THE UNIVERSITY OF TULSA

THE GRADUATE SCHOOL

ADJOINT-GRADIENT-BASED PRODUCTION OPTIMIZATION WITH THE AUGMENTED LAGRANGIAN METHOD

by
Chaohui Chen

A dissertation submitted in partial fulfillment of
the requirements for the degree of Doctor of Philosophy
in the Discipline of Petroleum Engineering

The Graduate School
The University of Tulsa

2011

THE UNIVERSITY OF TULSA

THE GRADUATE SCHOOL

ADJOINT-GRADIENT-BASED PRODUCTION OPTIMIZATION WITH THE AUGMENTED LAGRANGIAN METHOD

by
Chaohui Chen

A DISSERTATION

APPROVED FOR THE DISCIPLINE OF

PETROLEUM ENGINEERING

By Dissertation Committee

______________________________, Co-Chair
Gaoming Li

______________________________, Co-Chair
Albert C. Reynolds

____
Richard A. Redner

____
Ram S. Mohan

ABSTRACT

Chaohui Chen (Doctor of Philosophy in Petroleum Engineering)

Adjoint-Gradient-Based Production Optimization With The Augmented Lagrangian Method

Directed by Gaoming Li and Albert C. Reynolds

163 pp., Chapter 6: Conclusions

(452 words)

The production optimization step of the "closed-loop" reservoir management is an optimal well control problem determining optimal operating conditions to maximize hydrocarbon extraction or net present value (NPV) for the remaining expected life of a reservoir. The most challenging part of production optimization is to honor the nonlinear constraints or state-control constraints, such as WOR, GOR and production rates. In this research, we implemented an augmented Lagrangian method for solving the production optimization problem under linear and nonlinear constraints. In our implementation, the objective function to be maximized is defined as the augmented Lagrangian function consisting of the NPV and all constraints except the bound constraints. At each iteration of the optimization procedure, the objective function is approximated by a quadratic model based on the adjoint gradient and the approximate Hessian matrix obtained using a quasi-Newton method. The quadratic model is then maximized subject to the bound constraints using a gradient-projection trust-region method. This step ensures all the bound constraints are satisfied. Once the controls that maximize the quadratic function are obtained at this iteration, we update the Lagrange multipliers or penalty parameter depending on how well the constraints are satisfied, and move to the next iteration. The above process is repeated until convergence. The advantage of the above procedure is that the bound constraints are easily handled using the gradient-projection method for a quadratic approximation of the objective function. Compared to the generalized reduced gradient (GRG) method which is implemented in Eclipse 300, our method does not require the controls to be feasible at every iteration, but the constraints are satisfied within a reasonable tolerance at convergence.

We extend the augmented Lagrangian method to solve the robust production optimization problem. The technique is applied to synthetic reservoir problems to demonstrate its efficiency and robustness. When reservoir description is uncertain, experiments show that the optimal NPV obtained based on a single reservoir model may not be the optimal NPV for the true geology, whereas the application of robust optimization significantly reduces this risk. Another challenging problem for production optimization is to solve multi-objective optimization problems, such as long-term and short-term optimization. Robust long-term optimization maximizes the expected life-cycle net-present value (NPV) over a set of geological models, which represent the uncertainty of reservoir description. As the life-cycle optimal controls may be in conflict with the operator's objective of maximizing short-time production, the method is adapted to maximize the expectation of short-term NPV over the next one or two years subject to the constraint that the life-cycle NPV will not be substantially decreased. Experimental results also show robust sequential optimization on each short-term period is not able to achieve an expected life-cycle NPV as high as the one obtained with robust long-term optimization.

ACKNOWLEDGEMENTS

I would like to express my sincere appreciation to Dr. Gaoming Li, Associate Professor of Petroleum Engineering of The University of Tulsa, and Dr. Albert C. Reynolds, Jr., Professor of Petroleum Engineering and Mathematical Science of The University of Tulsa, for their support, guidance, insights and encouragement during my five years' Ph.D. study at The University of Tulsa. I specially thank Dr. Richard A. Redner, Professor of Mathematical Science of The University of Tulsa and Dr. Ram S. Mohan, Professor of Mechanical Engineering of The University of Tulsa, for serving on my committee and for their helpful suggestions and comments. I greatly appreciate the help of Mrs. Judy Teal and Mrs. Loreta M. Watkins during my study in The University of Tulsa. I would like to extend my thanks and appreciations to all the other faculty members of the Petroleum Engineering department of The University of Tulsa for their guidance through my courses of study as a graduate student.

I gratefully acknowledge financial support from the member companies of the University of Tulsa Petroleum Reservoir Exploitation Projects (TUPREP) and the Bellwether Fellowship offered by the Graduate School of The University of Tulsa. The reservoir simulator used in this study was provided by Chevron Petroleum Technology Company.

This work is dedicated to my parents, Mingyi Chen and Meihua Lin, and all the friends in Tulsa and Houston for their unconditional love and support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Production optimization is an optimal well control problem determining optimal operating conditions to maximize hydrocarbon extraction or net present value (NPV) for the remaining expected life of a reservoir. It is an important step in the context of closed-loop reservoir management [6, 23, 39, 40], which is defined as a combination of production optimization and data assimilation (history matching or model calibration). Fig. 1.1 demonstrates the work flow of the closed-loop reservoir management. The "reservoir system" represents the actual reservoir oil field from which we collect production data (e.g., well production rates, bottom hole pressure, etc.; The "data assimilation" step calibrates the geological models with available data (e.g., production and time-lapse seismic data) to reduce the geological uncertainty which is usually represented by a set of reservoir models shown in Fig. 1.1; Then a set of optimal well controls are determined based on the new geological model(s) with the production optimization process and these optimal well controls will be used to operate the field. For the data assimilation step, the current popular choice is the ensemble Kalman filter (EnKF), in which a set of geological models are all sequentially updated [18, 19, 48]. EnKF avoids calculation of sensitivities and iteration and is thus far more computationally efficient than the randomized likelihood method [16, 33, 52]. Model calibration and production optimization alternate in the "closed-loop" reservoir management process. However, the focus of this study is on the production optimization step of the closed-loop reservoir management.

Most of the past studies on production optimization [1, 2, 6, 7, 15, 23, 24, 29, 32, 39, 40, 47, 49] focus on optimizing the reservoir performance under waterflood-

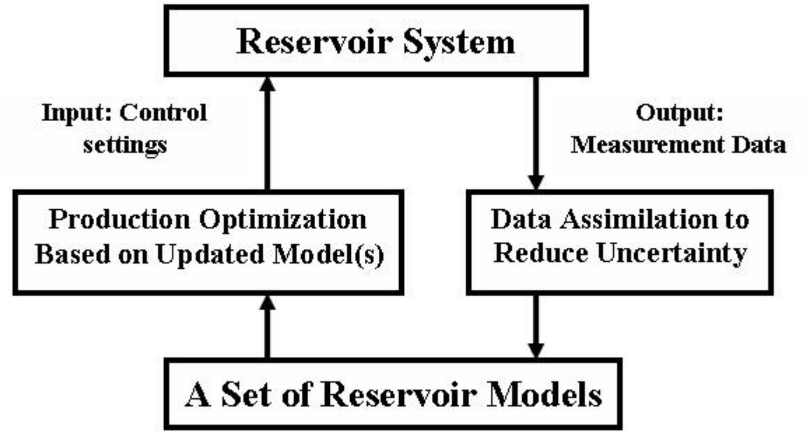

Reservoir System
Input: Control settings
Output: Measurement Data
Production Optimization Based on Updated Model(s)
Data Assimilation to Reduce Uncertainty
A Set of Reservoir Models

Figure 1.1: Closed-loop reservoir management.

ing, because waterflooding is by far the most commonly used method to enhance oil recovery after primary depletion. Poor recovery from waterflooding techniques may be caused by high permeability channels between injectors and producers and hence early water breakthrough. In early studies, the well controls are usually adjusted to maximize the recovery factor or sweep efficiency; see, for example, Asheim [2], who investigated the optimization of the net present value (NPV) of waterflooding with multiple vertical injectors and a vertical producer by rate allocation. There are two major challenges in production optimization: one is how to honor the physical constraints (e.g., the maximum allowable water injection rate or production liquid rate, the minimum allowable producer BHP constraint) and an economic limit on WOR or GOR when maximizing the NPV and another one is how to obtain a set of optimal control settings when reservoir uncertainty exists.

1.1 Constraint Types in Production Optimization

The constraints encountered in production optimization may be categorized into three types: 1) simple bound constraint which is the lower limit or the upper limit on the control variables; 2) control-only constraints which are explicit functions of the control variables only; 3) and state-control constraints which are functions of the control variable through the reservoir simulator, hence nonlinear functions of the primary variables (state) and control variables. Constraints on WOR, GOR and total fluid voidage are of the last type of constraints, which is the most challenging part of the production optimization problem.

1.1.1 Simple bound constraint

Simple bound constraints on the control variables are sometimes dealt with by limiting the step size during optimization [24]. Although the method is simple, only one variable reaches its bound at a time, which can be very inefficient for a problem with "Bang-Bang" control behavior [42, 43]. For the "Bang-Bang" control behavior, each component of the control vector takes either its minimum or maximum allowed values. As shown in [50], the optimal controls are "Bang-Bang" if the objective function is linear in the controls and there are only simple bound constraints for the control variables. Unfortunately, the objective function to be maximized is usually a non-linear function of control variables and there are other physical or economic limits in additional to the simple bound constraints. Instead of simply limiting the step size during optimization, Wang et al. [47] and Zhao et al. [54] applied a simple log-transformation to ensure the control variables stay within their bounds during the optimization. For the log-transformation method, a transformed new variable s_i is defined as

$$s_i = \ln\left(\frac{u_i - u_i^{\text{low}}}{u_i^{\text{up}} - u_i}\right), \tag{1.1}$$

for $i = 1, 2, ..., N_u$, where N_u is the total number of controls; and u_i^{low} and u_i^{up}, respectively, denote the lower bound and the upper bound of control variable u_i. The simple bound constrained problem can be transformed to an unconstrained optimization problem by mapping the control from the original domain to the log domain. When the log-transformation is applied during optimization, all the operations are done in the log domain and the actual control variables are obtained using the inverse log-transformation:

$$u_i = \frac{\exp(s_i)u_i^{\text{up}} + u_i^{\text{low}}}{1 + \exp(s_i)}. \tag{1.2}$$

The log-transformation is an easy way to deal with the bound constraints, which eliminates these constraints in the optimization process. The ith control variable u_i approaches its lower bound (or upper bound), the transformed variable s_i approaches $-\infty$ (or $+\infty$). Another method to handle the bound constraint is to simply truncate the variables when they are out of bound. Simple truncation sometimes gives good results although the generally recommended optimization procedure for enforcing bound constraints is the more general gradient-projection method [31].

The basic idea of the projection gradient method is to handle the bound constraints in two steps. Assume we have a minimization problem; without losing generality, the maximization problem is converted to a problem of minimizing the negative objective function of maximization problem. In the first step, we search along the steepest descent direction for minimization. When a bound is encountered, the search direction is "bent" or projected onto the bound constraint so that all the points along the search path are feasible. The search path is then the piecewise linear search path; see Fig. 1.2. Note that this piecewise search direction is still downhill [30]. The first local minimizer along this piecewise search direction is called the Cauchy point. Finding the Cauchy point along the piecewise search direction requires the evaluation of the objective function at each location where the search direction is "bent", so this method can only be efficiently implemented with models that do not require expensive function evaluations, e.g. a quadratic approximation model of the nonlinear function. The second step of the gradient-projection method is to search the face on which the Cauchy point is located for a new (improved) minimum. During the second step in optimization, the variables that form the active bound constraints at the Cauchy point are fixed and optimization is done in the space of the variables of the inactive bound constraints.

