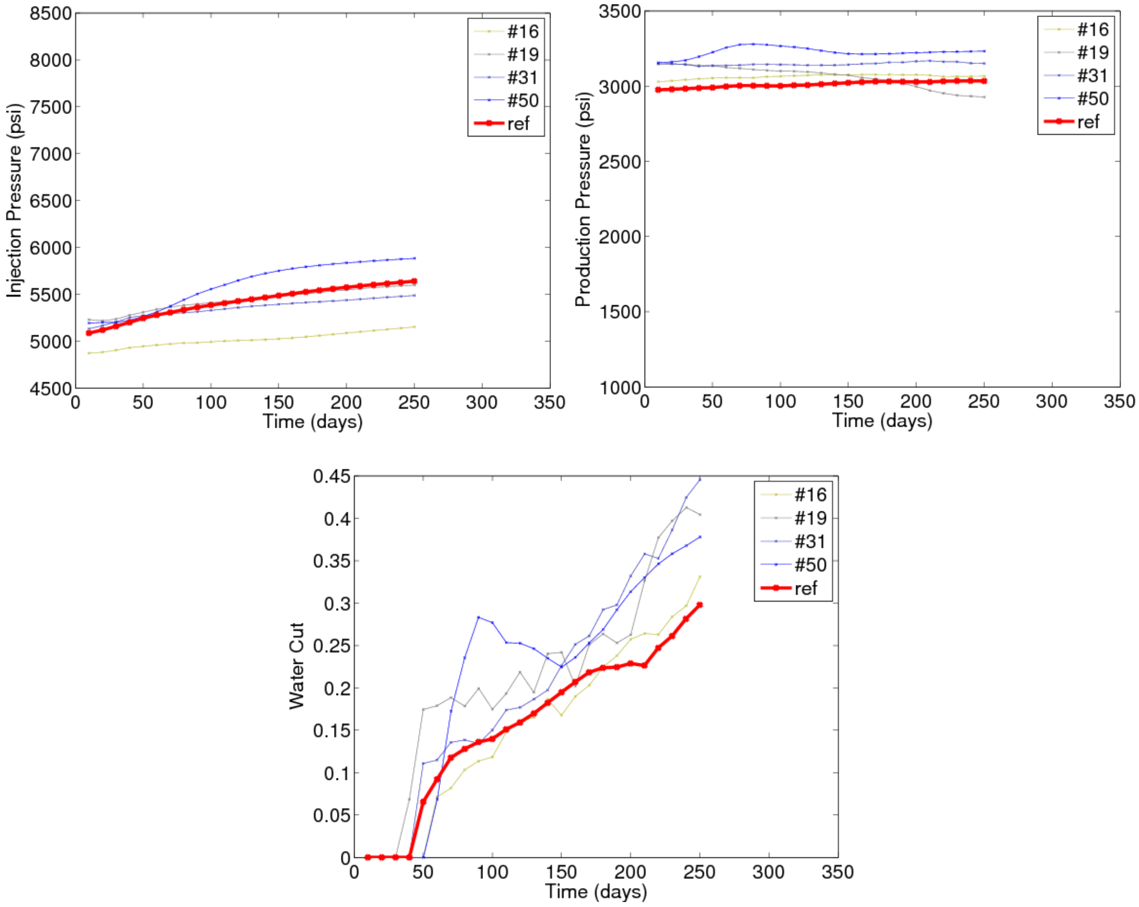

Figure 7-5 Production data for the "true" reservoir and 4 reservoir models from the reduced set at the end of the model selection process

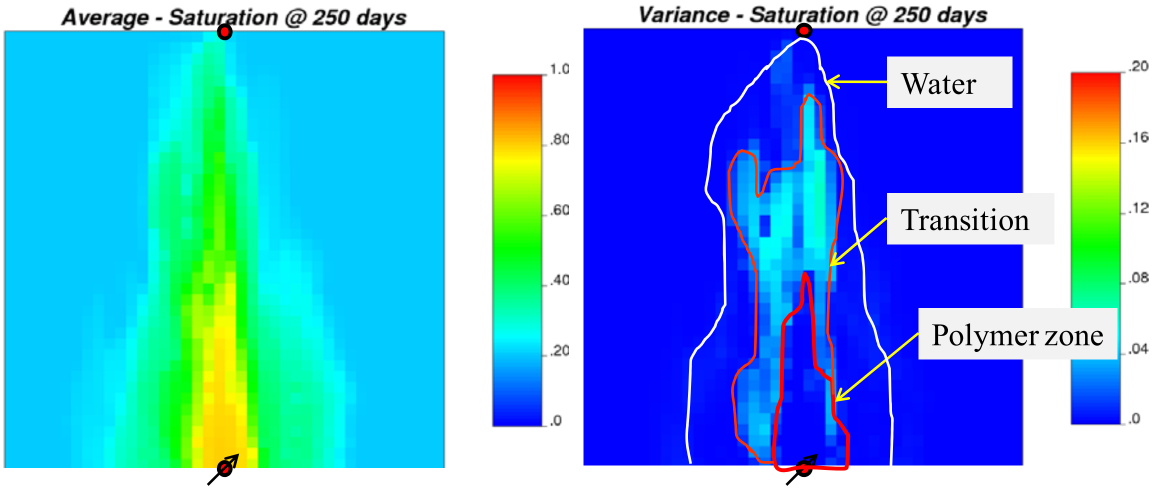

Figure 7-6. Assessment of water saturation profile in the selected reservoir models. The figure on the left shows the average water saturation over the 4 posterior reservoir models retained. The figure on the right shows the variances over the same set of reservoir models.

The dynamic state of the reservoir at the end of the first period is assessed through the simulation results of the four selected reservoir models. As evidenced from the early water production, the water bank already has reached the producer at that time. The pathway between the wells restricts the areal extension of the plume as seen in Figure 7-6 (left). The area with more uncertainty (higher variance) is the transition zone between polymer and water banks as can be seen in Figure 7-6 (right).

The production/injection strategy is then revised with the new set of reservoir models. The optimization algorithm looks to optimize rates only after 250 days. Thus, the number of control variables is 7 for the second step. Figure 7-7 shows that the optimal injection rate reduces gradually as the reservoir models predict high water production due to the high connectivity between wells. The optimization results are consistent with the objective of delaying water production as much as possible while producing oil to generate revenue. Comparison of Figure 7-3 and Figure 7-7 indicates that the variability in the optimal control settings has also been reduced.

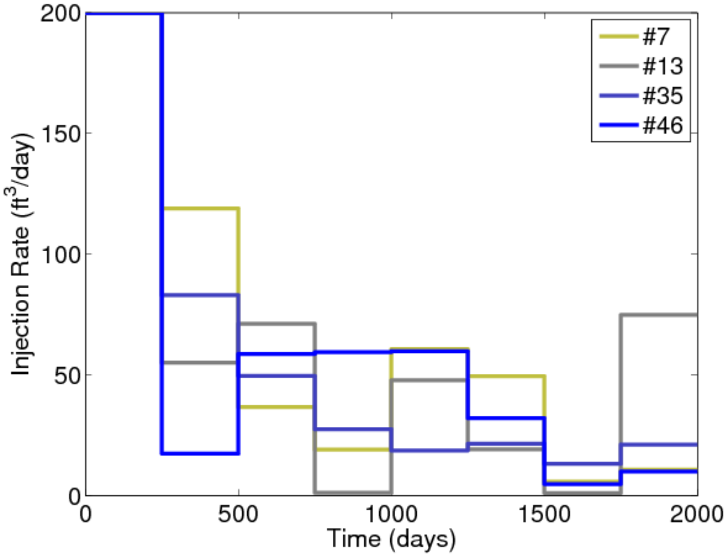

Figure 7-7 Optimal control settings for the reduced set of reservoir models after using production data for 250 days. The optimization starts from 250 days to 2000 days

If we have four realizations with similar production data but different control settings, what optimum control rate should be used for the next period? Figure 7-8 compares the NPV profile corresponding to the optimum control setting for each of the 4 reservoir posterior models against the profile obtained corresponding to the effective optimum control expressed as the average of the control strategy for the 4 realizations. Judging by the closeness of those two NPV profiles, it can be concluded that the expected value of the optimal control yields final NPV that is comparable to the optimal control for each of the four realizations as shown in Figure 7-8. Hence, the feedback control for the second period is calculated as in Equation 7.1.

$$\mathbf{u}^* = \frac{1}{4}\sum_{i=1}^{4}\mathbf{u}_i \qquad (7.1)$$

The effective or average control can also be calculated as a weighted sum of the control for each realization with the weight identified proportional to the economic gain or loss corresponding to over or under production of fluids.

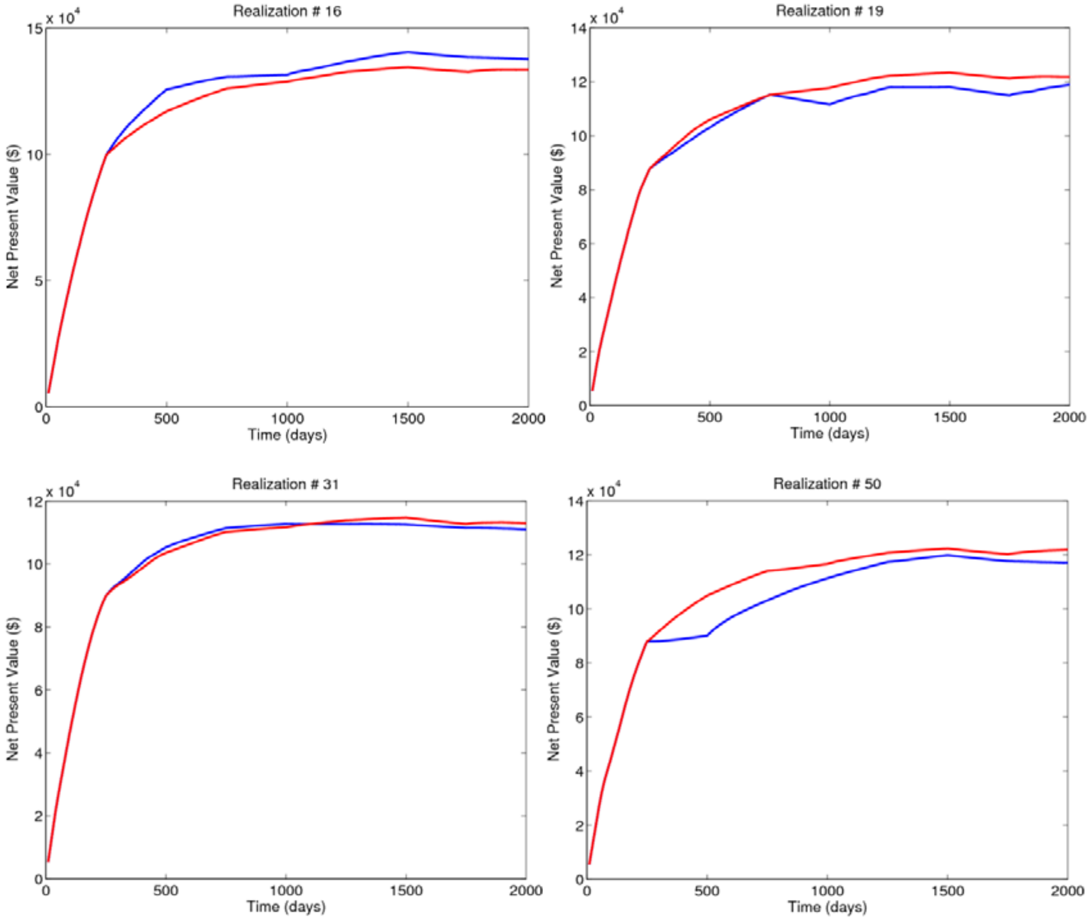

Figure 7-8 Net present value profile corresponding to the optimal control vector of each of the four selected realizations (blue) compared to the NPV profile corresponding to the expected value of the control vector (red) calculated using Equation 7.1.

Then, the injection and production rates for the second control period are implemented to control the reference reservoir from 250 to 500 days, and the well responses are further monitored. The production data for the four selected realizations is further simulated using UTCHEM corresponding to the updated control strategy and the simulated response is compared to the monitored reservoir response during the second period. Figure 7-9 shows that the well pressures continued along the same trend without diverging significantly from the reference, especially at the producer. But the overstimation of the water cut persists and consequently the mismatch in cumulative oil production gets progressively worse. Thus, the selection of models needs to be revised to incorporate the new production data.

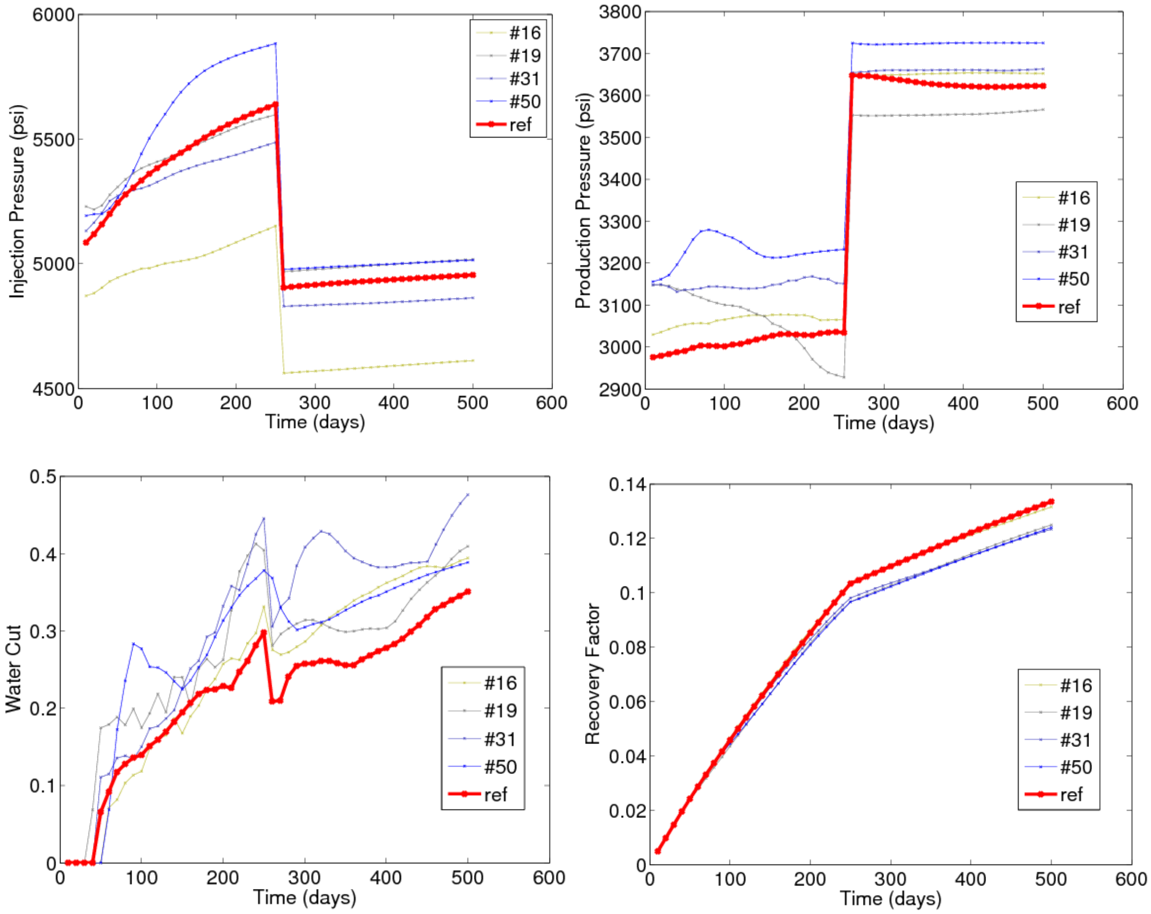

Figure 7-9 Production response of the reference reservoir compared to the simulated response for the 4 reservoir models over the first two control periods (500 days)

7.2.3 Second Updating of Model Selection

Before refining the suite of posterior models further, the production data is analyzed to explain the behavior of water production as the result of a physical process in the reservoir. Note that the permeability channel in the selected realizations is straight.

Once water breaks into the producer through a high permeability channel in the selected reservoir models, water production continues increasing because the flow continues mainly through the swept channel. In contrast in the reference reservoir, even though water breaks through approximately at the same time, the water production is less than the water production in the selected realizations. This indicates that water has found an alternate flow path such that not all water injected goes through the main path.

This particular behavior is explained by observing Figure 7-10, which shows the saturation front in the reference reservoir at the end of 500 days. The displacement front branches out along the path of the channel diverting the flow to a secondary channel. Hence, part of the injected fluids displaces oil in that region without being produced until late time.

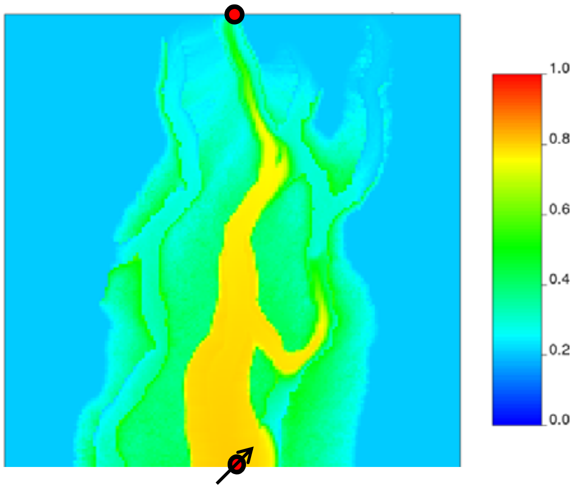

Figure 7-10 Water saturation in the reference model after 500 days of operation

However, in reality the saturation picture in the reference (or “true”) reservoir is not available to us. A geologic description with high permeability channels branching out is not present in the initial ensemble. Note that the lateral spread of the injected water occurs at a location away from the wells. Consequently, there is no possibility of depicting the branching characteristic of the channel based only on the well data. It is necessary that the analog reservoir models from which statistics to develop the prior suite of models be rich enough to provide a finite probability for such branching to occur. Therefore, the ensemble of reservoir models needs to be augmented to incorporate a wider geological description. To incorporate more plausible geologic scenarios, realizations that replicate the spatial template of the training image shown in Figure 7-11 were added to the augmented ensemble for a new model selection process. The model selection method allows expanding the representation of prior uncertainty at any time during the selection process if other geologic models need to be tested.

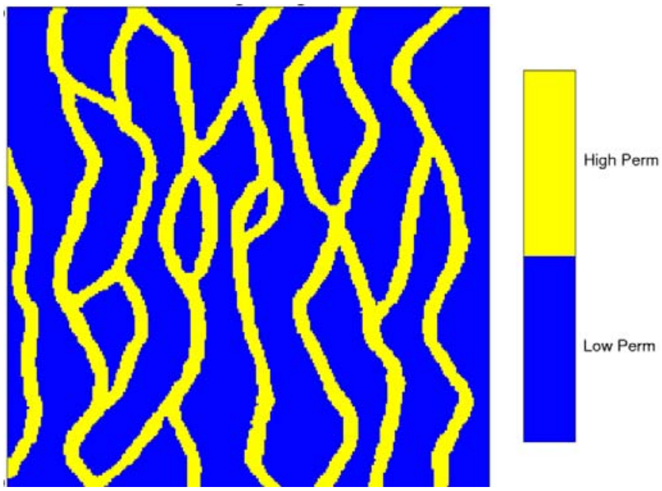

Figure 7-11 Fifty reservoir models in the augmented ensemble honor the spatial statistics of the above training image.

The augmented ensemble should also preserve the main characteristics of the reduced set from the previous step. The challenge now is how to generate a new suite of realizations with *similar* characteristics in well connectivity as the already reduced set. We approach this problem in two ways. First, the reduced set is analyzed to compute the proportion of each geologic description of the geology, and this proportion is maintained in the augmented ensemble. Second, the probability of rock type at each grid node is quantified by aggregating the models in the selected set thus far. These location dependent probabilities are introduced as a secondary piece of information for the generation of new realizations using sequential indicator simulation *sisim* and single normal equation simulation *snesim*. As a result, the new ensemble of realizations accounts for the correction in local probability of the rock type indicator. Figure 7-12 shows that in the final suite there is high probability of encountering high permeability channels between wells, but that probability reduces away from the wells.

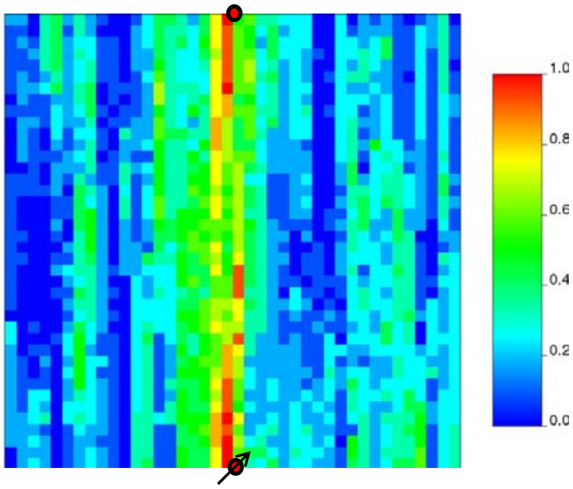

Figure 7-12 Map of probability of having a high permeability rock type from the selection of models at 250 days

Figure 7-13 shows that the introduction of the updated probability of the geologic scenarios and the correction in local probability yields a reduction in the posterior uncertainty in production data visualized over the models in the final selected set. These results are after doing the model selection using the response over the first 250 day production period. The distribution of the injection pressure profile of the posterior ensemble is not centered on the reference response, while the producer pressure and water cut responses are. The uncertainty in water cut predictions is still large, but most of the reservoir models with late breakthrough are not present in the posterior set.

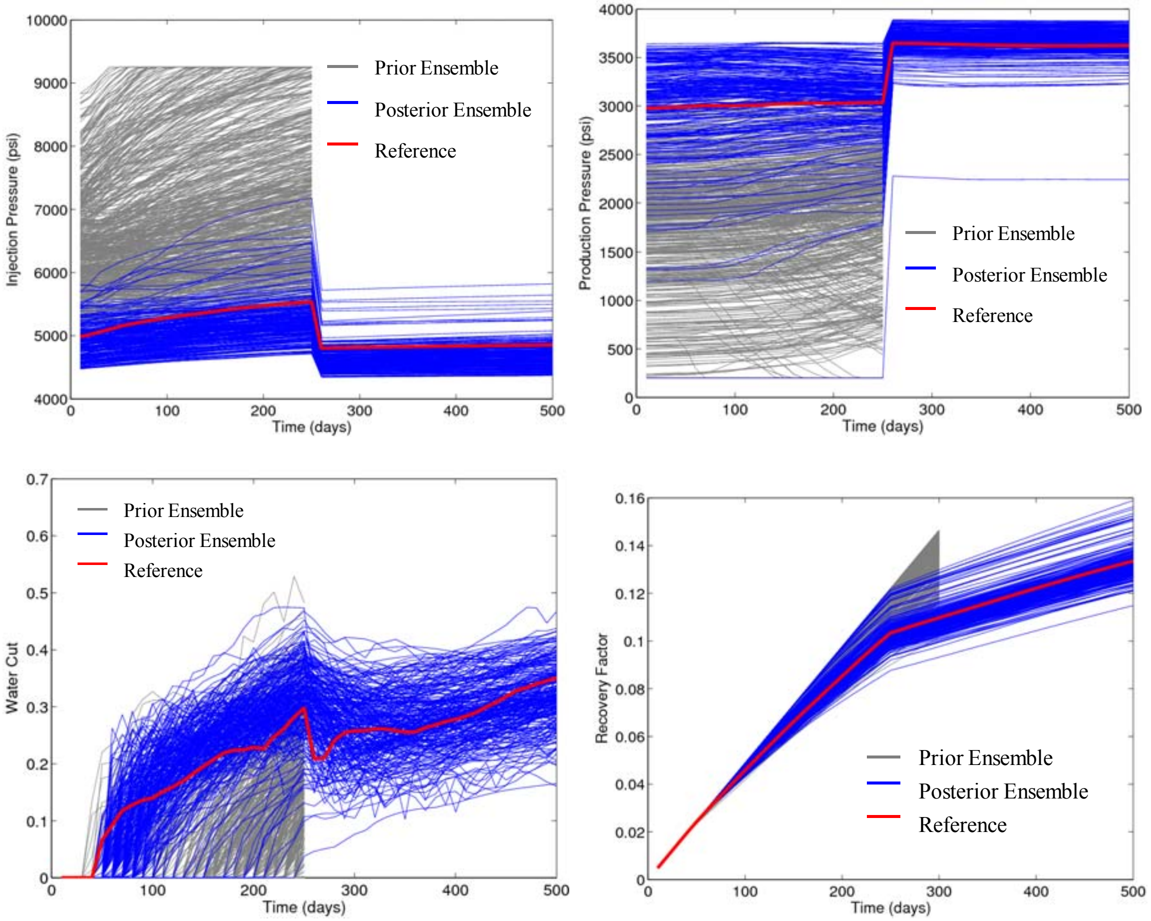

Figure 7-13 Comparison of production data of prior and posterior ensembles of reservoir models compared to the reference. These results were obtained after the prior ensemble was augmented so as to include additional models with channels exhibiting complex connectivity.

The model selection process is repeated using production data up to 500 days. The posterior uncertainty in the production data yielded by the reservoir models from the final set is tighter and more centered on the reference data set as shown in Figure 7-14. A few outliers with regard to the producer pressure remain in the final suite. Also, it is noticed that it is difficult to reduce the wide spread in the water cut profiles.

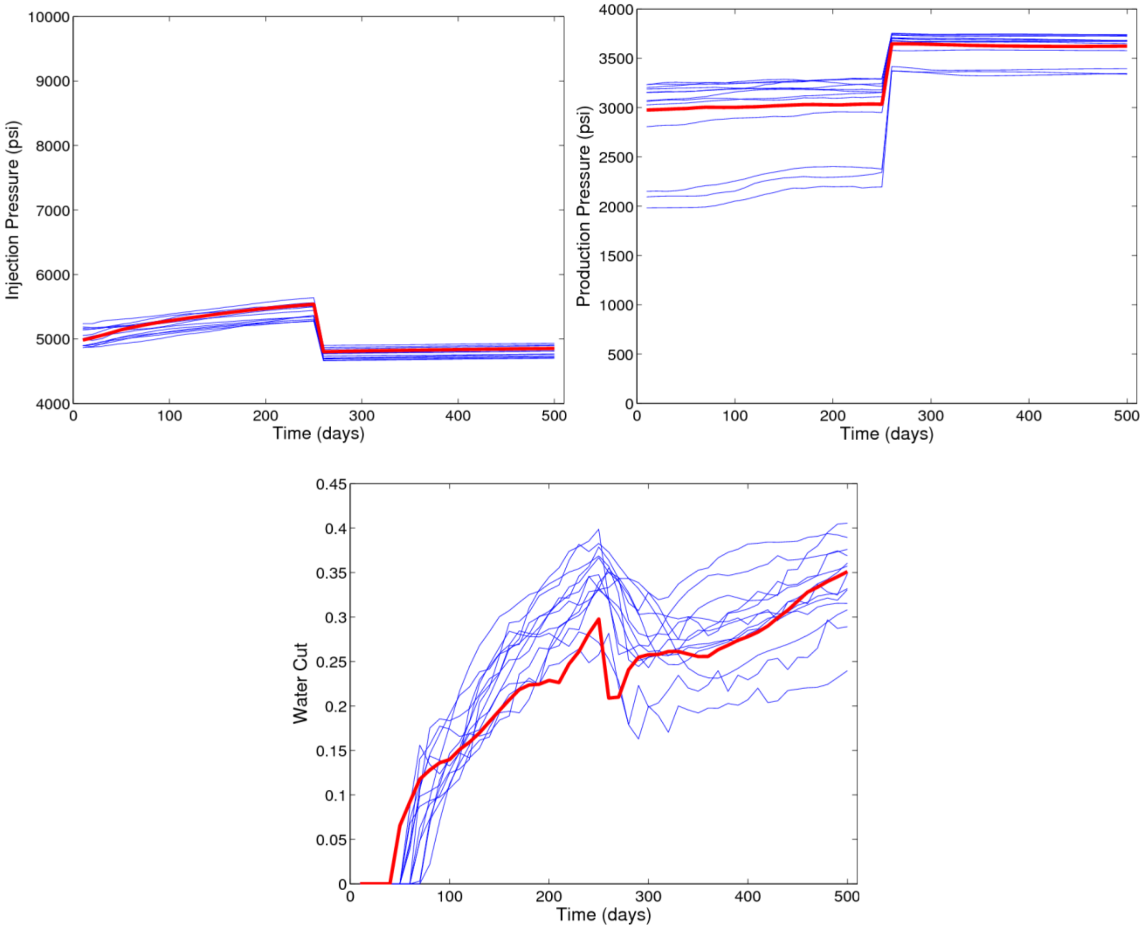

Figure 7-14 Production data for the final set of reservoir models

The four reservoir models with the best match in production data (Figure 7-15) among the final set were selected to reevaluate the control scheme for the remaining 1500 days. Note that the selected realizations have high well connectivity as introduced through the probability map shown in Figure 7-12, but the connection between wells is through channels with a sinusoidal and branched shape as reproduced from the training image shown in Figure 7-11. The selected realizations resemble better the underlying geology of the reference reservoir (Figure 5-20).

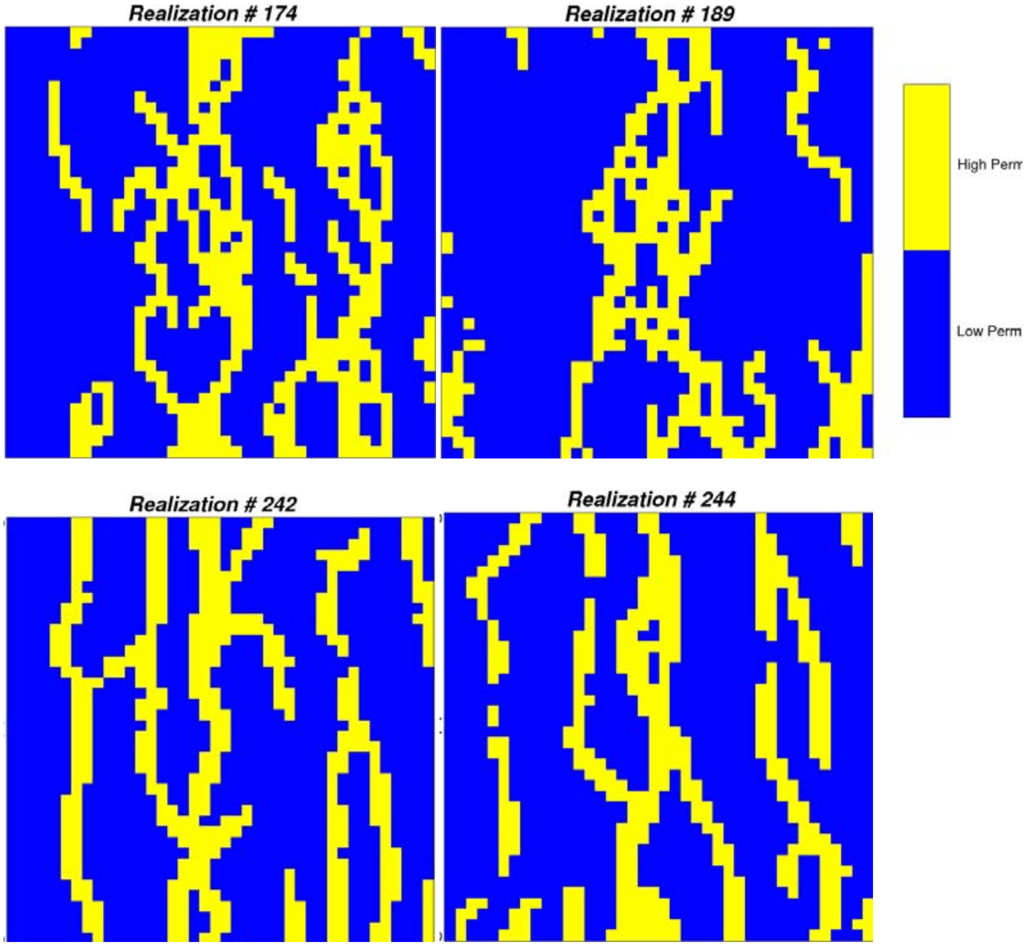

Figure 7-15 Four reservoir models selected from the final set after incorporating production data of 500 days

7.2.4 Feedback Control with Updated Selection of Models

The next step is to reevaluate the optimal control rates using the new selection of reservoir models. Again, the expected value of the control vector for the third period is used to control the injection and production rates in the reference reservoir, and production is monitored for the subsequent 250 days.

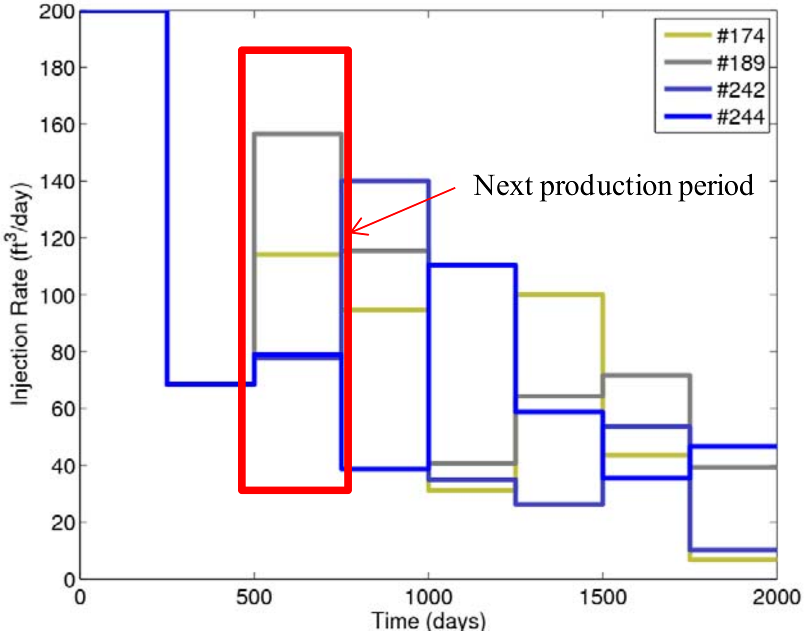

Figure 7-16 Optimal control settings for the selected four realizations after model selection using data gathered over 500 days.

Figure 7-17 shows that the production data from the four selected reservoir models continue being consistent with the observations from the reference. Although the residual uncertainty in water production remains, those predictions do not deviate significantly from the reference. As a result, no additional revisions to the control strategy are required after the two updates.

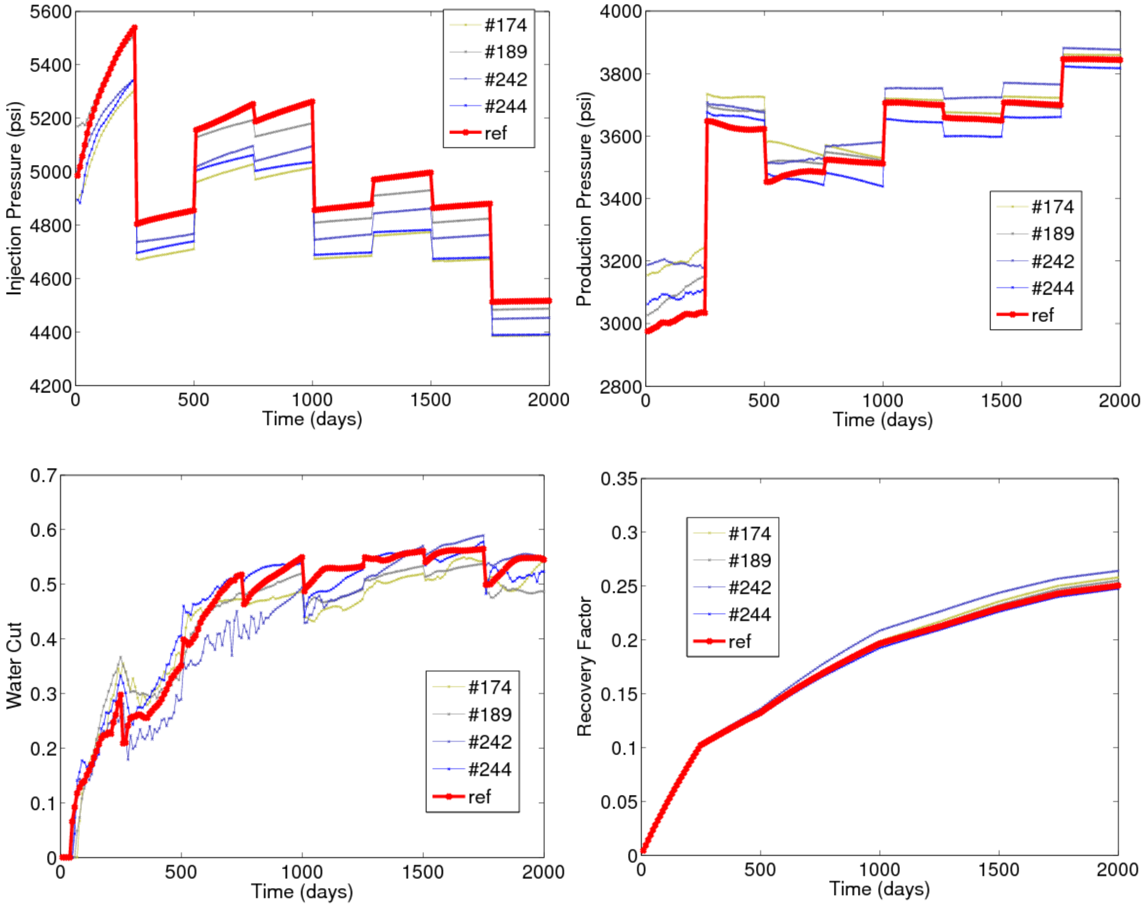

Figure 7-17 Production data after 8 control periods (2000 days) for the selected reservoir models and the reference.

The NPV obtained after feedback control and the NPV obtained in Chapter 6 using the reference reservoir model (called optimal NPV) are close as shown in Table 7-1. In order to quantify the value of the information, the NPV for the feedback control case is compared to the NPV obtained with the optimal control settings of a reservoir model chosen without updating the prior uncertainty (initial ensemble). Using the optimal NPV as a reference, the feedback control system results in a loss of 0.78 %, whereas the optimal control settings for a random reservoir model results in a loss of 7.9% as presented in Table 7-1.

Table 7-1 Results of control strategies for a random model, feedback control system and the reference reservoir

Control Settings	**Net present value ($)**	**Recovery Factor**
Random model	132660	0.2981
Feedback control	142510	0.2466
Reference Reservoir	143690	0.2431

It is interesting to note that the prior models are more optimistic about the reservoir recovery factor than the posterior suite. In addition, the feedback control system returns a higher NPV with less oil and water produced as can be inferred from Table 7-1, leaving more oil for future production. This example demonstrates the value of implementing a system for continuous monitoring and control of the operation.

7.3 IMPLEMENTATION OF THE FEEDBACK CONTROL SYSTEM TO THE FIVE-WELL RESERVOIR EXAMPLE

The feedback control framework is now demonstrated in the five-well reservoir. At this point the reservoir has been in operation for 250 days and the production data of that period have been used for the model selection as shown in Chapter 5, Section 5.5.

7.3.1 Optimizing the Second Period

Now, the selected reservoir models in Section 5.5 are used to reevaluate the production strategy for the next control period using the production optimization algorithm. The number of control variables reduces to 7x4: 7 control periods of 250 days are remaining until the final time (2000 days), and 4 for wells controlled independently, the fifth well (Prod#4) is constrained by the material balance. Figure 7-18 shows the optimal control settings computed for the current selection of reservoir models.

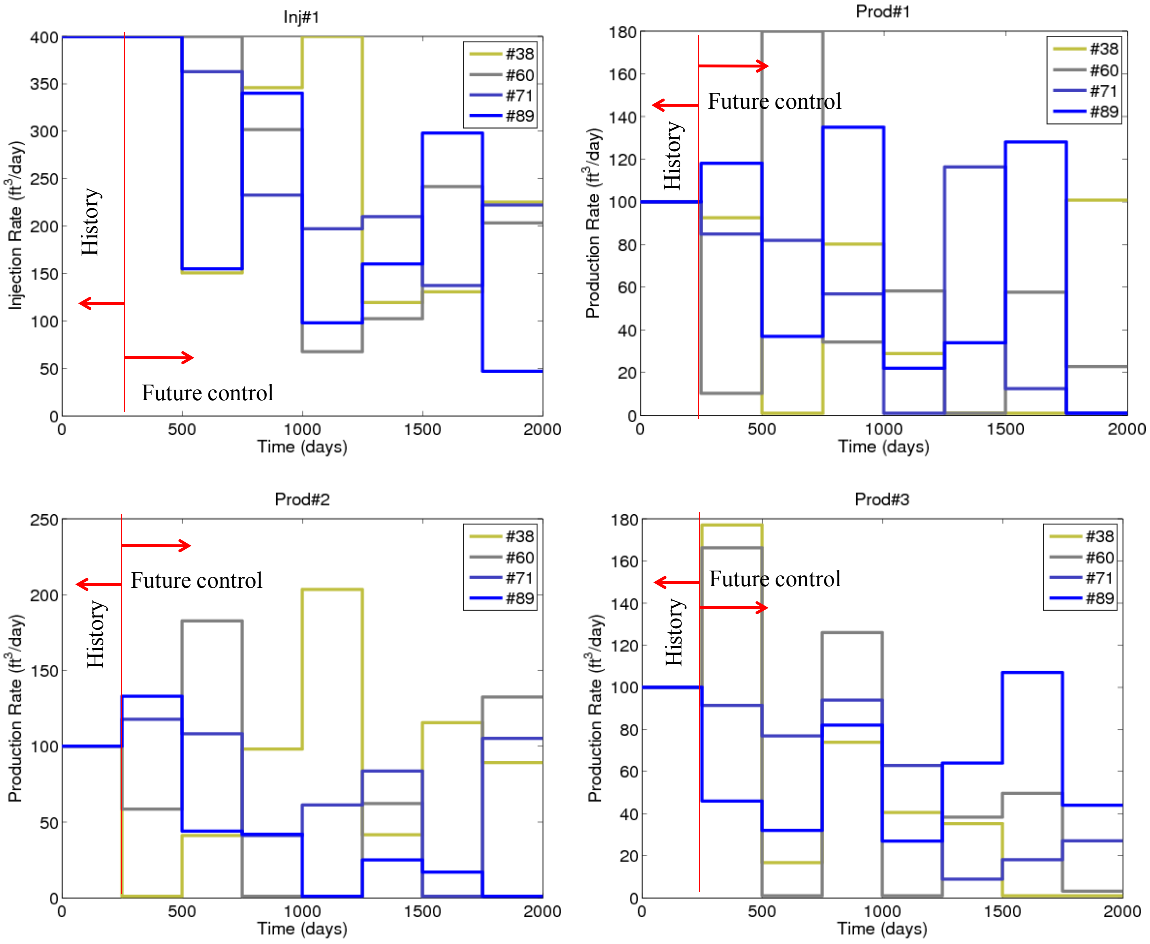

Figure 7-18 Optimal control settings for 4 realizations selected after updating the model selection with production data of 250 days. Control settings of Prod#4 are deduced from equilibrium between injection and production rates.

Table 7-2 shows the expected value of the control rates for the rest of the operation. Then, the reservoir is operated during the next production period (250-500 days) at the rates indicated in Table 7-2. The optimal injection rate is reduced over time to control the excessive water production. The variability of the optimal control rates is high because still there is high uncertainty in the reservoir models as the selection includes models exhibiting wide variability in connectivity characteristics.

Table 7-2 Expected value of optimal control settings for the five-well reservoir, starting at 250 days

Time (days)	Inj#1 (ft^3/day)	Prod#1 (ft^3/day)	Prod#2 (ft^3/day)	Prod#3 (ft^3/day)	Prod#4 (ft^3/day)
250-500	400	76	78	120	126
500-750	267	75	94	32	66
750-1000	305	77	46	94	89
1000-1250	191	28	67	33	64
1250-1500	148	38	53	37	20
1500-1750	202	50	34	44	75
1750-2000	174	31	82	19	42

After adjusting the injection and production rates corresponding to the expected value of the optimal control rates, and monitoring well pressure and water cut for the next control period (250-500 days), it is observed that water production has initiated at all wells (Figure 7-19). Water breaks first into Prod#2 (270 days=0.0938 PV), then into Prod#1 and Prod#4 (290 days=0.1 PV), finally into Prod#3 (330 days=0.108 PV). The selected models continue showing the same trend in pressure as the reference, but they fail to reproduce the water breakthrough in general. This implies that the model selection needs to be updated again to correct this mismatch. It is important to have a reliable water production forecast for the optimization of well controls. Furthermore, water breakthrough brings key information about well connectivity. The approximately equal water breakthrough suggests that there is no a preferential communication between the injector and the producers.

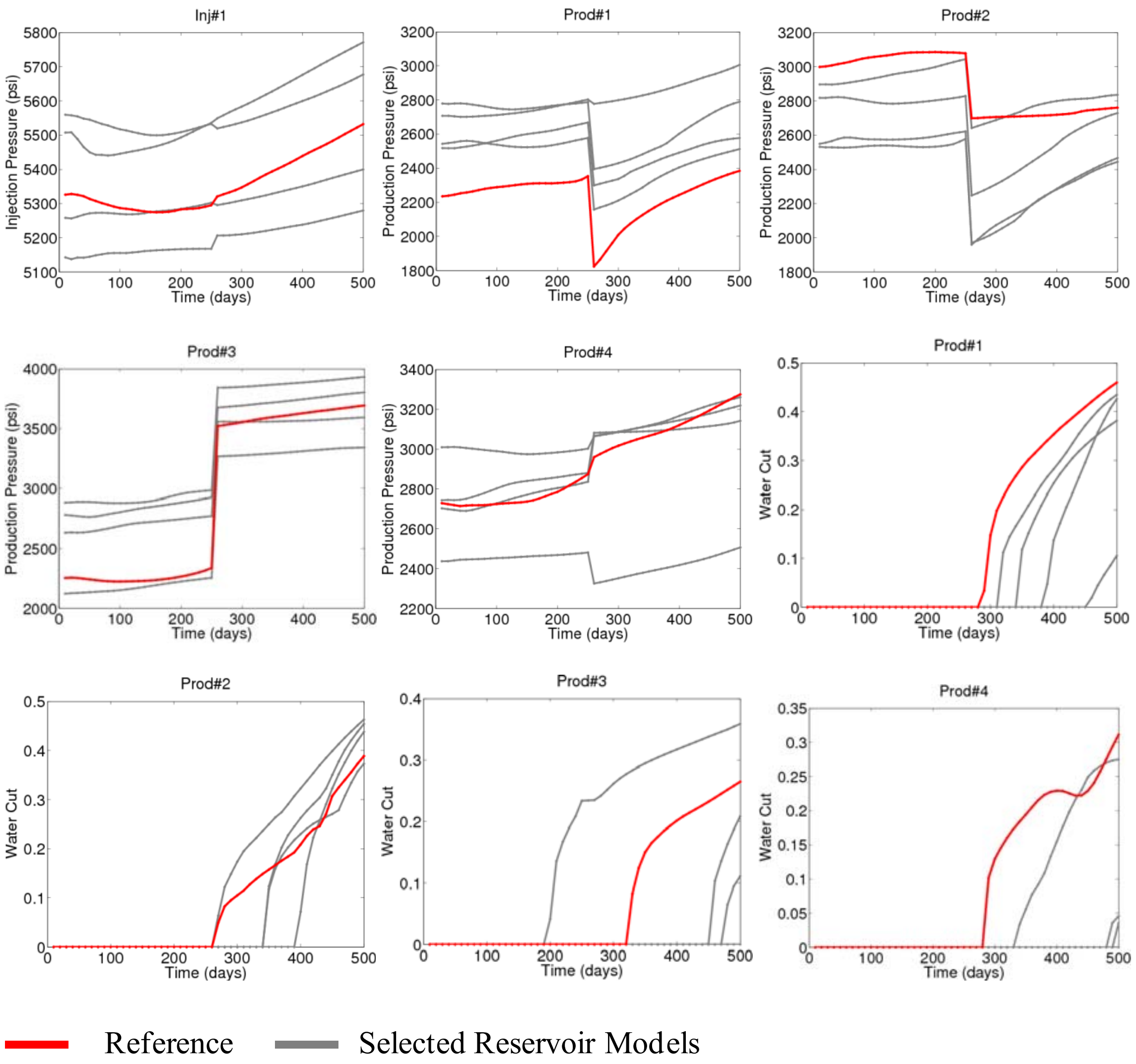

Figure 7-19 Production data for reference and set of reservoir models selected after the first control period (250 days)

7.3.2 Updating the Model Selection after Water Breakthrough

After observing the production data for 500 days, the objective becomes to refine the selection of models such that the water production is better reproduced. It was shown in Chapter 4 that the correlation between the actual breakthrough time and the breakthrough from the proxy model is high (ρ=0.89). Then, the strategy is to include in the new ensemble only reservoir models whose breakthrough from the proxy model corresponds to the breakthrough expected for the reference according to the observation. A linear regression is performed with the available data shown in Figure 4-6 (Eq 7.2.)

$$T_{BT-UTCHEM} = 0.25 \times 10^{-5} \times T_{BT-PROXY} + 0.059 \qquad (7.2)$$

Table 7-3 – Criteria used to screen reservoir models for the new ensemble

	Prod#1	Prod#2	Prod#3	Prod#4
Actual breakthrough	0.1007	0.0938	0.1146	0.1007
Min $T_{BT\text{-}proxy}$	3250	2977	3801	3250
$T_{BT\text{-}proxy}$ (Eq. 7.2)	3750	3477	4301	3750
Max $T_{BT\text{-}proxy}$	4250	3977	4801	4250

A tolerance range is defined (Table 7-3) around the estimated proxy response in order to define a threshold for accepting or rejecting a reservoir model into the new ensemble. Subsequently, a large number of realizations are generated using *sisim* and *snesim* using the probability of permeability around the wells as secondary data. This large number of realization is processed with the proxy model in order to screen them out criteria set in Table 7-3. As a result, the new ensemble is conditioned to the statistics of the breakthrough in the proxy model, and the probability of permeability around wells.

The model selection process is repeated on the new ensemble using the water cut and pressure profiles as production data simultaneously. Figure 7-19 shows the sequence of the selection process along with the water cut production evaluated for one of the well on each cluster. The figure shows the projection on the space of the top 3 eigenvectors (for visualization purposes), but the cluster analysis actually proceeded in the 5-dimensional space. For that reason, some of the clusters do not show a planar separation and look intermixed. The water production profiles can be discriminated according to each cluster.

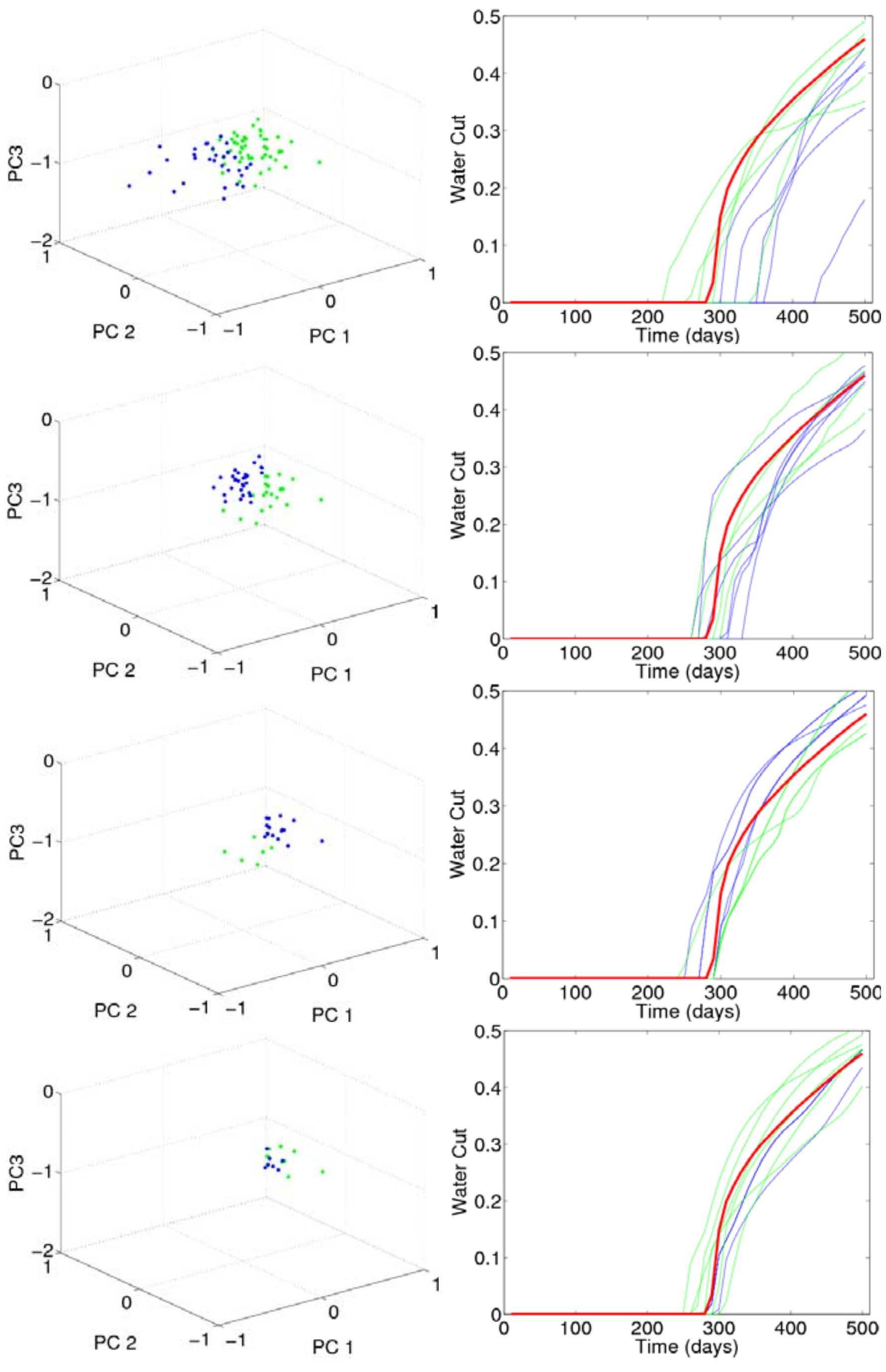

Figure 7-20 Illustration of the model selection process. Projection of reservoir models on top-3 eigenvectors (right) and water cut evaluated for reservoir models selected from the clusters in the right. Production data and clusters are color coded.

At the end of the selection process at this time, the field water production and thereby the oil recovered is well covered by the posterior uncertainty as shown in Figure 7-21. In addition, the production data for the final set was evaluated and compared to the reference as shown in Figure 7-22. A compromise in accuracy is made to conciliate production data in all wells simultaneously. Some reservoir models seem to overlap with members of another cluster, but overall the separation between the groups of models is acceptable

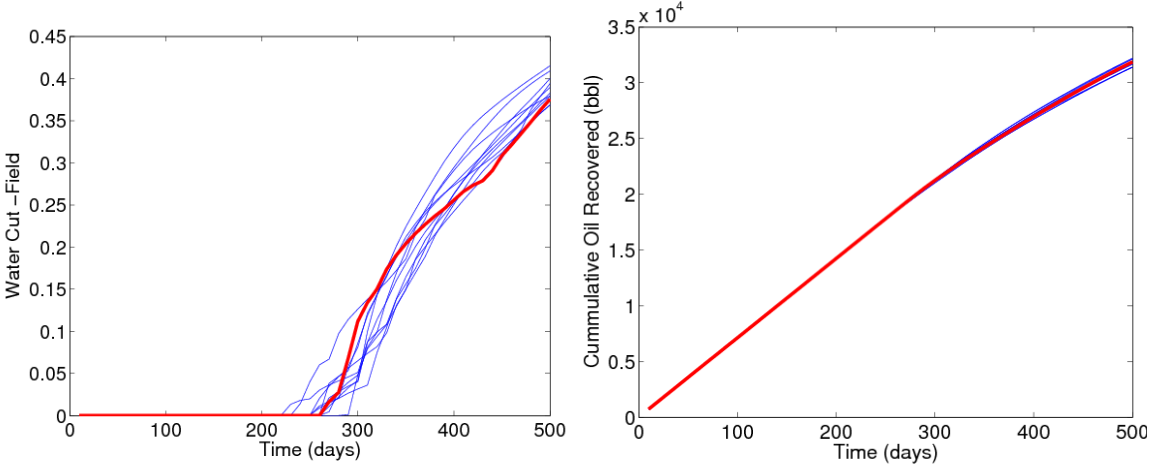

Figure 7-21 Field production data from selected models after 500 days of production is compared to the monitored water cut and cumulative oil production in the true reservoir

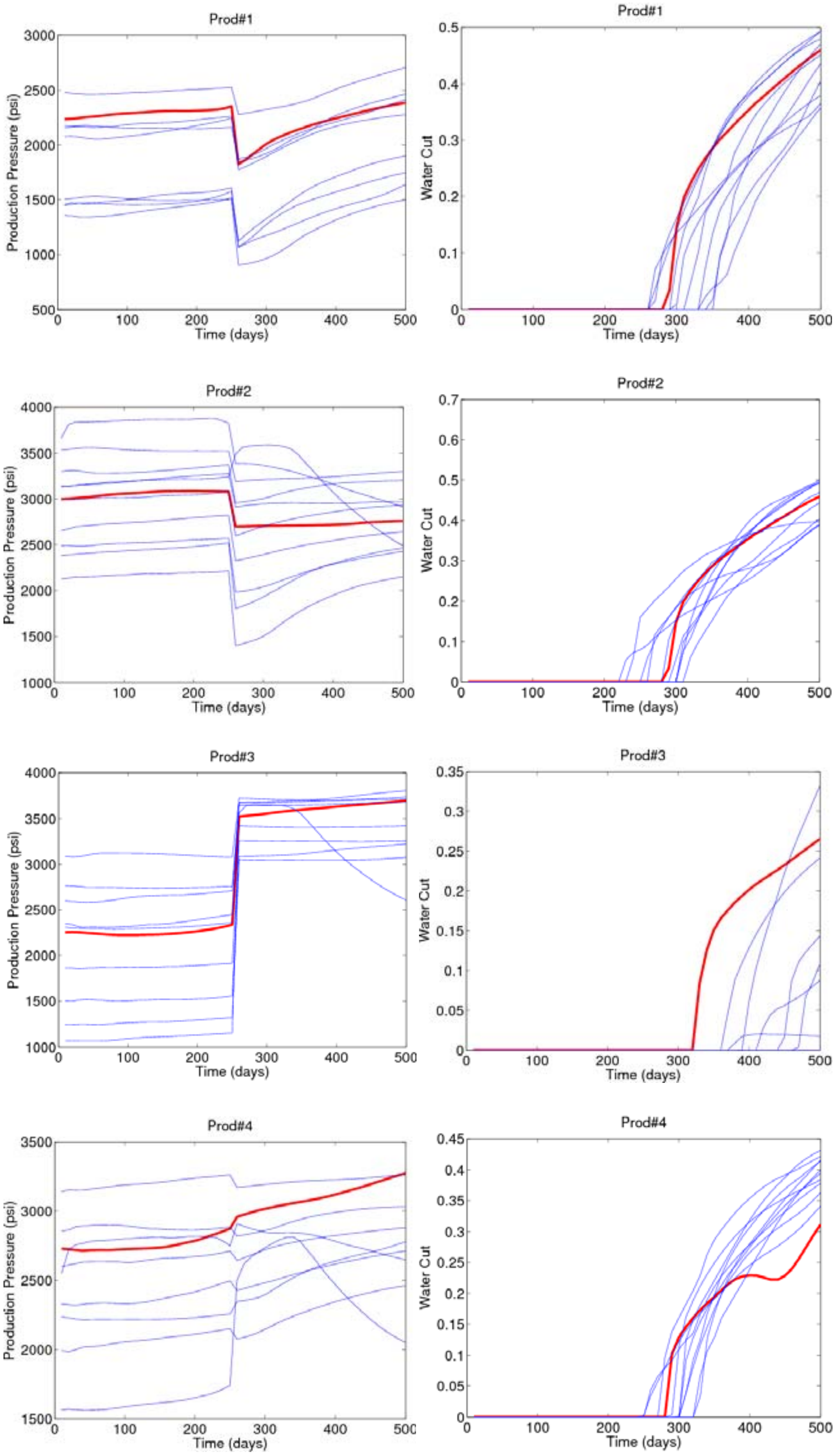

Figure 7-22 Production data simulated by the selected reservoir models selected (blue) is compared to the pressure and water cut observed from the reference (red)

The six reservoir models shown in Figure 7-23 are the closest to the center of the cluster and are presented as representative samples from the final set. It can be observed that there is no a direct connection between the injector and any of the producers. There is still uncertainty in the geologic scenario, but the sinusoidal training image appears more frequently than any other.

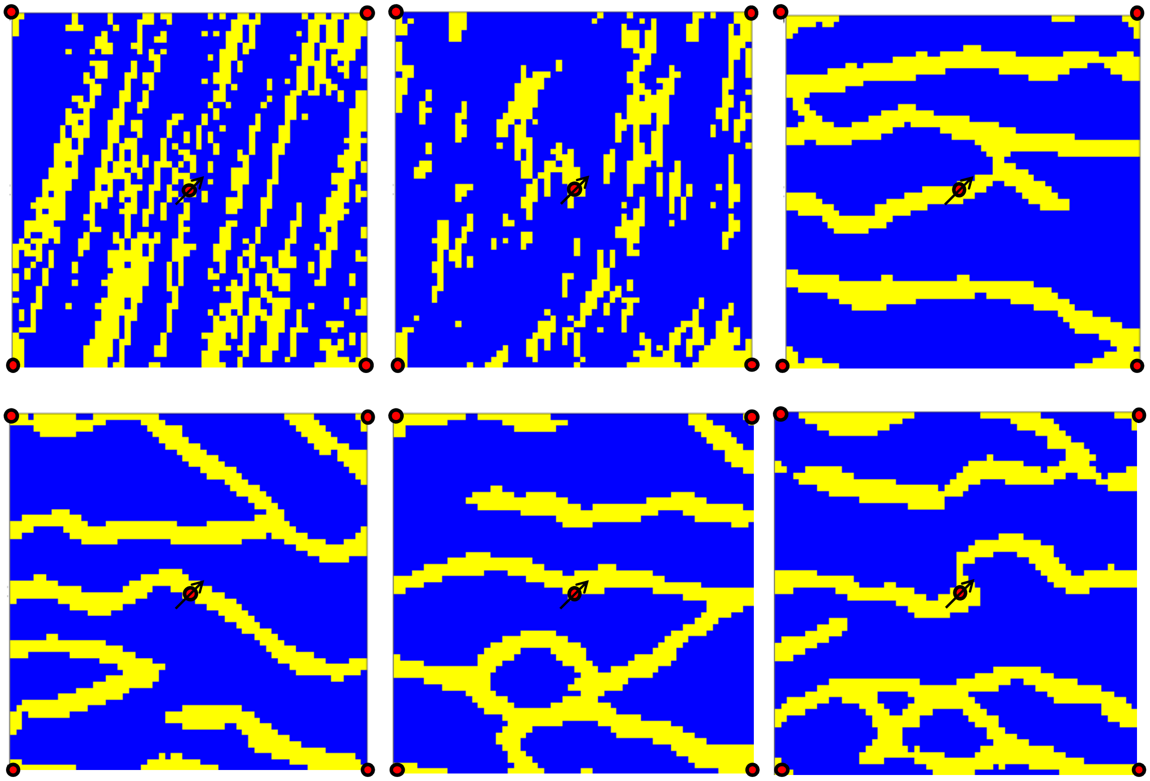

Figure 7-23 Six reservoir models from the selected set after updating with 500 days of production data

The first four reservoir models shown in Figure 7-23 were used to determine the new optimal control settings to continue the polymer flooding operation. Table 7-4 shows the expected value of the injection and production rates used in the next control periods.

Table 7-4 Expected value of optimal control settings for the five-well reservoir, starting at 500 days

Time (days)	Inj#1 (ft^3/day)	Prod#1 (ft^3/day)	Prod#2 (ft^3/day)	Prod#3 (ft^3/day)	Prod#4 (ft^3/day)
500-750	322	75	35	111	101
750-1000	303	76	71	53	103
1000-1250	179	16	47	65	51
1250-1500	140	28	58	15	39
1500-1750	104	23	26	28	27
1750-2000	160	13	62	25	60

The production responses continue being monitored for the rest of the operation until completing 2000 days. The selected reservoir models exhibited production responses consistent with the observations as can be seen in Figure 7-24; thus, no further updates are performed. Finally, the net present value curve obtained with the feedback control scheme is compared to the NPV that would have been obtained if the all producer were operated at constant rate equal to 100 ft^3/day (17.8 STB/day) in Figure 7-26. The difference in final NPV between these schemes is 9%. Therefore, the value of implementing this closed-loop control framework is quantifiable by the economic benefit.

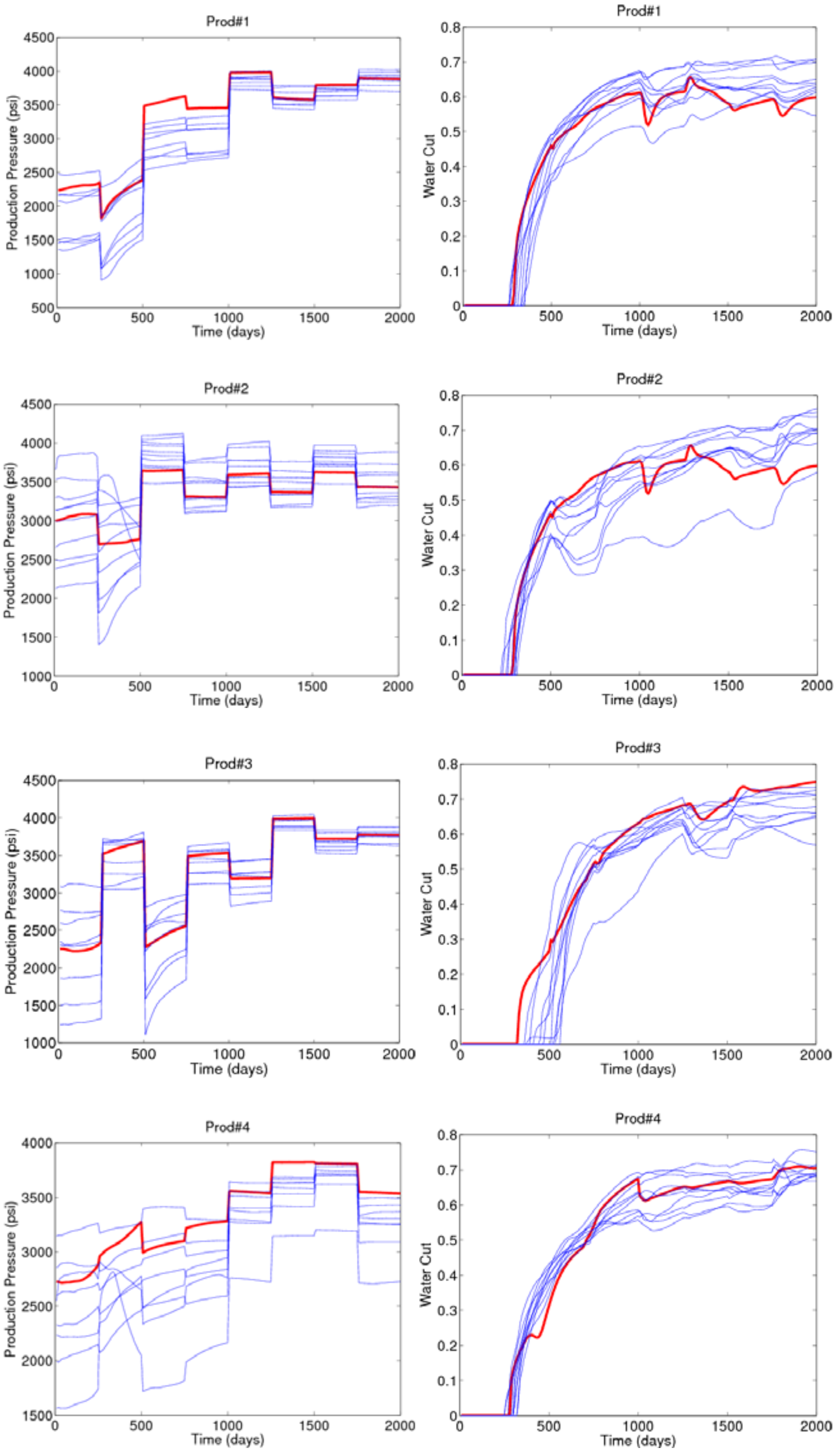

Figure 7-24 Production data simulated by the selected reservoir models selected (blue) is compared to the pressure and water cut observed from the reference (red)

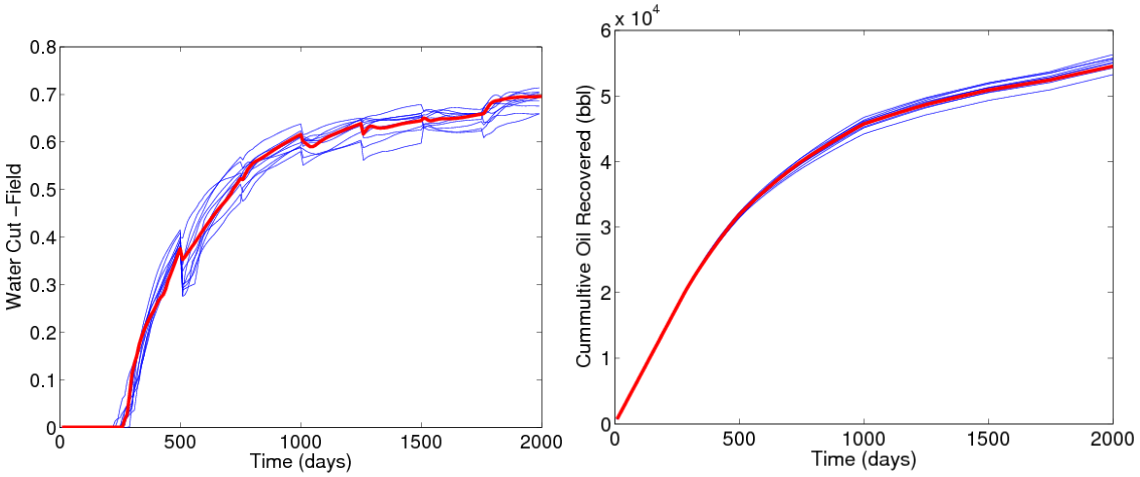

Figure 7-25 Field water cut and cumulative oil produced for selected reservoir models (blue) and reference (red)

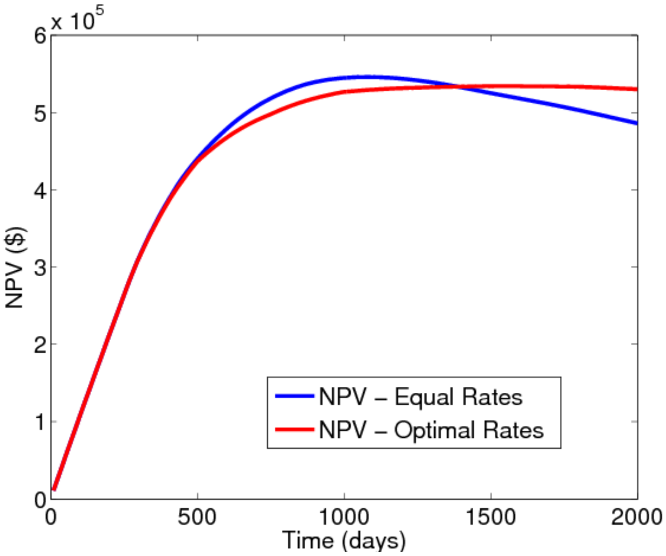

Figure 7-26 Comparison of net present value curves obtained producing 100 ft3/day at all producers and the optimal control settings

7.4 CONCLUSION

The integration of the methods for reservoir model selection and optimization presented in Chapter 5 and 6 completes the feedback control framework for polymer flooding. The control system is demonstrated with the two-well example introduced in Chapter 5. This example indicates that well pressures and water cut profiles at the early stage of the flooding process carry valuable information to refine the prior uncertainty in geologic heterogeneity. The reservoir models in the final selection exhibit similar heterogeneity between injector and producer, allowing a better assessment of the reservoir geology. As a result, the production forecasts from the selected reservoir models are less uncertain and in closer agreement with the field data. Consequently, the optimal control settings are also less uncertain. The value of the feedback control system is readily quantifiable by looking at the improvement in the economics of the project.