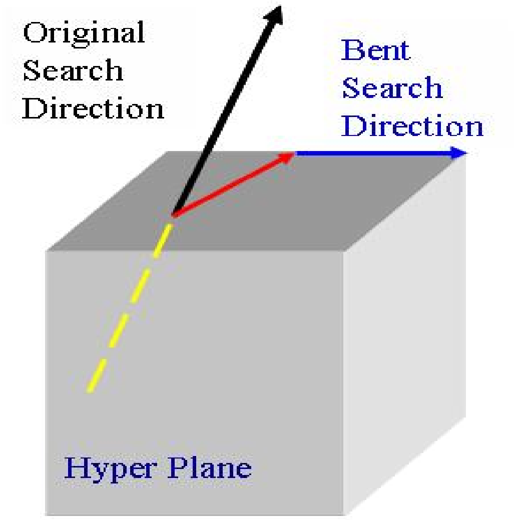


Figure 1.2: The piecewise-linear path for the gradient-projection method.

1.1.2 Control-only constraint

Control-only constraints consist of only control variables and are explicit linear function of control variables. For example, total water injection rate constraint is linear function of water injection rate controls. Control-only constraints are easily handled with Rosen's gradient-projection method [30, 37]. The general idea for Rosen's gradient-projection method is to project the unconstrained gradient onto the hyperplane of the linear equality constraints or the linear active inequality constraints, which ensures that any points on the search direction will satisfy the constraints. Zhang et al. [53] and Forouzanfar et al. [20] applied this method to handle a linear total injection rate constraint together with bound inequality constraints in their well placement problem.

1.1.3 State-control constraint

State-control constraints are nonlinear functions of the primary variables (state) and control variables. For example, the production liquid rate constraint is implicit nonlinear function of the producer BHP controls. The most troublesome constraints are the nonlinear state-control equality and inequality constraints. As the inequality constraints can be easily converted to equality constraints with active set method or slack variables [30, 31], here we only discuss the equality constraints without loss of generality. The only problem with this conversion is that it brings more variables into the optimization problem and increases the optimization problem size. In the production optimization literature, two adjoint-gradient-based strategies, generalized reduced gradient [15, 24, 49] and approximate feasible direction [40], have been applied to the nonlinear state-control equality of constraints.

In the generalized reduced gradient method (GRG), the control variables are divided into free (independent) and dependent control variables. In order to satisfy the constraints, the dependent variables are made dependent on the free variables which are completely independent. de Montleau et al. [15] claims the choice of the free variables is arbitrary. The search direction is composed of two parts: the free control part and the dependent control part. The free control part is the gradient of the objective function (NPV) with respect to the free control variables solved backward in time using adjoint formulation. The dependent control part of the search direction is then approximated using the linearized system (simulation) equations together with the linearized constraint equations forward in time at the current iteration. As the state-control constraints are nonlinear in nature, the approximation with linearization for the constraint equation cannot ensure the new controls along the search direction will satisfy the equality constraints and this forward solution may involve inversion of a large matrix [15, 49]. Kraaijevanger et al. [24] introduced slack variables to convert inequality constraints to equality constraint and these slack variables are treated as parameters to be optimized, which hence increases the size of the optimization problem. Zakirov et al. [49] and de Montleau et al. [15] only considered the situation when inequality constraints become active, which avoids using slack variables. As inactive constraints are not considered, it is possible to violate these constraints with a big step size. Therefore, Zakirov et al. [49] and de Montleau et al. [15] limited the step size to ensure that the constraints are satisfied during optimization iterations. The major disadvantage of the generalized reduced gradient method is that it requires modification of the Jacobian matrix in the reservoir simulator in order to get the coefficient matrix of the adjoint equations. Another disadvantage is that the initial control variables have to be carefully chosen to ensure they are in the feasible region. Otherwise, the inequality constraints are violated and cannot be converted to active equality constraints which will result in a premature stop in the optimization procedure.

The approximate feasible direction algorithm has its roots in the feasible direction methods [37]. Sarma et al. [40] compared different existing methods for nonlinear path constraints and proposed the approximate feasible direction algorithm. The basic idea for the feasible direction method is that it starts with a feasible point and then moves to a better feasible point with a usable feasible direction [37] for the iteration. If no constraints are active and there are no equality constraints, the search direction in the method is the steepest ascent direction, conjugate-gradient direction or the BFGS direction. However, in the presence of the equality constraints or active inequality constraints, the search direction (feasible direction) is obtained by solving a linear programming problem [37], which involves the gradient of the objective function and the gradient of the constraints to ensure the new points along the feasible direction do not violate the constraints within the neighborhood of the current points. The method requires the gradient of all the active constraints, which can be computationally expensive as each constraint needs one adjoint solution for gradient. Thus, Sarma et al. [40] lumped the constraints into one, which requires only one backward adjoint run for the gradient of the lumped constraint and saves computational time. Because of the constraint lumping, the individual constraints are not guaranteed to be satisfied at convergence.

The augmented Lagrangian method is implemented in LANCELOT [13, 31], where the authors suggest combining the penalty function method and Lagrangian multiplier method for the equality and inequality constraints and implement the gradient-projection method to deal with the bound constraints. The pure penalty method may result in ill-conditioning and numerical problems when the penalty parameter approaches 0 and hence the penalty term becomes significantly large. In Nocedal and Wright [31], it is shown that the augmented Lagrangian method avoids some numerical difficulties associated with the pure penalty method by introducing the Lagrangian function. Another advantage is the efficiency of gradient calculation. As all the constraints are incorporated into the objective function to be maximized with Lagrangian variables and penalty function, only one backward adjoint run is required for gradient calculation. At convergence, all the constraints are satisfied. This research will focus on the application of the augmented Lagrangian method to the production optimization problem. Note that our implementation of the augmented Lagrangian method is different from Doublet et al. [17], in which the authors proposed to combine flow equations with the NPV function into an augmented Lagrangian function. The computational saving of their algorithm arises from the fact that the simulation equations are not solved fully until convergence for each iteration of the optimization process, instead only an approximate solution for the simulation primary variables are obtained for the simulation equations. The simulation equations converge to the right solution when the optimization process reaches convergence.

1.2 Objective Functions of Production Optimization

Most work on production optimization [1, 2, 6, 7, 15, 23, 24, 29, 32, 39, 40, 47, 49] is based on a single reservoir model which may be the mean of the ensemble of reservoir models or any single reservoir model from an ensemble. Throughout, optimization based on a single reservoir model will be referred to as nominal optimization. As a single reservoir model may be far from the true geology, applying the optimal controls obtained from a single model to the actual field may not achieve the maximum NPV. To reduce the risk arising from the uncertainty in the geological description, van Essen et al. [46] proposed to optimize the expectation of NPV over a set of reservoir models. This procedure is referred to as robust optimization. van Essen et al. [46] compared robust optimization over 100 geological realizations with nominal optimization using a channelized synthetic reservoir example. They concluded that robust optimization yields not only a higher expected NPV but also a much smaller variance of NPV than is obtained with nominal optimization when the optimal controls obtained from nominal optimization are applied to the 100 realizations. Chen et al. [9, 10] applied robust optimization with an ensemble-based optimization scheme to closed-loop reservoir management. In the production optimization of [9, 10, 46], only simple linear constraints (e.g. constant total injection/production rates) and bound constraints are considered and their algorithms cannot easily be extended to handle general nonlinear constraints. In this study, we investigate the applicability of the robust optimization under bound, linear and nonlinear physical constraints (e.g. maximum allowable water-oil ratio) using the augmented Lagrangian method.

We define the long-term (or life-cycle) net-present-value (NPV), J_L, as

$$\begin{aligned} J_L(y_j, u, m_j) = & \sum_{n=1}^{N_L} \left[\sum_{i=1}^{N_p} \left(r_o^n q_{o,i}^n - r_w^n q_{w,i}^n + r_g^n q_{g,i}^n \right) - \sum_{i=1}^{N_{wi}} r_{wi}^n q_{wi,i}^n - \sum_{i=1}^{N_{gi}} r_{gi}^n q_{gi,i}^n \right] \\ & \frac{\Delta t^n}{(1+b)^{t^n/365}}, \end{aligned} \tag{1.3}$$

where m_j is a column vector of the reservoir model parameters for the jth geological realization of the reservoir; u is the control vector; y_j is the state vector consisting of the reservoir simulation primary variables for all simulation time steps calculated with the vector of well controls u and the jth realization of the vector of model parameters; N_L is the total number of simulation time steps; N_p is the total number of producers; N_{wi} is the total number of water injection wells; the superscript n denotes the nth simulation step; r_o is the oil revenue ($/STB); r_w is the water production cost (\$/STB); r_g is the gas production revenue (\$/SCF); r_{wi} is the water injection cost (\$/STB); r_{gi} is the gas injection cost (\$/SCF) $q_{o,i}^n$ is the average oil production rate over the nth time step of the ith producer (STB/day); $q_{w,i}^n$ is the average water production rate over the nth time step of the ith producer (STB/day); $q_{g,i}^n$ is the average gas production rate over the nth time step of the ith producer (SCF/day); $q_{wi,i}^n$ is the average water injection rate over the nth time step of the ith water injector (STB/day); $q_{gi,i}^n$ is the average gas injection rate over the nth time step of the ith gas injector (SCF/day); b is the annual discount rate; t^n is the cumulative time up to and including the nth simulator time step (days); Δt^n is the length of the nth simulator time step (days).

For the nominal life-cycle production optimization based on the jth realization m_j, we maximize the NPV, $J_L(y_j, u, m_j)$, i.e., the nominal optimization problem is

$$\max\ J_L(y_j, u, m_j), \tag{1.4}$$

subject to the following constraints,

$$\begin{aligned} e_i(y_j, u, m_j) &= 0, \quad i = 1, 2, ..., n_e; &(1.5)\\ c_i(y_j, u, m_j) &\leq 0, \quad i = 1, 2, ..., n_i; &(1.6)\\ u_i^{\text{low}} \leq u_i &\leq u_i^{\text{up}}, \quad i = 1, 2, ..., n_b. &(1.7) \end{aligned}$$

Here Eq. 1.5, Eq. 1.6 and Eq. 1.7, respectively, represent the equality, inequality and bound constraints.

For robust production optimization, we maximize the expectation of NPV, which is approximated as the average NPV over N_e reservoir models, i.e., the robust production optimization problem is given by

$$\max\ \mathrm{E}[J_L(y, u, m)] = \max \frac{1}{N_e} \sum_{j=1}^{N_e} J_L(y_j, u, m_j), \tag{1.8}$$

subject to the constraints in Eqs. 1.5 through 1.7 applied for $j = 1, 2, ..., N_e$, where N_e is the number of plausible reservoir models available. As pointed out in Ben-Tal and Nemirovski [5], in robust optimization, the constraints should be satisfied for all realizations.

Moreover, the objective function of production optimization can be divided into two types based on the length of production period: life-cycle optimization and short-term optimization. Typically, the objective function for production optimization is the NPV of production over the expected reservoir life. This is referred to as long-term (or life-cycle) production optimization. From the operator's point of view, it may be more important to maximize NPV over the next one or two years. The optimization of the NPV or its expectation over the next one or two years is referred to as short-term production optimization. Following long-term optimization, we maximize the expectation of the short-term NPV, where for a given m_j, the short-term NPV $J_s(y_j, u, m_j)$ is defined by

$$J_s(y_j, u, m_j) = \sum_{n=1}^{N_s} \left[\sum_{i=1}^{N_p} \left(r_o^n q_{o,i}^n - r_w^n q_{w,i}^n + r_g^n q_{g,i}^n \right) - \sum_{i=1}^{N_{wi}} r_{wi}^n q_{wi,i}^n \right] \frac{\Delta t^n}{(1+b)^{t^n/365}}, \tag{1.9}$$

with the total number of simulation time steps N_L replaced by N_s. N_s is the total number of simulation time steps for the short-term NPV and $N_s < N_L$. It is shown in [45] that long-term production optimization may be ill-posed and we may still be extra degrees of freedom in the control variables at the estimate of the maximum of NPV. van Essen et al. [45] proposed a hierarchical production optimization, in which they first maximize the life-cycle NPV and then maximize the short-term NPV using the extra degrees of the freedom in the estimate of optimal controls obtained by life-cycle optimization. As the hierarchical production optimization requires that we find an optimal control strategy for short-term NPV within the null space of the approximate Hessian estimated at the optimal controls obtained by life-cycle optimization, the procedure of [45] involves the computation of the Hessian matrix and its null space, which is very computationally expensive. The requirement for an accurate Hessian matrix makes the method infeasible for any realistic reservoir production optimization problems as current reservoir simulators are not capable of calculating the Hessian matrix and computing it from a finite-difference approximation is extremely expensive. van Essen et al. [45] proposed an alternative method (switching method) to avoid calculating explicitly the extra degrees of freedom in the control variables. Following the life-cycle optimization, the authors simply optimize the short-term NPV or the life-cycle NPV alternatively, i.e. if the life-cycle NPV is greater than or equal to the optimal life-cycle NPV, the short-term optimization is executed; otherwise, they switch to optimize the life-cycle NPV. The switching method is straightforward to implement; however, the authors point out the convergence of the method is slow due to the infeasible solution steps.

1.3 Algorithms Applied in Production Optimization

Based on how one calculates or approximates the gradient of the objective function with respect to the well controls, we categorize the current production optimization algorithms into two groups: gradient-free optimization algorithms and gradient-based optimization algorithms.

The gradient-free algorithms, such as finite-difference method, simultaneous perturbation stochastic approximation (SPSA) algorithm [41], ensemble-based optimization algorithm (EnOpt) [9, 10, 32], new unconstrained optimization algorithm (NEWUOA) [36] and quadratic interpolation model with an approximate gradient (QIM-AG) [54], treat the reservoir simulator as a black box and evaluate the approximate gradient based on the output of estimated objective values and the input of perturbations of control variables. The finite-difference method requires evaluation of at least $N_u + 1$ objective function values for each optimization iteration and the computational cost is extremely expensive. SPSA is a “simplified” finite-difference method as all the parameters are perturbed at one time stochastically and the SPSA gradient is then calculated from one-side or two-side perturbation. Although the SPSA gradient is stochastic, its expectation is proved to be the true gradient and always an uphill direction [47]. The SPSA algorithm has been applied in optimal well control [47, 54], optimal well placement [4] and history matching [22, 25]. EnOpt was first applied in the optimal well control problem in [29] and then developed by Nwaozo [32] and Chen et al. [9, 10]. EnOpt requires generating an ensemble of control vectors and running the reservoir simulator for each of these control vectors in order to calculate the EnOpt gradient from the cross-correlation between the control vector and NPV's. In EnOpt, the controls for each well are usually assumed to be correlated in time, which leads to the smooth optimal control settings. The implementation details of the EnOpt algorithm can be found in Appendix C. NEWUOA is a quadratic model-based derivative-free algorithm proposed by Powell [36]. In NEWUOA, constructing the quadratic model is based on quadratic interpolation, where the coefficients in the quadratic function are determined by a condition that the quadratic function and the objective function are equal at a set of interpolation points. To promote computational efficiency in large-scale optimization problems, the number of interpolation points is usually much less than the number of coefficients in the quadratic model. The extra degrees of freedom are used to minimize the Frobenius norm of the difference between the term representing the approximate Hessian matrix in the quadratic model at the previous iteration and the Hessian in the updated quadratic model for the current iteration, i.e.

$$\parallel G^l - G^{l-1} \parallel_F^2 = \sum_{i=1}^{N_u} \sum_{j=1}^{N_u} (G_{ij}^l - G_{ij}^{l-1})^2, \tag{1.10}$$

where the operator $\parallel \cdot \parallel_F^2$ represents the Frobenius norm, G^l is the approximate Hessian matrix at the lth iteration and the subscripts "ij" refer to the entry in the ith row and jth column of the matrix. The updated quadratic model is then maximized using a trust-region method. This quadratic model is updated during iteration as more and better interpolation points become available. As NEWUOA builds an initial quadratic model based on at least $N_u + 2$ interpolation points before the optimization starts, it is very inefficient when the number of control variables is very large. To overcome this limitation in NEWUOA, Zhao et al. [54] proposed the Quadratic Interpolation Modeling with Approximate Gradient (QIM-AG) algorithm based on a dynamic quadratic interpolation model for the maximization of the NPV. When this EnOpt preconditioned gradient is used, the algorithm is referred to as QIM-EnOpt, whereas when the SPSA gradient is used, the algorithm is referred to as QIM-SPSA. QIM-AG algorithm uses all available evaluated points to construct dynamically a quadratic interpolation model along the iterative process. For NEWUOA and QIM-AG, they all build a quadratic approximate model based on a set of interpolation points and then seek the optimum for the quadratic model. NEWUOA constructs the quadratic model by minimizing the Frobenious norm of the difference of the approximate Hessian between two consecutive iterations, but QIM-AG method constructs the quadratic model by minimizing the Frobenius norm of the approximate Hessian at the current iteration subject to the constraints that the quadratic model is equal to the objective function evaluated at all the interpolation points.

Currently, most of the gradient-free algorithms, such as SPSA and QIM-AG, are applied to handle the optimization problem with simple bound constraints in the literature of petroleum engineering. [9, 10] claim their ensemble-based optimization algorithm (EnOpt) is able to handle the problem with linear constraints as well as simple bound constraints. In the application of [9, 10], the total water injection rate and the total production liquid rate are linear functions of the control variables consisting of well injection rate and well production liquid rate. The total injection rate and total production liquid rate are truncated once they are violated, and then the total rate constraint is honored by reallocating the rates among wells proportionally according to the truncated values. Although this truncation technique seemed to give good results for their examples, it is not able to handle more complicated constraint types, e.g. state-control constraint.

Wang et al. [47] compared three different optimization algorithms: ensemble-based algorithm, SPSA algorithm, and the steepest ascent algorithm using finite-difference method and they concluded that the steepest ascent algorithm is the most efficient one and it gives reasonable results. In the examples of [47], using an average of 10-20 SPSA gradients as the search direction in the steepest ascent method, the SPSA method resulted in the same net present values (NPV) and well control obtained as using the true gradient, but the SPSA method required far more reservoir simulation runs. Zhao et al. [54] compared the QIM-AG algorithm with the adjoint-gradient-based gradient-projection method and several other gradient-free algorithms, including NEWUOA, SID-PSM (Pattern Search Method guided by Simplex Derivatives), PSO (Particle Swarm Optimization), EnOpt and SPSA. The optimization results in Zhao et al. [54] indicate that the QIM-AG algorithm is better than other gradient-free methods but the adjoint-gradient-based optimization method is still the most efficient method in terms of the number of simulation runs and also achieved the highest NPV for their examples.

The gradient based optimization algorithms require the computation of the gradient of the objective function with respect to control variables. As the NPV and the state-related constraints are implicit functions of the control variables, we are not able to calculate the gradient of the objective function explicitly. In the context of data assimilation (or history match), Li [26], Li et al. [27] and Gao [21] developed an adjoint procedure to obtain the gradient of the objective function by solving adjoint equations. de Montleau et al. [15], Kraaijevanger et al. [24] and Sarma et al. [40] developed the adjoint technique to calculate the gradient of NPV with respect to control variables and applied the adjoint gradient in the GRG algorithm and the approximate feasible direction method. The adjoint gradient calculation with respect to well controls has significant computational advantages when the number of variables is large, as the number of adjoint runs does not depend on the number of variables. Therefore, it is suitable for problems with a large number of well controls to be adjusted, which is often the case when many wells and/or many sections of wells (smart wells) are involved with many control steps during the expected reservoir life. In this study, we further develop the gradient-based optimization algorithm for production optimization problem.

1.4 Research Objectives and Dissertation Outline

1.4.1 Research Objectives

The major objective of this research is to develop practical optimization methods that can efficiently deal with large scale production optimization problems with bound, linear and nonlinear constraints. The specific items in plan are:

1. To develop the adjoint code that can efficiently calculate the gradient of the augmented Lagrangian function with respect to the control variables considering various constraints on WOR, GOR, production and injection rates.

2. To develop a large scale optimization code, in which the augmented Lagrangian method is used to ensure the general equality and inequality constraints and the gradient-projection trust-region method is used to ensure the bound constraints.

3. To develop methods dealing with scaling problems of different constraints, which can be a serious problem when the values of different constraints differ by several orders of magnitude, such as water oil ratio and rate constraints.

4. To develop methods to maximize the NPV of production on a set of reservoir models considering geological uncertainty.

5. To develop methods that can optimize multi-objective functions (e.g. life-cycle NPV and short-term NPV).

1.4.2 Dissertation Outline

There are six chapters and three appendices in this dissertation. In Chapter 2, we introduce the augmented Lagrangian method, including a discussion of updating the Lagrangian multipliers and the penalty parameter and a discussion of the gradient-projection trust-region algorithm. In Chapter 3, we briefly present the basic equations for a fully-implicit black oil reservoir simulator and the adjoint method for calculating the gradient of the augmented Lagrangian function. In Chapter 4, we use three synthetic reservoir models for production optimization and discuss the effect of scaling factors. For the case of only bound constraints, we also compare computational results of SPSA and EnOpt algorithms with gradient-based gradient-projection method in this chapter. In Chapter 5, we discuss the robust optimization for the life-cycle production optimization and the short-term production optimization. We extend the augmented Lagrangian method to multi-objective optimization, i.e., maximize the short-term objective without compromising the life-cycle objective. Chapter 6 presents the conclusions and summarizes the research contributions of this study. In Appendix A, we derive the equations for computing the derivatives of the augmented Lagrangian function with respect to the reservoir primary variables. In Appendix B, we briefly present the Generalized Reduced Gradient (GRG) method which is involved in Chapter 4. In Appendix C, we discuss the ensemble-based optimization method (EnOpt).