Chapter 8: Conclusions and Future Work

8.1 SUMMARY AND GENERAL CONCLUSIONS

A feedback control system for polymer flooding has been developed to optimize well operations under uncertainty in reservoir modeling. The general conclusions of this research work are organized according to the three main objectives laid down at the beginning of this research project.

- To characterize uncertainty in reservoir modeling and forecasting considering multiple geologic scenarios.

To achieve the first objective, the uncertainty in reservoir geology is represented by an ensemble of reservoir models. This uncertainty representation is mapped into a metric space equipped with a measure of dissimilarity between reservoir models that accounts for inter-well connectivity. The uncertainty map facilitates the assessment of uncertainty in past and future production data using a small number of flow simulations. An analysis of the uncertainty map concluded that multiple geologic scenarios can result in similar characteristics of well connectivity, and thereby similar sets of production data. Hence, limiting a history matching or production forecasting process to a single geologic description is a naïve mistake.

- To update the prior uncertainty using past production data such that the uncertainty in production forecasts is refined.

The second objective was achieved based on the hypothesis that similar well connectivity characteristics result in similar production data in polymer flooding. This hypothesis is specific to the reservoir process under study. Since the distance in the uncertainty map measures the dissimilarity in well connectivity, groups of reservoir models distinguished on the basis of distance to each other can be employed to assess the uncertainty in production forecasts. Furthermore, a comparison of simulated production data to the actual reservoir production history allows selecting a set of reservoir models with more accurate production forecasts. As a result, the posterior uncertainty in production forecasting is quantified. The entire methodology leads to a shift in the paradigm of history matching from perturbing a single reservoir model to selecting a set of reservoir models that represent the posterior uncertainty in geology and future production.

- To optimize well operating conditions based on the updated production forecasts in order to maximize the economic profit.

This challenging optimization problem was approached by considering the reservoir as a system that responds according to the input of some control variables. A response surface can be constructed upon evaluating the responses of the reservoir to different control inputs. A random-walk-based proxy model for polymer flooding is a major contribution of this research work because it yields a good estimation of recovery and captures the impact of reservoir heterogeneity on flow performance. This proxy model is utilized to construct the response surface, and a gradient-based procedure is implemented to solve the now simple optimization problem. The entire solution of the optimization problem is effective provided that the uncertainty in geology has been taken into consideration. It is impractical to attempt solving this optimization problem with a deterministic approach by using the classical method of optimal control theory. The added value of this feedback control framework is readily quantifiable by looking at the improved economic profit.

8.2 SPECIFIC CONCLUSIONS BASED ON THE RESEARCH

- Well pressure is well related to the permeability of the grid-blocks near the wells where most of the flow of water and polymer particles occurs. However, this indirect information is only local and it is difficult to extrapolate it to infer large scale heterogeneity features. On the other hand, the water breakthrough is very important to identify inter-well connectivity characteristics that have a great impact on the evolution of water production, which directly affects the economic performance of the project. For that reason, the water production must be carefully monitored and it is recommended to update the models of the reservoir immediately after breakthrough.
- After water breakthrough, even though water production has commenced, there is a chance to control its production by regulating the production rates. In fact, after water breakthrough the operation can be continued at profitable rates before the polymer bank arrives to the producers.
- After updating the reservoir models with production data that includes the water breakthrough and early water production, the production predictions are less uncertain and more consistent with the observations. In addition, the uncertainty in reservoir heterogeneity is reduced to a level where the main characteristics of the reservoir can be detected.
- The optimization scheme of dividing the total operational time into control periods results in improved net present value. This optimization scheme is proactive in the sense that the control settings are based on production forecasts; a reactive control scheme would adjust the rates as a reaction to high water production. Reliable production forecasts for updated reservoir models permit delaying water and polymer production while generating revenue from oil

production. A constant rate scheme is uneconomic after water production commences.

- Response surfaces are able to capture the sensitivity of the net present value with respect to the injection and production rates provided a good transfer function like the proxy model for polymer flooding developed in this research. Therefore, response surface equations are a surrogate to the full flow equations used by reservoir simulators. Hence, production optimization problems in enhanced oil recovery can be solved using response surface equations and well-established optimization algorithms either gradient-based or stochastic with considerably less effort than the rigorous optimal control theory.

8.3 RECOMMENDATIONS FOR FUTURE WORK

The recommendations for future work focus on foreseeable applications of the model selection method developed in this research work.

- The representation of uncertainty in a metric space is valid to any source of uncertainty and any physical model, provided that the measure of dissimilarity chosen (or developed) is correlated to the variable of interest. For example, a geomechanical model with uncertainty in the distribution of natural fractures can be assessed by taking multiple realizations of fracture networks and establishing a measure of dissimilarity related to the observable data.
- Additional control variables for polymer flooding that deserve to be investigated are: polymer viscosity, polymer concentration, slug size, placement of new wells. It is worthwhile to investigate the effect of uncertainty in remaining oil saturation after a secondary recovery to

decide the placement of new wells. However, it is to be noted that addition of new wells or perforation intervals can cause the resultant response surface to be non-smooth, rendering the optimization procedure challenging.

- In reservoir management and decision theory, the probability distribution of production forecasts is often required to make decisions about well placement, production facility sizing or project sanctioning. The uncertainty map constructed by the distance-based method, can be readily adapted to any decision making process.
- The proxy model developed for polymer flooding can be easily generalized to waterflooding or other common displacement processes. In fact, it would be interesting to study the evolution from water flooding to tertiary recovery processes, where the effective permeabilities at the tertiary recovery process have been altered by water flooding.
- In general, the random-walk algorithm is a fast transfer function analog that can be used as an alternative to expensive and time-consuming fine-scale computations, provided that the physics of the process are introduced into the probability rules for particle propagation.
- The optimization algorithm can be improved for accuracy by evaluating the objective function for the initial construction of the response surface, and once the region of optimal conditions has been identified, perform the functions evaluations using the reservoir simulator. In that way, the proxy model is used to guide the search for optimal conditions, the accurate solution is calculated using a full-physics model.

- With regards to optimization, it is worthwhile to introduce the uncertainty in reservoir heterogeneity within the optimization procedure itself. A possible avenue is to compute the distribution of the velocity vectors using the reduced set of models, and guide the random particles by sampling that distribution. Since the velocity vectors are a direct function of local reservoir heterogeneity, the uncertainty in reservoir heterogeneity would be introduced inside the random walk algorithm.
- Sensitivity studies about the quality of model selection with respect to the production data used to update the selection process would contribute to make the selection process more robust and less sensitive to possibly non-meaningful data. Furthermore, it may be necessary to investigate other control variables in polymer flooding like polymer concentration grading or slug size, which can potentially impact the economics of the process.
- The frequency of updating and controlling the reservoir is another topic that deserves further investigation. The ability to respond on time and maintain the controllability of the system depends on the frequency of the control periods. This frequency can be adapted to capture key episodes like water breakthrough while moved at fast pace when no changes in production data occur.

Appendix 1 – UTCHEM Deck Files for Cases Studied

A1-1. TWO-WELL RESERVOIR MODEL

```
CC**************************************************************
CC*                                                            *
CC*     BRIEF DESCRIPTION OF DATA SE : UTCHEM (VERSION 9.9)    *
CC*                                                            *
CC**************************************************************
CC*                                                            *
CC*                                                            *
CC*                                                            *
CC*   LENGTH (FT):              PROCESS :                      *
CC*   THICKNESS (FT):           INJ RATE (FT3/DAY) :           *
CC*   WIDTH (FT):               COORDINATES : CARTESIAN        *
CC*   POROSITY :                                               *
CC*   GRID BLOCKS :                                            *
CC*   DATE :                                                   *
CC*                                                            *
CC**************************************************************
CC*                                                            *
CC**************************************************************
CC*                                                            *
CC*   RESERVOIR DESCRIPTION                                    *
CC*                                                            *
CC**************************************************************
CC
CC Run Number
*----RUNNO
REF_40
CC
CC Title and run description
*----title(i)
POLYMER FLOODING
HETEROGENEOUS CASE
HORIZONTAL DISPLACEMENT
C
CC SIMULATION FLAGS
*---- IMODE     IMES    IDISCP  ICWM    ICAP    IREACT  IBIO    ICOORD  ITREAC  ITC
IGAS    IENG
       1       2      3       0       0      0       0      1       0       0      0
       0       0      0
CC
CC no. of gridblocks,flag specifies constant or variable grid size,unit
*---- NX    NY    NZ  IDXYZ  IUNIT
       40      40     1      0      0
CC
CC constant grid block size in x,y,and z
*---- dx1           dy1           dz1
       20.00   20.00  8.00
CC
CC total no. of components,no. of tracers,no. of gel components
*----n    no    ntw    nta    ngc    ng    noth
       6       0      0       0      0       0      0
CC
CC Name of the components
*----spname(i) for i=1 to n
Water
Oil
Surf
Polymer
Chloride
Sodium
```

```
CC
CC flag indicating if the component is included in calculations or not
*----icf(kc) for kc=1,n
1       1       0       1       1       1
CC
CC**************************************************************
CC*                                                            *
CC*   OUTPUT OPTIONS                                           *
CC*                                                            *
CC**************************************************************
CC
CC
CC FLAG TO WRITE TO UNIT 3,FLAG FOR PV OR DAYS TO PRINT OR TO STOP THE RUN
*---- ICUMTM  ISTOP  IOUTGMS
       0       0       0
CC
CC FLAG INDICATING IF THE PROFILE OF KCTH COMPONENT SHOULD BE WRITTEN
*---- IPRFLG(KC),KC=1,N
1       1       0       1       1       1
CC
CC FLAG FOR PRES.,SAT.,TOTAL CONC.,TRACER CONC.,CAP.,GEL, ALKALINE PROFILES
*---- IPPRES IPSAT IPCTOT IPBIO IPCAP IPGEL IPALK IPTEMP IPOBS
       1       1       1       0       0       0       0       0       0
CC
CC FLAG FOR WRITING SEVERAL PROPERTIES TO UNIT 4 (Prof)
*---- ICKL IVIS IPER ICNM ICSE IHYSTP IFOAMP INONEQ
       1       1       1       0       1       0       0       0
CC
CC FLAG  for variables to PROF output file
*---- IADS IVEL IRKF IPHSE
       1       1       1       0
CC
CC**************************************************************
CC*                                                            *
CC*   RESERVOIR PROPERTIES                                     *
CC*                                                            *
CC**************************************************************
CC
CC
CC MAX. SIMULATION TIME ( DAYS)
*---- TMAX
6000.00
CC
CC ROCK COMPRESSIBILITY (1/PSI), STAND. PRESSURE(PSIA)
*---- COMPR                PSTAND
0.000000E+00   1000.000000
CC
CC FLAGS INDICATING CONSTANT OR VARIABLE POROSITY, X,Y,AND Z PERMEABILITY
*---- IPOR1  IPERMX IPERMY IPERMZ IMOD ITRANZ INTG
       0       2       3       3       0       0       0
CC
CC CONSTANT POROSITY
*---- PORC
0.10
CC
CC VARIABLE PERMX
*---- PERMX(nx*ny*nz)
1148.22
…
30.57
CC
CC Y DIRECTION PERMEABILITY IS DEPENDENT ON X DIRECTION PERMEABILITY
*---- CONSTANT PERMEABILITY MULTIPLIER FOR Y DIRECTION PERMEABILITY
1.00
CC
Z DIRECTION PERMEABILITY IS DEPENDENT ON X DIRECTION PERMEABILITY
```

```
*---- CONSTANT PERMEABILITY MULTIPLIER FOR Z DIRECTION PERMEABILITY
0.10
CC
CC FLAG FOR CONSTANT OR VARIABLE DEPTH, PRESSURE, WATER SATURATION,INITIAL AQUEOUS PHASE
COMPOSITIONS
*----IDEPTH  IPRESS  ISWI  ICWI
        0      0       0      0
CC
CC CONSTANT DEPTH (FT)
*---- D111
3500.00
CC
CC CONSTANT PRESSURE (PSIA)
*---- PRESS1
2500.00
CC
CC CONSTANT INITIAL WATER SATURATION
*---- SWIC
0.20
CC
CC INITIAL CONCENTRATION OF SPECIES IN THE AQUEOUS PHASE
*---- CWI(KW) FOR KW=1,N(8+NO)
1.00
CC
CC INITIAL CONCENTRATION OF SPECIES IN THE AQUEOUS PHASE
*---- CWI(KW) FOR KW=1,N(8+NO)
0.00
CC
CC INITIAL CONCENTRATION OF SPECIES IN THE AQUEOUS PHASE
*---- CWI(KW) FOR KW=1,N(8+NO)
0.00
CC
CC INITIAL CONCENTRATION OF SPECIES IN THE AQUEOUS PHASE
*---- CWI(KW) FOR KW=1,N(8+NO)
0.00
CC
CC INITIAL CONCENTRATION OF SPECIES IN THE AQUEOUS PHASE
*---- CWI(KW) FOR KW=1,N(8+NO)
0.00
CC
CC**************************************************************
CC*                                                            *
CC*   PHYSICAL PROPERTY DATA                                   *
CC*                                                            *
CC**************************************************************
CC
CC
CC OIL CONC. AT PLAIT POINT FOR TYPE II(+)AND TYPE II(-), CMC
*---- c2plc  c2prc   epsme   ihand
0.000000       1.000000       0.000600       0
CC
CC flag indicating type of phase behavior parameters
*---- ifghbn
        0
CC SLOPE AND INTERCEPT OF BINODAL CURVE AT ZERO, OPT., AND 2XOPT SALINITY
CC FOR ALCOHOL 1
*---- hbns70   hbnc70   hbns71   hbnc71   hbns72   hbnc72
0.000000       0.010000       0.000000       0.016000       0.000000       0.100000
CC SLOPE AND INTERCEPT OF BINODAL CURVE AT ZERO, OPT., AND 2XOPT SALINITY
CC FOR ALCOHOL 2
*---- hbns80   hbnc80   hbns81   hbnc81   hbns82   hbnc82
0.000000       0.000000       0.000000       0.000000       0.000000       0.000000
CC
CC LOWER AND UPPER EFFECTIVE SALINITY FOR ALCOHOL 1 AND ALCOHOL 2
*---- csel7   cseu7   csel8   cseu8
0.550000       0.916000       0.000000       0.000000
```

```
CC
CC THE CSE SLOPE PARAMETER FOR CALCIUM AND ALCOHOL 1 AND ALCOHOL 2
*---- beta6    beta7    beta8
0.800000       -2.000000      0.000000
CC
CC FLAG FOR ALCOHOL PART. MODEL AND PARTITION COEFFICIENTS
*---- ialc   opsk7o   opsk7s   opsk8o   opsk8s
0       0.000000       0.000000       0.000000       0.000000
CC
CC NO. OF ITERATIONS, AND TOLERANCE
*---- nalmax     epsalc
20.000000      0.000100
CC
CC ALCOHOL 1 PARTITIONING PARAMETERS IF IALC=1
*---- akwc7     akws7    akm7     ak7      pt7
4.671000       1.790000       48.000000      35.310000      0.222000
CC
CC ALCOHOL 1 PARTITIONING PARAMETERS IF IALC=2
*---- akwc8     akws8    akm8     ak8      pt8
0.000000       0.000000       0.000000       0.000000       0.000000
CC
CC ift model flag
*----  ift
       0
CC
CC INTERFACIAL TENSION PARAMETERS
*----  g11     g12     g13     g21     g22      g23
       13.000000      -14.800000     0.007000       13.000000      -14.500000
       0.010000
CC
CC LOG10 OF OIL/WATER INTERFACIAL TENSION
*---- xiftw
       1.300000
CC
CC ORGANIC MASS TRANSFER FLAG
*---- imass icor
       0      0
CC
CC CAPILLARY DESATURATION PARAMETERS FOR PHASE 1, 2, AND 3
*---- itrap      t11      t22      t33
       1      0.000000       0.000000       364.200000
CC
CC  FLAG FOR RELATIVE PERMEABILITY AND CAPILLARY PRESSURE MODEL
*---- iperm irtype
       0      0
CC
CC FLAG FOR CONSTANT OR VARIABLE REL. PERM. PARAMETERS
*---- isrw    iprw    iew
       0      0      0
CC
CC CONSTANT RES. SATURATION OF PHASES 1,2,AND 3 AT LOW CAPILLARY NO.
*---- s1rwc    s2rwc     s3rwc
       0.200000       0.200000       0.370000
CC
CC ENDPOINT REL. PERM. OF PHASES 1,2,AND 3 AT LOW CAPILLARY NO.
*---- p1rwc   p2rwc   p3rwc
       0.200000       1.000000       0.200000
CC
CONSTANT REL. PERM. EXPONENT OF PHASES 1,2,AND 3 AT LOW CAPILLARY NO.
*---- e1wc     e2wc     e3wc
       1.500000       2.000000       1.500000
CC
CC RES. SATURATION OF PHASES 1,2,AND 3 AT HIGH CAPILLARY NO.
*---- s1rc   s2rc   s3rc
       0.000000       0.000000       0.000000
CC
```

```
CC ENDPOINT REL. PERM. OF PHASES 1,2,AND 3 AT HIGH CAPILLARY NO.
*---- p1rc   p2rc   p3rc
       1.000000        1.000000       0.000000
CC
CC REL. PERM. EXPONENT OF PHASES 1,2,AND 3 AT HIGH CAPILLARY NO.
*---- e13c   e23c   e31c
       1.500000        2.000000       0.000000
CC
CC WATER AND OIL VISCOSITY , RESERVOIR TEMPERATURE
*---- VIS1    VIS2   TSTAND
       0.730000        40.000000      0.000000
CC
CC COMPOSITIONAL PHASE VISCOSITY PARAMETERS
*----   ALPHAV1   ALPHAV2   ALPHAV3   ALPHAV4  ALPHAV5
       2.000000        2.000000       0.000000        0.900000       0.700000
CC
CC PARAMETERS TO CALCULATE POLYMER VISCOSITY AT ZERO SHEAR RATE
*---- AP1      AP2      AP3
       35.140000       1899.000000    0.000000
CC
CC PARAMETER TO COMPUTE CSEP,MIN. CSEP, AND SLOPE OF LOG VIS. VS. LOG CSEP
*---- BETAP    CSE1     SSLOPE
       20.000000       0.010000       -0.300000
CC
CC PARAMETER FOR SHEAR RATE DEPENDENCE OF POLYMER VISCOSITY
*---- GAMMAC   GAMHF   POWN   IPMOD
       130.000000      280.000000     2.200000         1
CC
CC FLAG FOR POLYMER PARTITIONING, PERM. REDUCTION PARAMETERS
*---- IPOLYM    EPHI3    EPHI4    BRK     CRK        RKCUT
       0       1.000000        0.850000       4.000000        0.150000       10.000000
CC
CC SPECIFIC WEIGHT FOR COMPONENTS 1,2,3,7,8 ,Coeffient of oil and GRAVITY FLAG
*---- DEN1     DEN2    DEN23     DEN3    DEN7    DEN8    IDEN
       0.433530        0.385840       0.368000        0.420000       0.346000
       0.000000        2
CC
CC FLAG FOR CHOICE OF UNITS ( 0:BOTTOMHOLE CONDITION , 1: STOCK TANK)
*----- ISTB
       0
CC
CC COMPRESSIBILITY FOR VOL. OCCUPYING COMPONENTS 1,2,3,7,AND 8
*---- COMPC(1)  COMPC(2)  COMPC(3)  COMPC(7)  COMPC(8)
       0.00000010      0.00000010     0.00000000      0.00000000     0.00000000
CC
CC CONSTANT OR VARIABLE PC PARAM., WATER-WET OR OIL-WET PC CURVE FLAG
*---- ICPC    IEPC   IOW
       0       0       0
CC
CC CAPILLARY PRESSURE PARAMETER, CPC0
*---- CPC0
       9.000000
CC
CC CAPILLARY PRESSURE PARAMETER, EPC0
*---- EPC0
       2.000000
CC
CC MOLECULAR DIFFUSION COEF. KCTH COMPONENT IN PHASE 1
*---- D(KC,1),KC=1,N
       0.000000        0.000000       0.000000        0.000000       0.000000
       0.000000
CC
CC MOLECULAR DIFFUSION COEF. KCTH COMPONENT IN PHASE 2
*---- D(KC,2),KC=1,N
       0.000000        0.000000       0.000000        0.000000       0.000000
       0.000000
```

```
CC
CC MOLECULAR DIFFUSION COEF. KCTH COMPONENT IN PHASE 3
*---- D(KC,3),KC=1,N
       0.000000       0.000000       0.000000       0.000000       0.000000
       0.000000
CC
LONGITUDINAL AND TRANSVERSE DISPERSIVITY OF PHASE 1
*---- ALPHAL(1)     ALPHAT(1)
       0.160000       0.040000
CC
LONGITUDINAL AND TRANSVERSE DISPERSIVITY OF PHASE 2
*---- ALPHAL(2)     ALPHAT(2)
       0.160000       0.040000
CC
LONGITUDINAL AND TRANSVERSE DISPERSIVITY OF PHASE 3
*---- ALPHAL(3)     ALPHAT(3)
       0.160000       0.040000
CC
CC flag to specify organic adsorption calculation
*---- iadso
       0
CC
CC SURFACTANT AND POLYMER ADSORPTION PARAMETERS
*---- AD31    AD32   B3D    AD41   AD42   B4D   IADK  IADS1   FADS   REFK
       1.500000       0.500000       1000.000000    0.800000       0.000000
       300.000000     1      0       0      100.000000
CC
CC PARAMETERS FOR CATION EXCHANGE OF CLAY AND SURFACTANT
*---- QV      XKC     XKS     EQW
       0.000000       0.000000       0.000000       804.000000
CC
CC**************************************************************
CC*                                                            *
CC*   WELL DATA                                                *
CC*                                                            *
CC**************************************************************
CC
CC
CC FLAG FOR SPECIFIED BOUNDARY AND ZONE IS MODELED
*---- IBOUND     IZONE
       0      0
CC
CC TOTAL NUMBER OF WELLS, WELL RADIUS FLAG, FLAG FOR TIME OR COURANT NO.
*---- NWELL   IRO    ITIME    NWREL
       2      2      1      2
CC
CC WELL ID,LOCATIONS,AND FLAG FOR SPECIFYING WELL TYPE, WELL RADIUS, SKIN
*---- IDW    IW    JW    IFLAG    RW     SWELL   IDIR   IFIRST   ILAST   IPRF
       1      20     1      1      0.250000       -1.000000      3      1      1
       0
CC
CC WELL NAME
*----  WELNAM
       INJ-1
CC
CC ICHEK , MAX. AND MIN. ALLOWABLE BOTTOMHOLE PRESSURE AND RATE
*---- ICHEK     PWFMIN     PWFMAX    QTMIN    QTMAX
       1      200.000000     9251.000000    0.000000       4000.000000
CC
CC WELL ID,LOCATIONS,AND FLAG FOR SPECIFYING WELL TYPE, WELL RADIUS, SKIN
*---- IDW    IW    JW    IFLAG    RW     SWELL   IDIR   IFIRST   ILAST   IPRF
       2      20     40     4      0.250000       0.000000       3      1      1
       0
CC
CC WELL NAME
*----  WELNAM
```

```
       PROD-1
CC
CC ICHEK , MAX. AND MIN. ALLOWABLE BOTTOMHOLE PRESSURE AND RATE
*---- ICHEK     PWFMIN     PWFMAX    QTMIN    QTMAX
       1       200.000000     9251.000000    0.000000       2807.000000
CC
CC  ID,INJ. RATE AND INJ. COMP. FOR RATE CONS. WELLS FOR EACH PHASE (L=1,3)
*----  ID     QI(M,L)    C(M,KC,L)
       1       50.000000      1.000000       0.000000       0.000000       0.075000
       0.006096       0.004075
       1       0.000000       0.000000       0.000000       0.000000       0.000000
       0.000000       0.000000
       1       0.000000       0.000000       0.000000       0.000000       0.000000
       0.000000       0.000000
CC
CC  ID, TOTAL PRODUCTION RATE
*----  ID    QI(M,1)
       2       -50.000000
CC
CC CUM. INJ. TIME , AND INTERVALS (PV OR DAY) FOR WRITING TO OUTPUT FILES
*---- TINJ     CUMPR1     CUMHI1     WRHPV     WRPRF      RSTC
       6000.000000    30.000000      30.000000      10.000000      30.000000
       60.000000
CC
CC FOR IMES=2 ,THE INI. TIME STEP,CONC. TOLERANCE,MAX.,MIN. courant numbers
*----  DT      DCLIM     CNMAX      CNMIN
       0.100000       0.003000       0.200000       0.010000
```

A1-2. FIVE-WELL RESERVOIR MODEL

```
CC**************************************************************
CC*                                                            *
CC*     BRIEF DESCRIPTION OF DATA SE : UTCHEM (VERSION 9.9)    *
CC*                                                            *
CC**************************************************************
CC*                                                            *
CC*                                                            *
CC*                                                            *
CC*   LENGTH (FT):              PROCESS :                      *
CC*   THICKNESS (FT):           INJ RATE (FT3/DAY) :           *
CC*   WIDTH (FT):               COORDINATES : CARTESIAN        *
CC*   POROSITY :                                               *
CC*   GRID BLOCKS :                                            *
CC*   DATE :                                                   *
CC*                                                            *
CC**************************************************************
CC*                                                            *
CC**************************************************************
CC*                                                            *
CC*   RESERVOIR DESCRIPTION                                    *
CC*                                                            *
CC**************************************************************
CC
CC Run Number
*----RUNNO
REF_60
CC
CC Title and run description
*----title(i)
POLYMER FLOODING
HETEROGENEOUS CASE
HORIZONTAL DISPLACEMENT
C
```

```
CC SIMULATION FLAGS
*---- IMODE     IMES    IDISCP  ICWM    ICAP    IREACT  IBIO    ICOORD  ITREAC  ITC
IGAS    IENG
       1       2       3      0       0      0       0      1       0      0       0
       0       0       0
CC
CC no. of gridblocks,flag specifies constant or variable grid size,unit
*---- NX    NY    NZ  IDXYZ  IUNIT
       60      60      1      0       0
CC
CC constant grid block size in x,y,and z
*---- dx1           dy1           dz1
       20.00   20.00  8.00
CC
CC total no. of components,no. of tracers,no. of gel components
*----n    no    ntw    nta    ngc    ng    noth
       6       0       0      0       0      0       0
CC
CC Name of the components
*----spname(i) for i=1 to n
Water
Oil
Surf
Polymer
Chloride
Sodium
CC
CC flag indicating if the component is included in calculations or not
*----icf(kc) for kc=1,n
1      1       0      1       1      1
CC
CC***************************************************************
CC*                                                             *
CC*   OUTPUT OPTIONS                                            *
CC*                                                             *
CC***************************************************************
CC
CC
CC FLAG TO WRITE TO UNIT 3,FLAG FOR PV OR DAYS TO PRINT OR TO STOP THE RUN
*---- ICUMTM  ISTOP  IOUTGMS
       0       0       0
CC
CC FLAG INDICATING IF THE PROFILE OF KCTH COMPONENT SHOULD BE WRITTEN
*---- IPRFLG(KC),KC=1,N
1      1       0      1       1      1
CC
CC FLAG FOR PRES.,SAT.,TOTAL CONC.,TRACER CONC.,CAP.,GEL, ALKALINE PROFILES
*---- IPPRES IPSAT IPCTOT IPBIO IPCAP IPGEL IPALK IPTEMP IPOBS
       1       1       1      0       0      0       0      0       0
CC
CC FLAG FOR WRITING SEVERAL PROPERTIES TO UNIT 4 (Prof)
*---- ICKL IVIS IPER ICNM ICSE IHYSTP IFOAMP INONEQ
       1       1       1      0       1      0       0      0
CC
CC FLAG  for variables to PROF output file
*---- IADS IVEL IRKF IPHSE
       1       1       1      0
CC
CC***************************************************************
CC*                                                             *
CC*   RESERVOIR PROPERTIES                                      *
CC*                                                             *
CC***************************************************************
CC
CC
CC MAX. SIMULATION TIME ( DAYS)
```

```
*---- TMAX
2000.00
CC
CC ROCK COMPRESSIBILITY (1/PSI), STAND. PRESSURE(PSIA)
*---- COMPR                PSTAND
0.000000E+00   1000.000000
CC
CC FLAGS INDICATING CONSTANT OR VARIABLE POROSITY, X,Y,AND Z PERMEABILITY
*---- IPOR1  IPERMX IPERMY IPERMZ IMOD ITRANZ INTG
       0      2      3      3      0      0      0
CC
CC CONSTANT POROSITY
*---- PORC
0.10
CC
CC VARIABLE PERMX
*---- PERMX(nx*ny*nz)
885.00
…
CC
CC Y DIRECTION PERMEABILITY IS DEPENDENT ON X DIRECTION PERMEABILITY
*---- CONSTANT PERMEABILITY MULTIPLIER FOR Y DIRECTION PERMEABILITY
1.00
CC
Z DIRECTION PERMEABILITY IS DEPENDENT ON X DIRECTION PERMEABILITY
*---- CONSTANT PERMEABILITY MULTIPLIER FOR Z DIRECTION PERMEABILITY
0.10
CC
CC FLAG FOR CONSTANT OR VARIABLE DEPTH, PRESSURE, WATER SATURATION,INITIAL AQUEOUS PHASE
COMPOSITIONS
*----IDEPTH  IPRESS  ISWI  ICWI
       0      0      0      0
CC
CC CONSTANT DEPTH (FT)
*---- D111
3500.00
CC
CC CONSTANT PRESSURE (PSIA)
*---- PRESS1
2500.00
CC
CC CONSTANT INITIAL WATER SATURATION
*---- SWIC
0.20
CC
CC INITIAL CONCENTRATION OF SPECIES IN THE AQUEOUS PHASE
*---- CWI(KW) FOR KW=1,N(8+NO)
1.00
CC
CC INITIAL CONCENTRATION OF SPECIES IN THE AQUEOUS PHASE
*---- CWI(KW) FOR KW=1,N(8+NO)
0.00
CC
CC INITIAL CONCENTRATION OF SPECIES IN THE AQUEOUS PHASE
*---- CWI(KW) FOR KW=1,N(8+NO)
0.00
CC
CC INITIAL CONCENTRATION OF SPECIES IN THE AQUEOUS PHASE
*---- CWI(KW) FOR KW=1,N(8+NO)
0.00
CC
CC INITIAL CONCENTRATION OF SPECIES IN THE AQUEOUS PHASE
*---- CWI(KW) FOR KW=1,N(8+NO)
0.00
CC
CC****************************************************************
```

```
CC*                                                            *
CC*   PHYSICAL PROPERTY DATA                                   *
CC*                                                            *
CC**************************************************************
CC
CC
CC OIL CONC. AT PLAIT POINT FOR TYPE II(+)AND TYPE II(-), CMC
*---- c2plc  c2prc   epsme   ihand
0.000000       1.000000       0.000600       0
CC
CC flag indicating type of phase behavior parameters
*---- ifghbn
       0
CC SLOPE AND INTERCEPT OF BINODAL CURVE AT ZERO, OPT., AND 2XOPT SALINITY
CC FOR ALCOHOL 1
*---- hbns70   hbnc70   hbns71   hbnc71   hbns72   hbnc72
0.000000       0.010000       0.000000       0.016000       0.000000       0.100000
CC SLOPE AND INTERCEPT OF BINODAL CURVE AT ZERO, OPT., AND 2XOPT SALINITY
CC FOR ALCOHOL 2
*---- hbns80   hbnc80   hbns81   hbnc81   hbns82   hbnc82
0.000000       0.000000       0.000000       0.000000       0.000000       0.000000
CC
CC LOWER AND UPPER EFFECTIVE SALINITY FOR ALCOHOL 1 AND ALCOHOL 2
*---- csel7   cseu7   csel8   cseu8
0.550000       0.916000       0.000000       0.000000
CC
CC THE CSE SLOPE PARAMETER FOR CALCIUM AND ALCOHOL 1 AND ALCOHOL 2
*---- beta6    beta7    beta8
0.800000       -2.000000      0.000000
CC
CC FLAG FOR ALCOHOL PART. MODEL AND PARTITION COEFFICIENTS
*---- ialc   opsk7o   opsk7s   opsk8o   opsk8s
0      0.000000       0.000000       0.000000       0.000000
CC
CC NO. OF ITERATIONS, AND TOLERANCE
*---- nalmax     epsalc
20.000000      0.000100
CC
CC ALCOHOL 1 PARTITIONING PARAMETERS IF IALC=1
*---- akwc7     akws7    akm7     ak7      pt7
4.671000       1.790000       48.000000      35.310000      0.222000
CC
CC ALCOHOL 1 PARTITIONING PARAMETERS IF IALC=2
*---- akwc8     akws8    akm8     ak8      pt8
0.000000       0.000000       0.000000       0.000000       0.000000
CC
CC ift model flag
*----  ift
       0
CC
CC INTERFACIAL TENSION PARAMETERS
*----  g11     g12     g13     g21     g22      g23
       13.000000      -14.800000     0.007000       13.000000      -14.500000
       0.010000
CC
CC LOG10 OF OIL/WATER INTERFACIAL TENSION
*---- xiftw
       1.300000
CC
CC ORGANIC MASS TRANSFER FLAG
*---- imass icor
       0       0
CC
CC CAPILLARY DESATURATION PARAMETERS FOR PHASE 1, 2, AND 3
*---- itrap      t11      t22      t33
       1       0.000000       0.000000       364.200000
```

```
CC
CC  FLAG FOR RELATIVE PERMEABILITY AND CAPILLARY PRESSURE MODEL
*---- iperm irtype
       0       0
CC
CC FLAG FOR CONSTANT OR VARIABLE REL. PERM. PARAMETERS
*---- isrw    iprw    iew
       0       0      0
CC
CC CONSTANT RES. SATURATION OF PHASES 1,2,AND 3 AT LOW CAPILLARY NO.
*---- s1rwc    s2rwc     s3rwc
       0.200000       0.200000       0.370000
CC
CC ENDPOINT REL. PERM. OF PHASES 1,2,AND 3 AT LOW CAPILLARY NO.
*---- p1rwc   p2rwc   p3rwc
       0.200000       1.000000       0.200000
CC
CONSTANT REL. PERM. EXPONENT OF PHASES 1,2,AND 3 AT LOW CAPILLARY NO.
*---- e1wc     e2wc     e3wc
       1.500000       2.000000       1.500000
CC
CC RES. SATURATION OF PHASES 1,2,AND 3 AT HIGH CAPILLARY NO.
*---- s1rc   s2rc   s3rc
       0.000000       0.000000       0.000000
CC
CC ENDPOINT REL. PERM. OF PHASES 1,2,AND 3 AT HIGH CAPILLARY NO.
*---- p1rc   p2rc   p3rc
       1.000000       1.000000       0.000000
CC
CC REL. PERM. EXPONENT OF PHASES 1,2,AND 3 AT HIGH CAPILLARY NO.
*---- e13c   e23c   e31c
       1.500000       2.000000       0.000000
CC
CC WATER AND OIL VISCOSITY , RESERVOIR TEMPERATURE
*---- VIS1    VIS2   TSTAND
       0.730000       40.000000      0.000000
CC
CC COMPOSITIONAL PHASE VISCOSITY PARAMETERS
*----   ALPHAV1   ALPHAV2   ALPHAV3   ALPHAV4  ALPHAV5
       2.000000       2.000000       0.000000       0.900000       0.700000
CC
CC PARAMETERS TO CALCULATE POLYMER VISCOSITY AT ZERO SHEAR RATE
*---- AP1      AP2      AP3
       35.140000      1899.000000    0.000000
CC
CC PARAMETER TO COMPUTE CSEP,MIN. CSEP, AND SLOPE OF LOG VIS. VS. LOG CSEP
*---- BETAP    CSE1     SSLOPE
       20.000000      0.010000       -0.300000
CC
CC PARAMETER FOR SHEAR RATE DEPENDENCE OF POLYMER VISCOSITY
*---- GAMMAC   GAMHF   POWN   IPMOD
       130.000000     280.000000     2.200000       1
CC
CC FLAG FOR POLYMER PARTITIONING, PERM. REDUCTION PARAMETERS
*---- IpOLYM    EPHI3    EPHI4    BRK     CRK        RKCUT
       0      1.000000       0.850000       4.000000       0.150000       10.000000
CC
CC SPECIFIC WEIGHT FOR COMPONENTS 1,2,3,7,8 ,Coeffient of oil and GRAVITY FLAG
*---- DEN1     DEN2    DEN23     DEN3    DEN7    DEN8    IDEN
       0.433530       0.385840       0.368000       0.420000       0.346000
       0.000000       2
CC
CC FLAG FOR CHOICE OF UNITS ( 0:BOTTOMHOLE CONDITION , 1: STOCK TANK)
*----- ISTB
       0
CC
```

```
CC COMPRESSIBILITY FOR VOL. OCCUPYING COMPONENTS 1,2,3,7,AND 8
*---- COMPC(1)  COMPC(2)  COMPC(3)  COMPC(7)  COMPC(8)
       0.00000010     0.00000010     0.00000000     0.00000000     0.00000000
CC
CC CONSTANT OR VARIABLE PC PARAM., WATER-WET OR OIL-WET PC CURVE FLAG
*---- ICPC    IEPC   IOW
       0      0      0
CC
CC CAPILLARY PRESSURE PARAMETER, CPC0
*---- CPC0
       9.000000
CC
CC CAPILLARY PRESSURE PARAMETER, EPC0
*---- EPC0
       2.000000
CC
CC MOLECULAR DIFFUSION COEF. KCTH COMPONENT IN PHASE 1
*---- D(KC,1),KC=1,N
       0.000000       0.000000       0.000000       0.000000       0.000000
       0.000000
CC
CC MOLECULAR DIFFUSION COEF. KCTH COMPONENT IN PHASE 2
*---- D(KC,2),KC=1,N
       0.000000       0.000000       0.000000       0.000000       0.000000
       0.000000
CC
CC MOLECULAR DIFFUSION COEF. KCTH COMPONENT IN PHASE 3
*---- D(KC,3),KC=1,N
       0.000000       0.000000       0.000000       0.000000       0.000000
       0.000000
CC
LONGITUDINAL AND TRANSVERSE DISPERSIVITY OF PHASE 1
*---- ALPHAL(1)     ALPHAT(1)
       0.160000       0.040000
CC
LONGITUDINAL AND TRANSVERSE DISPERSIVITY OF PHASE 2
*---- ALPHAL(2)     ALPHAT(2)
       0.160000       0.040000
CC
LONGITUDINAL AND TRANSVERSE DISPERSIVITY OF PHASE 3
*---- ALPHAL(3)     ALPHAT(3)
       0.160000       0.040000
CC
CC flag to specify organic adsorption calculation
*---- iadso
       0
CC
CC SURFACTANT AND POLYMER ADSORPTION PARAMETERS
*---- AD31    AD32   B3D    AD41   AD42   B4D   IADK  IADS1   FADS   REFK
       1.500000       0.500000       1000.000000    0.800000       0.000000
       300.000000     1      0      0      100.000000
CC
CC PARAMETERS FOR CATION EXCHANGE OF CLAY AND SURFACTANT
*---- QV      XKC     XKS     EQW
       0.000000       0.000000       0.000000       804.000000
CC
CC***************************************************************
CC*                                                             *
CC*   WELL DATA                                                 *
CC*                                                             *
CC***************************************************************
CC
CC
CC FLAG FOR SPECIFIED BOUNDARY AND ZONE IS MODELED
*---- IBOUND     IZONE
       0      0
```

```
CC
CC TOTAL NUMBER OF WELLS, WELL RADIUS FLAG, FLAG FOR TIME OR COURANT NO.
*---- NWELL   IRO    ITIME    NWREL
       5       2       1       5
CC
CC WELL ID,LOCATIONS,AND FLAG FOR SPECIFYING WELL TYPE, WELL RADIUS, SKIN
*---- IDW    IW    JW    IFLAG    RW     SWELL   IDIR   IFIRST   ILAST   IPRF
       1       30      30      1      0.250000       -1.000000      3       1       1
       0
CC
CC WELL NAME
*----  WELNAM
       INJ-1
CC
CC ICHEK , MAX. AND MIN. ALLOWABLE BOTTOMHOLE PRESSURE AND RATE
*---- ICHEK     PWFMIN     PWFMAX    QTMIN    QTMAX
       1       200.000000     9251.000000    0.000000       4000.000000
CC
CC WELL ID,LOCATIONS,AND FLAG FOR SPECIFYING WELL TYPE, WELL RADIUS, SKIN
*---- IDW    IW    JW    IFLAG    RW     SWELL   IDIR   IFIRST   ILAST   IPRF
       2       1       1       4      0.250000       0.000000       3       1       1
       0
CC
CC WELL NAME
*----  WELNAM
       PROD-1
CC
CC ICHEK , MAX. AND MIN. ALLOWABLE BOTTOMHOLE PRESSURE AND RATE
*---- ICHEK     PWFMIN     PWFMAX    QTMIN    QTMAX
       1       200.000000     9251.000000    0.000000       2807.000000
CC
CC WELL ID,LOCATIONS,AND FLAG FOR SPECIFYING WELL TYPE, WELL RADIUS, SKIN
*---- IDW    IW    JW    IFLAG    RW     SWELL   IDIR   IFIRST   ILAST   IPRF
       3       60      1       4      0.250000       0.000000       3       1       1
       0
CC
CC WELL NAME
*----  WELNAM
       PROD-2
CC
CC ICHEK , MAX. AND MIN. ALLOWABLE BOTTOMHOLE PRESSURE AND RATE
*---- ICHEK     PWFMIN     PWFMAX    QTMIN    QTMAX
       1       200.000000     9251.000000    0.000000       2807.000000
CC
CC WELL ID,LOCATIONS,AND FLAG FOR SPECIFYING WELL TYPE, WELL RADIUS, SKIN
*---- IDW    IW    JW    IFLAG    RW     SWELL   IDIR   IFIRST   ILAST   IPRF
       4       1       60      4      0.250000       0.000000       3       1       1
       0
CC
CC WELL NAME
*----  WELNAM
       PROD-3
CC
CC ICHEK , MAX. AND MIN. ALLOWABLE BOTTOMHOLE PRESSURE AND RATE
*---- ICHEK     PWFMIN     PWFMAX    QTMIN    QTMAX
       1       200.000000     9251.000000    0.000000       2807.000000
CC
CC WELL ID,LOCATIONS,AND FLAG FOR SPECIFYING WELL TYPE, WELL RADIUS, SKIN
*---- IDW    IW    JW    IFLAG    RW     SWELL   IDIR   IFIRST   ILAST   IPRF
       5       60      60      4      0.250000       0.000000       3       1       1
       0
CC
CC WELL NAME
*----  WELNAM
       PROD-4
CC
```

```
CC ICHEK , MAX. AND MIN. ALLOWABLE BOTTOMHOLE PRESSURE AND RATE
*---- ICHEK     PWFMIN     PWFMAX    QTMIN    QTMAX
       1       200.000000     9251.000000    0.000000       2807.000000
CC
CC  ID,INJ. RATE AND INJ. COMP. FOR RATE CONS. WELLS FOR EACH PHASE (L=1,3)
*----  ID     QI(M,L)     C(M,KC,L)
       1       400.000000     1.000000       0.000000       0.000000       0.060000
       0.006096       0.004075
       1       0.000000       0.000000       0.000000       0.000000       0.000000
       0.000000       0.000000
       1       0.000000       0.000000       0.000000       0.000000       0.000000
       0.000000       0.000000
CC
CC  ID, TOTAL PRODUCTION RATE
*----  ID    QI(M,1)
       2       -100.000000
CC
CC  ID, TOTAL PRODUCTION RATE
*----  ID    QI(M,1)
       3       -100.000000
CC
CC  ID, TOTAL PRODUCTION RATE
*----  ID    QI(M,1)
       4       -100.000000
CC
CC  ID, TOTAL PRODUCTION RATE
*----  ID    QI(M,1)
       5       -100.000000
CC
CC CUM. INJ. TIME , AND INTERVALS (PV OR DAY) FOR WRITING TO OUTPUT FILES
*---- TINJ     CUMPR1     CUMHI1     WRHPV     WRPRF      RSTC
       250.000000     30.000000      30.000000      10.000000      30.000000
       60.000000
CC
CC FOR IMES=2 ,THE INI. TIME STEP,CONC. TOLERANCE,MAX.,MIN. courant numbers
*----  DT      DCLIM     CNMAX      CNMIN
       0.100000       0.003000       0.200000       0.010000
```

Appendix 2 – Library of MATLAB® Functions to Interact with UTCHEM

A2-1. READING WELL AND PRODUCTION REPORTS FROM UTCHEM

```
%% This function reads the history file of the well No. 2 and returns an
%% array HIST02 with the variables
function [OVERAL]=read_overal(RUNNO);
if size(RUNNO,2)>6
    strfile=RUNNO(1:6);
else
    strfile=RUNNO;
end
str='xxxxxx.OVERAL';
filename=[strfile str(size(strfile,2)+1:end)];
fid1=fopen(filename,'r');
M=textscan(fid1,'%f','headerlines',16,'delimiter',',');
field={'x'};
Y=cell2struct(M,field,1);
X=Y.x;
NT=size(X,1)/(23);
OVERAL=reshape(X,23,NT)';
fclose(fid1);
```

```
function [HIST]=read_hist(RUNNO,ID,TYPE,NPERF,NWELLS)
% This function reads the production history of a well in UTCHEM
% input parameters: -RUNNO: identifier of UT-CHEM run
%                   -ID: well id number
%                   -TYPE: 'INJ' for injector or 'PRO' for producer
%                   -NPERF: number of perforations
%                   -NWELLS: total number of wells in UTCHEM file
% output: Array with the production history
if size(RUNNO,2)>6
    strfile=RUNNO(1:6);
else
    strfile=RUNNO;
end
IDstr=num2str(ID);
if ID<10
    str='xxxxxx.HIST0';
else
    str='xxxxxx.HIST';
end
filename=[strfile str(size(strfile,2)+1:end) IDstr];
fid=fopen(filename,'r');
i=0;
flg=0;
while feof(fid) ==0
    s=fgetl(fid);
    matches = findstr(s,'TOTAL NO. OF ');
    if matches == 4
        position=ftell(fid);
        fseek(fid,position-3,'bof');
        s=fgets(fid,2);
        NCOL=str2num(s);
    end
end
fclose(fid);
fid=fopen(filename,'r');
if TYPE=='INJ'
    M=textscan(fid,'%f','headerlines',15,'delimiter',',');
```

```
else
    M=textscan(fid,'%f','headerlines',21,'delimiter',',');
end
field={'x'};
Y=cell2struct(M,field,1);
X=Y.x;
if NWELLS==2
    nv=2;
else
    nv=1;
end
if TYPE=='INJ'
    %NT=size(X,1)/(4+NPERF*nv);
    NT=size(X,1)/NCOL;
    HIST=reshape(X,NCOL,NT)';
    %HIST=reshape(X,4+NPERF*nv,NT)';
else
    NCOL=30;
    %NT=size(X,1)/(29+NPERF);
    NT=size(X,1)/NCOL;
    %HIST=reshape(X,(29+NPERF),NT)';
    HIST=reshape(X,NCOL,NT)';
end
fclose(fid);
```