CHAPTER 2

CONSTRAINED OPTIMIZATION WITH AUGMENTED LAGRANGIAN METHOD

Augmented Lagrangian method is commonly used in the general literature dealing with equality and inequality constraints. In our implementation, the objective function to be maximized is defined as the augmented Lagrangian function consisting of the NPV and all constraints except the bound constraints. Although the augmented Lagrangian function is implemented in the LANCELOT package, the gradient of each term in the augmented Lagrangian function needs to be specified for its input deck, which is not practical as each gradient of the specified term requires one backward adjoint solution for the adjoint gradient calculation. In our implementation, the computation of the gradient of the augmented Lagrangian function (objective function) requires only one adjoint solution for one specified geological model.

The augmented Lagrangian method has two loops: outer-loop iterations and inner-loop iterations. At each outer-loop iteration of the optimization procedure, we only update either the Lagrange multipliers (λ's) or the penalty parameter (μ) presented in the formulation of augmented Lagrangian function. Once the controls that maximize the augmented Lagrangian function are obtained in the inner-loop iterations, we update in the outer-loop iteration the Lagrange multipliers or penalty parameter, depending on how well the constraints are satisfied, and move to the next inner-loop iteration. The above process is repeated until convergence. We discuss the procedure of the outer-loop iteration in the section 2.1. In the inner-loop iterations, both λ's and μ are fixed and the augmented Lagrangian function is to be maximized within the bound constraints. At each inner-loop iteration, the objective function is approximated by a quadratic model based on the adjoint gradient and the approximate Hessian matrix obtained using a quasi-Newton method. The quadratic model is then maximized subject to the bound constraints using a gradient-projection trust-region method [8, 13, 31, 44]. This step ensures all the bound constraints are satisfied. We discuss the gradient-projection trust-region method in the section 2.2. The advantage of the above procedure is that the bound constraints are easily handled using the gradient-projection method for a quadratic approximation of the objective function. Unlike the generalized reduced gradient method [15, 24, 49] and the approximate feasible direction method [40], the augmented Lagrangian method does not require the control vector to be feasible during optimization and hence one can start with any initial guess. However, the constraints are satisfied within a specified tolerance at convergence. We summarize our implementation of the augmented Lagrangian method in section 2.3.

2.1 The Augmented Lagrangian Function

For nominal optimization with a single reservoir model, we incorporate the general equality and inequality constraints into an augmented Lagrangian function, which is formulated as

$$\begin{aligned}\beta_L = & J_L[y,u] - \sum_{i=1}^{n_e}\left[\lambda_{e,i}e_i(y,u) + \frac{s_{e,i}}{2\mu}(e_i(y,u))^2\right] \\ & - \sum_{i=1}^{n_i}\left[\lambda_{c,i}(c_i(y,u)+v_i) + \frac{s_{c,i}}{2\mu}(c_i(y,u)+v_i)^2\right],\end{aligned} \tag{2.1}$$

where $J_L[y,u]$ is the life-cycle NPV defined in Eq. 1.3; The v_i is a positive slack variable used to convert the ith inequality constraint ($c_i \leq 0$) into an equality constraint; The λ's are the Lagrange multipliers; μ is the penalty parameter. The subscripts, e and c, respectively, represent the equality and inequality constraints. As the constraints may be of different scales, we rescale the constraints with scaling factors $s_{e,i}$ and $s_{c,i}$. We observed from the results of experiments shown in Chapter 4 that the convergence rate of the method can be slowed appreciably by poor scaling of the constraints. In this study, we use the following scaling factors:

$$s_{e,i} = \frac{1}{E_i^2}, \tag{2.2}$$

and

$$s_{c,i} = \frac{1}{C_i^2}, \tag{2.3}$$

where E_i and C_i, respectively, are the nonzero constraint values of the ith equality and the ith inequality constraints. For example, if the ith inequality constraint is WOR < 9.0, C_i is set to 9.0. If the ith equality constraint on the total water injection rate is 4000 STB/D, E_i is set to 4000.0. The E_i's and C_i's do not change during the optimization process. In fact, using the scaling factors is equivalent to using different penalty parameters for the constraints of different scales. It can be clearly seen that if we use

$$\mu_{e,i} = \mu / s_{e,i} \tag{2.4}$$

and

$$\mu_{c,i} = \mu / s_{c,i}, \tag{2.5}$$

the augmented Lagrangian function in the formulation of Eq. 2.1 can be rewritten as

$$\begin{aligned} \beta_L =& J_L[y,u] - \sum_{i=1}^{n_e} \left[\lambda_{e,i} e_i(y,u) + \frac{1}{2\mu_{c,i}} (e_i(y,u))^2 \right] \\ & - \sum_{i=1}^{n_i} \left[\lambda_{c,i}(c_i(y,u) + v_i) + \frac{1}{2\mu_{c,i}} (c_i(y,u) + v_i)^2 \right]. \end{aligned} \tag{2.6}$$

Eq. 2.6 indicates that we can also rescale the constraints by using different penalty parameters.

In the formulation of Eq. 2.1, the slack variables (v_i's) are additional adjustable parameters for optimization. The algorithm used in LANCELOT [14] treats the slack variables as the optimization variables, which requires an update of the slack variables at every iteration. However, it is possible to have inequality constraints on a group of wells and the field for a general production optimization problem (e.g., the produced fluids from a group of wells or the wells from the whole field flow into a separator, which has some certain maximum allowable capacity). Therefore, the number of slack variables may be equal to or even greater than the number of actual controls in the vector u, which increases the size of the production optimization problem using the augmented Lagrangian function defined in Eq. 2.1. Following Nocedal and Wright [31], we eliminate the slack variables for the inequality constraints in the augmented Lagrangian function. Note that the Lagrangian function of Eq. 2.1 is a concave quadratic function of the slack variable v_i. We denote the term in the last bracket in Eq. 2.1 as ψ_i, i.e.,

$$\psi_i = \lambda_{c,i}(c_i(y,u) + v_i) + \frac{s_{c,i}}{2\mu}(c_i(y,u) + v_i)^2. \tag{2.7}$$

The optimal value for v_i satisfies,

$$\frac{\partial \beta_L}{\partial v_i} = -\frac{\partial \psi_i}{\partial v_i} = -\lambda_{c,i} - \frac{s_{c,i}}{\mu}(c_i(y,u) + v_i) = 0. \tag{2.8}$$

Solving Eq. 2.8 for the slack variable v_i yields,

$$v_i = -c_i(y,u) - \frac{\mu \lambda_{c,i}}{s_{c,i}}. \tag{2.9}$$

If $c_i(y,u) > -\mu\lambda_{c,i}/s_{c,i}$, the slack variable calculated in Eq. 2.9 is negative and we set it equal to its lower bound of 0, i.e.,

$$v_i = 0, \text{ when } c_i(y,u) > -\mu\lambda_{c,i}/s_{c,i}. \tag{2.10}$$

Using Eqs. 2.9 and 2.10 in Eq. 2.7 yields,

$$\psi_i = \begin{cases} -\dfrac{\mu}{2s_{c,i}}\lambda_{c,i}^2, & c_i(y,u) \leq -\mu\lambda_{c,i}/s_{c,i}, \\ \lambda_{c,i}c_i(y,u) + \dfrac{s_{c,i}}{2\mu}(c_i(y,u))^2, & \text{otherwise.} \end{cases} \tag{2.11}$$

Eq. 2.11 indicates that the inequality constraint term ψ_i will become a constant if $c_i \leq -\mu\lambda_{c,i}/s_{c,i}$ is satisfied, otherwise the term ψ_i will become a quadratic function of c_i. With ψ_i given by Eq. 2.11, Eq. 2.1 can be rewritten as

$$\beta_L = J_L(y,u) - \sum_{i=1}^{n_e}[\lambda_{e,i}e_i(y,u) + \frac{s_{e,i}}{2\mu}(e_i(y,u))^2] - \sum_{i=1}^{n_i}\psi_i(y,u), \tag{2.12}$$

In the augmented Lagrangian method, we usually need to update the Lagrange multipliers (λ's) or the penalty parameter (μ) a few times before the optimization converges. This procedure of updating the λ's or the μ is referred to as the outer loop. With fixed value of the Lagrangian multipliers and a fixed value of the penalty parameter, we maximize the augmented Lagrangian function in Eq. 2.12 subject to the bound constraints, i.e., we solve the subproblem

$$\begin{aligned} &\max\ \beta_L(u), \\ &\text{subject to } u_i^{\text{low}} \leq u_i \leq u_i^{\text{up}}, i = 1, 2, ..., n_b. \end{aligned} \tag{2.13}$$

This maximization is performed using a gradient-projection trust-region method, where the objective function β_L is approximated by a quadratic function and the bound constraints are enforced explicitly when the quadratic function is maximized using the gradient-projection method. The quadratic function is updated iteratively to yield a more accurate representation of the objective function β_L to be maximized. This procedure is referred to as the inner loop.

2.2 Outer-loop Iteration

For the outer-loop iterations, we update the Lagrange multipliers λ's or the penalty parameter μ depending on the violation of the constraints. When the constraint violation is small, we update the Lagrange multipliers and do not change the value of the penalty parameter for the next step. When the constraint violation is large, we keep the Lagrange multipliers fixed and decrease the penalty parameter μ, which allows the algorithm to minimize the constraint violation during the next inner-loop step. Conn et al. [13] used an infinity norm of the constraint violation to determine whether the Lagrange multipliers or the penalty parameter is updated. Here, for production optimization with a single reservoir model, we define an average constraint violation parameter σ_{c_v} by

$$\sigma_{c_v} = \begin{cases} \sqrt{\dfrac{1}{n_v}\left[\displaystyle\sum_{i=1}^{n_e} s_{e,i}e_i^2 + \sum_{i=1}^{n_i} s_{c,i}(\max[0, c_i])^2\right]}, & n_v > 0, \\ 0, & n_v = 0, \end{cases} \tag{2.14}$$

where n_v is the total number of violated constraints. If σ_{c_v} is less than or equal to the tolerance η^ℓ [12, 13], which is also updated in the outer loop, we update the Lagrange multipliers λ's and η without changing μ as

$$\lambda_{e,i}^{\ell+1} = \lambda_{e,i}^{\ell} + \frac{s_{e,i}e_i(y^\ell, u^\ell)}{\mu^\ell}, \quad i = 1, 2, ..., n_e, \tag{2.15}$$

$$\lambda_{c,i}^{\ell+1} = \max[0, \lambda_{c,i}^{\ell} + \frac{s_{c,i}c_i(y^\ell, u^\ell)}{\mu^\ell}], \quad i = 1, 2, ..., n_i, \tag{2.16}$$

and

$$\eta^{\ell+1} = \eta^\ell(\mu^\ell)^{\beta_\eta}, \tag{2.17}$$

where ℓ is the outer-loop iteration index and β_η is a free parameter. Conn et al. [12] suggests to using $\beta_\eta = 0.9$ for well-scaled problems. Conn et al. [12] proved that by updating η with Eq. 2.17, the algorithm is guaranteed to achieve convergence without driving the penalty parameter to zero. The updating equations for the λ's in Eqs. 2.15 and 2.16 are obtained from comparing the gradient of the augmented Lagrangian function and the first-order necessary Karush-Kuhn-Tucker (KKT) conditions [31]. The gradient of β at the ℓth iteration can be written as

$$\nabla_u \beta_L^\ell = \nabla_u J_L^\ell - \sum_{i=1}^{n_e} \left[\left(\lambda_{e,i}^\ell + \frac{s_{e,i}}{\mu} e_i(y^\ell, u^\ell) \right) \nabla_u e_i(y^\ell, u^\ell) \right] - \sum_{i=1}^{n_i} \nabla_u \psi_i(y^\ell, u^\ell), \quad (2.18)$$

where

$$\nabla_u \psi_i(y^\ell, u^\ell) = \begin{cases} 0, & c_i(y^\ell, u^\ell) \le -\mu \lambda_{c,i}^\ell / s_{c,i}, \\ [\lambda_{c,i}^\ell + \dfrac{s_{c,i}}{\mu} c_i(y^\ell, u^\ell)] \nabla_u c_i(y^\ell, u^\ell), & \text{otherwise.} \end{cases} \quad (2.19)$$