A2-2. WRITING UTCHEM INPUT FILE

```
function UTCHEM_FILE(filename);
%% This function writes a UTCHEM input file given a Matlab structure of data written in
file = ‘filename’;
load(filename);
fid=fopen('INPUT','wt');
fprintf(fid,'%s\n','CC***********************************');
fprintf(fid,'%s\n','CC*                                                            *');
fprintf(fid,'%s\n','CC*     BRIEF DESCRIPTION OF DATA SE : UTCHEM (VERSION 9.9)    *');
fprintf(fid,'%s\n','CC*                                                            *');
fprintf(fid,'%s\n','CC********************************************');
fprintf(fid,'%s\n','CC*                                                            *');
fprintf(fid,'%s\n','CC*                                                            *');
fprintf(fid,'%s\n','CC*                                                            *');
fprintf(fid,'%s\n','CC*   LENGTH (FT):              PROCESS :                      *');
fprintf(fid,'%s\n','CC*   THICKNESS (FT):       INJ RATE (FT3/DAY) :          *');
fprintf(fid,'%s\n','CC*   WIDTH (FT):   COORDINATES : CARTESIAN        *');
fprintf(fid,'%s\n','CC*   POROSITY :                                               *');
fprintf(fid,'%s\n','CC*   GRID BLOCKS :                                            *');
fprintf(fid,'%s\n','CC*   DATE :                                                   *');
fprintf(fid,'%s\n','CC*                                                            *');
fprintf(fid,'%s\n','CC***********************************************');
fprintf(fid,'%s\n','CC*                                                            *');
fprintf(fid,'%s\n','CC**********************************************');
fprintf(fid,'%s\n','CC*                                                            *');
fprintf(fid,'%s\n','CC*   RESERVOIR DESCRIPTION                                    *');
fprintf(fid,'%s\n','CC*                                                            *');
fprintf(fid,'%s\n','CC***********************************************');
fprintf(fid,'%s\n','CC');
fprintf(fid,'%s\n','CC Run Number');
fprintf(fid,'%s\n','*----RUNNO');
fprintf(fid,'%s\n',S.RUNNO);
fprintf(fid,'%s\n','CC');
fprintf(fid,'%s\n','CC Title and run description');
fprintf(fid,'%s\n','*----title(i)');
fprintf(fid,'%s\n',S.TITLE1);
fprintf(fid,'%s\n',S.TITLE2);
fprintf(fid,'%s\n',S.TITLE3);
fprintf(fid,'%s\n','C');
fprintf(fid,'%s\n','CC SIMULATION FLAGS');
fprintf(fid,'%s\n','*---- IMODE     IMES    IDISCP  ICWM    ICAP    IREACT  IBIO
ICOORD  ITREAC  ITC     IGAS    IENG');
fprintf(fid,'\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\n',S.IMODE,S.IMES,S
.IDISCP,S.ICWM,S.ICAP,S.IREACT,S.IBIO,S.ICOORD,S.ITREAC,S.ITC,S.IGAS,S.IENG,0,0);
fprintf(fid,'%s\n','CC');
fprintf(fid,'%s\n','CC no. of gridblocks,flag specifies constant or variable grid
size,unit');
fprintf(fid,'%s\n','*---- NX    NY    NZ  IDXYZ  IUNIT');
fprintf(fid,'\t%d\t%d\t%d\t%d\t%d\n',S.NX,S.NY,S.NZ,S.IDXYZ,S.IUNIT);
fprintf(fid,'%s\n','CC');
fprintf(fid,'%s\n','CC constant grid block size in x,y,and z');
fprintf(fid,'%s\n','*---- dx1           dy1           dz1');
fprintf(fid,'\t%3.2f\t%3.2f\t%3.2f\n',S.DX,S.DY,S.DZ);
fprintf(fid,'%s\n','CC');
fprintf(fid,'%s\n','CC total no. of components,no. of tracers,no. of gel components');
fprintf(fid,'%s\n','*----n    no    ntw    nta    ngc    ng    noth');
fprintf(fid,'\t%d\t%d\t%d\t%d\t%d\t%d\t%d\n',S.n,S.no,S.ntw,S.nta,S.ngc,S.ng,S.noth);
fprintf(fid,'%s\n','CC');
fprintf(fid,'%s\n','CC Name of the components');
fprintf(fid,'%s\n','*----spname(i) for i=1 to n');
for kc=1:S.n
    fprintf(fid,'%s\n',S.components(kc).names);
end
fprintf(fid,'%s\n','CC');
fprintf(fid,'%s\n','CC flag indicating if the component is included in calculations or
not');
fprintf(fid,'%s\n','*----icf(kc) for kc=1,n');
for kc=1:S.n
    fprintf(fid,'%d\t',S.icf(kc));
end
fprintf(fid,'\n');
fprintf(fid,'%s\n','CC');
fprintf(fid,'%s\n','CC***********************************************');
fprintf(fid,'%s\n','CC*                                                             *');
fprintf(fid,'%s\n','CC*   OUTPUT OPTIONS                                            *');
fprintf(fid,'%s\n','CC*                                                             *');
fprintf(fid,'%s\n','CC**********************************************');
fprintf(fid,'%s\n','CC');
fprintf(fid,'%s\n','CC');
fprintf(fid,'%s\n','CC FLAG TO WRITE TO UNIT 3,FLAG FOR PV OR DAYS TO PRINT OR TO STOP
THE RUN');
fprintf(fid,'%s\n','*---- ICUMTM  ISTOP  IOUTGMS');
fprintf(fid,'\t%d\t%d\t%d\n',S.ICUMTM,S.ISTOP,S.IOUTGMS);
fprintf(fid,'%s\n','CC');
fprintf(fid,'%s\n','CC FLAG INDICATING IF THE PROFILE OF KCTH COMPONENT SHOULD BE
WRITTEN');
fprintf(fid,'%s\n','*---- IPRFLG(KC),KC=1,N');
for kc=1:S.n
    fprintf(fid,'%d\t',S.IPRFLG(kc));
end
fprintf(fid,'\n');
fprintf(fid,'%s\n','CC');
fprintf(fid,'%s\n','CC FLAG FOR PRES.,SAT.,TOTAL CONC.,TRACER CONC.,CAP.,GEL, ALKALINE
PROFILES');
fprintf(fid,'%s\n','*---- IPPRES IPSAT IPCTOT IPBIO IPCAP IPGEL IPALK IPTEMP IPOBS');
fprintf(fid,'\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\n',S.IPPRES,S.IPSAT,S.IPCTOT,S.IPBIO,S.
IPCAP,S.IPGEL,S.IPALK,S.IPTEMP,S.IPOBS);
fprintf(fid,'%s\n','CC');
fprintf(fid,'%s\n','CC FLAG FOR WRITING SEVERAL PROPERTIES TO UNIT 4 (Prof)');
fprintf(fid,'%s\n','*---- ICKL IVIS IPER ICNM ICSE IHYSTP IFOAMP INONEQ');
fprintf(fid,'\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\n',S.ICKL,S.IVIS,S.IPER,S.ICNM,S.ICSE,S.IHY
STP,S.IFOAMP,S.INONEQ);
fprintf(fid,'%s\n','CC');
fprintf(fid,'%s\n','CC FLAG  for variables to PROF output file');
fprintf(fid,'%s\n','*---- IADS IVEL IRKF IPHSE');
fprintf(fid,'\t%d\t%d\t%d\t%d\n',S.IADS,S.IVEL,S.IRKF,S.IPHSE);
fprintf(fid,'%s\n','CC');
fprintf(fid,'%s\n','CC***********************************************');
fprintf(fid,'%s\n','CC*                                                            *');
fprintf(fid,'%s\n','CC*   RESERVOIR PROPERTIES                                     *');
fprintf(fid,'%s\n','CC*                                                            *');
fprintf(fid,'%s\n','CC***********************************************');
fprintf(fid,'%s\n','CC');
fprintf(fid,'%s\n','CC');
fprintf(fid,'%s\n','CC MAX. SIMULATION TIME ( DAYS)');
fprintf(fid,'%s\n','*---- TMAX ');
fprintf(fid,'%5.2f\n',S.TMAX);
fprintf(fid,'%s\n','CC');
fprintf(fid,'%s\n','CC ROCK COMPRESSIBILITY (1/PSI), STAND. PRESSURE(PSIA)');
fprintf(fid,'%s\n','*---- COMPR                PSTAND');
fprintf(fid,'%E\t%4f\n',S.COMPR,S.PSTAND);
fprintf(fid,'%s\n','CC');
fprintf(fid,'%s\n','CC FLAGS INDICATING CONSTANT OR VARIABLE POROSITY, X,Y,AND Z 
PERMEABILITY');
fprintf(fid,'%s\n','*---- IPOR1  IPERMX IPERMY IPERMZ IMOD ITRANZ INTG');
fprintf(fid,'\t%d\t%d\t%d\t%d\t%d\t%d\t%d\n',S.IPOR1,S.IPERMX,S.IPERMY,S.IPERMZ,S.IMOD,S.
ITRANZ,S.INTG);
fprintf(fid,'%s\n','CC');
if S.IPOR1==0
    fprintf(fid,'%s\n','CC CONSTANT POROSITY');
    fprintf(fid,'%s\n','*---- PORC');
    fprintf(fid,'%1.2f\n',S.PORC);
end
if S.IPOR1==1
    fprintf(fid,'%s\n','CC LAYER POROSITY');
    fprintf(fid,'%s\n','*---- POR');
    for z=1:S.NZ
        fprintf(fid,'%1.2f\t',S.PORL(z));
    end
    fprintf(fid,'\n');
end
if S.IPOR1==2
    fprintf(fid,'%s\n','CC VARIABLE POROSITY');
    fprintf(fid,'%s\n','*---- POR');
    for i=1:S.NX*S.NY*S.NZ
        fprintf(fid,'%1.2f\n',S.POR(i));
    end
end
if S.IPERMX==0
    fprintf(fid,'%s\n','CC');
    fprintf(fid,'%s\n','CC CONSTANT PERMX');
    fprintf(fid,'%s\n','*---- PERMXC');
    fprintf(fid,'%1.2f\n',S.PERMXC);
end
if S.IPERMX==1
    fprintf(fid,'%s\n','CC');
    fprintf(fid,'%s\n','CC LAYER PERMX');
    fprintf(fid,'%s\n','*---- PERMX(nz)');
    for z=1:S.NZ
        fprintf(fid,'%1.2f\t',S.PERMXL(z));
    end
    fprintf(fid,'\n');
end
if S.IPERMX==2
    fprintf(fid,'%s\n','CC');
    fprintf(fid,'%s\n','CC VARIABLE PERMX');
    fprintf(fid,'%s\n','*---- PERMX(nx*ny*nz)');
    for i=1:S.NX*S.NY*S.NZ
        fprintf(fid,'%1.2f\n',S.PERMX(i));
    end
end
```

```
if S.IPERMY==0
    fprintf(fid,'%s\n','CC');
    fprintf(fid,'%s\n','CC CONSTANT PERMY');
    fprintf(fid,'%s\n','*---- PERMYC');
    fprintf(fid,'%1.2f\n',S.PERMYC);
end
if S.IPERMY==1
    fprintf(fid,'%s\n','CC');
    fprintf(fid,'%s\n','CC LAYER PERMY');
    fprintf(fid,'%s\n','*---- PERMY(nz)');
    for z=1:S.NZ
        fprintf(fid,'%1.2f\t',S.PERMYL(z));
    end
    fprintf(fid,'\n');
end
if S.IPERMY==2
    fprintf(fid,'%s\n','CC');
    fprintf(fid,'%s\n','CC VARIABLE PERMY');
    fprintf(fid,'%s\n','*---- PERMY(nx*ny*nz)');
    for i=1:S.NX*S.NY*S.NZ
        fprintf(fid,'%1.2f\n',S.PERMY(i));
    end
end
if S.IPERMY==3
    fprintf(fid,'%s\n','CC');
    fprintf(fid,'%s\n','CC Y DIRECTION PERMEABILITY IS DEPENDENT ON X DIRECTION
PERMEABILITY');
    fprintf(fid,'%s\n','*---- CONSTANT PERMEABILITY MULTIPLIER FOR Y DIRECTION
PERMEABILITY');
    fprintf(fid,'%1.2f\n',S.PERMYMULT);
end
if S.IPERMZ==0
    fprintf(fid,'%s\n','CC');
    fprintf(fid,'%s\n','CC CONSTANT PERMZ');
    fprintf(fid,'%s\n','*---- PERMZC');
    fprintf(fid,'%1.2f\n',S.PERMZC);
end
if S.IPERMZ==1
    fprintf(fid,'%s\n','CC');
    fprintf(fid,'%s\n','CC LAYER PERMZ');
    fprintf(fid,'%s\n','*---- PERMZ(nz)');
    for z=1:S.NZ
        fprintf(fid,'%1.2f\t',S.PERMZL(z));
    end
    fprintf(fid,'\n');
end
if S.IPERMZ==2
    fprintf(fid,'%s\n','CC');
    fprintf(fid,'%s\n','CC VARIABLE PERMZ');
    fprintf(fid,'%s\n','*---- PERMZ(nx*ny*nz)');
    for i=1:S.NX*S.NY*S.NZ
        fprintf(fid,'%1.2f\n',S.PERMZ(i));
    end
end
if S.IPERMZ==3
    fprintf(fid,'%s\n','CC');
    fprintf(fid,'%s\n','Z DIRECTION PERMEABILITY IS DEPENDENT ON X DIRECTION
PERMEABILITY');
    fprintf(fid,'%s\n','*---- CONSTANT PERMEABILITY MULTIPLIER FOR Z DIRECTION
PERMEABILITY');
    fprintf(fid,'%1.2f\n',S.PERMZMULT);
end
fprintf(fid,'%s\n','CC');
fprintf(fid,'%s\n','CC FLAG FOR CONSTANT OR VARIABLE DEPTH, PRESSURE, WATER
SATURATION,INITIAL AQUEOUS PHASE COMPOSITIONS');
fprintf(fid,'%s\n','*----IDEPTH  IPRESS  ISWI  ICWI');
fprintf(fid,'\t%d\t%d\t%d\t%d\n',S.IDEPTH,S.IPRESS,S.ISWI,S.ICWI);
if S.IDEPTH==0
    fprintf(fid,'%s\n','CC');
    fprintf(fid,'%s\n','CC CONSTANT DEPTH (FT)');
    fprintf(fid,'%s\n','*---- D111');
    fprintf(fid,'%1.2f\n',S.D111);
end
if S.IDEPTH==1
    fprintf(fid,'%s\n','CC');
    fprintf(fid,'%s\n','CC LAYER DEPHT');
    fprintf(fid,'%s\n','*---- DEPHT');
    for z=1:S.NX*S.NY
        fprintf(fid,'%1.2f\n',S.DEPHT(z));
    end
end
if S.IPRESS==0
    fprintf(fid,'%s\n','CC');
    fprintf(fid,'%s\n','CC CONSTANT PRESSURE (PSIA) ');
    fprintf(fid,'%s\n','*---- PRESS1');
    fprintf(fid,'%1.2f\n',S.PRESS1);
end
if S.IPRESS==1
    fprintf(fid,'%s\n','CC');
    fprintf(fid,'%s\n','CC PRESSURE (PSIA)');
    fprintf(fid,'%s\n','*---- PRES');
    fprintf(fid,'%1.2f%1.2f\n',S.PINIT,S.HINIT);
    fprintf(fid,'\n');
end
if S.IPRESS==2
    fprintf(fid,'%s\n','CC');
    fprintf(fid,'%s\n','CC VARIABLE PRESSURE');
    fprintf(fid,'%s\n','*---- PRESSURE');
    for i=1:S.NX*S.NY*S.NZ
        fprintf(fid,'%1.2f\n',S.PRESS(i));
    end
end
if S.ISWI==0
    fprintf(fid,'%s\n','CC');
    fprintf(fid,'%s\n','CC CONSTANT INITIAL WATER SATURATION');
    fprintf(fid,'%s\n','*---- SWIC');
    fprintf(fid,'%1.2f\n',S.SWIC);
end
if S.ISWI==1
    fprintf(fid,'%s\n','CC');
    fprintf(fid,'%s\n','CC INITIAL WATER SATURATION');
    fprintf(fid,'%s\n','*---- SWI');
    for z=1:S.NZ
        fprintf(fid,'%1.2f\t',S.SWI(z));
    end
    fprintf(fid,'\n');
end
if S.ISWI==2
    fprintf(fid,'%s\n','CC');
    fprintf(fid,'%s\n','CC VARIABLE INITIAL WATER SATURATION');
    fprintf(fid,'%s\n','*---- SWI');
    for i=1:S.NX*S.NY*S.NZ
        fprintf(fid,'%1.2f\n',S.SWI(i));
    end
end
for kw=1:S.n-S.nta-S.ntw
    if S.icf(kw)==1
        fprintf(fid,'%s\n','CC');
        fprintf(fid,'%s\n','CC INITIAL CONCENTRATION OF SPECIES IN THE AQUEOUS PHASE');
        fprintf(fid,'%s\n','*---- CWI(KW) FOR KW=1,N(8+NO)');
        fprintf(fid,'%1.2f\n',S.CWI(kw));
    end
```

```
end
fprintf(fid,'%s\n','CC');
fprintf(fid,'%s\n','CC************************************************');
fprintf(fid,'%s\n','CC*                                                             *');
fprintf(fid,'%s\n','CC*   PHYSICAL PROPERTY DATA                                    *');
fprintf(fid,'%s\n','CC*                                                             *');
fprintf(fid,'%s\n','CC************************************************');
fprintf(fid,'%s\n','CC');
fprintf(fid,'%s\n','CC');
fprintf(fid,'%s\n','CC OIL CONC. AT PLAIT POINT FOR TYPE II(+)AND TYPE II(-), CMC');
fprintf(fid,'%s\n','*---- c2plc  c2prc   epsme   ihand ');
fprintf(fid,'%f\t%f\t%f\t%d\n',S.c2plc,S.c2prc,S.epsme,S.ihand);
fprintf(fid,'%s\n','CC');
fprintf(fid,'%s\n','CC flag indicating type of phase behavior parameters');
fprintf(fid,'%s\n','*---- ifghbn  ');
fprintf(fid,'\t%d\n',S.ifghbn);
fprintf(fid,'%s\n','CC SLOPE AND INTERCEPT OF BINODAL CURVE AT ZERO, OPT., AND 2XOPT
SALINITY');
fprintf(fid,'%s\n','CC FOR ALCOHOL 1');
fprintf(fid,'%s\n','*---- hbns70   hbnc70   hbns71   hbnc71   hbns72   hbnc72 ');
fprintf(fid,'%f\t%f\t%f\t%f\t%f\t%f\n',S.hbns70,S.hbnc70,S.hbns71,S.hbnc71,S.hbns72,S.hbn
c72);
fprintf(fid,'%s\n','CC SLOPE AND INTERCEPT OF BINODAL CURVE AT ZERO, OPT., AND 2XOPT
SALINITY');
fprintf(fid,'%s\n','CC FOR ALCOHOL 2');
fprintf(fid,'%s\n','*---- hbns80   hbnc80   hbns81   hbnc81   hbns82   hbnc82 ');
fprintf(fid,'%f\t%f\t%f\t%f\t%f\t%f\n',S.hbns80,S.hbnc80,S.hbns81,S.hbnc81,S.hbns82,S.hbn
c82);
fprintf(fid,'%s\n','CC');
fprintf(fid,'%s\n','CC LOWER AND UPPER EFFECTIVE SALINITY FOR ALCOHOL 1 AND ALCOHOL 2');
fprintf(fid,'%s\n','*---- csel7   cseu7   csel8   cseu8');
fprintf(fid,'%f\t%f\t%f\t%f\n',S.csel7,S.cseu7,S.csel8,S.cseu8);
fprintf(fid,'%s\n','CC');
fprintf(fid,'%s\n','CC THE CSE SLOPE PARAMETER FOR CALCIUM AND ALCOHOL 1 AND ALCOHOL 2');
fprintf(fid,'%s\n','*---- beta6    beta7    beta8 ');
fprintf(fid,'%f\t%f\t%f\n',S.beta6,S.beta7,S.beta8);
fprintf(fid,'%s\n','CC');
fprintf(fid,'%s\n','CC FLAG FOR ALCOHOL PART. MODEL AND PARTITION COEFFICIENTS');
fprintf(fid,'%s\n','*---- ialc   opsk7o   opsk7s   opsk8o   opsk8s ');
fprintf(fid,'%d\t%f\t%f\t%f\t%f\n',S.ialc,S.opsk7o,S.opsk7s,S.opsk8o,S.opsk8s);
fprintf(fid,'%s\n','CC');
fprintf(fid,'%s\n','CC NO. OF ITERATIONS, AND TOLERANCE');
fprintf(fid,'%s\n','*---- nalmax     epsalc ');
fprintf(fid,'%f\t%f\n',S.nalmax,S.epsalc);
fprintf(fid,'%s\n','CC');
fprintf(fid,'%s\n','CC ALCOHOL 1 PARTITIONING PARAMETERS IF IALC=1');
fprintf(fid,'%s\n','*---- akwc7     akws7    akm7     ak7      pt7 ');
fprintf(fid,'%f\t%f\t%f\t%f\t%f\n',S.akwc7,S.akws7,S.akm7,S.ak7,S.pt7);
fprintf(fid,'%s\n','CC');
fprintf(fid,'%s\n','CC ALCOHOL 1 PARTITIONING PARAMETERS IF IALC=2');
fprintf(fid,'%s\n','*---- akwc8     akws8    akm8     ak8      pt8 ');
fprintf(fid,'%f\t%f\t%f\t%f\t%f\n',S.akwc8,S.akws8,S.akm8,S.ak8,S.pt8);
fprintf(fid,'%s\n','CC');
fprintf(fid,'%s\n','CC ift model flag');
fprintf(fid,'%s\n','*----  ift ');
fprintf(fid,'\t%d\n',S.ift);
fprintf(fid,'%s\n','CC');
fprintf(fid,'%s\n','CC INTERFACIAL TENSION PARAMETERS');
fprintf(fid,'%s\n','*----  g11     g12     g13     g21     g22      g23 ');
fprintf(fid,'\t%f\t%f\t%f\t%f\t%f\t%f\n',S.g11,S.g12,S.g13,S.g21,S.g22,S.g23);
fprintf(fid,'%s\n','CC');
fprintf(fid,'%s\n','CC LOG10 OF OIL/WATER INTERFACIAL TENSION ');
fprintf(fid,'%s\n','*---- xiftw');
fprintf(fid,'\t%f\n',S.xiftw);
fprintf(fid,'%s\n','CC');
fprintf(fid,'%s\n','CC ORGANIC MASS TRANSFER FLAG');
fprintf(fid,'%s\n','*---- imass icor');
fprintf(fid,'\t%d\t%d\n',S.imass,S.icor);
fprintf(fid,'%s\n','CC');
fprintf(fid,'%s\n','CC CAPILLARY DESATURATION PARAMETERS FOR PHASE 1, 2, AND 3');
fprintf(fid,'%s\n','*---- itrap      t11      t22      t33');
fprintf(fid,'\t%d\t%f\t%f\t%f\n',S.itrap,S.t11,S.t22,S.t33);
fprintf(fid,'%s\n','CC');
fprintf(fid,'%s\n','CC  FLAG FOR RELATIVE PERMEABILITY AND CAPILLARY PRESSURE MODEL');
fprintf(fid,'%s\n','*---- iperm irtype');
fprintf(fid,'\t%d\t%d\n',S.iperm,S.irtype);
fprintf(fid,'%s\n','CC');
fprintf(fid,'%s\n','CC FLAG FOR CONSTANT OR VARIABLE REL. PERM. PARAMETERS');
fprintf(fid,'%s\n','*---- isrw    iprw    iew ');
fprintf(fid,'\t%d\t%d\t%d\n',S.isrw,S.iprw,S.iew);
switch S.isrw
    case 0
        fprintf(fid,'%s\n','CC');
        fprintf(fid,'%s\n','CC CONSTANT RES. SATURATION OF PHASES 1,2,AND 3 AT LOW
CAPILLARY NO.');
        fprintf(fid,'%s\n','*---- s1rwc    s2rwc     s3rwc ');
        fprintf(fid,'\t%f\t%f\t%f\n',S.s1rwc,S.s2rwc,S.s3rwc);
    case 1
        for ph=1:3
            fprintf(fid,'%s\n','CC');
            fprintf(fid,'%s%d%s\n','CC RES. SATURATION OF PHASE',ph,'AT LOW CAPILLARY NO.
BY LAYER');
            fprintf(fid,'%s\n','*---- s1rw(nz)');
            for z=1:S.NZ
                fprintf(fid,'%1.2f\n',S.srwc(ph,z));
            end
        end
    case 2
        for ph=1:3
            fprintf(fid,'%s\n','CC');
            fprintf(fid,'%s%d%s\n','CC RES. SATURATION OF PHASE',ph,'AT LOW CAPILLARY NO.
BY LAYER');
            fprintf(fid,'%s\n','*---- srw(nx*ny*nz)');
            for i=1:S.NX*S.NY*S.NZ
                fprintf(fid,'%1.2f\n',S.srwc(ph,i));
            end
        end
end
switch S.iprw
    case 0
        fprintf(fid,'%s\n','CC');
        fprintf(fid,'%s\n','CC ENDPOINT REL. PERM. OF PHASES 1,2,AND 3 AT LOW CAPILLARY
NO.');
        fprintf(fid,'%s\n','*---- p1rwc   p2rwc   p3rwc');
        fprintf(fid,'\t%f\t%f\t%f\n',S.p1rwc,S.p2rwc,S.p3rwc);
    case 1
        for ph=1:3
            fprintf(fid,'%s\n','CC');
            fprintf(fid,'%s%d%s\n','CC RES. SATURATION OF PHASE ',ph,' AT LOW CAPILLARY
NO. BY LAYER');
            fprintf(fid,'%s\n','*---- s1rw(nz)');
            for z=1:S.NZ
                fprintf(fid,'%1.2f\n',S.prwc(ph,z));
            end
        end
    case 2
        for ph=1:3
            fprintf(fid,'%s\n','CC');
            fprintf(fid,'%s%d%s\n','CC RES. SATURATION OF PHASE ',ph,' AT LOW CAPILLARY
NO. BY LAYER');
            fprintf(fid,'%s\n','*---- srw(nx*ny*nz)');
            for i=1:S.NX*S.NY*S.NZ
```

```
                fprintf(fid,'%1.2f\n',S.prwc(ph,i));
            end
        end
end
switch S.iew
    case 0
        fprintf(fid,'%s\n','CC');
        fprintf(fid,'%s\n','CONSTANT REL. PERM. EXPONENT OF PHASES 1,2,AND 3 AT LOW
CAPILLARY NO.');
        fprintf(fid,'%s\n','*---- e1wc     e2wc     e3wc ');
        fprintf(fid,'\t%f\t%f\t%f\n',S.e1wc,S.e2wc,S.e3wc);
    case 1
        for ph=1:3
            fprintf(fid,'%s\n','CC');
            fprintf(fid,'%s%d%s\n','REL. PERM. EXPONENT OF PHASE ',ph,' AT LOW CAPILLARY
NO. BY LAYER');
            fprintf(fid,'%s\n','*---- s1rw(nz)');
            for z=1:S.NZ
                fprintf(fid,'%1.2f\n',S.ewc(ph,z));
            end
        end
    case 2
        for ph=1:3
            fprintf(fid,'%s\n','CC');
            fprintf(fid,'%s%d%s\n','CC RES. SATURATION OF PHASE ',ph,' AT LOW CAPILLARY
NO. BY LAYER');
            fprintf(fid,'%s\n','*---- srw(nx*ny*nz)');
            for i=1:S.NX*S.NY*S.NZ
                fprintf(fid,'%1.2f\n',S.ewc(ph,i));
            end
        end
end
fprintf(fid,'%s\n','CC');
if S.itrap==1
    fprintf(fid,'%s\n','CC RES. SATURATION OF PHASES 1,2,AND 3 AT HIGH CAPILLARY NO.');
    fprintf(fid,'%s\n','*---- s1rc   s2rc   s3rc');
    fprintf(fid,'\t%f\t%f\t%f\n',S.s1rc,S.s2rc,S.s3rc);
    fprintf(fid,'%s\n','CC');
    fprintf(fid,'%s\n','CC ENDPOINT REL. PERM. OF PHASES 1,2,AND 3 AT HIGH CAPILLARY
NO.');
    fprintf(fid,'%s\n','*---- p1rc   p2rc   p3rc');
    fprintf(fid,'\t%f\t%f\t%f\n',S.p1rc,S.p2rc,S.p3rc);
    fprintf(fid,'%s\n','CC');
    fprintf(fid,'%s\n','CC REL. PERM. EXPONENT OF PHASES 1,2,AND 3 AT HIGH CAPILLARY
NO.');
    fprintf(fid,'%s\n','*---- e13c   e23c   e31c');
    fprintf(fid,'\t%f\t%f\t%f\n',S.e13c,S.e23c,S.e31c);
    fprintf(fid,'%s\n','CC');
end
fprintf(fid,'%s\n','CC WATER AND OIL VISCOSITY , RESERVOIR TEMPERATURE');
fprintf(fid,'%s\n','*---- VIS1    VIS2   TSTAND');
fprintf(fid,'\t%f\t%f\t%f\n',S.VIS1,S.VIS2,S.TSTAND);
fprintf(fid,'%s\n','CC');
fprintf(fid,'%s\n','CC COMPOSITIONAL PHASE VISCOSITY PARAMETERS');
fprintf(fid,'%s\n','*----   ALPHAV1   ALPHAV2   ALPHAV3   ALPHAV4  ALPHAV5');
fprintf(fid,'\t%f\t%f\t%f\t%f\t%f\n',S.ALPHAV1,S.ALPHAV2,S.ALPHAV3,S.ALPHAV4,S.ALPHAV5);
fprintf(fid,'%s\n','CC');
fprintf(fid,'%s\n','CC PARAMETERS TO CALCULATE POLYMER VISCOSITY AT ZERO SHEAR RATE');
fprintf(fid,'%s\n','*---- AP1      AP2      AP3');
fprintf(fid,'\t%f\t%f\t%f\n',S.AP1,S.AP2,S.AP3);
fprintf(fid,'%s\n','CC');
fprintf(fid,'%s\n','CC PARAMETER TO COMPUTE CSEP,MIN. CSEP, AND SLOPE OF LOG VIS. VS. LOG
CSEP ');
fprintf(fid,'%s\n','*---- BETAP    CSE1     SSLOPE');
fprintf(fid,'\t%f\t%f\t%f\n',S.BETAP,S.CSE1,S.SSLOPE);
fprintf(fid,'%s\n','CC');
fprintf(fid,'%s\n','CC PARAMETER FOR SHEAR RATE DEPENDENCE OF POLYMER VISCOSITY');
fprintf(fid,'%s\n','*---- GAMMAC   GAMHF   POWN IPMOD');
fprintf(fid,'\t%f\t%f\t%f\t%d\n',S.GAMMAC,S.GAMHF,S.POWN,S.IPMOD);
fprintf(fid,'%s\n','CC');
fprintf(fid,'%s\n','CC FLAG FOR POLYMER PARTITIONING, PERM. REDUCTION PARAMETERS');
fprintf(fid,'%s\n','*---- IpOLYM    EPHI3    EPHI4    BRK     CRK   RKCUT');
fprintf(fid,'\t%d\t%f\t%f\t%f\t%f\t%f\n',S.IpOLYM,S.EPHI3,S.EPHI4,S.BRK,S.CRK,S.RKCUT);
fprintf(fid,'%s\n','CC');
fprintf(fid,'%s\n','CC SPECIFIC WEIGHT FOR COMPONENTS 1,2,3,7,8 ,Coeffient of oil and
GRAVITY FLAG');
fprintf(fid,'%s\n','*---- DEN1     DEN2    DEN23     DEN3    DEN7    DEN8    IDEN ');
fprintf(fid,'\t%f\t%f\t%f\t%f\t%f\t%f\t%d\n',S.DEN1,S.DEN2,S.DEN23,S.DEN3,S.DEN7,S.DEN8,S
.IDEN);
fprintf(fid,'%s\n','CC');
fprintf(fid,'%s\n','CC FLAG FOR CHOICE OF UNITS ( 0:BOTTOMHOLE CONDITION , 1: STOCK
TANK)');
fprintf(fid,'%s\n','*----- ISTB');
fprintf(fid,'\t%d\n',S.ISTB);
fprintf(fid,'%s\n','CC');
fprintf(fid,'%s\n','CC COMPRESSIBILITY FOR VOL. OCCUPYING COMPONENTS 1,2,3,7,AND 8 ');
fprintf(fid,'%s\n','*---- COMPC(1)  COMPC(2)  COMPC(3)  COMPC(7)  COMPC(8)');
fprintf(fid,'\t%1.8f\t%1.8f\t%1.8f\t%1.8f\t%1.8f\n',S.COMPC(1),S.COMPC(2),S.COMPC(3),S.CO
MPC(7),S.COMPC(8));
fprintf(fid,'%s\n','CC');
fprintf(fid,'%s\n','CC CONSTANT OR VARIABLE PC PARAM., WATER-WET OR OIL-WET PC CURVE FLAG
');
fprintf(fid,'%s\n','*---- ICPC    IEPC   IOW ');
fprintf(fid,'\t%d\t%d\t%d\n',S.ICPC,S.IEPC,S.IOW);
fprintf(fid,'%s\n','CC');
fprintf(fid,'%s\n','CC CAPILLARY PRESSURE PARAMETER, CPC0 ');
fprintf(fid,'%s\n','*---- CPC0');
fprintf(fid,'\t%f\n',S.CPC0);
fprintf(fid,'%s\n','CC');
fprintf(fid,'%s\n','CC CAPILLARY PRESSURE PARAMETER, EPC0 ');
fprintf(fid,'%s\n','*---- EPC0');
fprintf(fid,'\t%f\n',S.EPC0);
fprintf(fid,'%s\n','CC');
fprintf(fid,'%s\n','CC MOLECULAR DIFFUSION COEF. KCTH COMPONENT IN PHASE 1 ');
fprintf(fid,'%s\n','*---- D(KC,1),KC=1,N');
for i=1:S.n
    fprintf(fid,'\t%f',S.D1(i));
end
fprintf(fid,'\n');
fprintf(fid,'%s\n','CC');
fprintf(fid,'%s\n','CC MOLECULAR DIFFUSION COEF. KCTH COMPONENT IN PHASE 2 ');
fprintf(fid,'%s\n','*---- D(KC,2),KC=1,N');
for i=1:S.n
    fprintf(fid,'\t%f',S.D2(i));
end
fprintf(fid,'\n');
fprintf(fid,'%s\n','CC');
fprintf(fid,'%s\n','CC MOLECULAR DIFFUSION COEF. KCTH COMPONENT IN PHASE 3 ');
fprintf(fid,'%s\n','*---- D(KC,3),KC=1,N');
for i=1:S.n
    fprintf(fid,'\t%f',S.D3(i));
end
fprintf(fid,'\n');
fprintf(fid,'%s\n','CC');
fprintf(fid,'%s\n','LONGITUDINAL AND TRANSVERSE DISPERSIVITY OF PHASE 1');
fprintf(fid,'%s\n','*---- ALPHAL(1)     ALPHAT(1)');
fprintf(fid,'\t%f\t%f\n',S.ALPHAL(1),S.ALPHAT(1));
fprintf(fid,'%s\n','CC');
fprintf(fid,'%s\n','LONGITUDINAL AND TRANSVERSE DISPERSIVITY OF PHASE 2');
fprintf(fid,'%s\n','*---- ALPHAL(2)     ALPHAT(2)');
fprintf(fid,'\t%f\t%f\n',S.ALPHAL(2),S.ALPHAT(2));
fprintf(fid,'%s\n','CC');
fprintf(fid,'%s\n','LONGITUDINAL AND TRANSVERSE DISPERSIVITY OF PHASE 3');
fprintf(fid,'%s\n','*---- ALPHAL(3)     ALPHAT(3)');
fprintf(fid,'\t%f\t%f\n',S.ALPHAL(3),S.ALPHAT(3));
fprintf(fid,'%s\n','CC');
fprintf(fid,'%s\n','CC flag to specify organic adsorption calculation');
fprintf(fid,'%s\n','*---- iadso');
fprintf(fid,'\t%d\n',S.iadso);
fprintf(fid,'%s\n','CC');
fprintf(fid,'%s\n','CC SURFACTANT AND POLYMER ADSORPTION PARAMETERS');
fprintf(fid,'%s\n','*---- AD31    AD32   B3D    AD41   AD42   B4D   IADK  IADS1   FADS
REFK');
fprintf(fid,'\t%f\t%f\t%f\t%f\t%f\t%f\t%d\t%d\t%d\t%f\n',S.AD31,S.AD32,S.B3D,S.AD41,S.AD4
2,S.B4D,S.IADK,S.IADS1,S.FADS,S.REFK);
fprintf(fid,'%s\n','CC');
fprintf(fid,'%s\n','CC PARAMETERS FOR CATION EXCHANGE OF CLAY AND SURFACTANT');
fprintf(fid,'%s\n','*---- QV      XKC     XKS     EQW');
fprintf(fid,'\t%f\t%f\t%f\t%f\n',S.QV,S.XKC,S.XKS,S.EQW);
fprintf(fid,'%s\n','CC');
if S.nta+S.ntw>0
    fprintf(fid,'%s\n','CC TRACER PARTITIONING COEFFICIENT');
    fprintf(fid,'%s\n','*---- TK(I),I=1,NTW + NTA');
    for nt=1:S.ntw+S.nta
        fprintf(fid,'%f\t',S.TK(nt));
    end
    fprintf(fid,'\n');
    fprintf(fid,'%s\n','CC');
    fprintf(fid,'%s\n','CC TRACER PARTITION COEFFICIENT SALINITY PARAMETER (1/MEQ/ML)');
    fprintf(fid,'%s\n','*---- TKS(I),I=1,NTW');
    for nt=1:S.ntw
        fprintf(fid,'%f\t',S.TKS(nt));
    end
    fprintf(fid,'%f\t',S.C5INI);
    fprintf(fid,'\n');
    fprintf(fid,'%s\n','CC');
    fprintf(fid,'%s\n','CC RADIOACTIVE DECAY COEFFICIENT');
    fprintf(fid,'%s\n','*---- RDC(I),I=1,NTW + NTA');
    for nt=1:S.ntw+S.nta
        fprintf(fid,'%f\t',S.RDC(nt));
    end
    fprintf(fid,'\n');
    fprintf(fid,'%s\n','CC');
    fprintf(fid,'%s\n','CC TRACER ADSORPTION PARAMETER');
    fprintf(fid,'%s\n','*---- RET(I),I=1,NTW + NTA');
    for nt=1:S.ntw+S.nta
        fprintf(fid,'%f\t',S.RET(nt));
    end
    fprintf(fid,'\n');
    fprintf(fid,'%s\n','CC');
end
fprintf(fid,'%s\n','CC***********************************************');
fprintf(fid,'%s\n','CC*                                                            *');
fprintf(fid,'%s\n','CC*   WELL DATA                                                *');
fprintf(fid,'%s\n','CC*                                                            *');
fprintf(fid,'%s\n','CC****************************************');
fprintf(fid,'%s\n','CC');
fprintf(fid,'%s\n','CC');
fprintf(fid,'%s\n','CC FLAG FOR SPECIFIED BOUNDARY AND ZONE IS MODELED');
fprintf(fid,'%s\n','*---- IBOUND     IZONE');
fprintf(fid,'\t%d\t%d\n',S.IBOUND,S.IZONE);
fprintf(fid,'%s\n','CC');
fprintf(fid,'%s\n','CC TOTAL NUMBER OF WELLS, WELL RADIUS FLAG, FLAG FOR TIME OR COURANT
NO.');
fprintf(fid,'%s\n','*---- NWELL   IRO    ITIME    NWREL');
fprintf(fid,'\t%d\t%d\t%d\t%d\n',S.NWELL,S.IRO(1),S.ITIME(1),S.NWREL);
for nw=1:S.NWELL
    fprintf(fid,'%s\n','CC');
    fprintf(fid,'%s\n','CC WELL ID,LOCATIONS,AND FLAG FOR SPECIFYING WELL TYPE, WELL
RADIUS, SKIN');
    fprintf(fid,'%s\n','*---- IDW    IW    JW    IFLAG    RW     SWELL   IDIR   IFIRST
ILAST   IPRF ');

fprintf(fid,'\t%d\t%d\t%d\t%d\t%f\t%f\t%d\t%d\t%d\t%d\n',S.WELL(nw).IDW,S.WELL(nw).IW,S.W
ELL(nw).JW,S.WELL(nw).IFLAG,  ...

S.WELL(nw).RW,S.WELL(nw).SWELL,S.WELL(nw).IDIR,S.WELL(nw).IFIRST,S.WELL(nw).ILAST,S.WELL(
nw).IPRF);
    fprintf(fid,'%s\n','CC');
    fprintf(fid,'%s\n','CC WELL NAME');
    fprintf(fid,'%s\n','*----  WELNAM');
    fprintf(fid,'\t%s\n',S.WELL(nw).NAM);
    fprintf(fid,'%s\n','CC');
    fprintf(fid,'%s\n','CC ICHEK , MAX. AND MIN. ALLOWABLE BOTTOMHOLE PRESSURE AND
RATE');
    fprintf(fid,'%s\n','*---- ICHEK     PWFMIN     PWFMAX    QTMIN    QTMAX');

fprintf(fid,'\t%d\t%f\t%f\t%f\t%f\n',S.WELL(nw).ICHEK,S.WELL(nw).PWFMIN,S.WELL(nw).PWFMAX
,S.WELL(nw).QTMIN,S.WELL(nw).QTMAX);
end
t=1;
for nw=1:S.NWELL
    if S.WELL(nw).IFLAG ==1 || S.WELL(nw).IFLAG ==3
        fprintf(fid,'%s\n','CC');
        fprintf(fid,'%s\n','CC  ID,INJ. RATE AND INJ. COMP. FOR RATE CONS. WELLS FOR EACH
PHASE (L=1,3)');
        fprintf(fid,'%s\n','*----  ID     QI(M,L)     C(M,KC,L)');
        for l=1:3
            fprintf(fid,'\t%d\t%f',S.WELL(nw).IDW,S.QI(nw,l,t));
            for kc=1:S.n
                fprintf(fid,'\t%f',S.C(nw,kc,l,t));
            end
            fprintf(fid,'\n');
        end
    end
    if  S.WELL(nw).IFLAG ==2 || S.WELL(nw).IFLAG ==3
        fprintf(fid,'%s\n','CC');
        fprintf(fid,'%s\n','CC ID, BOTTOM HOLE PRESSURE FOR PRESSURE CONSTRAINT WELL
(IFLAG=2 OR 3)');
        fprintf(fid,'%s\n','*----  ID    PWF');
        fprintf(fid,'\t%d\t%f\n',S.WELL(nw).IDW,S.WELL(nw).T(t).PWF);
    end
    if  S.WELL(nw).IFLAG ==4
        fprintf(fid,'%s\n','CC');
        fprintf(fid,'%s\n','CC  ID, TOTAL PRODUCTION RATE');
        fprintf(fid,'%s\n','*----  ID    QI(M,1)');
        fprintf(fid,'\t%d\t%f\n',S.WELL(nw).IDW,S.WELL(nw).T(t).PWF);
    end
end
fprintf(fid,'%s\n','CC');
fprintf(fid,'%s\n','CC CUM. INJ. TIME , AND INTERVALS (PV OR DAY) FOR WRITING TO OUTPUT
FILES');
fprintf(fid,'%s\n','*---- TINJ     CUMPR1     CUMHI1     WRHPV     WRPRF      RSTC');
fprintf(fid,'\t%f\t%f\t%f\t%f\t%f\t%f\n',S.TINJ(t),S.CUMPR1(t),S.CUMHI1(t),S.WRHPV(t),S.W
RPRF(t),S.RSTC(t));
fprintf(fid,'%s\n','CC');
fprintf(fid,'%s\n','CC FOR IMES=2 ,THE INI. TIME STEP,CONC. TOLERANCE,MAX.,MIN. courant
numbers');
fprintf(fid,'%s\n','*----  DT      DCLIM     CNMAX      CNMIN');
fprintf(fid,'\t%f\t%f\t%f\t%f\n',S.DT(t),S.DCLIM(t),S.CNMAX(t),S.CNMIN(t));
t=2;
while S.TINJ(t-1)<S.TMAX
    fprintf(fid,'%s\n','CC');
    fprintf(fid,'%s\n','CC FLAG FOR INDICATING BOUNDARY CHANGE');
    fprintf(fid,'%s\n','*---- IBMOD');
    fprintf(fid,'\t%d\n',S.IBMOD(t));
    fprintf(fid,'%s\n','CC');
    fprintf(fid,'%s\n','CC IRO, ITIME, NEW FLAGS FOR ALL THE WELLS');
    fprintf(fid,'%s\n','*----  IRO    ITIME     IFLAG ');
    fprintf(fid,'\t%d\t%d',S.IRO(t),S.ITIME(t));
    for nw=1:S.NWELL
        fprintf(fid,'\t%d',S.WELL(nw).T(t).IFLAG);
    end
    fprintf(fid,'\n');
    fprintf(fid,'%s\n','CC');
    fprintf(fid,'%s\n','CC NUMBER OF WELLS CHANGES IN LOCATION OR SKIN OR PWF');
    fprintf(fid,'%s\n','*----  NWEL1');
    fprintf(fid,'\t%d\n',S.NWEL1(t));
    for nw=1:S.NWEL1(t)
        fprintf(fid,'%s\n','CC');
        fprintf(fid,'%s\n','CC WELL ID,LOCATIONS,AND FLAG FOR SPECIFYING WELL TYPE, WELL 
RADIUS, SKIN');
        fprintf(fid,'%s\n','*---- IDW    IW    JW    RW     SWELL   IDIR   IFIRST   ILAST 
IPRF ');

fprintf(fid,'\t%d\t%d\t%d\t%f\t%f\t%d\t%d\t%d\t%d\n',S.NWELL1(nw,t).IDW,S.NWELL1(nw,t).IW
,S.NWELL1(nw,t).JW,  ...

S.NWELL1(nw,t).RW,S.NWELL1(nw,t).SWELL,S.NWELL1(nw,t).IDIR,S.NWELL1(nw,t).IFIRST,S.NWELL1
(nw,t).ILAST,S.NWELL1(nw,t).IPRF);
        fprintf(fid,'%s\n','CC');
        fprintf(fid,'%s\n','CC WELL NAME');
        fprintf(fid,'%s\n','*----  WELNAM');
        fprintf(fid,'\t%s\n',S.NWELL1(nw,t).NAM);
        fprintf(fid,'%s\n','CC');
        fprintf(fid,'%s\n','CC ICHEK , MAX. AND MIN. ALLOWABLE BOTTOMHOLE PRESSURE AND 
RATE');
        fprintf(fid,'%s\n','*---- ICHEK     PWFMIN     PWFMAX    QTMIN    QTMAX');

fprintf(fid,'\t%d\t%f\t%f\t%f\t%f\n',S.NWELL1(nw,t).ICHEK,S.NWELL1(nw,t).PWFMIN,S.NWELL1(
nw,t).PWFMAX,S.NWELL1(nw,t).QTMIN,S.NWELL1(nw,t).QTMAX);
    end
    fprintf(fid,'%s\n','CC');
    fprintf(fid,'%s\n','CC NUMBER OF WELLS WITH RATE CHANGES, ID');
    fprintf(fid,'%s\n','*----  NWEL2');
    fprintf(fid,'\t%d',S.NWEL2(t));
    for i=1:S.NWEL2(t)
        fprintf(fid,'\t%d',S.NWEL2ID(t,i));
    end
    fprintf(fid,'\n');
    for nw=1:S.NWELL
        if  (S.WELL(nw).IFLAG ==1 || S.WELL(nw).IFLAG ==3) && 
S.NWEL2ID(t,nw)==S.WELL(nw).IDW
            fprintf(fid,'%s\n','CC');
            fprintf(fid,'%s\n','CC  ID,INJ. RATE AND INJ. COMP. FOR RATE CONS. WELLS FOR 
EACH PHASE (L=1,3)');
            fprintf(fid,'%s\n','*----  ID     QI(M,L)     C(M,KC,L)');
            for l=1:3
                fprintf(fid,'\t%d\t%f',S.WELL(nw).IDW,S.QI(nw,l,t));
                for kc=1:S.n
                    fprintf(fid,'\t%f',S.C(nw,kc,l,t));
                end
                fprintf(fid,'\n');
            end
        end
        if (S.WELL(nw).IFLAG ==2 || S.WELL(nw).IFLAG ==4) && 
S.NWEL2ID(t,nw)==S.WELL(nw).IDW
            fprintf(fid,'%s\n','CC');
            fprintf(fid,'%s\n','CC ID, BOTTOM HOLE PRESSURE FOR PRESSURE CONSTRAINT WELL 
(IFLAG=2 OR 3)');
```

```
            fprintf(fid,'%s\n','*----  ID    PWF');
            fprintf(fid,'\t%d\t%f\n',S.WELL(nw).IDW,S.WELL(nw).T(t).PWF);
        end
    end
    fprintf(fid,'%s\n','CC');
    fprintf(fid,'%s\n','CC CUM. INJ. TIME , AND INTERVALS (PV OR DAY) FOR WRITING TO
OUTPUT FILES');
    fprintf(fid,'%s\n','*---- TINJ     CUMPR1     CUMHI1     WRHPV     WRPRF      RSTC');

fprintf(fid,'\t%f\t%f\t%f\t%f\t%f\t%f\n',S.TINJ(t),S.CUMPR1(t),S.CUMHI1(t),S.WRHPV(t),S.W
RPRF(t),S.RSTC(t));
    fprintf(fid,'%s\n','CC');
    fprintf(fid,'%s\n','CC FOR IMES=2 ,THE INI. TIME STEP,CONC. TOLERANCE,MAX.,MIN.
courant numbers');
    fprintf(fid,'%s\n','*----  DT      DCLIM     CNMAX      CNMIN');
    fprintf(fid,'\t%f\t%f\t%f\t%f\n',S.DT(t),S.DCLIM(t),S.CNMAX(t),S.CNMIN(t));
    t=t+1;
end
fclose(fid);
```

A2-3. READING A GRID PROPERTY

```
function [P1,P2,P3]=read_pres(rootname,T,NX,NY,NZ);
% read_pres reads the pressure field for a series of times
% returns three arrays of pressure, one for each phase
% [P1,P2,P3]=read_pres(rootname,T,NX,NY,NZ);
filename=[rootname '.PRESP'];
fid=fopen(filename);
for i=1:5
    S=fgetl(fid);
end
for t=1:T
    S=fgetl(fid);
    for k=1:NZ
        for phase=1:3
            S=fgetl(fid);
            for j=1:NY
                for i=1:NX
                    if phase==1
                        P1(i,j,k,t) = fscanf(fid, '%f', [1 1]);
                    end
                    if phase==2
                        P2(i,j,k,t) = fscanf(fid, '%f', [1 1]);
                    end
                    if phase==3
                        P3(i,j,k,t) = fscanf(fid, '%f', [1 1]);
                    end
                end
            end
            fseek(fid,ftell(fid)+1,'bof');
        end
    end
end
fclose(fid);
```