Let u^* denotes the local solution of the maximization problem. If the optimal solution is achieved at the $(\ell+1)$th iteration, i.e. $u^{\ell+1} = u^*$, the first-order necessary KKT condition requires the gradient of a Lagrangian function of the NPV and the equality and inequality constraints at the local solution u^* to be zero, i.e.

$$\begin{aligned} &\nabla_u J_L(y^*, u^*) - \sum_{i=1}^{n_e} [\lambda_{e,i}^* \nabla_u e_i(y^*, u^*)] - \sum_{i=1}^{n_i} [\lambda_{c,i}^* \nabla_u c_i(y^*, u^*)] = 0, \\ &\lambda_{c,i}^* \ge 0, \end{aligned} \quad (2.20)$$

where y^* is the reservoir primary variables evaluate at u^* and $\lambda_{e,i}^*$ and $\lambda_{c,i}^*$ are the corresponding Lagrangian multipliers that satisfy the KKT condition. Conn et al. [12] show that if we use the updating equations defined in Eqs. 2.15, then $\lambda_{e,i}^{\ell+1}$ approaches $\lambda_{e,i}^*$. As inequality constraints can be converted to equality constraints with slack variables, we can deduce that $\lambda_{c,i}^{\ell+1}$ approaches $\lambda_{c,i}^*$ when using the updating equations defined in Eqs. 2.16.

Currently, there does not appear to be any theoretical results on how to initialize the Lagrange multipliers. One choice is to set the initial λ's equal to 0 so the augmented Lagrangian function in the first outer-loop iteration is actually the NPV plus a penalty term. Another choice follows from Eq. 2.15 and Eq. 2.16, where we estimate the initial λ's as

$$\lambda_{e,i}^0 = \frac{s_{e,i} e_i(y^0, u^0)}{\mu^0}, \quad i = 1, 2, ..., n_e, \tag{2.21}$$

and

$$\lambda_{c,i}^0 = \max[0, \frac{s_{c,i} c_i(y^0, u^0)}{\mu^0}], \quad i = 1, 2, ..., n_i. \tag{2.22}$$

This requires one simulation run to obtain values for $e_i(y^0, u^0)$ and $c_i(y^0, u^0)$ with the initial guess of the control vector u^0. Chapter 4 will show the experimental results on the effect of the initial values of Lagrange multipliers.

If σ_{c_v} is larger than η^ℓ, we update μ and η without changing λ's as

$$\mu^{\ell+1} = \tau \mu^\ell, \tag{2.23}$$

and

$$\eta^{\ell+1} = \eta^0 (\mu^{\ell+1})^{0.1}, \tag{2.24}$$

where τ is a constant between 0.1 and 0.5 and η^0 is the initial value of η. We use $\tau = 0.1$ and $\eta^0 = 0.1$ in the results presented in this dissertation. Overall, once

$$\sigma_{c_v} \leq 0.01 \tag{2.25}$$

or

$$n_v/(n_e + n_i) \leq 0.001, \tag{2.26}$$

we stop the optimization.

How to choose the initial guess for the penalty parameter μ can be a challenging task and the choice can have an impact on the performance of the algorithm. We have found that when doing robust optimization a significant saving in the overall computational time can be achieved by first examining the effect of the initial value of μ on a related nominal optimization problem. The initial value of μ for robust optimization is based on the results of experiments. We illustrate this procedure later in Chapter 5.

2.3 Inner-loop Iteration

This subsection details the inner-loop iteration. As mentioned earlier, the inner-loop maximizes the augmented Lagrangian function with fixed Lagrangian multipliers and penalty parameter. As the code was built for solving a minimization problem, β ($\beta = -\beta_L$ for long-term production optimization) is minimized with upper and lower bound constraints with fixed Lagrange multipliers and penalty parameter, i.e.,

$$\begin{aligned} \text{Min} \quad & \beta(u), \\ \text{Subject to} \quad & u^{\text{low}} \leq u \leq u^{\text{up}}, \end{aligned} \tag{2.27}$$

where u denotes the control vector with its lower and upper bound constraints u^{low} and u^{up}. For each inner-loop iteration, $\beta(u)$ is first approximated with a quadratic function using the gradient information. The quadratic function is then minimized using a projection-gradient trust-region method, which ensures the solution within the bounds.

The quadratic function at the kth inner-loop iteration is

$$Q^k(u) = \beta(u^k) + (u - u^k)^T g^k + \frac{1}{2}(u - u^k)^T B^k (u - u^k), \tag{2.28}$$

where g^k is the gradient of the objective function β with respect to u evaluated at u^k and B^k is the approximate Hessian matrix which is an $N_u \times N_u$ symmetric matrix obtained using the BFGS method. The update equation for B^k at the kth iteration is

$$B^k = B^{k-1} + \frac{y^{k-1}(y^{k-1})^T}{(y^{k-1})^T s^{k-1}} - \frac{B^{k-1}s^{k-1}(s^{k-1})^T B^{k-1}}{(s^{k-1})^T B^{k-1} s^{k-1}} \tag{2.29}$$

where $s^{k-1} = u^k - u^{k-1}$ and $y^{k-1} = g^k - g^{k-1}$ and $B^0 = I$. If the curvature condition $(s^{k-1})^T y^{k-1} > 0$ and B^{k-1} is real symmetric positive definite, then B^k is real symmetric positive definite [30, 51]. To precisely ensure that B^k is positive definite, [11] suggests to update B^k only when

$$(y^{k-1})^T s^{k-1}/(y^{k-1})^T y^{k-1} \geq 10^{-8}. \tag{2.30}$$

The gradient-projection trust-region method solves the following constrained minimization problem,

$$\begin{aligned} &\min\, Q^k(u), \\ &\text{subject to } \max[u_i^k - \Delta^k, u_i^{\text{low}}] \leq u_i \leq \min[u_i^k + \Delta^k, u_i^{\text{up}}], \quad i = 1, 2, ..., N_u \end{aligned} \tag{2.31}$$

where Δ^k is the trust-region radius at the kth iteration. Note that we use the infinity norm to define the shape of the trust region. This minimization problem is done in two stages [31]. During the first stage, a local minimizer (Cauchy point) is obtained by searching along the projected gradient direction. Starting with the steepest descent search direction from the current point, the search direction is “bent” when a bound is encountered so that the control vector stays within the trust-region and the upper and lower bound constraints. The first local minimizer of Q^k along the piecewise linear search path is referred to as the Cauchy point. In the second stage, the elements that have encountered the bounds during the Cauchy point calculation are fixed at their bounds and the quadratic function is then minimized in the subspace of the free variables, which are not at the bound, using a conjugate gradient method.

2.3.1 Cauchy point calculation

For the Cauchy point calculation, the search path before any bound is encountered is defined as

$$u(t) = u^k - tg^k, \tag{2.32}$$

where t is the step size and g^k is the gradient of the quadratic model evaluated at u^k, which is equal to the gradient of β at u^k. The step size for the ith entry of u to reach its bound is calculated by

$$\widetilde{t}_i = \begin{cases} (u_i^k - \min[u_i^k + \Delta^k, u_i^{\mathrm{up}}])/g_i^k, & g_i^k < 0, \\ (u_i^k - \max[u_i^k - \Delta^k, u_i^{\mathrm{low}}])/g_i^k, & g_i^k > 0, \\ \infty, & g_i^k = 0. \end{cases} \tag{2.33}$$

where u_i^k is the ith component of control vector u^k. These step sizes $\{\widetilde{t}_i\}$ $(i = 1, 2, ..., N_u)$ are then sorted in increasing order into a series $\{t_j\}$ $(j = 1, 2, ..., N_u)$ where $t_j \leq t_{j+1}$. Along the search path, $u(t_j)$ is called a break point at which a control variable reaches its bound. The line segment of the search path on the interval $[t_{j-1}, t_j)$ can be written as

$$u(\Delta t) = u^k(t_{j-1}) + \Delta t p_{j-1}^k, \tag{2.34}$$

where $u^k(t_{j-1})$ is the $(j-1)$st break point at the kth iteration, $\Delta t = t - t_{j-1} > 0$ and p_{j-1}^k is the projected gradient after encountering $(j-1)$ bounds. The projected gradient p_{j-1}^k is defined as

$$p_{j-1}^k = -A_{j-1} \circ g^k, \tag{2.35}$$

where $\circ$ is the Schur product and A_{j-1} is an N_u-dimensional vector. The vector A_{j-1} has $(j-1)$ "0" elements corresponding to the $(j-1)$ control variables that have encountered the bounds and the rest of the components in A_{j-1} are "1".

Substituting Eq. 2.34 into the quadratic model (Eq. 2.28) for the interval $[t_{j-1}, t_j)$ yields

$$\begin{aligned} Q^k\big(u(\Delta t)\big) =& \beta(u^k) + (u^k(t_{j-1}) + \Delta t\, p^k_{j-1} - u^k)^T g^k \\ &+ \frac{1}{2}(u^k(t_{j-1}) + \Delta t\, p^k_{j-1} - u^k)^T B^k (u^k(t_{j-1}) + \Delta t\, p^k_{j-1} - u^k) \\ =& f_{j-1} + f'_{j-1}\Delta t + \frac{1}{2} f''_{j-1} \Delta t^2, \end{aligned} \tag{2.36}$$

where $\Delta t \in [0, t_j - t_{j-1}]$; f_{j-1}, f'_{j-1} and f''_{j-1}, respectively, are given by

$$\begin{aligned} f_{j-1} &= \beta(u^k) + (u^k(t_{j-1}) - u^k)^T g^k + \frac{1}{2}(u^k(t_{j-1}) - u^k)^T B^k (u^k(t_{j-1}) - u^k), \\ f'_{j-1} &= (g^k)^T p^k_{j-1} + (u^k(t_{j-1}) - u^k)^T B^k (p^k_{j-1}), \\ f''_{j-1} &= (p^k_{j-1})^T B^k (p^k_{j-1}). \end{aligned} \tag{2.37}$$

Note that f''_{j-1} is always positive as B^k is positive definite.

Suppose that no local minimizer is found yet after searching all the line segments before t_{j-1}. To check if there is a local minimizer at the interval $[t_{j-1}, t_j)$, differentiating $Q^k\big(u(\Delta t)\big)$ with respect to Δt and setting it equal to zero yields,

$$\Delta t^* = -f'_{j-1}/f''_{j-1}. \tag{2.38}$$

If $\Delta t^* \in [0, t_j - t_{j-1})$, there is a local minimizer of $Q^k\big(u(\Delta t)\big)$ at $t^* = t_{j-1} + \Delta t^*$. If $f'_{j-1} > 0$, then $Q^k(\Delta t)$ is a monotone increasing function in $\Delta t^* \in [0, t_j - t_{j-1})$. Therefore, there is a local minimizer at $t^* = t_{j-1}$ when $f'_{j-1} > 0$. Then the Cauchy point is $u^c = u(t^*)$. In all other cases, there is no local minimizer in this interval and the search is moved to the next interval $[t_j,\, t_{j+1})$.

If the Cauchy point is identified in the interval $[t_{j-1}, t_j)$ and $j < N_u$, there are still some free variables and the subspace minimization is solved using a conjugate gradient method to obtain a better minimizer within the intersection of the trust region and the original bounds. If $j = N_u$, all the control variables are fixed at their bounds and no subspace minimization is necessary.

2.3.2 Subspace minimization

The components of the control vector that have reached the bounds (including upper and lower bounds and the trust-region bounds) at the Cauchy point u^c are fixed at their bounds. These components are defined as the active bound constraint set

$$\mathcal{A}(u^c) = \{i | u_i^c = \min[u_i^k + \Delta^k, u_i^{\text{up}}] \quad \text{or} \quad u_i^c = \max[u_i^k - \Delta^k, u_i^{\text{low}}]\}, \tag{2.39}$$

where u_i denotes the ith component in the vector of u. The subspace minimization problem minimizes $Q^k(u)$ by adjusting the control variables that are not in the active bound constraint set $\mathcal{A}(u^c)$ subject to the trust-region and bound constraints. Let $\hat{u}$ denote a column vector which consists all the components of u that are not in $\mathcal{A}(u^c)$. The quadratic function $Q^k(\hat{u})$ can be rewritten as

$$Q^k(\hat{u}) = \beta(u^k) + (\hat{u} - u^k)^T \hat{g}^k + \frac{1}{2}(\hat{u} - u^k)^T \hat{B}^k (\hat{u} - u^k), \tag{2.40}$$

where $\hat{g}^k$ and $\hat{B}^k$ are, respectively, the gradient and the Hessian of β with respective to $\hat{u}^k$. Note that it is not necessary to recalculate the gradient $\hat{g}^k$ and the Hessian matrix $\hat{B}^k$, as $\hat{g}^k$ is a sub-vector of the gradient g^k and $\hat{B}^k$ is a sub-matrix of the approximate Hessian matrix B^k. The components of $\hat{g}^k$ and $\hat{B}^k$ correspond to the control sub-vector $\hat{u}$. Then this minimization problem becomes

$$\begin{gathered} \min \, Q^k(\hat{u}), \\ \text{subject to } \max[\hat{u}_j^k - \Delta^k, \hat{u}_j^{\text{low}}] \le \hat{u}_j \le \min[\hat{u}_j^k + \Delta^k, \hat{u}_j^{\text{up}}], \end{gathered} \tag{2.41}$$

where $\widehat{u}_j$ is the jth component in the vector of $\widehat{u}$. Note that the components in the active set are fixed at the Cauchy point, i.e., $u_i = u_i^c$, $i \in \mathcal{A}(u^c)$ during subspace minimization. The subspace minimization is performed using the constrained conjugate-gradient algorithm in [11]. Let $\hat{r}$ denote the gradient of the quadratic function $Q^k(\hat{u})$,

$$\hat{r} = \hat{g}^k + \hat{B}^k(\hat{u} - \hat{u}^k). \tag{2.42}$$

For an unconstrained problem, the minimizer is found when the gradient of the objective function is zero. In this case, the minimization problem becomes a problem of solving the linear system of equations,

$$\hat{g}^k + \hat{B}^k(\hat{u} - \hat{u}^k) = 0. \tag{2.43}$$

Therefore, $\hat{r}$ is also referred to as the "residual". However, for the constrained problem in Eq. 2.41, the minimizer is found when the ℓ_2 norm of $\hat{r}$ is smaller than certain convergence tolerance.

In the constrained conjugate gradient algorithm [11], the conjugate gradient search direction at the (ν)th iteration is a combination of the current residual $\hat{r}^\nu$ and the previous conjugate gradient search direction $\hat{p}^{\nu-1}$,

$$\hat{p}^\nu = -\hat{r}^\nu + \frac{(\hat{r}^\nu)^T(\hat{r}^\nu)}{(\hat{r}^{\nu-1})^T(\hat{r}^{\nu-1})}\hat{p}^{\nu-1}. \tag{2.44}$$

Note that for initialization, $\hat{p}^0 = 0$, i.e., the search direction is the negative gradient direction of the quadratic model. The control sub-vector is then updated as

$$\hat{u}^\nu = \hat{u}^{\nu-1} + \alpha\hat{p}^\nu, \tag{2.45}$$

where α is the step size. If the curvature is not positive in the direction $\hat{p}^\nu$, i.e., $(\hat{p}^\nu)^T\hat{B}^k(\hat{p}^\nu) \leq 0$, the minimum of the quadratic function should be found at either the upper bound or the lower bound. Precisely, α takes the maximum allowable step size α_1 and the subspace minimization is terminated at $\hat{u}^* = \hat{u}^{\nu-1} + \alpha_1\hat{p}^\nu$. The maximum allowable step size α_1 is calculated such that the trust-region bounds and the original control vector bounds are not violated.

$$\alpha_1 = \min\left(\alpha_{\max,j}\right), \tag{2.46}$$

where

$$\alpha_{\max,j} = \begin{cases} (\max[\hat{u}_j^k - \Delta^k, \hat{u}_j^{\text{low}}] - \hat{u}_j^{\nu-1})/\hat{p}_j^\nu, & \text{if} \quad \hat{p}_j^\nu < 0, \\ (\min[\hat{u}_j^k + \Delta^k, \hat{u}_j^{\text{up}}] - \hat{u}_j^{\nu-1})/\hat{p}_j^\nu, & \text{if} \quad \hat{p}_j^\nu > 0. \end{cases} \tag{2.47}$$

If $(\hat{p}^\nu)^T \hat{B}^k (\hat{p}^\nu) > 0$, then the step size α is calculated as

$$\alpha = \min(\alpha_1, \, \alpha_2), \tag{2.48}$$

where α_2 is calculated by

$$\alpha_2 = \frac{(\hat{r}^\nu)^T (\hat{r}^\nu)}{(\hat{p}^\nu)^T \hat{B}^k (\hat{p}^\nu)}. \tag{2.49}$$

If the step size takes the control vector to the bound, i.e., $\alpha_1 \leq \alpha_2$, the subspace minimization is terminated at $\hat{u}^* = \hat{u}^\nu$. Otherwise, the subspace minimization continues to the next iteration $\nu + 1$. If no bound is encountered during the subspace minimization, the conjugate gradient algorithm is terminated when the square root of the ℓ_2 norm of the residual is smaller than a prescribed value ϵ^k,

$$\sqrt{(\hat{r}^\nu)^T (\hat{r}^\nu)} \leq \epsilon^k. \tag{2.50}$$

To assure a superlinear convergence rate, Conn et al. [11] suggests to using the convergence criterion parameter ϵ^k determined by

$$\epsilon^k = \min\left(0.1, \sqrt{\parallel \overline{g}^k \parallel_\infty}\right) \parallel \overline{g}^k \parallel_\infty, \tag{2.51}$$

where $\overline{g}^k$ is the projected gradient

$$\overline{g}_j^k = \begin{cases} \hat{u}_j^k - \max[\hat{u}_j^k - \Delta^k, \hat{u}_j^{\text{low}}], & \text{if } \hat{u}_j^k - \hat{g}_j^k \leq \max[\hat{u}_j^k - \Delta^k, \hat{u}_j^{\text{low}}], \\ \hat{u}_j^k - \min[\hat{u}_j^k + \Delta^k, \hat{u}_j^{\text{up}}], & \text{if } \hat{u}_j^k - \hat{g}_j^k \geq \min[\hat{u}_j^k + \Delta^k, \hat{u}_j^{\text{up}}], \\ -\hat{g}_j^k, & \text{Otherwise.} \end{cases} \tag{2.52}$$

2.3.3 The non-monotone trust-region method

Once the minimizer of the quadratic function in Eq. 2.31 is found using the above gradient-projection trust-region method, the objective function β is evaluated by running the reservoir simulator. Before moving to the $(k+1)$st iteration, the trust-region radius for the next iteration Δ^{k+1} is updated according to the quality of the quadratic approximation to the objective function. The quality of the quadratic approximation is defined as

$$\rho_1^{(k)} = \frac{\beta(u^k) - \beta(\widetilde{u}^{k+1})}{Q^k(u^k) - Q^k(\widetilde{u}^{k+1})}, \tag{2.53}$$

where $\widetilde{u}^{k+1}$ is the minimizer of quadratic function. Large positive $\rho_1^{(k)}$ value indicates that the objective function β decreases as the quadratic function decreases and the quadratic function is a good approximation to the objective function. A small or negative value of $\rho_1^{(k)}$ value indicates the quality of the quadratic function is poor. The method which directly uses the quality parameter, $\rho_1^{(k)}$, to obtain the trust-region radius for the next iteration is called a monotone type method, in which the algorithm only accepts $\widetilde{u}^{k+1}$ if $\rho_1^{(k)}$ is greater than or equal to a specified positive value, e.g. 0.2. However, for a large-scale problem with a large number of control variables, the quadratic model may be a very poor approximation to the objective function and the quality parameter $\rho_1^{(k)}$ can be significantly smaller than the specified value, e.g. 0.2, for two consecutive iteration. In this case, the trust-region radius will shrink very quickly and hence the optimization converges quickly to a local solution. Toint

[44] proposed a non-monotone trust-region method, which checks the quality of the quadratic approximation using not only two successive iterations as in Eq. 2.53 but also a quadratic approximation quality parameter $\rho_2^{(k)}$ within the past few iterations. The quality parameter, $\rho_2^{(k)}$, is defined as

$$\rho_2^{(k)} = \frac{\beta_h - \beta(\widetilde{u}^{k+1})}{Q_h - Q^k(\widetilde{u}^{k+1})}, \tag{2.54}$$

where β_h is the maximum objective function value between the $(k-h)$th iteration and the kth iteration and Q_h is the value of quadratic model corresponding to the iteration where β_h is computed. In Toint [44], the trust-region radius is updated according to the quadratic approximation quality parameter $\rho^k = \max[\rho_1^{(k)}, \rho_2^{(k)}]$ as follows

$$\Delta^{k+1} = \begin{cases} \alpha_1 \Delta^k, & \rho^k < \xi_1, \\ \Delta^k, & \xi_1 \leq \rho^k < \xi_2, \\ \alpha_2 \Delta^k, & \text{Otherwise}, \end{cases} \tag{2.55}$$

where $0 < \xi_1 \leq \xi_2 < 1$ and $0 < \alpha_1 < 1.0 < \alpha_2$. If $\rho^k \geq \xi_1$, the trial point $\widetilde{u}^{k+1}$ is accepted, i.e. $u^{k+1} = \widetilde{u}^{k+1}$; otherwise, $u^{k+1} = u^k$. For the examples in this dissertation, the following values are used: ξ_1=0.2, ξ_2=0.8, α_1=0.0625, α_2=2.0.

2.4 Summary of Algorithm

Here we summarize the steps of the augmented Lagrangian algorithm.