Appendix 3- Model Selection Algorithm

A3-1. MATLAB CODE FOR MODEL SELECTION ALGORITHM

```
addpath '/home/cmantilla/Research/polymer/matlab_scripts/';
clc
clear all;
close all;
set(0,'defaultaxesfontsize',14)
%%***************INPUT FILES***************************************
Base_dir='/home/cmantilla/DB/Real-2/';
ref_input_file=[Base_dir,'REF_40ELLIP.mat'];    %
ref_prod_file=[Base_dir,'REFDATA_40b.mat'];  %
prod_data_real=[Base_dir,'PRODDATA_40b.mat'];  %
proxy_resp_file1=[Base_dir,'proxy_perm_wells.dat'];   %
proxy_resp_file2=[Base_dir,'Arrival.dat'];  %
Permfile=[Base_dir,'All_Real_Perm.dat'];   %
Faciefile=[Base_dir,'All_Facies.dat'];
%%**************END INPUT FILES************************************
%%********************PREPROCESING DATA****************************
load(ref_input_file);
load(ref_prod_file);
load(prod_data_real);
well_num=[1,2];
NX=S.NX;  % Number of blocks in Y
NY=S.NY;  % Number of blocks in Y
NZ=S.NZ;   % Number of blocks in Z
NCELLS=NX*NY*NZ; % Total number of cells
NW=S.NWELL; % Number of wells
NWI=0; % Number of injectors
NWP=0; % Number of producers
NT=25;
for nw=1:NW
  if S.WELL(nw).IFLAG==1 || S.WELL(nw).IFLAG==3
  NWI=NWI+1;
  else
  NWP=NWP+1;
  end
end
ND=2;
Dref2=Dref;
clear Dref;
for nd=1:ND
  if well_num(nd)<=NWI
  Dref.Data(:,nd)=Dref2.WELL(well_num(nd)).DATA(1:NT,6);
  else
  Dref.Data(:,nd)=Dref2.WELL(well_num(nd)).DATA(1:NT,10);
  end
end
NReal=size(REAL,2);
REAL2=REAL;
clear REAL
for nr=1:NReal
  for nd=1:ND
  if well_num(nd)<=NWI
    REAL(nr).DATA(:,nd)=REAL2(nr).WELL(well_num(nd)).DATA(1:NT,6);
  else
    REAL(nr).DATA(:,nd)=REAL2(nr).WELL(well_num(nd)).DATA(1:NT,10);
  end
  end
end
NM=12;   % Number of reservoir model types
NR=50;  % Number of realizations of each model type
```

```
%%***************************END OF DATA PREPROCESSING********************
%% Phase 1 : Scan Reservoir Models through Proxy Models
delete *.ps
NRmin=10;
DT=10; % Time step size.
for nt=1:NT
  Time(nt)=DT*nt;
end
%% Phase 2 : Compute Covariance Matrix and PCA
% Normalizes the outputs from proxies
system('~/Research/polymer/Random_cpp/Cluster_Normal/cluster_normal.exe');   % Executes
the proxy model
system('~/Research/polymer/Random_cpp/Cluster_Polymer/cluster_polymer.exe');  % Executes
the proxy model
ncols=[1,2];  % Number of columns from the proxy output file
[PR1]=read_proxy_response(proxy_resp_file1,ncols);
ncols=[1,2];  % Number of columns from the proxy output file
[PR2]=read_proxy_response(proxy_resp_file2,ncols);
PR=[PR1]; % Number of columns from the proxy output file
%stores minimum and maximum values from the proxy responses
minPR=min(PR);
maxPR=max(PR);
PR=(PR-ones(NM*NR,1)*minPR)./(ones(NM*NR,1)*(maxPR-minPR));
% Normalizes the outputs from proxies
CovarMat=cov(PR); % Covariance Matrix
[U,Sval,V]=svd(CovarMat); % Singular Value Decomposition
PCscores=PR*U; % Gets the principal component scores (PC scores)
thres=0.8; % number of eigen values selected
eval=diag(Sval); % Eigen Values
for i=1:size(PR,2)
  if sum(eval(1:i))/sum(eval)>thres
  break;
  end
end
nval=i;
nval=2;
PC1=PR*U(:,1);  % Dot product with first eigenvector
PC2=PR*U(:,2);  % Dot product with second eigenvector
PC=PCscores(:,1:nval); % PC matrix has the first n PC scores
nf=1;
figure(nf)
for nm=1:NM
  switch nm
  case 1
    color='ob';
    face='b';
  case 2
    color='or';
    face='r';
  case 3
    color='og';
    face='r';
  end
  i1=(nm-1)*50+1;
  i2=nm*50;
  plot(PC1(i1:i2),PC2(i1:i2),color,'markersize',3,'markerface',face);
  hold on;
end
C={};
for nm=1:NM
  str=['Model ',int2str(nm)];
  C=[C,str];
end
title('Projection on top 2 Eigenvectors','fontsize',16);
legend(C);
xlabel('Projection on 1st Eigenvector','fontsize',16);
```

```
ylabel('Projection on 2nd Eigenvector','fontsize',16);
print -dpsc PCScores_Models.ps
hold off;
xlim_orig=xlim;
ylim_orig=ylim;
%% PHASE 3: CLUSTER ANALYSIS
NC=best_clusters(PC);
it=0;
nsmin=3;
fid_perm_cent=fopen('Perm_centers.dat','wt');
fprintf(fid_perm_cent,'%s\n%s\n%s\n%s\n','Permeability of cluster centers','3','Average',
'Variance','Indicator');
fid_res=fopen('Cluster_Probability.txt','wt');
for nc=1:NC
  fprintf(fid_res,'%s %d\t','NR',nc);
end
for nc=1:NC
  fprintf(fid_res,'%s %d\t','PrCD',nc);
end
for nr=1:NReal
  Real_Indx(nr)=nr;
end
while NReal>NRmin
  it=it+1;
  obj_fun=zeros(NC,1);
  flag=0;
  while flag==0
  flag=1;
  %[clus,centers]=k_means(PC,NC);
  y=kMeansCluster(PC,NC,0);
  clus=y(:,3);
  centers=zeros(NC,2);
  for nc=1:NC
    cluster(nc).NRClus=length(find(clus==nc));
  end
  for nr=1:NReal
    nc=clus(nr);
    centers(nc,:)=centers(nc,:)+PC(nr,:)/cluster(nc).NRClus;
  end
  for nc=1:NC
    x=find(clus==nc);
    if size(x,1)==0
    flag=0;
    break;
    end
  end
  end
  nf=nf+1;
  for nc=1:NC
  cluster(nc).ns=0;
  cluster(nc).NRClus=length(find(clus==nc)); % Number of realizations on each cluster
  cluster(nc).PrC=cluster(nc).NRClus/NReal;
  cluster(nc).center=centers(nc,:);
  count=0;
  for nr=1:NReal
    if clus(nr)==nc
    count=count+1;
    cluster(nc).Real_Indx(count)=Real_Indx(nr);
    end
  end
  for nm=1:NM
    count=0;
    m1=(nm-1)*NR+1;
    m2=nm*NR;
    for nr=1:cluster(nc).NRClus
    if cluster(nc).Real_Indx(nr)>=m1 && cluster(nc).Real_Indx(nr)<=m2
```

```
      cluster(nc).ModelType(nr)=nm;
      count=count+1;
    end
    end
    cluster(nc).PrCM(nm)=count/NR;
    cluster(nc).PrMC(nm)=cluster(nc).PrCM(nm)*(1/NM)/cluster(nc).PrC; %
Pr{Model=nm|Cluster=nc}
  end
  count=0;
  figure(nf);
  for nr=1:NReal
    if nc==clus(nr)
    count=count+1;
    cluster(nc).PC(count,:)=PC(nr,:);
    col=[nc/NC,1-nc/NC,nc/NC];
    switch nc
      case 1
      col=[0,0,1];
      case 2
      col=[0,1,0];
      case 3
      col=[0.5,0.5,0.5];
    end
    plot(PC(nr,1),PC(nr,2),'MarkerFaceColor',col,...
      'MarkerSize',4,'Marker','o','LineStyle','none','Color',col);
    hold on;
    end
  end
  dc_weight=[];
  for nr=1:cluster(nc).NRClus
    dc_weight(nr)=1/norm(cluster(nc).PC(nr,:)-centers(nc,:));
  end
  cluster(nc).dc_weight=dc_weight/sum(dc_weight);
  for ns=1:nsmin
    cluster(nc).ns=cluster(nc).ns+1;
    cdf=0;
    x=rand;
    for n=1:cluster(nc).NRClus
    cdf=cdf+cluster(nc).dc_weight(n);
    % the probability of selecting a realization within a cluster is proportional to the
distance to its center
    if x<cdf
      break;
    end
    end
    [a,ix]=sort(cluster(nc).dc_weight);
    cluster(nc).samples(ns).nrs=cluster(nc).Real_Indx(n);
    x=cluster(nc).samples(ns).nrs;
    %% RUNS UTCHEM TO EXTRACT PRODUCTION DATA
    S.PERMX=textread(Permfile,'%f','headerlines',3+(x-1)*NCELLS);
    save temp.mat S
    UTCHEM_FILE('temp.mat');
    UTCHEM_HEAD('temp.mat');
    system('time ~/Research/polymer/UTCHEM10/utchem10.exe');
    H1=read_hist(S.RUNNO,1,'INJ',S.WELL(1).NPERF,S.NWELLS);
    %%
    for nd=1:ND
    cluster(nc).samples(ns).DataType(nd).Data=REAL(x).DATA(1:NT,nd);
    obj_fun(nc)=obj_fun(nc)+norm(cluster(nc).samples(ns).DataType(nd).Data-...
      Dref.Data(1:NT,nd)); % mismatch between the sampled and the reference production
data
    end
  end
  end
  for nc=1:NC
  x2=0.8/6+(cluster(nc).center(1)-xlim_orig(1))/(xlim_orig(2)-xlim_orig(1))*4.625/6;
  y2=0.5/4.5+(cluster(nc).center(2)-ylim_orig(1))/(ylim_orig(2)-ylim_orig(1))*3.7/4.5;
  annotation('textbox',[x2,y2,0.03,0.01],'string',num2str(nc),'LineStyle','none');
  end
  xlim(xlim_orig);
  ylim(ylim_orig);
  title('K-means sub-clustering in principal component space','fontsize',14);
  xlabel('Projection on 1st Eigenvector','fontsize',16);
  ylabel('Projection on 2nd Eigenvector','fontsize',16);
  str=['PCScores_Clusters_it_',num2str(it),'.ps'];
  print('-dpsc',str);
  hold off;
  fprintf(fid_res,'\n');
  NS=min(100,max(10,ceil(NReal/30))); % total number of samples
  for ns1=nsmin*NC+1:NS
  x=rand; % samples from pixelplt_para uniform distribution
  cdf=0;
  for nc=1:NC
    cdf=cdf+cluster(nc).PrC;  % Probability of selecting each cluster
    if x<cdf
    break; % nc is the number of cluster selected for this reservoir
    end
  end
  cluster(nc).ns=cluster(nc).ns+1; % counter for the number of samples drawn from the
selected cluster
  ns=cluster(nc).ns;
  % selects a sample according to the distance to the center
  cdf=0;
  x=rand;
  for n=1:cluster(nc).NRClus
    cdf=cdf+cluster(nc).dc_weight(n);
    % the probability of selecting a realization within a cluster is proportional to the
distance to its center
    if x<cdf
    break;
    end
  end
  [a,ix]=sort(cluster(nc).dc_weight);
  %n=ix(ns);
  cluster(nc).samples(ns).nrs=cluster(nc).Real_Indx(n);
  % realization index refered to the initial set of realizations
  x=cluster(nc).samples(ns).nrs;
  S.PERMX=textread(Permfile,'%f','headerlines',3+(x-1)*NCELLS);
  save temp.mat S
  UTCHEM_FILE('temp.mat');
  UTCHEM_HEAD('temp.mat');
  system('time ~/Research/polymer/UTCHEM10/utchem10.exe');
  H1=read_hist(S.RUNNO,1,'INJ',1,2);
  for nd=1:ND
    cluster(nc).samples(ns).DataType(nd).Data=REAL(x).DATA(1:NT,nd);
    obj_fun(nc)=obj_fun(nc)+norm(cluster(nc).samples(ns).DataType(nd).Data-...
    Dref.Data(1:NT,nd)); % mismatch between the sampled and the reference production data
  end
  end
  for nd=1:ND % plots the sampled production data
  figure(nf+nd);
  for nc=1:NC
    col=[nc/NC,1-nc/NC,nc/NC];
    switch nc
    case 1
      col=[0,0,1];
    case 2
      col=[0,1,0];
    case 3
      col=[0.5,0.5,0.5];
    end
    for ns=1:cluster(nc).ns
    plot(Time,cluster(nc).samples(ns).DataType(nd).Data,'Color',col);
    hold on; % retrives the production data from the pre-run
    end
  end
  plot(Time,Dref.Data(1:NT,nd),'r','LineWidth',3);
  xlim([0,(NT+1)*DT]);
  if it==1
    ylim_prod(nd,:)=ylim;
  else
    ylim([0,ylim_prod(nd,2)]);
  end
  ylim([0,10000]);
  % labeling of production data figure
  str=['Production data ',num2str(nd)];
  title(str,'fontsize',16);
  xlabel('Time (days)','fontsize',16);
  ylabel('Pressure drop (psi)','fontsize',16);
  str=['Prod_Data_it_',num2str(it),'.ps'];
  print('-dpsc',str);
  hold off;
  end
  nf=nf+ND;
  for nc=1:NC
  obj_fun(nc)=obj_fun(nc)/cluster(nc).ns; % normalizes the objective function
  cluster(nc).upfactor=0;
  end
  NS=NS+NC;
  % In this loop it is calculated the probability of each cluster
  for nt=1:NT
  PC1Plot=[];
  PC2Plot=[];
  for nd=1:ND
    % calculates the average production data
    ave=0;
    for nc=1:NC
    NSC=cluster(nc).ns;
    for nsc=1:NSC
      ave=ave+cluster(nc).samples(nsc).DataType(nd).Data(nt);
    end
    end
    % average of production data in all samples
    ave=ave/NS;
    var=0;
    for nc=1:NC
    NSC=cluster(nc).ns;
    for nsc=1:NSC
      var=var+(cluster(nc).samples(nsc).DataType(nd).Data(nt)-ave)^2;
    end
    end
    % variance of production data in all samples
    var=var/NS;
    % Probability of Dref within all realizations
    PrD(nd)=exp(-(ave-Dref.Data(nt,nd))^2/(2*var))/(sqrt(2*pi*var));
    PrD(nd)=1/(sqrt(2*pi*var));
    for nc=1:NC
    % number of samples in cluster
    NSC=cluster(nc).ns;
    avec=0;
    varc=0;
    for nsc=1:NSC
      avec=avec+cluster(nc).samples(nsc).DataType(nd).Data(nt);
    end
    avec=avec/NSC;
    % average production data within cluster
    for nsc=1:NSC
      varc=varc+(cluster(nc).samples(nsc).DataType(nd).Data(nt)-avec)^2;
    end
    % variance of prodution data within cluster
    varc=varc/NSC;
    % Probability of Dref within cluster nc
    PrDC(nd)=exp(-(avec-Dref.Data(nt,nd))^2/(2*varc))/(sqrt(2*pi*varc));
    PrDC(nd)=exp(-(avec-Dref.Data(nt,nd))^2/(2*var))/(sqrt(2*pi*var));
    upfactor=PrDC(nd)/PrD(nd);
    cluster(nc).upfactor=cluster(nc).upfactor+upfactor/NT;
    end
  end
  end
  sum1=0;
  % Updates the probability of each cluster
  for nc=1:NC
  fprintf(fid_res,'%d\t',cluster(nc).NRClus);
  cluster(nc).PrC=cluster(nc).PrC*cluster(nc).upfactor;
  sum1=sum1+cluster(nc).PrC;
  end
  % Standarizes the probability of each cluster
  for nc=1:NC
  cluster(nc).PrC=cluster(nc).PrC/sum1;
  fprintf(fid_res,'%1.2f\t',cluster(nc).PrC);
  end
  % Selection of cluster according to their probability
  cdf=0;
  x=rand;
  for nc_sel=1:NC
  cdf=cdf+cluster(nc_sel).PrC;
  if x<cdf
    break;
  end
  end
  for nc=1:NC
  PrC(nc)=cluster(nc).PrC;
  end
  [x,nc_sel]=max(PrC);
  if it==1
  nc_sel1=nc_sel;
  centers1=centers;
  end
  fprintf(fid_res,'%d\n',nc_sel);
  NReal=cluster(nc_sel).NRClus;
  Real_Indx=cluster(nc_sel).Real_Indx;
  PC=cluster(nc_sel).PC;
  NC=best_clusters(PC);
  nrc=zeros(NC,1);
  PrM=cluster(nc_sel).PrMC;
  dc=cluster(nc_sel).dc_weight;
  clear cluster;   % clears all data refering to the previous level
end
sum1=0;
for nm=1:NM
  sum1=sum1+PrM(nm);
end
PrM=PrM/sum1;
% Find the common characteristics in the restricted cluster
FAC=zeros(NX*NY*NZ,1);
for nr=1:NReal
  x=Real_Indx(nr);
  for nd=1:ND
  Data=REAL(x).DATA(1:NT,nd);
  figure(nf+nd);
  plot(Time,Data);
  hold on;
  end
end
```

```
for nd=1:ND
  figure(nf+nd);
  plot(Time,Dref.Data(1:NT,nd),'r','LineWidth',3);
  xlim([0,(NT+1)*DT]);
  ylim([0,ylim_prod(nd,2)]);
  str=['Production data ',num2str(nd)];
  title(str,'fontsize',16);
  xlabel('Time Step','fontsize',16);
  ylabel('Production data','fontsize',16);
  str=['Prod_Data_final',num2str(nd),'.ps'];
  print('-dpsc',str);
  hold off;
end
nf=nf+ND;
```

A3-2. MATLAB CODE FOR OPTIMIZATION USING RESPONSE SURFACE

```
function Uopt_all=Optimize(Real_Indx,Permfile,Proxyfile,Uhist,nt,NT,S,UB,LB,DT,Tol)
% Looks for the optimal control settings that maximizes the NPV
% Sintax:  Uopt_all=Optimize(Real_Indx,Permfile,Proxyfile,Uhist,nt,NT,S,UB,LB,DT,Tol)
% - Input : Real_Indx = Vector witth indexes of the realizations contained
% in the permeability file
%        Permfile = File with the permeability values, gslib format
%        Proxyfile = Parameter file for proxy model. Matlab type.
%        Uhist = NC x NT control matrix with the history of the control
%        variables, columns are rates at wells and rows is time
%        nt = Time step at which the control starts
%        NT = Total number of time steps incluiding the history
%        S = Matlab structure with the parameters of the reservoir
%        UB = Upper bound for control. Maximum injection/production
%        rates
%        LB = Lower bound for control.
%        DT = Control time step size.
%        Tol = Tolerance for convergence on control variables
% - Output : Vector with the optimal control settings
load(Proxyfile);
PROXY.Permfile='perm_temp.dat';
Tf=DT*NT;
dt=(DT)/(floor(100/NT));
NT2=Tf/dt;
PROXY.NT=NT2;
for t=1:NT2
  PROXY.T(t)=t*dt;
end
NReal=size(Real_Indx,2);
NX=S.NX;
NY=S.NY;
NZ=S.NZ;
NW=S.NWELL;
NWI=0;
NWP=0;
for nw=1:NW
  if S.WELL(nw).IFLAG==1
    NWI=NWI+1;
  else
    NWP=NWP+1;
  end
end
for t1=1:nt-1
  for t2=DT/dt*(t1-1)+1:NT2
    time=dt*t2;
    if time<=(t1)*DT
      Inj=0;
      for nw=1:NWI
        PROXY.SOUR(nw).Q(t2)=Uhist(t1,nw);
        Inj=Inj+Uhist(t1,nw);
      end
      Prd=0;
      for nw=1:NWP-1
        PROXY.SINK(nw).Q(t2)=Uhist(t1,nw+NWI);
        Prd=Prd+Uhist(t1,nw+NWI);
      end
      PROXY.SINK(NW-NWI).Q(t2)=(Inj-Prd);
    end
  end
end
NC=(NW-1)*(NT-nt+1);
NRUmin=(NC-1)*NC/2+2*NC+1; % NRU realizations of the control vector
NRUmax=NRUmin+100;
lb=LB*ones(NC,1);
ub=UB*ones(NC,1);
for nr=1:NRUmin
  for t=nt:NT
    Prodlast=LB-1;
    while ((Prodlast<LB) || (Prodlast>UB))
      U1=round((LB+(UB-LB).*rand(1,NW-1))*10)/10;
      Inj=sum(U1(1:NWI));
      if NWP==1
        Prodlast=Inj;
      else
        Prodlast=Inj-sum(U1(NWI+1:NWI+NWP-1));
      end
    end
    t2=t-nt;
    U(nr,t2*(NW-1)+1:(t2+1)*(NW-1))=U1;
  end
end
Uopt_all=zeros(NC,1);
NT3=DT/dt;
for nr=1:NReal
  ind1=(Real_Indx(nr)-1)*NX*NY*NZ+3;
  [K]=textread(Permfile,'%f',NX*NY*NZ,'headerlines',ind1);
  write_gslib(K,NX,NY,NZ,PROXY.Permfile);
  NPV_PR=0;
  for nru=1:NRUmin
    for t=nt:NT
      for nw=1:NWI
        PROXY.SOUR(nw).Q(NT3*(t-1)+1:NT3*t)=ones(NT3,1)*U(nru,(t-nt)*(NW-1)+nw);
      end
      for nw=NWI+1:NW-1
        PROXY.SINK(nw-NWI).Q(NT3*(t-1)+1:NT3*t)=ones(NT3,1)*U(nru,(t-nt)*(NW-1)+nw);
      end
      PROXY.SINK(NW-NWI).Q(NT3*(t-1)+1:NT3*t)=ones(NT3,1)*(sum(U(nru,(t-nt)*(NW-1)+1:(t-
nt)*(NW-1)+NWI))-sum(U(nru,(t-nt)*(NW-1)+NWI+1:(t-nt)*(NW-1)+NW-1)));
    end
    write_proxy(PROXY);
    system('/home/cmantilla/Research/polymer/Optimization/Proxy-Opt/Proxy-Polymer.exe
proxy-40.par echo=off');
    X=read_gslib('Production.dat');
    N=X(:,5);
    NPV_PR(nru,1)=N(end);
    p=ceil(nru/NRUmin*100);
    display(['Phase 1, ',int2str(p),' % completed']);
  end
  U1=U(1,:);
  Ainj=ones(1,NWI);
  Aprod=ones(1,NWP-1);
  A=zeros(2*(NT-nt+1),(NW-1)*(NT-nt+1));
  A2=[Ainj,-Aprod;-Ainj,Aprod];
  b=[];
  for t=1:NT-nt+1
```

```
    A(2*(t-1)+1:2*t,(t-1)*(NW-1)+1:t*(NW-1))=A2;
    b=[b;[UB;-LB]];
  end
  for nru=NRUmin+1:NRUmax
    [w,NPV_poly]=polyreg(U,NPV_PR);


[Uopt,Jopt]=fmincon(@(U3)proxy_polynomial(U3,w),U1',A,b,[],[],ones(NC,1)*LB,ones(NC,1)*UB
);
    U=[U;Uopt'];
    for t=nt:NT
      for nw=1:NWI
        PROXY.SOUR(nw).Q(NT3*(t-1)+1:NT3*t)=ones(NT3,1)*round(U(nru,(t-nt)*(NW-
1)+nw)*10)/10;
      end
      for nw=NWI+1:NW-1
        PROXY.SINK(nw-NWI).Q(NT3*(t-1)+1:NT3*t)=ones(NT3,1)*round(U(nru,(t-nt)*(NW-
1)+nw)*10)/10;
      end
      PROXY.SINK(NW-NWI).Q(NT3*(t-1)+1:NT3*t)=ones(NT3,1)*round((sum(U(nru,(t-nt)*(NW-
1)+1:(t-nt)*(NW-1)+NWI))-sum(U(nru,(t-nt)*(NW-1)+NWI+1:(t-nt)*(NW-1)+NW-1)))*10)/10;
    end
    write_proxy(PROXY);
    system('/home/cmantilla/Research/polymer/Optimization/Proxy-Opt/Proxy-Polymer.exe
proxy-40.par echo=off');
    X=read_gslib('Production.dat');
    N=X(:,5);
    NPV_PR(nru,1)=N(end);
    if max(abs(U(nru,:)-U(nru-1,:)))<Tol;
      break;
    end
  end
  Uopt_all=Uopt_all+Uopt/NReal;
  U=U(1:NRUmin,:);
end
```

Appendix -4 Computer Code of Proxy Model

A4-1. PROGRAM FOR OBTAINING THE PARTICLE LENGTH DISTRIBUTION

A4-1.1. Parameter File

```
**Random Walker Clustering Parameter File
40  40  1    -NX, NY, NZ
5        -Number of realizations
All_Real.dat  2    -Permeability file, column number
1        -Number of injectors
20  1  1    -Location of injectos
1        -Number of producers
20  40  1    -Location of producers
```

A4-1.2. Computer Code

```
#include <iostream>
#include <iomanip>
#include <fstream>
using namespace std;
#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#include <time.h>
#include "matrix.h"
#include "cg.h"
#include "functions.h"
#include "single_phase.h"
//global parameters
const int NP_max=1000;
const int nx_max=80, ny_max=110, nz_max=1;                          // Maximum number of cells allowed in each direction
const int nsour_max=5, nsink_max=5;                   // Maximum number of wells
int NX,NY,NZ,NSOUR,NSINK,NM,NF,NC,ND,NR;         //number of blocks in x, y  and z direction
int i,j,k,nw,nr,ns;
double VX[nx_max+1][ny_max+1][nz_max+1], VY[nx_max+1][ny_max+1][nz_max+1], VZ[nx_max+1][ny_max+1][nz_max+1];
double K1[nx_max+1][ny_max+1][nz_max+1], Kbar_sour[nsour_max+1], Kbar_sink[nsink_max+1], K_temp[11], Temp[11];
int Sources[nsour_max+1][4], Sinks[nsink_max+1][4], flag;
char tempch[150];
double x, T[NP_max+1][nsour_max+1][nsink_max+1],tvar,perm_ave;
int NP[nsour_max+1][nsink_max+1];
char Pfile[20], permfile[50];
// global variables for K-means
int nchg, nc, ncol, NCOL;
//const int nn=1000, mm=2, kk=10;
//double data[nn+1][mm+1], means[kk+1][mm+1], maxdata[mm+1], mindata[mm+1];
//int assign[nn+1], count_kmeans[kk+1];
// sub functions
void random_walker();
void Kmeans();
int estep();
void mstep();
void read_parameters(char Pfile[20]);
double DistanceFromLine(double cx, double cy, double ax, double ay ,double bx, double by);
void write_wellfile();
double *kx = new double[(nx_max+2)*(ny_max+2)*(nz_max+2)];  //permeability in x-direction
double *ky = new double[(nx_max+2)*(ny_max+2)*(nz_max+2)];  //permeability in x-direction
double *vx = new double[nx_max*ny_max*nz_max];
double *vy = new double[nx_max*ny_max*nz_max];
double *vz = new double[nx_max*ny_max*nz_max];
double Tmean[nsour_max+1][nsink_max+1];
double Tvar[nsour_max+1][nsink_max+1];
int main(int argc, char *argv[])
{
 int nsink, nsour;
 int NPT=NP_max;
 for (i=1;i<=(nx_max+2)*(ny_max+2);i++)
 {
  kx[i]=0;
  ky[i]=0;
 }
 cout<<"Clustering based on random Walkers"<<endl;
 if (argc<2)
 {
  cout<<"Enter parameter file"<<endl;
  cin>>Pfile;
  read_parameters(Pfile);
 }
 else
 {
  read_parameters(argv[1]);
 }
 write_wellfile();
 ofstream Arrival("Arrival.dat", ios::trunc);
 Arrival<<"Random walker arrival times"<<endl<<NSOUR*NSINK*3<<endl;
 for (nsink=1;nsink<=NSINK;nsink++)
  for (nsour=1;nsour<=NSOUR;nsour++)
  {
   Arrival<<"Tmean_"<<nsour<<"_"<<nsink<<endl;
   Arrival<<"Tvar_"<<nsour<<"_"<<nsink<<endl;
   Arrival<<"No_Part_"<<nsour<<"_"<<nsink<<endl;
  }
 ifstream perm_file(permfile);
 perm_file.getline(tempch,150);
 perm_file>>NCOL;
 perm_file.getline(tempch,150);
 for (i=1;i<=NCOL;i++)
  perm_file.getline(tempch,150);
 for (nr=1;nr<=NR;nr++)
 {
  for (k=1;k<=NZ;k++)
   for (j=1;j<=NY;j++)
    for (i=1;i<=NX;i++)
    {
     for (nc=1;nc<=NCOL;nc++)
     {
      perm_file>>x;
      if (nc==ncol)
       K1[i][j][k]=x;  //permeability in y-direction
     }
     kx[j*(NX+2)+i]=K1[i][j][k];
     kx[j*(NX+2)+i]=kx[j*(NX+2)+i]*0.9869E-15;  // Conversion to SI units
     ky[j*(NX+2)+i]=kx[j*(NX+2)+i];
     vx[(k-1)*NX*NY+(j-1)*NX+i] = 0;
     vy[(k-1)*NX*NY+(j-1)*NX+i] = 0;
     vz[(k-1)*NX*NY+(j-1)*NX+i] = -1;
    }
  single_phase(NX,NY,20*0.3048,20*0.3048,8*0.3048,kx,ky,0.1,vx,vy,vz);
  for (k=1;k<=NZ;k++)
   for (j=1;j<=NY;j++)
    for (i=1;i<=NX;i++)
    {
     VX[i][j][k]=vx[(k-1)*NX*NY+(j-1)*NX+i];
     VY[i][j][k]=vy[(k-1)*NX*NY+(j-1)*NX+i];
     VZ[i][j][k]=vz[(k-1)*NX*NY+(j-1)*NX+i];
    }
  cout<<"Working on Realization # "<<nr<<endl;
  random_walker();
  for (nsink=1;nsink<=NSINK;nsink++)
   for (nsour=1;nsour<=NSOUR;nsour++)
   {
    Arrival<<Tmean[nsour][nsink]<<"  "<<Tvar[nsour][nsink]<<"  ";
    Arrival<<double(NP[nsour][nsink])/double(NP_max)<<"  ";
   }
  Arrival<<endl;
 }
 Arrival.close();
 return 0;
}
void random_walker()
{
 srand ( time(NULL) );
 int i,j,k, dist;
 double cdf, pdf,sum, d;
 double Pr[7],x,t1;
 double T[NP_max+1][nsour_max+1][nsink_max+1];
 int pos1[NP_max+1][4], IALL[nx_max+1][ny_max+1][nz_max+1];
 int ns,p,np, arrive, I[nx_max+1][nx_max+1][nz_max+1],nsink,nsour,cnt;
 t1=time(NULL);
 for (nsink=1;nsink<=NSINK;nsink++)
  for (nsour=1;nsour<=NSOUR;nsour++)
  {
   Tmean[nsour][nsink]=0;
   Tvar[nsour][nsink]=0;
   NP[nsour][nsink]=0;
   for (np=1;np<=NP_max;np++)
    T[np][nsour][nsink]=0;
  }

 for (k=1; k<=NZ; k++)
  for (j=1; j<=NY; j++)
   for (i=1; i<=NX; i++)
    {IALL[i][j][k]=0;}
 for (np=1; np<=NP_max; np++)
 {
  ns=fmod(np,NSOUR)+1;
  pos1[np][1]=Sources[ns][1];
  pos1[np][2]=Sources[ns][2];
  pos1[np][3]=Sources[ns][3];
  arrive=0;
  for (k=1; k<=NZ; k++)
   for (j=1; j<=NY; j++)
    for (i=1; i<=NX; i++)
     {I[i][j][k]=0;}
  cnt=0;
  do
  {
   cnt++;
   for (p=1; p<=6; p++)
    Pr[p]=0;
   i=pos1[np][1];
   j=pos1[np][2];
   k=pos1[np][3];
   I[i][j][k]=1;
   if (i>1) {  if (VX[i-1][j][k]<0) Pr[1]=-VX[i-1][j][k];}
   if (i<NX) {  if (VX[i][j][k]>0) Pr[2]=VX[i][j][k];}
   if (j>1) {  if (VY[i][j-1][k]<0) Pr[3]=-VY[i][j-1][k];}
   if (j<NY) {  if (VY[i][j][k]>0) Pr[4]=VY[i][j][k];}
   if (k>1) {  if (VZ[i][j][k-1]<0) Pr[5]=-VZ[i][j][k-1];}
   if (k<NZ) {  if (VZ[i][j][k+1]>0) Pr[6]=VZ[i][j][k];}
   sum=0;
```

```
for (p=1; p<=6; p++)
  {sum+=Pr[p];}
if (sum>0)
{
  x=rand()%100;
  cdf=0;
  for (p=1; p<=6; p++)
  {
    pdf=Pr[p]/sum;
    cdf=cdf+pdf;
    if (x<cdf*100) {break;}
  }
  switch (p)
  {
    case 1:
      pos1[np][1]=i-1;
      break;
    case 2:
      pos1[np][1]=i+1;
      break;
    case 3:
      pos1[np][2]=j-1;
      break;
    case 4:
      pos1[np][2]=j+1;
      break;
    case 5:
      pos1[np][3]=k-1;
      break;
    case 6:
      pos1[np][3]=k+1;
      break;
  }
}
for (nsink=1; nsink<=NSINK; nsink++)
{
  if ((pos1[np][1]==Sinks[nsink][1]) && (pos1[np][2]==Sinks[nsink][2]) && (pos1[np][3]==Sinks[nsink][3]))
  {
    d=sqrt((double)(pow((double)(Sinks[nsink][1]-Sources[ns][1]),2)+pow((double)(Sinks[nsink][2]-Sources[ns][2]),2)
      +pow((double)(Sinks[nsink][3]-Sources[ns][3]),2)));
    arrive=1;
    sum=0;
    dist=0;
    for (k=1; k<=NZ; k++)
      for (j=1; j<=NY; j++)
        for (i=1; i<=NX; i++)
        {
            sum+=I[i][j][k];
            IALL[i][j][k]=IALL[i][j][k]+I[i][j][k];
            if (I[i][j][k]==1)
            {
              dist+=DistanceFromLine(double(i),double(j),double(Sources[ns][1]), double(Sources[ns][2]),
                double(Sinks[nsink][1]), double(Sinks[nsink][2]));
            }
        }
    dist/=sum;
    NP[ns][nsink]++;
    T[NP[ns][nsink]][ns][nsink]=dist;
  }
}
if (cnt>3000)
{
  cout<<"Particle lost"<<endl;
  ofstream track("Trajectory.dat", ios::trunc);
  track<<"Trajectory"<<endl<<"1"<<endl<<"Path"<<endl;
```

```
    for (k=1;k<=NZ;k++)
     for (j=1;j<=NY;j++)
      for (i=1;i<=NX;i++)
       track<<I[i][j][k]<<endl;
    track.close();
    exit(1);
   }
  }
  while (arrive==0);
 }
 for (nsink=1;nsink<=NSINK;nsink++)
  for (nsour=1;nsour<=NSOUR;nsour++)
  {
   Tmean[nsour][nsink]=0;
   for (np=1;np<=NP[nsour][nsink];np++)
    Tmean[nsour][nsink]+=T[np][nsour][nsink]/NP[nsour][nsink];
  }
 for (nsink=1;nsink<=NSINK;nsink++)
  for (nsour=1;nsour<=NSOUR;nsour++)
  {
   Tvar[nsour][nsink]=0;
   for (np=1;np<=NP[nsour][nsink];np++)
    Tvar[nsour][nsink]+=(T[np][nsour][nsink]-Tmean[nsour][nsink])*(T[np][nsour][nsink]-Tmean[nsour][nsink])/NP[nsour][nsink];
  }
 t1=time(NULL)-t1;
 ofstream track("Trajectory.dat", ios::trunc);
 track<<"Trajectory"<<endl<<"1"<<endl<<"Path"<<endl;
 sum=0;
 perm_ave=0;
 for (k=1;k<=NZ;k++)
  for (j=1;j<=NY;j++)
   for (i=1;i<=NX;i++)
   {
    track<<IALL[i][j][k]<<endl;
    sum+=IALL[i][j][k];
    perm_ave+=K1[i][j][k]*IALL[i][j][k];
   }
 perm_ave=perm_ave/sum;
 track.close();
 return;
}
/*void Kmeans()
{
 int k,m;
 for (k=1;k<=kk;k++)
  for (m=1;m<=mm;m++)
   means[k][m]=mindata[m]+(maxdata[m]-mindata[m])*(k-1)/kk;
 do
 {
  estep();
  mstep();
 }
 while (nchg>0);
}

int estep()
{
 int k,m,n,kmin;
 double dmin,d;
 nchg = 0;
 for (k=1;k<=NC;k++) count_kmeans[k] = 0;
 for (n=1;n<=ND;n++) {
  dmin = 9.99e99;
  for (k=1;k<=NC;k++) {
   for (d=0.,m=1; m<=mm; m++) d += pow((double)data[n][m]-means[k][m],2);
   if (d < dmin) {dmin = d; kmin = k;}
  }
  if (kmin != assign[n]) nchg++;
  assign[n] = kmin;
  count_kmeans[kmin]++;
 }
 return nchg;
}
void mstep() {
 int n,k,m;
 for (k=1;k<=NC;k++) for (m=1;m<=mm;m++) means[k][m] = 0.;
 for (n=1;n<=ND;n++) for (m=1;m<=mm;m++) means[assign[n]][m] += data[n][m];
 for (k=1;k<=NC;k++) {
  if (count_kmeans[k] > 0) for (m=1;m<=mm;m++) means[k][m] /= count_kmeans[k];
 }
}*/
void read_parameters(char Pfile[20])
{
 int ns;
 ifstream param_file(Pfile);
 if (!param_file)
 {
  cout<<"There is no Data File"<<endl;
  exit(1);
 }
 param_file.getline(tempch,150);
 param_file>>NX>>NY>>NZ;
 param_file.getline(tempch,150,'\n');
 param_file>>NR;
 param_file.getline(tempch,150,'\n');
 param_file>>permfile;
 ifstream perm_file(permfile);
 if (!perm_file)
 {
  cout<<"File "<<permfile<<" does not exist"<<endl;
  exit(1);
 }
 param_file>>ncol;
 param_file.getline(tempch,150,'\n');
 param_file>>NSOUR;
 param_file.getline(tempch,150,'\n');
 for (ns=1;ns<=NSOUR;ns++)
 {
  param_file>>Sources[ns][1]>>Sources[ns][2]>>Sources[ns][3];
  param_file.getline(tempch,150,'\n');
 }
 param_file>>NSINK;
 param_file.getline(tempch,150,'\n');
 for (ns=1;ns<=NSINK;ns++)
 {
  param_file>>Sinks[ns][1]>>Sinks[ns][2]>>Sinks[ns][3];
  param_file.getline(tempch,150,'\n');
 }
 param_file.close();
}

void write_wellfile()   // write well data file for simulator
{
 ofstream wellfl("well.dat", ios::trunc);
 wellfl<<"0"<<endl;   // start time
 wellfl<<"1"<<endl;  // number of time step entries
 wellfl<<"1  5"<<endl;  // entry # 1 - 2 steps of 1 day
 wellfl<<NSOUR+NSINK<<endl;    // total number of wells
 for (ns=1;ns<=NSOUR;ns++)
  wellfl<<Sources[ns][1]<<"  "<<Sources[ns][2]<<endl;      // Location of sources (injetors)
  for (ns=1;ns<=NSINK;ns++)
    wellfl<<Sinks[ns][1]<<"  "<<Sinks[ns][2]<<endl;      // Location of sinks (producers)
  for (ns=1;ns<=NSOUR;ns++)
    wellfl<<"0  "<<ns<<"  6.048e-5  Q  "<<20/35.314666<<endl;
  for (ns=1;ns<=NSINK;ns++)
    wellfl<<"0  "<<ns+NSOUR<<"  6.048e-5  Q  "<<-20/35.314666<<endl;
  wellfl.close();
}

double DistanceFromLine(double cx, double cy, double ax, double ay ,
            double bx, double by)
{
  double distanceLine, distanceSegment;
  //
  // find the distance from the point (cx,cy) to the line
  // determined by the points (ax,ay) and (bx,by)
  //
  // distanceSegment = distance from the point to the line segment
  // distanceLine = distance from the point to the line (assuming
  //          infinite extent in both directions
  //
  double r_numerator = (cx-ax)*(bx-ax) + (cy-ay)*(by-ay);
  double r_denomenator = (bx-ax)*(bx-ax) + (by-ay)*(by-ay);
  double r = r_numerator / r_denomenator;
//
    double px = ax + r*(bx-ax);
    double py = ay + r*(by-ay);
//
    double s =  ((ay-cy)*(bx-ax)-(ax-cx)*(by-ay) ) / r_denomenator;
  distanceLine = fabs(s)*sqrt(r_denomenator);
// (xx,yy) is the point on the lineSegment closest to (cx,cy)
//
  double xx = px;
  double yy = py;

  if ( (r >= 0) && (r <= 1) )
  {
    distanceSegment = distanceLine;
  }
  else
  {
    double dist1 = (cx-ax)*(cx-ax) + (cy-ay)*(cy-ay);
    double dist2 = (cx-bx)*(cx-bx) + (cy-by)*(cy-by);
    if (dist1 < dist2)
    {
      xx = ax;
      yy = ay;
      distanceSegment = sqrt(dist1);
    }
    else
    {
      xx = bx;
      yy = by;
      distanceSegment = sqrt(dist2);
    }
  }
  return distanceLine;
}
```