Step 1: Initialize the outer-loop by setting $\ell = 0$, where ℓ is the outer-loop iteration index. Set $\eta^0 = 0.1$ and choose an initial value for the penalty parameter μ. Note that the initial Lagrange multipliers are either set equal to zero or computed using Eqs. 2.21 and 2.22.

Step 2: Inner-loop optimization. We minimize a function β^ℓ, which is the negative of the augmented Lagrangian function β_L defined by Eq. 2.12 with the

Lagrange multipliers and the penalty parameter determined in the outer loop. The minimization problem in the inner-loop is given by Eq. 2.27.

Step 2.1: Initialization of inner loop. Set the initial trust-region radius Δ^0 and $k = 0$, where k is the iteration index for the inner-loop iterations. The initial Hessian matrix B^0 is set to the identity matrix.

Step 2.2: Calculate the gradient of $\beta^{k,\ell}$ and update the quadratic model Q^k which approximates $\beta^{k,\ell}$. Note that we update the approximate Hessian matrix B^k using the BFGS method.

Step 2.3: Use the infinity norm to define the shape of the trust region. Apply the gradient-projection trust-region method to find an optimal solution $u^{k,\ell}$ for the minimization problem given by

$$\begin{aligned} &\min Q^k(u), \\ &\text{subject to } \max[u_i^{k,\ell} - \Delta^k, u_i^{\text{low}}] \leq u_i \leq \min[u_i^{k,\ell} + \Delta^k, u_i^{\text{up}}], \quad i = 1, 2, ..., N_u. \end{aligned} \tag{2.56}$$

Step 2.4: Check the convergence criteria for the inner-loop iterations. The stopping criteria used here are: k is greater than the maximum allowable iteration number; Or the relative change in the function β is less than or equal to 10^{-4}, i.e.

$$\frac{\beta(u^{k,\ell}) - \beta(u^{k+1,\ell})}{\beta(u^{k,\ell})} \leq 10^{-4}, \tag{2.57}$$

and the relative change of the ℓ_2 norm of control vector is smaller than or equal to 10^{-3}, i.e.

$$\frac{\|u^{k+1,\ell} - u^{k,\ell}\|}{\max(\|u^{k,\ell}\|, 1.0)} \leq 10^{-3}; \tag{2.58}$$

Or the trust-region radius is smaller than a minimum allowable value $r_{\min}$; $r_{\min}$ is generally set to a very small value, e.g. 10^{-5}; however, a very small trust-region radius is not a good choice for production optimization, as the objective function may not be sensitive to a small change of control variables. In this study, we choose to use $r_{\min}$ defined by

$$r_{\min} = 0.001 \times \min_{1 \leq i \leq N_u} [u_i^{\text{up}} - u_i^{\text{low}}]. \tag{2.59}$$

If the criteria are not satisfied, then set $k = k + 1$, update trust region radius with Eq. 2.55 and go back to Step 2.2; otherwise, exit the inner loop and continue to Step 3.

Step 3: Check the average constraint violation parameter σ_{c_v} defined in Eq. 2.14. If $\sigma_{c_v} \leq 0.01$ or $n_v/[N_e \times (n_e + n_i)]$ is less than 0.001, the outer loop converges and the whole algorithm is terminated; otherwise, then do the following:

Step 3.1: If $0.01 \leq \sigma_{c_v} \leq \eta^\ell$, keep the penalty parameter unchanged, i.e. $\mu^{\ell+1} = \mu^\ell$, update the Lagrange multipliers based on Eq. 2.15 and Eq. 2.16 and update $\eta^{\ell+1} = \eta^\ell(\mu^\ell)^{0.9}$.

Step 3.2: If $\sigma_{c_v} > \eta^\ell$, keep the Lagrange multipliers unchanged and update the penalty parameter $\mu^{\ell+1} = \tau\mu^\ell$ and $\eta^{\ell+1} = \eta^0(\mu^{\ell+1})^{0.1}$.

Step 3.3: Set $\ell = \ell + 1$, go back to Step 2.

The above algorithm follows the idea in the LANCELOT [13]. The major differences are:

1. The inequality constraints are handled following the idea introduced in [31] as shown in Eqs. 2.11 and 2.12. We do not introduce the slack variable explicitly. However, LANCELOT uses the slack variables to convert the inequality constraints to equality constraints and treats the slack variables as pseudo control variables. Although the gradient of objective function with respect to the slack variables is easy to calculate, it will inevitably enlarge the dimension of the gradient and the approximate Hessian matrix and possibly make the problem more ill-posed.

2. To check convergence of the inner-loop iteration, LANCELOT only check whether

the ℓ_2 or infinity norm of projected gradient to the original bounds, u^{up} and u^{low}, is smaller than a tolerance (e.g. 10^{-3}). The projected gradient onto the original bounds can be formulated as

$$\overline{g}_i = \begin{cases} u_i - u_i^{\text{low}}, & \text{if } u_i^k - g_i^k \leq u_i^{\text{low}}, \\ u_i - u_i^{\text{up}}, & \text{if } u_i^k - g_i^k \geq u_i^{\text{up}}, \\ -g_i, & \text{otherwise.} \end{cases} \tag{2.60}$$

In many experiments, the trust-region radius reduces to a small value due to the poor approximation of quadratic model to original objective function. The trust-region radius restricts the change of control variables for the next iteration. In this case, the $\ell - 2$ or infinity norm of the projected gradient onto original bounds may be much larger than the distance between the current control vector and the trust-region bounds. As a result, the inner-loop iteration will not stop until the convergence criteria is satisfied. The gradient projection onto the original bound as in Eq. 2.60 makes the inner-loop convergence criterion very stringent, especially at the early outer-loop iterations, where an accurate solution to an inaccurate approximation is not necessary. This criterion makes the algorithm very inefficient, as many simulation runs are required for these inner-loop iterations. In this study, we relax the convergence criterion for the inner-loop iteration and use the stopping criteria described in Step 2.4. Our limited experiments show this modification results in a more efficient algorithm for production optimization.

3. Another difference is that LANCELOT requires the gradient of each constraint, which requires one adjoint solution per constraint, whereas our algorithm only requires the gradient of the whole augmented Lagrangian function, which requires only one adjoint run.

CHAPTER 3

ADJOINT GRADIENT CALCULATION

In this chapter, we develop the adjoint procedure for the gradient calculation of the augmented Lagrangian function with respect to the control variables for a black-oil simulator. Although all our examples pertain to a two-phase water-oil system, the adjoint procedure is developed for a general three-phase system.

3.1 The Reservoir Simulator

We use an in-house simulator, Chevron Limited Application Simulation System (CLASS), for this research. This simulator is based on a fully-implicit, finite-difference formulation of three-phase flow black-oil equations which is considered in a three-dimensional ($x-y-z$ coordinate) system. We suppose there are N_x, N_y and N_z gridblocks in the $x-$, $y-$ and $z-$ directions, respectively. Let N denote the total number of grid blocks so $N = N_x \times N_y \times N_z$. Capillary pressures and gravity effects are included in the simulator but will be neglected in the adjoint formulation. At the ith gridblock, there are three unknown primary variables, i.e., the grid block pressure (p_i), oil saturation ($S_{o,i}$) and gas saturation ($S_{g,i}$) for $S_{g,i} > 0$ or dissolved gas oil ratio ($R_{s,i}$) for $S_{g,i} = 0$. Throughout, we let x denote gas saturation or dissolved gas oil ratio at the ith gridblock, i.e.,

$$x_i = \begin{cases} S_{g,i}, & \text{for} \quad S_{g,i} > 0, \\ R_{s,i}, & \text{for} \quad S_{g,i} = 0. \end{cases} \tag{3.1}$$

In addition, the flowing wellbore pressure ($p_{wf,l}$) at the lth well at a specified depth is also considered as a primary variable. At each gridblock, we apply three basic finite-difference equations, each of which represents the mass balance for each of the three components, i.e., oil, water and gas. In addition, a constraint is applied at each of the N_w wells to yield N_w additional equations. At each well and each simulation time step, either an individual phase flow rate, the total flow rate or the wellbore pressure may be specified as a well constraint.

Throughout, we let the superscript n denote the simulation time step and y^n denote a column vector consisting of the set of primary variables at time t^n. The total number of simulation time steps is denoted by N_L. The total number of primary variables in y^n is $3 \times N + N_w$. For each time step t^n, we obtain the primary variable vector y^n from the simulator CLASS and compute all the derivatives required for constructing the adjoint system based on the rock and fluid property tables. Let U^n be a column vector which contains a set of control variables at simulation time step n. Note that the control vector u in Eq. 1.3 contains control variables at each control step which may have several corresponding simulation steps; however, $U = [(U^1)^T, (U^2)^T, ..., (U^{N_L})^T]^T$ contains control variables at each simulation time step, so the dimension of u is not the same as the dimension of U. We first calculate the gradient with respect to U and then convert to the gradient with respect to control vector u, which will be discussed later. Following Zhang [51], the finite difference equations for phase π ($\pi = o, w, g$) at gridblock i can be written as

$$f_{\pi,i}^{n+1} = f_{\pi,i}(y^{n+1}, y^n, U^{n+1}) = 0, \tag{3.2}$$

for $n = 0, 1, ..., N_L - 1$. The well constraints are represented by

$$f_{wf,l}^{n+1} = f_{wf,l}(y^{n+1}, y^n, U^{n+1}) = 0, \tag{3.3}$$

for $l = 1, 2, ..., N_w$ and $n = 0, 1, ..., N_L - 1$. If the flowing wellbore pressure $p_{wf,l}^{n+1}$ at the datum depth is specified to be equal to $p_{wf,l,0}^{n+1}$, which can be a component in U^{n+1}, the well constraint equations can be simplified to

$$f_{wf,l}^{n+1} = p_{wf,l}^{n+1} - p_{wf,l,0}^{n+1} = 0. \tag{3.4}$$

In CLASS, the three equations for three-phase problem that are solved at gridblock i are

$$f_{1,i}^{n+1} = f_{o,i}^{n+1} + f_{w,i}^{n+1} + f_{g,i}^{n+1} = 0, \tag{3.5}$$

$$f_{2,i}^{n+1} = f_{o,i}^{n+1} = 0, \tag{3.6}$$

$$f_{3,i}^{n+1} = f_{g,i}^{n+1} = 0, \tag{3.7}$$

where $n = 0, 1, ..., N_L - 1$. As the rock properties are specified or assumed to be known, the complete system of simulation equations can be formally written as

$$f^{n+1} = f(y^{n+1}, y^n, U^{n+1}) = \begin{bmatrix} f_{1,1}^{n+1} \\ f_{o,1}^{n+1} \\ f_{g,1}^{n+1} \\ f_{1,2}^{n+1} \\ \vdots \\ f_{g,N}^{n+1} \\ f_{wf,1}^{n+1} \\ \vdots \\ f_{wf,N_w}^{n+1} \end{bmatrix} = 0. \tag{3.8}$$

In CLASS, Eq. 3.8 is solved by the Newton-Raphson method [3].

3.2 Adjoint Gradient Calculation

Maximization of the augmented Lagrangian function defined Eq. 2.12 in the inner loop iterations using the gradient-projection trust-region method requires computation of the gradient of the augmented Lagrangian function, $\beta[y, u]$, for the specified geological model m_j. As the augmented Lagrangian function consists of the NPV and terms representing constraints, the objective function, $\beta[y,u]$, is an expression of the well flow rates, which are functions of some primary variables, $y^1, ..., y^{N_L}$, and the control variables, $u^1, u^2, ..., u^{N_c}$, where N_c is the number of control steps. The primary variables and the control variables are independent variables during adjoint gradient derivation.