A4-2. COMPUTER CODE FOR ESTIMATION OF RECOVERY FACTOR AND BREAKTHROUGH TIME

```
#include <iostream>
#include <iomanip>
#include <fstream>
using namespace std;
#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#include <time.h>
//global parameters
int NX,NY,NZ,NSOUR,NSINK,NR;        //number of blocks in x, y  and z direction
int i,j,k,nw,nr,count1,n,ns;
int ip,jp,kp,iw,jw,kw;
const int nx_max=100, ny_max=100, nz_max=1, nsour_max=10, nsink_max=10;
const int nreal_max=1000, np_max=50000;
double                                                          VX[nx_max+1][ny_max+1][nz_max+1],
VY[nx_max+1][ny_max+1][nz_max+1],VZ[nx_max+1][ny_max+1][nz_max+1];
int Sources[nsour_max+1][4], Sinks[nsink_max+1][4], NCOL, ncol,nc;
char tempch[150];
double x, RF[nreal_max+1],TBT[nsour_max*nsink_max+1],TD;
char Pfile[20], permfile[60];
// Variables for Random Walker
double dmax, d, cdf, pdf,sum;
int p,np,IP[nx_max+1][ny_max+1][nz_max+1],IPW[nx_max+1][ny_max+1][nz_max+1],nsink,t,tt,flag;
int NPTotal, pos1[np_max+1][4];
double Pr[7];
double *kx = new double[(nx_max+2)*(ny_max+2)*(nz_max+2)];
double *ky = new double[(nx_max+2)*(ny_max+2)*(nz_max+2)];
double *vx = new double[nx_max*ny_max*nz_max];
double *vy = new double[nx_max*ny_max*nz_max];
double *vz = new double[nx_max*ny_max*nz_max];
int NPmax=7,NPmaxW=2;
// sub functions
void random_walker(double& rf);
void read_parameters(char Pfile[20]);
void move_water_particle();
void move_polymer_particle();
void write_wellfile();
void single_phase(int nx, int ny, double DX, double DY, double DZ, double *kx, double *ky, double Poro,
     double *vx, double *vy, double *vz);
int main(int argc, char *argv[])
{
 double rf;
 for (i=1;i<=(nx_max+2)*(ny_max+2);i++)
 {
  kx[i]=0;
  ky[i]=0;
 }
 cout<<"Clustering based on random Walkers"<<endl<<"Name of parameter file "<<endl;
 if (argc<2)
 {
  cout<<"Enter parameter file"<<endl;
  cin>>Pfile;
  read_parameters(Pfile);
 }
 else
 {
  read_parameters(argv[1]);
 }
 write_wellfile();
 dmax=sqrt((double)(pow((double)NX,2)+pow((double)NY,2)+pow((double)NZ,2)));
 ofstream Recovery("Recovery.dat", ios::trunc);
```

```
  Recovery<<"Random Walker Recovery Factor"<<endl<<NSOUR*NSINK+1<<endl<<"RF"<<endl;
  for (ns=1;ns<=NSOUR*NSINK;ns++)
    Recovery<<"TBT_"<<ns<<endl;
  ifstream k_file(permfile);
  k_file.getline(tempch,150);
  k_file>>NCOL;
  k_file.getline(tempch,150);
  ofstream VelfileX("VelX.dat",ios::trunc);
  ofstream VelfileY("VelY.dat",ios::trunc);
  ofstream VelfileZ("VelZ.dat",ios::trunc);
  VelfileX<<"VelX"<<endl<<"1"<<endl<<"Values"<<endl;
  VelfileY<<"VelY"<<endl<<"1"<<endl<<"Values"<<endl;
  VelfileZ<<"VelZ"<<endl<<"1"<<endl<<"Values"<<endl;
  for (i=1;i<=NCOL;i++)
    k_file.getline(tempch,150);
  for (nr=1;nr<=NR;nr++)
  {
    cout<<"Working on Realization # "<<nr<<endl;
    for (k=1;k<=NZ;k++)
      for (j=1;j<=NY;j++)
        for (i=1;i<=NX;i++)
        {
          for (nc=1;nc<=NCOL;nc++)
          {
            k_file>>x;
            if (nc==ncol)
              kx[j*(NX+2)+i]=x;  //permeability in y-direction
          }
          kx[j*(NX+2)+i]=kx[j*(NX+2)+i]*0.9869E-15;  // Conversion to SI units
          ky[j*(NX+2)+i]=kx[j*(NX+2)+i];
          vx[(k-1)*NX*NY+(j-1)*NX+i] = 0;
          vy[(k-1)*NX*NY+(j-1)*NX+i] = 0;
          vz[(k-1)*NX*NY+(j-1)*NX+i] = -1;
        }
    single_phase(NX,NY,20*0.3048,20*0.3048,20*0.3048,kx,ky,0.1,vx,vy,vz);
    for (k=1;k<=NZ;k++)
      for (j=1;j<=NY;j++)
        for (i=1;i<=NX;i++)
        {
          VX[i][j][k]=vx[(k-1)*NX*NY+(j-1)*NX+i];
          VY[i][j][k]=vy[(k-1)*NX*NY+(j-1)*NX+i];
          VZ[i][j][k]=vz[(k-1)*NX*NY+(j-1)*NX+i];
          VelfileX<<VX[i][j][k]<<endl;
          VelfileY<<VY[i][j][k]<<endl;
          VelfileZ<<VY[i][j][k]<<endl;
        }
    random_walker(rf);
    RF[nr]=rf;
    Recovery<<RF[nr]<<"  ";
    for (ns=1;ns<=NSOUR;ns++)
      for (nsink=1;nsink<=NSINK;nsink++)
        Recovery<<TBT[NSOUR*(ns-1)+nsink]<<"  ";
    Recovery<<endl;
  }
  Recovery.close();
  VelfileX.close();
  VelfileY.close();
  VelfileZ.close();
  return 0;
}

void random_walker(double& rf)
{
  const int nsour_max=10, nsink_max=10;
  int SW_INI=1, arrive_w=0, flag_particle, flag_tbt[nsour_max*nsink_max+1];
  srand ( time(NULL) );
  dmax=sqrt(pow((double)NX,2)+pow((double)NY,2)+pow((double)NZ,2));
  for (k=1;k<=NZ;k++)
    for (j=1;j<=NY;j++)
      for (i=1;i<=NX;i++)
      {
        IP[i][j][k]=0;
        IPW[i][j][k]=SW_INI;
      }
  NPTotal=ceil(NX*NY*NZ*NPmax*TD);  //Total number of particles to be launched, 5 is the maximum capacity of each
node
  for (ns=1;ns<=NSOUR;ns++)
    for (nsink=1;nsink<=NSINK;nsink++)
    {
      flag_tbt[NSOUR*(ns-1)+nsink]=0;
      TBT[NSOUR*(ns-1)+nsink]=15000+rand()%300;
    }
  for (n=1;n<=NPTotal;n++)
  {
    ns=fmod(n,NSOUR)+1;
    pos1[n][1]=Sources[ns][1];
    pos1[n][2]=Sources[ns][2];
    pos1[n][3]=Sources[ns][3];
    IP[pos1[n][1]][pos1[n][2]][pos1[n][3]]++;
    flag_particle=0;
    while (flag_particle==0)
    {
      if ((IPW[pos1[n][1]][pos1[n][2]][pos1[n][3]]+IP[pos1[n][1]][pos1[n][2]][pos1[n][3]]>NPmax)
        || (IPW[pos1[n][1]][pos1[n][2]][pos1[n][3]]>NPmaxW))
      {
        if (IPW[pos1[n][1]][pos1[n][2]][pos1[n][3]]==0)
        {
        //Move polymer particle
          move_polymer_particle();
          IP[pos1[n][1]][pos1[n][2]][pos1[n][3]]--;
          IP[ip][jp][kp]++;
          if (IP[ip][jp][kp]+IPW[ip][jp][kp]<=NPmax)
          {
            flag_particle=1;
          }
          for (nsink=1; nsink<=NSINK; nsink++)
          {
            if ((ip==Sinks[nsink][1]) && (jp==Sinks[nsink][2]) && (kp==Sinks[nsink][3]))
            {
              IP[ip][jp][kp]--;
            }
          }
          pos1[n][1]=ip;
          pos1[n][2]=jp;
          pos1[n][3]=kp;

        }
        else
        {
          //Move water particle
          move_water_particle();
          IPW[pos1[n][1]][pos1[n][2]][pos1[n][3]]--;
          IPW[iw][jw][kw]++;
          if ((IPW[iw][jw][kw]<=NPmaxW) &&
            (IP[iw][jw][kw]+IPW[iw][jw][kw]<=NPmax))
          {
            flag_particle=1;
          }
          for (nsink=1; nsink<=NSINK; nsink++)
          {
```

```
      if ((iw==Sinks[nsink][1]) && (jw==Sinks[nsink][2]) && (kw==Sinks[nsink][3]))
      {
       IPW[iw][jw][kw]--;
       if (flag_tbt[NSOUR*(ns-1)+nsink]==0)
       {
        TBT[NSOUR*(ns-1)+nsink]=n;
        flag_tbt[NSOUR*(ns-1)+nsink]=1;
       }
      }
     }
     pos1[n][1]=iw;
     pos1[n][2]=jw;
     pos1[n][3]=kw;
     }
   }
   else
   {
    flag_particle=1;
   }
  }
 }
 sum=0;
 for (k=1;k<=NZ;k++)
  for (j=1;j<=NY;j++)
   for (i=1;i<=NX;i++)
    sum+=IP[i][j][k]+IPW[i][j][k];
 sum-=SW_INI*NX*NY*NZ;
 rf=sum/(NX*NY*NZ*NPmax);
}

void read_parameters(char Pfile[20])
{
 int ns;
 ifstream param_file(Pfile);
 if (!param_file)
 {
  cout<<"There is no Data File"<<endl;
  exit(1);
 }
 param_file.getline(tempch,150);
 param_file>>NX>>NY>>NZ;
 param_file.getline(tempch,150,'\n');
 param_file>>NR;
 param_file.getline(tempch,150,'\n');
 param_file>>permfile;
 ifstream k_file(permfile);
 if (!k_file)
 {
  cout<<"Permeability file "<<permfile<<" does not exist"<<endl;
  exit(1);
 }
 k_file.close();
 param_file>>ncol;
 param_file.getline(tempch,150,'\n');
 param_file>>NSOUR;
 param_file.getline(tempch,150,'\n');
 for (ns=1;ns<=NSOUR;ns++)
 {
  param_file>>Sources[ns][1]>>Sources[ns][2]>>Sources[ns][3];
  param_file.getline(tempch,150,'\n');
 }
 param_file>>NSINK;
 param_file.getline(tempch,150,'\n');
 for (ns=1;ns<=NSINK;ns++)
 {
  param_file>>Sinks[ns][1]>>Sinks[ns][2]>>Sinks[ns][3];
  param_file.getline(tempch,150,'\n');
 }
 param_file>>TD;
 param_file.getline(tempch,150,'\n');
 param_file.close();
}

void move_polymer_particle()
{
 ip=pos1[n][1];
 jp=pos1[n][2];
 kp=pos1[n][3];
 for (p=1; p<=6; p++)
  Pr[p]=0;
 if (ip>1) {  if (VX[ip-1][jp][kp]<0) Pr[1]=-VX[ip-1][jp][kp];}
 if (ip<NX) {  if (VX[ip][jp][kp]>0) Pr[2]=VX[ip][jp][kp];}
 if (jp>1) {  if (VY[ip][jp-1][kp]<0) Pr[3]=-VY[ip][jp-1][kp];}
 if (jp<NY) {  if (VY[ip][jp][kp]>0) Pr[4]=VY[ip][jp][kp];}
 if (kp>1) {  if (VZ[ip][jp][kp-1]<0) Pr[5]=-VZ[ip][jp][kp-1];}
 if (kp<NZ) {  if (VZ[ip][jp][kp+1]>0) Pr[6]=VZ[ip][jp][kp];}
 sum=0;
 for (p=1; p<=6; p++)
  {sum+=Pr[p];}
 if (sum>0)
 {
  x=rand()%100;
  cdf=0;
  for (p=1; p<=6; p++)
  {
   pdf=Pr[p]/sum;
   cdf=cdf+pdf;
   if (x<cdf*100) {break;}
  }
  switch (p)
  {
   case 1:
    ip=ip-1;
    break;
   case 2:
    ip=ip+1;
    break;
   case 3:
    jp=jp-1;
    break;
   case 4:
    jp=jp+1;
    break;
   case 5:
    kp=kp-1;
    break;
   case 6:
    kp=kp+1;
    break;
  }
 }
}
void move_water_particle()
{
 iw=pos1[n][1];
 jw=pos1[n][2];
 kw=pos1[n][3];
 for (p=1; p<=6; p++)
  Pr[p]=0;
 if (iw>1) {  if (VX[iw-1][jw][kw]<0) Pr[1]=-VX[iw-1][jw][kw];}
 if (iw<NX) {  if (VX[iw][jw][kw]>0) Pr[2]=VX[iw][jw][kw];}
 if (jw>1) {  if (VY[iw][jw-1][kw]<0) Pr[3]=-VY[iw][jw-1][kw];}
 if (jw<NY) {  if (VY[iw][jw][kw]>0) Pr[4]=VY[iw][jw][kw];}
 if (kw>1) {  if (VZ[iw][jw][kw-1]<0) Pr[5]=-VZ[iw][jw][kw-1];}
 if (kw<NZ) {  if (VZ[iw][jw][kw+1]>0) Pr[6]=VZ[iw][jw][kw];}
 sum=0;
 for (p=1; p<=6; p++)
  {sum+=Pr[p];}
 if (sum>0)
 {
  x=rand()%100;
  cdf=0;
  for (p=1; p<=6; p++)
  {
   pdf=Pr[p]/sum;
   cdf=cdf+pdf;
   if (x<cdf*100) {break;}
  }
  switch (p)
  {
   case 1:
    iw=iw-1;
    break;
   case 2:
    iw=iw+1;
    break;
   case 3:
    jw=jw-1;
    break;
   case 4:
    jw=jw+1;
    break;
   case 5:
    kw=kw-1;
    break;
   case 6:
    kw=kw+1;
    break;
  }
 }
}
void write_wellfile()   // write well data file for simulator
{
 ofstream wellfl("well.dat", ios::trunc);
 wellfl<<"0"<<endl;   // start time
 wellfl<<"1"<<endl;  // number of time step entries
 wellfl<<"1  5"<<endl;  // entry # 1 - 2 steps of 1 day
 wellfl<<NSOUR+NSINK<<endl;    // total number of wells
 for (ns=1;ns<=NSOUR;ns++)
  wellfl<<Sources[ns][1]<<"  "<<Sources[ns][2]<<endl;      // Location of sources (injetors)
 for (ns=1;ns<=NSINK;ns++)
  wellfl<<Sinks[ns][1]<<"  "<<Sinks[ns][2]<<endl;      // Location of sinks (producers)
 for (ns=1;ns<=NSOUR;ns++)
  wellfl<<"0  "<<ns<<"  6.048e-5  Q  "<<2.83168/NSOUR<<endl;
 for (ns=1;ns<=NSINK;ns++)
  wellfl<<"0  "<<ns+NSOUR<<"  6.048e-5  Q  "<<-2.83168/NSINK<<endl;
 wellfl.close();
}
```

A4-3. PROGRAM FOR ECONOMIC PARAMETERS

A4-3.1. Parameter File

```
40  40  1       - Grid size
20  20  8      - DX, DY, DZ
2          - No. Sources
40  1  1      - Location of sources
1  1  1
1          - No. sinks
20  40  1      - Location of sinks
0.1          - Porosity
100          - No. Time steps
30  100  0  100    - Rate Schedule
…
3000  100  0  100
3.9          - Water Velocity factor
0.65          - Polymer Velocity factor
0.3  0.78        - Swf, Spf
0.2  0.2        - Swr, Sor
0.2  1.5  0.73      - Krw_max, exp_w, Vw
1.0  2.0  40      - Kro_max, exp_o, Vo
0.2  1.5  20      - Krp_max, exp_p, Vp
 Perm_3W.dat        - Permeability File
```

A4-3.2. Computer Code

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <algorithm>
using namespace std;
#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#include <time.h>
#include "matrix.h"
#include "cg.h"
#include "functions.h"
#include "single_phase.h"

const int nx_max=100, ny_max=110, nz_max=1;                         // Maximum number of cells allowed in each direction
const int nsour_max=10, nsink_max=10;                  // Maximum number of wells
const int NP=10000, nt_max=100;    // Number of particles
double Vx[nx_max+1][ny_max+1][nz_max+1], Vy[nx_max+1][ny_max+1][nz_max+1], Vz[nx_max+1][ny_max+1][nz_max+1];
double                              VxT[nx_max+1][ny_max+1][nz_max+1],                              VyT[nx_max+1][ny_max+1][nz_max+1],
VzT[nx_max+1][ny_max+1][nz_max+1];
double DX, DY, DZ, xP[NP+1], yP[NP+1], zP[NP+1], TimeP[NP+1];
double xW[NP+1], yW[NP+1], zW[NP+1], TimeW[NP+1], I[nx_max+1][ny_max+1][nz_max+1];
double t1, Poro, T[nt_max+1],Qi[nsour_max+1][nt_max+1], Qp[nsink_max+1][nt_max+1], Qw,Qo, Vwfac, Vpfac;
double Qi_tot,Sw_ini,RF,Qinj[nsour_max+1];
double Sw[nsink_max+1], So, Sp, Krw_max, Kro_max, Krp_max, Krw, Kro, Krp, Vw, Vo, Vp, Swf, Spf, Sor;
double exp_w, exp_o, exp_p, Sowf, Sopf, Krofw, Krofp, Krwf, Krpf, Fw, Fp;
int NX, NY, NZ, NSOUR, NSINK, NT;
int nt, np, read_flag;
int Sources[nsour_max+1][4], Sinks[nsink_max+1][4];
int ns, narg;
double conc=750E-6, tax=0.385, royal=0.125, apr=0.1, wfcost=1000, watinjcost=1, watprodcost=10;
double oilprodcost=4, polymasscost=1, oilprice=35,tslug[3]={0,3000};
// SUBROUTINES

void read_parameters(char *Pfile);
void frac_flow();
```

```
double NetPresent(double& NPV, double wc);
void write_wellfile();
double min(double a, double b)
{
  if (a<b) {return a;}
  else {return b;}
}

double *kx = new double[(nx_max+2)*(ny_max+2)*(nz_max+2)];  //permeability in x-direction
double *ky = new double[(nx_max+2)*(ny_max+2)*(nz_max+2)];  //permeability in x-direction
double *vx = new double[nx_max*ny_max*nz_max];
double *vy = new double[nx_max*ny_max*nz_max];
double *vz = new double[nx_max*ny_max*nz_max];

int main(int argc, char *argv[])
{
  int i, j, k, ip, jp, kp, p,nsink, nsour, Npw, Npp, NPWleft, NPPleft;
  double Pr[7], sum, cdf, pdf, dt, r, wc, NPV, dx1, dx2, dx, dy, dz, dmin, d;
  double rate_factor, x,y, frac[nsour_max+1], Qinj_all,wc_sink[nsink_max+1];
  double Sw_sink[nsink_max+1], FracW_sink[nsink_max+1], FracP_sink[nsink_max+1];
  bool flag, get_velocity;
  int arriveW[NP+1], arriveP[NP+1],part_countW[nsink_max+1][nsour_max+1],part_countP[nsink_max+1][nsour_max+1];
  int NPartW_sink[nsink_max+1], NPartP_sink[nsink_max+1];
  char Pfile[20];
  narg=argc;
   for (i=1;i<=(nx_max+2)*(ny_max+2);i++)
  {
    kx[i]=0;
    ky[i]=0;
  }
  if (argc<2)
  {
    cout<<"Enter parameter file"<<endl;
    cin>>Pfile;
    read_parameters(Pfile);
  }
  else
  {
    read_parameters(argv[1]);
  }
  srand ( time(NULL) );
  t1=time(NULL);
  T[0]=0;
  frac_flow();
  for (np=1; np<=NP; np++)
  {
    ns=fmod(np,NSOUR)+1;
    xP[np]=(Sources[ns][1]-0.5)*DX;
    yP[np]=(Sources[ns][2]-0.5)*DY;
    zP[np]=(Sources[ns][3]-0.5)*DZ;
    xW[np]=(Sources[ns][1]-0.5)*DX;
    yW[np]=(Sources[ns][2]-0.5)*DY;
    zW[np]=(Sources[ns][3]-0.5)*DZ;
    arriveW[np]=0;
    arriveP[np]=0;
  }
  wc=0;
  ofstream positionfl("Positions.dat",ios::trunc);
  positionfl<<"Particle Positions"<<endl;
  positionfl<<"4"<<endl<<"X"<<endl<<"Y"<<endl<<"Z"<<endl<<"Particle Indicator"<<endl;
  ofstream trajfile("Trajectory.dat",ios::trunc);
  trajfile<<"Trajectory"<<endl;
  trajfile<<"1"<<endl<<"Number of Particles"<<endl;
  ofstream prodfl("Production.dat",ios::trunc);
  prodfl<<"Production File"<<endl<<5+NSINK<<endl<<"Time (days)"<<endl<<"Qw (cu.ft/day)"<<endl<<"Qo (cu.ft/day)"<<endl;
```

```
prodfl<<"Water Cut"<<endl<<"Net Present Value"<<endl;
for (nsink=1;nsink<=NSINK;nsink++)
 prodfl<<"Water Cut Well # "<<nsink<<endl;
for (i=1; i<=NX; i++)
 for (j=1; j<=NY; j++)
  for (k=1; k<=NZ; k++)
    I[i][j][k]=Sw_ini;
for (np=1; np<=NP; np++)
{
 TimeP[np]=0;
 TimeW[np]=0;
}
NPV=0;
get_velocity=true;
Qinj_all=0;
Qi_tot=0;
for (nsour=1;nsour<=NSOUR;nsour++)
{
 Qi_tot+=Qi[ns][1];
}

for (nsink=1;nsink<=NSINK;nsink++)
{
 Sw[nsink]=Sw_ini;
 FracW_sink[nsink]=Qp[nsink][1]/Qi_tot;
 FracP_sink[nsink]=Qp[nsink][1]/Qi_tot;
 for (nsour=1;nsour<=NSOUR;nsour++)
 {
  part_countW[nsink][nsour]=0;
  part_countP[nsink][nsour]=0;
 }
}
NPWleft=NP*NSOUR;
NPPleft=NP*NSOUR;
for (nt=1; nt<=NT; nt++)
{
 rate_factor=1;
 Qi_tot=0;
 for (ns=1;ns<=NSOUR;ns++)
 {
  Qi_tot+=Qi[ns][nt];
  Qinj[ns]+=Qi[ns][nt]*(T[nt]-T[nt-1]);
  Qinj_all+=Qi[ns][nt]*(T[nt]-T[nt-1]);
  frac[ns]=0;
 }
 if (nt>1)
 {
  rate_factor=Qi[1][nt]/Qi[1][nt-1];
  if (Qi[1][nt-1]==0)
  {
   if (Qi[1][nt]==0)
   {  rate_factor=1;}
   else
   {
    rate_factor=1;
    get_velocity=true;}
  }
  for (ns=1;ns<=NSOUR;ns++)
   if (Qi[ns][nt]!=rate_factor*Qi[ns][nt-1])
   {
    rate_factor=1;
    get_velocity=true;
   }
  for (ns=1;ns<=NSINK;ns++)
   if (Qp[ns][nt]!=rate_factor*Qp[ns][nt-1])
   {
    rate_factor=1;
    get_velocity=true;
   }
 }
 if (get_velocity)
 {
  for (nsink=1;nsink<=NSINK;nsink++)
  {
   Sw_sink[nsink]=min(Sw[nsink],Swf);
   for (nsour=1;nsour<=NSOUR;nsour++)
   {
    NPWleft-=part_countW[nsink][nsour];
    NPPleft-=part_countP[nsink][nsour];
    part_countW[nsink][nsour]=0;
    part_countP[nsink][nsour]=0;
    NPartW_sink[nsink]=0;
    NPartP_sink[nsink]=0;
   }
  }
  write_wellfile();
  single_phase(NX,NY,DX*0.3048,DY*0.3048,DZ*0.3048,kx,ky,Poro,vx,vy,vz);
  if (nt>1)
  {
   for (np=1;np<=NP;np++)
   {
    if (arriveW[np]==0)
    {
     ip=(int)floor((xW[np]+0.5*DX)/DX);
     jp=(int)floor((yW[np]+0.5*DY)/DY);
     kp=(int)floor((zW[np]+0.5*DZ)/DZ);
     dmin=99999;
     for (nsink=1;nsink<=NSINK;nsink++)
     {
      d=pow((ip-Sinks[nsink][1])*(ip-Sinks[nsink][1])+(jp-Sinks[nsink][2])*(jp-Sinks[nsink][2])
       +(kp-Sinks[nsink][2])*(kp-Sinks[nsink][3]),0.5);
      if (dmin>d)
      {
       dmin=d;
       ns=nsink;
      }
     }
     NPartW_sink[ns]++;
    }
    if (arriveP[np]==0)
    {
     ip=(int)floor((xP[np]+0.5*DX)/DX);
     jp=(int)floor((yP[np]+0.5*DY)/DY);
     kp=(int)floor((zP[np]+0.5*DZ)/DZ);
     dmin=99999;
     for (nsink=1;nsink<=NSINK;nsink++)
     {
      d=pow((ip-Sinks[nsink][1])*(ip-Sinks[nsink][1])+(jp-Sinks[nsink][2])*(jp-Sinks[nsink][2])
       +(kp-Sinks[nsink][2])*(kp-Sinks[nsink][3]),0.5);
      if (dmin>d)
      {
       dmin=d;
       ns=nsink;
      }
     }
     NPartP_sink[ns]++;
    }
   }
   for (nsink=1;nsink<=NSINK;nsink++)
   {
    FracW_sink[nsink]=double(NPartW_sink[nsink])/NPWleft;
    FracP_sink[nsink]=double(NPartP_sink[nsink])/NPPleft;
   }
  }
 }
 get_velocity=false;
 for (i=1; i<=NX; i++)
  for (j=1; j<=NY; j++)
   for (k=1; k<=NZ; k++)
   {
    VxT[i][j][k]=vx[(k-1)*NX*NY+(j-1)*NX+i]*rate_factor*Vwfac/Poro;  // conversion from vector to matrix
    VyT[i][j][k]=vy[(k-1)*NX*NY+(j-1)*NX+i]*rate_factor*Vwfac/Poro;  // conversion from vector to matrix
    VzT[i][j][k]=vz[(k-1)*NX*NY+(j-1)*NX+i]*rate_factor*Vwfac/Poro;  // conversion from vector to matrix
   }
 for (np=1; np<=NP; np++)
 {
  flag = false;
  if (arriveW[np]==0)
  {
   while (TimeW[np]<T[nt])
   {
    ip=(int)floor((xW[np]+0.5*DX)/DX);
    jp=(int)floor((yW[np]+0.5*DY)/DY);
    kp=(int)floor((zW[np]+0.5*DZ)/DZ);
    I[ip][jp][kp]=Swf;
    for (ns=1; ns<=NSINK; ns++)
     if ((ip==Sinks[ns][1]) && (jp==Sinks[ns][2]) && (kp==Sinks[ns][3]))
     {  nsink=ns;
      flag = true;}
    if (flag)
    {
     nsour=fmod(np,NSOUR)+1;
     arriveW[np]=nsour;
     arriveW[np]=1;
     part_countW[nsink][nsour]++;
     break;}
    for (p=1; p<=6; p++)
     Pr[p]=0;
    x=floor((xW[np]+0.5*DX)/DX);
    y=(xW[np]+0.5*DX)/DX;
    if (x!=y)
    {
     if (VxT[ip][jp][kp]>0)
     {
      dx1=(ip+0.5)*DX-xW[np];
      dt=T[nt]-TimeW[np];
      dx2=VxT[ip][jp][kp]*dt;
      xW[np]=xW[np]+min(dx1,dx2);
      if (dx1<dx2) {TimeW[np]=TimeW[np]+dx1/VxT[ip][jp][kp];}
      else {TimeW[np]=TimeW[np]+dt;}
     }
     else
     {
      dx1=xW[np]-(ip-0.5)*DX;
      dt=T[nt]-TimeW[np];
      dx2=-VxT[ip][jp][kp]*dt;
      xW[np]=xW[np]-min(dx1,dx2);
      if (dx1<dx2) {TimeW[np]=TimeW[np]+dx1/VxT[ip][jp][kp];}
      else {TimeW[np]=TimeW[np]+dt;}
     }
     break;
    }
    x=floor((yW[np]+0.5*DY)/DY);
    y=(yW[np]+0.5*DY)/DY;
    if (x!=y)
```

```
{
 if (VyT[ip][jp][kp]>0)
 {
  dx1=(jp+0.5)*DY-yW[np];
  dt=T[nt]-TimeW[np];
  dx2=VyT[ip][jp][kp]*dt;
  yW[np]=yW[np]+min(dx1,dx2);
  if (dx1<dx2) {TimeW[np]=TimeW[np]+dx1/VyT[ip][jp][kp];}
  else {TimeW[np]=TimeW[np]+dt;}
 }
 else
 {
  dx1=yW[np]-(jp-0.5)*DY;
  dt=T[nt]-TimeW[np];
  dx2=-VyT[ip][jp][kp]*dt;
  yW[np]=yW[np]-min(dx1,dx2);
  if (dx1<dx2) {TimeW[np]=TimeW[np]+dx1/VyT[ip][jp][kp];}
  else {TimeW[np]=TimeW[np]+dt;}
 }
 break;
}
x=floor((zW[np]+0.5*DZ)/DZ);
y=(zW[np]+0.5*DZ)/DZ;
if (x!=y)
{
 if (VzT[ip][jp][kp]>0)
 {
  dx1=(kp+0.5)*DZ-zW[np];
  dt=T[nt]-TimeW[np];
  dx2=VzT[ip][jp][kp]*dt;
  zW[np]=zW[np]+min(dx1,dx2);
  if (dx1<dx2) {TimeW[np]=TimeW[np]+dx1/VzT[ip][jp][kp];}
  else {TimeW[np]=TimeW[np]+dt;}
 }
 else
 {
  dx1=zW[np]-(kp-0.5)*DZ;
  dt=T[nt]-TimeW[np];
  dx2=-VyT[ip][jp][kp]*dt;
  zW[np]=zW[np]-min(dx1,dx2);
  if (dx1<dx2) {TimeW[np]=TimeW[np]+dx1/VzT[ip][jp][kp];}
  else {TimeW[np]=TimeW[np]+dt;}
 }
 break;
}
if (ip>1) {  if ((VxT[ip-1][jp][kp]<0)) Pr[1]=-VxT[ip-1][jp][kp];}
if (ip<NX) {  if ((VxT[ip][jp][kp]>0)) Pr[2]=VxT[ip][jp][kp];}
if (jp>1) {  if ((VyT[ip][jp-1][kp]<0)) Pr[3]=-VyT[ip][jp-1][kp];}
if (jp<NY) {  if ((VyT[ip][jp][kp]>0)) Pr[4]=VyT[ip][jp][kp];}
if (kp>1) {  if ((VzT[ip][jp][kp-1]<0)) Pr[5]=-VzT[ip][jp][kp-1];}
if (kp<NZ) {  if ((VzT[ip][jp][kp+1]>0)) Pr[6]=VzT[ip][jp][kp];}
sum=0;
for (p=1; p<=6; p++)
 {sum+=Pr[p];}
if (sum>0)
{
 r=rand()%100;
 cdf=0;
 for (p=1; p<=6; p++)
 {
  pdf=Pr[p]/sum;
  cdf=cdf+pdf;
  if (r<cdf*100) {break;}
 }
 switch (p)
```

```
        {
         case 1:
          dt=-DX/VxT[ip-1][jp][kp];
          if ((TimeW[np]+dt)<=T[nt])
          {
           xW[np]=xW[np]-DX;
           TimeW[np]=TimeW[np]+dt;
          }
          else
          {
           dt=T[nt]-TimeW[np];
           TimeW[np]=T[nt];
           dx=-dt*VxT[ip-1][jp][kp];
           xW[np]=xW[np]-dx;
          }
          break;
         case 2:
          dt=DX/VxT[ip][jp][kp];
          if ((TimeW[np]+dt)<=T[nt])
          {
           xW[np]=xW[np]+DX;
           TimeW[np]=TimeW[np]+dt;
          }
          else
          {
           dt=T[nt]-TimeW[np];
           TimeW[np]=T[nt];
           dx=dt*VxT[ip][jp][kp];
           xW[np]=xW[np]+dx;
          }
          break;
         case 3:
          dt=-DY/VyT[ip][jp-1][kp];
          if ((TimeW[np]+dt)<=T[nt])
          {
           yW[np]=yW[np]-DY;
           TimeW[np]=TimeW[np]+dt;
          }
          else
          {
           dt=T[nt]-TimeW[np];
           TimeW[np]=T[nt];
           dy=-dt*VyT[ip][jp-1][kp];
           yW[np]=yW[np]-dy;
          }
          break;
         case 4:
          dt=DY/VyT[ip][jp][kp];
          if ((TimeW[np]+dt)<=T[nt])
          {
           yW[np]=yW[np]+DY;
           TimeW[np]=TimeW[np]+dt;
          }
          else
          {
           dt=T[nt]-TimeW[np];
           TimeW[np]=T[nt];
           dy=dt*VyT[ip][jp][kp];
           yW[np]=yW[np]+dy;
          }
          break;
         case 5:
          dt=-DZ/VzT[ip][jp][kp-1];
          if ((TimeW[np]+dt)<=T[nt])
          {
```

```
       zW[np]=zW[np]-DZ;
       TimeW[np]=TimeW[np]+dt;
      }
      else
      {
       dt=T[nt]-TimeW[np];
       TimeW[np]=T[nt];
       dz=-dt*VzT[ip][jp][kp-1];
       zW[np]=zW[np]-dz;
      }
      break;
     case 6:
      dt=DZ/VzT[ip][jp][kp];
      if ((TimeW[np]+dt)<=T[nt])
      {
       zW[np]=zW[np]+DZ;
       TimeW[np]=TimeW[np]+dt;
      }
      else
      {
       dt=T[nt]-TimeW[np];
       TimeW[np]=T[nt];
       dz=-dt*VzT[ip][jp][kp];
       zW[np]=zW[np]+dz;
      }
      break;
    }
   }
  }
 }
}
for (i=1; i<=NX; i++)
 for (j=1; j<=NY; j++)
  for (k=1; k<=NZ; k++)
  {
   VxT[i][j][k]=vx[(k-1)*NX*NY+(j-1)*NX+i]*rate_factor*Vpfac/Poro;  // conversion from vector to matrix
   VyT[i][j][k]=vy[(k-1)*NX*NY+(j-1)*NX+i]*rate_factor*Vpfac/Poro;  // conversion from vector to matrix
   VzT[i][j][k]=vz[(k-1)*NX*NY+(j-1)*NX+i]*rate_factor*Vpfac/Poro;  // conversion from vector to matrix
  }
for (np=1; np<=NP; np++)
{
 flag = false;
 if (arriveP[np]==0)
 {
  while (TimeP[np]<T[nt])
  {
   ip=(int)floor((xP[np]+0.5*DX)/DX);
   jp=(int)floor((yP[np]+0.5*DY)/DY);
   kp=(int)floor((zP[np]+0.5*DZ)/DZ);
   I[ip][jp][kp]=Spf;
   for (ns=1; ns<=NSINK; ns++)
   if ((ip==Sinks[ns][1]) && (jp==Sinks[ns][2]) && (kp==Sinks[ns][3]))
   {flag = true;}
   if (flag)
   {
    nsour=fmod(np,NSOUR)+1;
    arriveP[np]=nsour;
    arriveP[np]=1;
    part_countP[nsink][nsour]++;
    break;}
   for (p=1; p<=6; p++)
    Pr[p]=0;
   x=floor((xP[np]+0.5*DX)/DX);
   y=(xP[np]+0.5*DX)/DX;
   if (x!=y)
```

```
{
 if (VxT[ip][jp][kp]>0)
 {
  dx1=(ip+0.5)*DX-xP[np];
  dt=T[nt]-TimeP[np];
  dx2=VxT[ip][jp][kp]*dt;
  xP[np]=xP[np]+min(dx1,dx2);
  if (dx1<dx2) {TimeP[np]=TimeP[np]+dx1/VxT[ip][jp][kp];}
  else {TimeP[np]=TimeP[np]+dt;}
 }
 else
 {
  dx1=xP[np]-(ip-0.5)*DX;
  dt=T[nt]-TimeP[np];
  dx2=-VxT[ip][jp][kp]*dt;
  xP[np]=xP[np]-min(dx1,dx2);
  if (dx1<dx2) {TimeP[np]=TimeP[np]+dx1/VxT[ip][jp][kp];}
  else {TimeP[np]=TimeP[np]+dt;}
 }
 break;
}
x=floor((yP[np]+0.5*DY)/DY);
y=(yP[np]+0.5*DY)/DY;
if (x!=y)
{
 if (VyT[ip][jp][kp]>0)
 {
  dx1=(jp+0.5)*DY-yP[np];
  dt=T[nt]-TimeP[np];
  dx2=VyT[ip][jp][kp]*dt;
  yP[np]=yP[np]+min(dx1,dx2);
  if (dx1<dx2) {TimeP[np]=TimeP[np]+dx1/VyT[ip][jp][kp];}
  else {TimeP[np]=TimeP[np]+dt;}
 }
 else
 {
  dx1=yP[np]-(jp-0.5)*DY;
  dt=T[nt]-TimeP[np];
  dx2=-VyT[ip][jp][kp]*dt;
  yP[np]=yP[np]-min(dx1,dx2);
  if (dx1<dx2) {TimeP[np]=TimeP[np]+dx1/VyT[ip][jp][kp];}
  else {TimeP[np]=TimeP[np]+dt;}
 }
 break;
}
x=floor((zP[np]+0.5*DZ)/DZ);
y=(zP[np]+0.5*DZ)/DZ;
if (x!=y)
{
 if (VzT[ip][jp][kp]>0)
 {
  dx1=(kp+0.5)*DZ-zP[np];
  dt=T[nt]-TimeP[np];
  dx2=VzT[ip][jp][kp]*dt;
  zP[np]=zP[np]+min(dx1,dx2);
  if (dx1<dx2) {TimeP[np]=TimeP[np]+dx1/VzT[ip][jp][kp];}
  else {TimeP[np]=TimeP[np]+dt;}
 }
 else
 {
  dx1=zP[np]-(kp-0.5)*DZ;
  dt=T[nt]-TimeP[np];
  dx2=-VyT[ip][jp][kp]*dt;
  zP[np]=zP[np]-min(dx1,dx2);
  if (dx1<dx2) {TimeP[np]=TimeP[np]+dx1/VzT[ip][jp][kp];}
  else {TimeP[np]=TimeP[np]+dt;}
 }
 break;
}
if (ip>1) {  if ((VxT[ip-1][jp][kp]<0)) Pr[1]=-VxT[ip-1][jp][kp];}
if (ip<NX) {  if ((VxT[ip][jp][kp]>0)) Pr[2]=VxT[ip][jp][kp];}
if (jp>1) {  if ((VyT[ip][jp-1][kp]<0)) Pr[3]=-VyT[ip][jp-1][kp];}
if (jp<NY) {  if ((VyT[ip][jp][kp]>0)) Pr[4]=VyT[ip][jp][kp];}
if (kp>1) {  if ((VzT[ip][jp][kp-1]<0)) Pr[5]=-VzT[ip][jp][kp-1];}
if (kp<NZ) {  if ((VzT[ip][jp][kp+1]>0)) Pr[6]=VzT[ip][jp][kp];}
sum=0;
for (p=1; p<=6; p++)
 {sum+=Pr[p];}
if (sum>0)
{
 r=rand()%100;
 cdf=0;
 for (p=1; p<=6; p++)
 {
  pdf=Pr[p]/sum;
  cdf=cdf+pdf;
  if (r<cdf*100) {break;}
 }
 switch (p)
 {
  case 1:
   dt=-DX/VxT[ip-1][jp][kp];
   if ((TimeP[np]+dt)<=T[nt])
   {
    xP[np]=xP[np]-DX;
    TimeP[np]=TimeP[np]+dt;
   }
   else
   {
    dt=T[nt]-TimeP[np];
    TimeP[np]=T[nt];
    dx=-dt*VxT[ip-1][jp][kp];
    xP[np]=xP[np]-dx;
   }
   break;
  case 2:
   dt=DX/VxT[ip][jp][kp];
   if ((TimeP[np]+dt)<=T[nt])
   {
    xP[np]=xP[np]+DX;
    TimeP[np]=TimeP[np]+dt;
   }
   else
   {
    dt=T[nt]-TimeP[np];
    TimeP[np]=T[nt];
    dx=dt*VxT[ip][jp][kp];
    xP[np]=xP[np]+dx;
   }
   break;
  case 3:
   dt=-DY/VyT[ip][jp-1][kp];
   if ((TimeP[np]+dt)<=T[nt])
   {
    yP[np]=yP[np]-DY;
    TimeP[np]=TimeP[np]+dt;
   }
   else
   {
    dt=T[nt]-TimeP[np];
```

```
         TimeP[np]=T[nt];
         dy=-dt*VyT[ip][jp-1][kp];
         yP[np]=yP[np]-dy;
        }
        break;
       case 4:
        dt=DY/VyT[ip][jp][kp];
        if ((TimeP[np]+dt)<=T[nt])
        {
         yP[np]=yP[np]+DY;
         TimeP[np]=TimeP[np]+dt;
        }
        else
        {
         dt=T[nt]-TimeP[np];
         TimeP[np]=T[nt];
         dy=dt*VyT[ip][jp][kp];
         yP[np]=yP[np]+dy;
        }
        break;
       case 5:
        dt=-DZ/VzT[ip][jp][kp-1];
        if ((TimeP[np]+dt)<=T[nt])
        {
         zP[np]=zP[np]-DZ;
         TimeP[np]=TimeP[np]+dt;
        }
        else
        {
         dt=T[nt]-TimeP[np];
         TimeP[np]=T[nt];
         dz=-dt*VzT[ip][jp][kp-1];
         zP[np]=zP[np]-dz;
        }
        break;
       case 6:
        dt=DZ/VzT[ip][jp][kp];
        if ((TimeP[np]+dt)<=T[nt])
        {
         zP[np]=zP[np]+DZ;
         TimeP[np]=TimeP[np]+dt;
        }
        else
        {
         dt=T[nt]-TimeP[np];
         TimeP[np]=T[nt];
         dz=-dt*VzT[ip][jp][kp];
         zP[np]=zP[np]+dz;
        }
        break;
      }
     }
    }
   }
  }
  Qw=0;
  for (nsink=1;nsink<=NSINK;nsink++)
  {
   Npw=0;
   Npp=0;
   for (nsour=1;nsour<=NSOUR;nsour++)
   {
    wc+=(part_countW[nsink][nsour]-
part_countP[nsink][nsour])*Fw*Qi[nsour][nt]/Qp[nsink][nt]/(double)(NP/NSOUR)*Qinj[nsour]/Qinj_all
     +part_countP[nsink][nsour]/(double)(NP/NSOUR)*Qinj[nsour]/Qinj_all*Fp*Qi[nsour][nt]/Qp[nsink][nt];
```

```
        Npw+=part_countW[nsink][nsour];
        Npp+=part_countP[nsink][nsour];

      }
      Sw[nsink]=Sw_sink[nsink]+(Swf-Sw_sink[nsink])*(Npw-Npp)/(FracW_sink[nsink]*NPWleft);
      Sw[nsink]+=(Spf-Swf)*Npp/(FracP_sink[nsink]*NPPleft);
      So=1-Sw[nsink];
      Krw=Krw_max*pow((Sw[nsink]-Sw_ini)/(1-Sw_ini-Sor),exp_w);
      Kro=Kro_max*pow((So-Sor)/(1-Sw_ini-Sor),exp_o);
      if (Sw[nsink]>Swf)
        wc=(Fp-Fw)*(Sw[nsink]-Swf)/(Spf-Swf)+Fw;
      else
      {  if (Krw>0)
          wc=1/(1+Kro*Vw/(Krw*Vo));
        else
          wc=0;}
      wc_sink[nsink]=wc;
      Qw+=Qp[nsink][nt]*wc;
    }
    wc=Qw/Qi_tot;
    Qo=(1-wc)*Qi_tot;
    NetPresent(NPV,wc);
    prodfl<<T[nt]<<"  "<<Qw*35.314666<<"  "<<Qo*35.314666<<"  "<<wc<<"  "<<NPV<<"  ";
    for (nsink=1;nsink<=NSINK;nsink++)
      prodfl<<wc_sink[nsink]<<"  ";
    prodfl<<endl;
  }
  for (np=1; np<=NP; np++)
  {
    nsour=fmod(np,NSOUR)+1;
    if (nsour==1)
    {  positionfl<<xP[np]<<"  "<<yP[np]<<"  "<<zP[np]<<"  "<<"2"<<endl;}
    else
    {  positionfl<<xP[np]<<"  "<<yP[np]<<"  "<<zP[np]<<"  "<<"3"<<endl;}
  }
  for (np=1; np<=NP; np++)
  {
    nsour=fmod(np,NSOUR)+1;
    if (nsour==1)
    {  positionfl<<xW[np]<<"  "<<yW[np]<<"  "<<zW[np]<<"  "<<"0"<<endl;}
    else
    {  positionfl<<xW[np]<<"  "<<yW[np]<<"  "<<zW[np]<<"  "<<"1"<<endl;}
  }
  positionfl.close();
  for (k=1; k<=NZ; k++)
    for (j=1; j<=NY; j++)
      for (i=1; i<=NX; i++)
        trajfile<<I[i][j][k]<<endl;
  trajfile.close();
  prodfl.close();
  if (argc<3)
  {
    cout<<"Read production data in 'Production.dat'"<<endl;
    cout<<"Computer Time "<<time (NULL)-t1<<" seconds"<<endl;
  }

  return 0;
}


void read_parameters(char Pfile[20])
{
  char chartemp[120];
  char Kfile[80];
  int i, j, k;
```

```
read_flag=1;
ifstream Param(Pfile);
//ifstream Param("proxy-40.par");
if (!Param)
{
  cout<<"Parameter file does not exist"<<endl;
  exit(1);
}
Param>>NX;
if (NX>nx_max) { cout<<"Number of blocks in X-direction exceeded, max"<<nx_max<<endl; exit(1);}
Param>>NY;
if (NY>ny_max) { cout<<"Number of blocks in Y-direction exceeded, max"<<ny_max<<endl; exit(1);}
Param>>NZ;
if (NZ>nz_max) { cout<<"Number of blocks in Z-direction exceeded, max"<<nz_max<<endl; exit(1);}
Param.getline(chartemp,120,'\n');
Param>>DX;
Param>>DY;
Param>>DZ;
Param.getline(chartemp,120,'\n');
Param>>NSOUR;
Param.getline(chartemp,120,'\n');
if (NSOUR>nsour_max) { cout<<"Number of sources exceeded, max "<<nsour_max<<endl; exit(1);}
for (ns=1; ns<=NSOUR; ns++)
{
  Param>>Sources[ns][1];
  Param>>Sources[ns][2];
  Param>>Sources[ns][3];
  Param.getline(chartemp,120,'\n');
}
Param>>NSINK;
Param.getline(chartemp,120,'\n');
if (NSINK>nsink_max) { cout<<"Number of sinks exceeded, max "<<nsink_max<<endl; exit(1);}
for (ns=1; ns<=NSINK; ns++)
{
  Param>>Sinks[ns][1];
  Param>>Sinks[ns][2];
  Param>>Sinks[ns][3];
  Param.getline(chartemp,120,'\n');
}
Param>>Poro;
Param.getline(chartemp,120,'\n');
Param>>NT;
Param.getline(chartemp,120,'\n');
if (NT>100) { cout<<"Number of time steps exceeded, max 100"<<endl; exit(1);}
for (i=1; i<=NT; i++)
{
  Param>>T[i];
  for (ns=1;ns<=NSOUR;ns++)
  {
    Param>>Qi[ns][i];
    Qi[ns][i]=Qi[ns][i]/35.314666; // Conversion to SI units
  }
  for (ns=1;ns<=NSINK;ns++)
  {
    Param>>Qp[ns][i];
    Qp[ns][i]=Qp[ns][i]/35.314666; // Conversion to SI units
  }
}
Param>>Vwfac;
Param.getline(chartemp,120,'\n');
Param>>Vpfac;
Param.getline(chartemp,120,'\n');
Param>>Swf>>Spf;
Param.getline(chartemp,120,'\n');
Param>>Sw_ini>>Sor;
```

```
  Param.getline(chartemp,120,'\n');
  Param>>Krw_max>>exp_w>>Vw;
  Param.getline(chartemp,120,'\n');
  Param>>Kro_max>>exp_o>>Vo;
  Param.getline(chartemp,120,'\n');
  Param>>Krp_max>>exp_p>>Vp;
  Param.getline(chartemp,120,'\n');
  Param>>Kfile;
  Param.getline(chartemp,120,'\n');
  Param.close();
  ifstream Kfl(Kfile);
  if (!Kfl)
  {
    cout<<"Permeability file does not exist"<<endl;
    exit(1);
  }
  for (i=1; i<=3; i++)
    Kfl.getline(chartemp,120);
  for (k=1; k<=NZ; k++)
    for (j=1; j<=NY; j++)
      for (i=1; i<=NX; i++)
      {
        Kfl>>kx[j*(NX+2)+i];  //permeability in y-direction
        kx[j*(NX+2)+i]=kx[j*(NX+2)+i]*0.9869E-15;  // Conversion to SI units
        ky[j*(NX+2)+i]=kx[j*(NX+2)+i];
        vx[(k-1)*NX*NY+(j-1)*NX+i] = 0;
        vy[(k-1)*NX*NY+(j-1)*NX+i] = 0;
        vz[(k-1)*NX*NY+(j-1)*NX+i] = -1;
      }
  Kfl.close();
  ifstream Econ("Econ.par");
  if (!Econ)
  {
      if (narg<3)  {cout<<"No economic parameter file found ... using default parameters"<<endl;}
  }
  else
  {
    if (narg<3) {cout<<"Reading economic parameter file"<<endl;}
    Econ.getline(chartemp,120);
    Econ>>conc;
    conc=conc*1E-6;
    Econ>>tslug[1]>>tslug[2];
    Econ>>tax;
    Econ>>royal;
    Econ>>apr;
    Econ>>wfcost;
    Econ>>watinjcost;
    Econ>>watprodcost;
    Econ>>oilprodcost;
    Econ>>polymasscost;
    Econ>>oilprice;
    Econ.close();
  }
}

double NetPresent(double& NPV, double wc)
{
  double watden;
  double oilprod, watprod, injcost;
  double dt, winjec, oilcost, watcost, oilrevenue, royalty, netrevenue;
  double wf, cost, advalorem, tax_income, tax_paid, netincome;
  double polycost;
  watden=350;
  dt=T[nt]-T[nt-1];
  winjec=Qi_tot*dt*6.28981077;
  oilprod=Qi_tot*(1-wc)*dt*6.28981077;
  watprod=Qi_tot*wc*dt*6.28981077;
  injcost=watinjcost*winjec;
  oilcost=oilprod*oilprodcost;
  watcost=watprod*watprodcost;
  oilrevenue=oilprod*oilprice;
  royalty=oilrevenue*royal;
  netrevenue=oilrevenue-royalty;
  if ((T[nt]>=tslug[1]) && (T[nt]<=tslug[2]))
  {  polycost=polymasscost*winjec*watden*conc;}
  else
  {  polycost=0;}
  wf=wfcost*dt/30;
  cost=injcost+oilcost+watcost+polycost+wf;
  advalorem=oilprod*0.0046;
  tax_income=netrevenue-cost-advalorem;
  tax_paid=tax_income*tax;
  netincome=tax_income-tax_paid;
  NPV=NPV+netincome/pow((1+apr),T[nt]/365);
  return 0;
}
void write_wellfile()   // write well data file for simulator
{
  ofstream wellfl("well.dat", ios::trunc);
  wellfl<<"0"<<endl;   // start time
  wellfl<<"1"<<endl;  // number of time step entries
  wellfl<<"1  5"<<endl;  // entry # 1 - 2 steps of 1 day
  wellfl<<NSOUR+NSINK<<endl;    // total number of wells
  for (ns=1;ns<=NSOUR;ns++)
    wellfl<<Sources[ns][1]<<"  "<<Sources[ns][2]<<endl;      // Location of sources (injetors)
  for (ns=1;ns<=NSINK;ns++)
    wellfl<<Sinks[ns][1]<<"  "<<Sinks[ns][2]<<endl;      // Location of sinks (producers)
  for (ns=1;ns<=NSOUR;ns++)
    wellfl<<"0  "<<ns<<"  6.048e-5  Q  "<<Qi[ns][nt]<<endl;
  for (ns=1;ns<=NSINK;ns++)
    wellfl<<"0  "<<ns+NSOUR<<"  6.048e-5  Q  "<<-Qp[ns][nt]<<endl;
  wellfl.close();
}

void frac_flow()
{
  Sowf=1-Swf;
  Sopf=1-Spf;
  Krofw=Kro_max*pow((Sowf-Sor)/(1-Sor-Sw_ini),exp_o);
  Krofp=Kro_max*pow((Sopf-Sor)/(1-Sor-Sw_ini),exp_o);
  Krwf=Krw_max*pow((Swf-Sw_ini)/(1-Sor-Sw_ini),exp_w);
  Krpf=Krp_max*pow((Spf-Sw_ini)/(1-Sor-Sw_ini),exp_w);
  Fw=1/(1+Krofw*Vw/(Krwf*Vo));
  Fp=1/(1+Krofp*Vp/(Krpf*Vo));
}
```

References

Anderson, G. A., Delshad, M., King, C. B., Mohammadi, H., and Pope, G. A. 2006. Optimization of Chemical Flooding in a Mixed-Wet Dolomite Reservoir, SPE 100082, presented at the SPE/DOE Symposium on Improved Oil Recovery, Tulsa, Apr 22-26.

Arpat, B.G. and Caers, J. 2005. A Multiple-scale, Pattern-based Approach to Sequential Simulation. Quantitative Geology and Geostatistics Springer, Dordrecht, v1 p. 255-264

Bishop, C. 2006. *Pattern Recognition and Machine Learning*, Springer, New York

Brouwer, D. R., Nævdal, R., Jansen, J. D., Vefring E. H., and van Kruijsdijk, J. 2004. Improved Reservoir Management through Optimal Control and Continuous Model Updating, SPE 90149, SPE Annual Technical Conference and Exhibition, Houston, Oct 2-5.

Burns, R. 2001. *Advanced Control Engineering*, Butterworth and Heinemann, Oxford.

Caers, J. Park, K. and Scheidt, C. 2009. Modeling Uncertainty in Metric Space, International Association of Mathematical Geology Meeting, Stanford University.

Canella, W.J., Huh, C. And Seright, R. S. 1988. Prediction of Xanthan Rheology in Porous Media, SEP 18089, presented at the SPE Annual Technical Conference and Exhibition, Houston, Oct 2-5.

Chaudhri, M., Phale, H. A., Liu, N., Oliver, D. S. 2009. An Improved Approach for Ensemble-Based Production Optimization, SPE 121305, presented at the SPE Western Regional Meeting, San Jose, March 23-26.

Chen, Y. and Oliver D. S. 2009. Ensemble-Based Closed-Loop Optimization Applied to Brugge Field, SPE 118926, presented at the SPE Reservoir Simulation Symposium, Feb 2-4.

Chen, Y., Oliver, D. S. and Zhang, D. 2008. Efficient Ensemble-Based Closed-Loop Production Optimization, SPE 112873, presented at SPE/DOE Improved Oil Recovery Symposium, Tulsa.

Dakhlia, H., 1995. A Simulation Study of Polymer Flooding and Surfactant Flooding using Horizontal Wells, Ph. D. Dissertation, The University of Texas at Austin.

Deustch, C. V. 1997. *Geostatisctical Software Library and User's Guide*, Oxford University Press, New York

Evensen, G. 2007. Data assimilation: the Ensemble Kalman Filter, Springer, Berlin, NY.

Evensen, G., 1994. Sequential Data Assimilation with a Non-linear Quasi Geostrophic Model Using Monte Carlo Methods to Forecast Error Statistics, Journal of Geophysics Res. 99, 143.

Fathi, Z. and Ramirez, W. 1984 Optimal Injection Policies for Enhanced Oil Recovery: Part 1 Theory and Computational Strategies, SPE Journal, Volume 24, Number 3, pp 333-341.

Franklin, G. F., Powell, J. D. and Emani-Naeini, A., 2010. Feedback Control of Dynamic Systems, Upper Saddle River, NY.

Friedmann, F., Chawathè, A., and Larud, D. K., 2001. Assessing Uncertainty in Channelized Reservoirs Using Experimental Designs, SPE 71622, presented at the SPE Annual Technical Conference and Exhibition, New Orleans, Louisiana, Sept. 30-Oct. 3.

Green, D. W., and Willhite, G. P. 1998. Enhanced Oil Recovery, Society of Petroleum Engineers, Richardson.

Hofman, B. T. and Caers, J. History Matching under Geological Control: Application to a North Sea Reservoir, Geostatistics Banff 2004, O. Leuangthong and C.V. Deutsch (eds), 1067-1076. Kluwer Academic Publishers, Dordrecht.

Huh, C., Landis, L. H., Maer, N. K., McKinney, P. H. and Dougherty, N. A. 1990b. Simulation to Support Interpretation of the Loudon Surfactant Pilot Tests, SPE 20245, presented at the SPE Annual Technical Conference and Exhibition, New Orleans, Sept 23-26.

Huh, C., Lange, E. A., and Canella, W. J. 1990a. Polymer Retention in Porous Media, SPE 20235, presented at the 7th Symposium on Enhanced Oil Recovery, Tulsa, Apr 22-25.

Hull, D., 2003. Optimal Control Theory for Applications, Springer, New York.

Jablonowski, C. and MacEarchen D. P. 2009. Developing probabilistic well construction estimates using Regression Analysis, Energy Exploration and Exploitation, Vol. 27 No. 6, pp 439-452

Journel, A. G. 2002. Combining Knowledge from Diverse Sources: An alternative to Traditional Data Independence Hypothesis, Mathematical Geology, V. 34, No. 5, 573-596.

Kalman, R. E., 1961, On the General Theory of Control Systems. Automatic and Remote Control, Butterworth and Co., London, pp 481-492.

Kalman, R.E. 1960. A New Approach to Linear Filtering and Prediction Problems. Journal of Basic Engineering 82 (1): 35–45.

Kim, D. H., Lee, S., Ahn, C.H., Huh, C. and Pope, G. A. 2010. Development of a Viscoelastic Property Database for EOR Polymers, SPE 129971 presented at the SPE Improved Oil Recovery Symposium, Tulsa, April 24-28

Koval, E. J. 1963. A Method for Predicting the Performance of Unstable Miscible Displacements in Heterogeneous Media, SPE J, June, 145-154

Kraaijevanger, J.F.B.M., Egberts, P.J.P., Valstar, J. and Buurman H. W. 2007. Optimal Waterflood Design Using the Adjoint Method, SPE 105764, presented at the 2007 Reservoir Simulation Symposium, Houston, Feb 26-28.

Lake, L. W., 1989. *Enhanced Oil Recovery,* Englewood Cliffs: Prentice Hall.

Lawler, G. F. and Limic, V. 2010. Random Walk: A Modern Introduction. Cambridge University Press, New York

Lee, S. 2009. Development of a Comprehensive Rheological Property Database for EOR Polymers, M. Sc. thesis, The University of Texas at Austin.

Levitt, D. B. and Pope, G. A. 2008. Selection and Screening of Polymers for Enhanced Oil Recovery, SPE 113845 presented at the SPE Improved Oil Recovery Symposium, Tulsa, April 24-28

Lorentzen, R. J., Fjelde, K. K., Froyen, J., Lage, A. C., Naevdal, G. and Vefring, E. H. 2001. Underbalanced and Low-Head Drilling Operations: Real Time Interpretation of Measured Data and Operational Support, SPE 71384 presented at the SPE Annual Technical Conference and Exhibition, New Orleans.

Mantilla, C. A. 2010, Proposal of a Rapid Model Updating and Feedback Control Scheme for Polymer Flooding Processes, M.S.E. Report, The University of Texas at Austin.

Mantilla, C., Bhowmick, S., and Srinivasan, S. Tracking CO_2 Plume Migration during geologic Sequestration using a Probabilistic History Matching Approach, presented at the 8th International Conference on Geostatistics for Environmental Applications, Ghent.

Mantilla, C., Nguyen, Q. P. and Srinivasan, S. 2007. Model Updating and Optimum Control for Water Coning: an Application of Ensemble Kalman Filter, presented at the International Petroleum Technology Conference, Kuala Lumpur.

Mehos, G. J. and Ramirez W. F. 1989. Use of Optimal Control Theory to Optimize Carbon Dioxide Miscible Flooding enhanced Oil Recovery, Journal of Petroleum Science and Engineering, 2, pp 247-60.

Myers, R. H. and Montgomery D. C. 2002. *Response Surface Methodology: Process and Product in Optimization using Designed Experiments*, Wiley, New York.

Naiki, M. 1979. Numerical Simulation of Polymer Flooding Including the Effects of Salinity, Ph. D. Dissertation, The University of Texas at Austin.

Patton, J. T., Coats, K. H. and Colegrove, G. T. 1971. Prediction of Polymer Flood Performance, SPEJ, March, pp 72-84.

Pearson, K. 1905. The problem of the Random Walk. Nature. 72, 294

Peters, E., Arts, R.J., Brouwer, G.K. and Geel, C.R. 2009. Results of the Brugge Benchmark Study for Flooding Optimisation and History Matching, presented at the SPE Reservoir Simulation Symposium, The Woodlands

Pope, G. A. 1980. Application of Fractional Flow Theory to Enhanced Oil Recovery, SPEJ June 1980, pp 191-205.

Pope, G.A., and Nelson, R.C. 1978. A Chemical Flooding Compositional Simulator. SPE Journal Volume 18 Number 5, pp 339-354.

Porzucek, C. and Ramirez, W. F. 1990. Optimal Injection Strategies for Surfactant Flooding Enhanced Oil Recovery, Journal of Petroleum Science and Engineering, 4, pp 323-334.

Prasanphanich, J. 2009. Gas Reserves Estimation by Monte Carlo Simulation and Chemical Flooding Optimization using Experimental Design and Response Surface Methodology, Master Report, The University of Texas at Austin.

Putz, A. G., Lecourtier, J. M. and Bruckert, L. 1988. Interpretation of High Recovery Obtained in a New Polymer Flood in the Chateaurenard Field, SPE paper 18093 presented at the Annual Technical Conference and Exhibition, Houston.

Ramirez, J. 2009. Coupled CO2 Sequestration and Enhanced Oil Recovery Optimization using Experimental Design and Response Surface Methodology, Master Report, The University of Texas at Austin.

Ramirez, W. F. 1986 *Application of Optimal Control Theory to Enhanced Oil Recovery*, Elsevier Science Publishers B.V., Amsterdam.

Ray, S. and Turi, R. H. 1999. Determination of Number of Clusters in K-Means Clustering and Application in Colour Image Segmentation, Proceedings of Proceedings of the 4th International Conference on Advances in Pattern Recognition and Digital Techniques, Calcutta.

Sangvaree, T. 2008. Chemical Flooding Optimization using the Experimental Design and Response Surface Method, M. Sc. thesis, The University of Texas at Austin.

Sarma, P, Aziz, K. and Durlofsky, L. J. 2005. Implementation of Adjoint Solution for Optimal Control of Smart Wells, presented at the SPE Reservoir Simulation Symposium, Houston.

Sarma, P, Aziz, K. and Durlofsky, L. J. 2005. Implementation of Adjoint Solution for Optimal Control of Smart Wells, presented at the SPE Reservoir Simulation Symposium, Houston.

Sarma, P., Durlofsky, L. J., Aziz, K. and Chen, W. H. 2006. Efficient Real-Time Reservoir Management Using Adjoint-Based Optimal Control and Model Updating, Computational Geosciences, 10:3-36.

Scheidt, C. and Caers, J. 2009. Representing Spatial Uncertainty Using Distances and Kernels. Math Geoscience: 41, pp 397-417

Shook, G. M. and Mitchell, K. M. 2009. A Robust Measure of Heterogeneity for Ranking Earth Models: the F-PHI Curve and Dynamic Lorentz Coefficient, SPE paper 124625 presented at the Annual Technical Conference and Exhibition, New Orleans.

Smith, F. W. 1970. The Behaviour of Partially Hydrolized Polyacrylamide Solutions in Porous Media, Journal of Petroleum Technology, 22, pp 146-156

Srinivasan, S. and Bryant, S. Integrating Dynamic Data in Reservoir Models using Parallel Computing Approach. SPE paper 89444 presented at the SPE/DOE 13th Symposium on Improved Oil Recovery, Tulsa, Oklahoma, 17-21 April 2004

Strebelle, S. and Journel, A. 2001. Reservoir Modeling Using Multiple Point Statistics, SPE paper 71324 presented at the Annual Technical Conference and Exhibition, New Orleans.

Sudaryanto, B. and Yortsos, Y. C. 2001. Optimization of Displacement in Porous Media using Rate Control, presented at the SPE Annual Technical Conference, New Orleans, Sep-30, Oct-3.

UTCHEM-9.0, 2000 Volume II: Technical Documentation, Center for Petroleum and Geosystems Engineering, The University of Texas at Austin

van Essen, G.M., Zandvliet, M.J., Van den Hof, P.M.J., and Bosgra O.H., and Jansen, J.D. 2009. Robust Waterflooding Optimization of Multiple Geological Scenarios, SPE Journal Volume 14, Number 1

Vaskas, A. J.: 1996. Optimization of Surfactant Flooding: An Economic Approach, M.A. Thesis, The University of Texas at Austin.

Yadav, S., Srinivasan, S, Bryant, S.L. and Barrera, A. 2005. History Matching using Probabilistic Approach in a Distributed Parallel Computing Environment, SPE paper 93399 presented at the SPE Reservoir Simulation Symposium, Houston.

Yamali, N., Nguyen, Q. P. and Srinivasan, S. 2007. Optimum Control of Unwanted Water Production in Stratified Gas Reservoirs, SPE 106640, presented at SPE Production and Operations Symposium, Oklahoma City, Mar 31-Apr 3.

Zhou, H., Gomez-Hernandez, J.J. and Li, L. 2010. Performance of Modified Ensemble Kalman Filter in non-Gaussian Heterogeneous Media, presented at the 8th International Conference on Geostatistics for Environmental Applications, Ghent.

Vita

Cesar A. Mantilla, the son of Martha Lucia Uribe, was born in Bucaramanga, Colombia on September 18, 1976. In 1991, Cesar's family moved to Pasto, Colombia where he obtained his High School Diploma from Colegio San Francisco Javier in 1993. In the same year, he started his studies at Universidad Industrial de Santander, School of Petroleum Engineering, where he received his degree of Bachelor of Science in Petroleum Engineering in 2000. He worked for Halliburton Energy Services as field engineer from 2001 to 2006. In 2004, he married Adriana Arevalo. In 2006, he started his graduate studies at The University of Texas at Austin. In 2010, he received the degree of Master in Engineering at this university.

Permanent address: Calle 38 # 35-15, Apt. 101
Bucaramanga, Colombia

This report was typed by the author.