As each control step may contain several simulation time steps, the control variables at a specified control step are identical to the control variables at the corresponding simulation time steps. We first calculate the gradient of β with respect to U and then convert to the gradient of β with respect to u. If we use k_i to denote the number of simulation steps for the ith control step, then

$$\sum_{i=1}^{N_c} k_i = N_L, \tag{3.9}$$

and

$$\begin{aligned} u^1 =& U^0 = U^1 = ... = U^{k_1}, \\ u^2 =& U^{k_1+1} = ... = U^{k_1+k_2}, \\ &\vdots \\ u^{N_c} =& U^{k_1+...+k_{(N_c-1)}} = ... = U^{N_L}. \end{aligned} \tag{3.10}$$

Therefore, $\beta[y,u]$ can also be written as $\beta[y,U]$ which is the function of the control variables, U^n, at each simulation time t^n, $n = 1, 2, ..., N_L$. To calculate the gradient of the the functional for the specified geological model m_j, we define the Lagrangian function as

$$\Phi = \beta[y,U] + \sum_{n=0}^{N_L} (\lambda^{n+1})^T f^{n+1}, \tag{3.11}$$

where λ^{n+1} is a vector of adjoint variables at simulation time t^{n+1}, and is given by

$$\lambda^{n+1} = [\lambda_1^{n+1}, \lambda_2^{n+1}, ..., \lambda_{3\times N+N_w}^{n+1}]^T. \tag{3.12}$$

The total derivatives of Φ is

$$\begin{aligned}d\Phi =& d\beta + \sum_{n=0}^{N_L} \Big\{ (\lambda^{n+1})^T [\nabla_{y^{n+1}} (f^{n+1})^T]^T dy^{n+1} + (\lambda^{n+1})^T [\nabla_{U^{n+1}} (f^{n+1})^T]^T dU^{n+1} \Big\} \\ &+ \sum_{n=0}^{N_L} (\lambda^{n+1})^T [\nabla_{y^n} (f^{n+1})^T]^T dy^n \\ =& d\beta + \sum_{n=1}^{N_L} \Big\{ [(\lambda^n)^T [\nabla_{y^n} (f^n)^T]^T + (\lambda^{n+1})^T [\nabla_{y^n} (f^{n+1})^T]^T] dy^n \\ &+ (\lambda^n)^T [\nabla_{U^n} (f^n)^T]^T dU^n \Big\} + (\lambda^1)^T [\nabla_{y^0} (f^1)^T]^T dy^0 \\ &+ (\lambda^{N_L+1})^T [\nabla_{y^{N_L+1}} (f^{N_L+1})^T]^T dy^{N_L+1} + (\lambda^{N_L+1})^T [\nabla_{U^{N_L+1}} (f^{N_L+1})^T]^T dU^{N_L+1}.\end{aligned} \tag{3.13}$$

The initial conditions are fixed and independent of the model parameters and the control variables, i.e., $dy^0 = 0$. Choosing $\lambda^{N_L+1} = 0$ in Eq. 3.13 yields

$$\begin{aligned}d\Phi =& d\beta + \sum_{n=1}^{N_L} \Big\{ \big[(\lambda^n)^T [\nabla_{y^n} (f^n)^T]^T + (\lambda^{n+1})^T [\nabla_{y^n} (f^{n+1})^T]^T \big] dy^n \\ &+ (\lambda^n)^T [\nabla_{U^n} (f^n)^T]^T dU^n \Big\}.\end{aligned} \tag{3.14}$$

The total differential of β can be written as

$$d\beta = \sum_{n=1}^{N_L} \{ [\nabla_{y^n} \beta]^T dy^n + [\nabla_{U^n} \beta]^T dU^n \}. \tag{3.15}$$

Therefore, the total derivatives of Φ can be rewritten as

$$\begin{aligned}d\Phi =& \sum_{n=1}^{N_L} \{ [\nabla_{y^n} \beta]^T + (\lambda^n)^T [\nabla_{y^n} (f^n)^T]^T + (\lambda^{n+1})^T [\nabla_{y^n} (f^{n+1})^T]^T \} dy^n \\ &+ \sum_{n=1}^{N_L} \left\{ (\lambda^n)^T [\nabla_{U^n} (f^n)^T]^T + [\nabla_{U^n} \beta]^T \right\} dU^n.\end{aligned} \tag{3.16}$$

To obtain the adjoint gradient of β with respect to U^n, we first choose λ's as the solution of

$$[\nabla_{y^n}\beta]^T + (\lambda^n)^T[\nabla_{y^n}(f^n)^T]^T + (\lambda^{n+1})^T[\nabla_{y^n}(f^{n+1})^T]^T = 0. \tag{3.17}$$

Taking the transpose of Eq. 3.17 gives the adjoint system

$$[\nabla_{y^n}(f^n)^T]\lambda^n = -[\nabla_{y^n}(f^{n+1})^T]\lambda^{n+1} - \nabla_{y^n}\beta, \tag{3.18}$$

where

$$\nabla_{y^n}(f^n)^T = \begin{pmatrix} \frac{\partial f^n_{1,1}}{\partial p^n_1} & \frac{\partial f^n_{o,1}}{\partial p^n_1} & \frac{\partial f^n_{g,1}}{\partial p^n_1} & \cdots & \frac{\partial f^n_{g,N}}{\partial p^n_1} & \frac{\partial f^n_{wf,1}}{\partial p^n_1} & \cdots & \frac{\partial f^n_{wf,N_w}}{\partial p^n_1} \\ \frac{\partial f^n_{1,1}}{\partial S^n_{o,1}} & \frac{\partial f^n_{o,1}}{\partial S^n_{o,1}} & \frac{\partial f^n_{g,1}}{\partial S^n_{o,1}} & \cdots & \frac{\partial f^n_{g,N}}{\partial S^n_{o,1}} & \frac{\partial f^n_{wf,1}}{\partial S^n_{o,1}} & \cdots & \frac{\partial f^n_{wf,N_w}}{\partial S^n_{o,1}} \\ \frac{\partial f^n_{1,1}}{\partial x^n_1} & \frac{\partial f^n_{o,1}}{\partial x^n_1} & \frac{\partial f^n_{g,1}}{\partial x^n_1} & \cdots & \frac{\partial f^n_{g,N}}{\partial x^n_1} & \frac{\partial f^n_{wf,1}}{\partial x^n_1} & \cdots & \frac{\partial f^n_{wf,N_w}}{\partial x^n_1} \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \cdots & \vdots \\ \frac{\partial f^n_{1,1}}{\partial x^n_N} & \frac{\partial f^n_{o,1}}{\partial x^n_N} & \frac{\partial f^n_{g,1}}{\partial x^n_N} & \cdots & \frac{\partial f^n_{g,N}}{\partial x^n_N} & \frac{\partial f^n_{wf,1}}{\partial x^n_N} & \cdots & \frac{\partial f^n_{wf,N_w}}{\partial x^n_N} \\ \frac{\partial f^n_{1,1}}{\partial p^n_{wf,1}} & \frac{\partial f^n_{o,1}}{\partial p^n_{wf,1}} & \frac{\partial f^n_{g,1}}{\partial p^n_{wf,1}} & \cdots & \frac{\partial f^n_{g,N}}{\partial p^n_{wf,1}} & \frac{\partial f^n_{wf,1}}{\partial p^n_{wf,1}} & \cdots & \frac{\partial f^n_{wf,N_w}}{\partial p^n_{wf,1}} \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \cdots & \vdots \\ \frac{\partial f^n_{1,1}}{\partial p^n_{wf,N_w}} & \frac{\partial f^n_{o,1}}{\partial p^n_{wf,N_w}} & \frac{\partial f^n_{g,1}}{\partial p^n_{wf,N_w}} & \cdots & \frac{\partial f^n_{g,N}}{\partial p^n_{wf,N_w}} & \frac{\partial f^n_{wf,1}}{\partial p^n_{wf,N_w}} & \cdots & \frac{\partial f^n_{wf,N_w}}{\partial p^n_{wf,N_w}} \end{pmatrix}, \tag{3.19}$$

$$\nabla_{y^n}(f^{n+1})^T = \begin{pmatrix} \frac{\partial f^{n+1}_{1,1}}{\partial p^n_1} & \frac{\partial f^{n+1}_{o,1}}{\partial p^n_1} & \frac{\partial f^{n+1}_{g,1}}{\partial p^n_1} & \cdots & \frac{\partial f^{n+1}_{g,N}}{\partial p^n_1} & \frac{\partial f^{n+1}_{wf,1}}{\partial p^n_1} & \cdots & \frac{\partial f^{n+1}_{wf,N_w}}{\partial p^n_1} \\ \frac{\partial f^{n+1}_{1,1}}{\partial S^n_{o,1}} & \frac{\partial f^{n+1}_{o,1}}{\partial S^n_{o,1}} & \frac{\partial f^{n+1}_{g,1}}{\partial S^n_{o,1}} & \cdots & \frac{\partial f^{n+1}_{g,N}}{\partial S^n_{o,1}} & \frac{\partial f^{n+1}_{wf,1}}{\partial S^n_{o,1}} & \cdots & \frac{\partial f^{n+1}_{wf,N_w}}{\partial S^n_{o,1}} \\ \frac{\partial f^{n+1}_{1,1}}{\partial x^n_1} & \frac{\partial f^{n+1}_{o,1}}{\partial x^n_1} & \frac{\partial f^{n+1}_{g,1}}{\partial x^n_1} & \cdots & \frac{\partial f^{n+1}_{g,N}}{\partial x^n_1} & \frac{\partial f^{n+1}_{wf,1}}{\partial x^n_1} & \cdots & \frac{\partial f^{n+1}_{wf,N_w}}{\partial x^n_1} \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \cdots & \vdots \\ \frac{\partial f^{n+1}_{1,1}}{\partial x^n_N} & \frac{\partial f^{n+1}_{o,1}}{\partial x^n_N} & \frac{\partial f^{n+1}_{g,1}}{\partial x^n_N} & \cdots & \frac{\partial f^{n+1}_{g,N}}{\partial x^n_N} & \frac{\partial f^{n+1}_{wf,1}}{\partial x^n_N} & \cdots & \frac{\partial f^{n+1}_{wf,N_w}}{\partial x^n_N} \\ \frac{\partial f^{n+1}_{1,1}}{\partial p^n_{wf,1}} & \frac{\partial f^{n+1}_{o,1}}{\partial p^n_{wf,1}} & \frac{\partial f^{n+1}_{g,1}}{\partial p^n_{wf,1}} & \cdots & \frac{\partial f^{n+1}_{g,N}}{\partial p^n_{wf,1}} & \frac{\partial f^{n+1}_{wf,1}}{\partial p^n_{wf,1}} & \cdots & \frac{\partial f^{n+1}_{wf,N_w}}{\partial p^n_{wf,1}} \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \cdots & \vdots \\ \frac{\partial f^{n+1}_{1,1}}{\partial p^n_{wf,N_w}} & \frac{\partial f^{n+1}_{o,1}}{\partial p^n_{wf,N_w}} & \frac{\partial f^{n+1}_{g,1}}{\partial p^n_{wf,N_w}} & \cdots & \frac{\partial f^{n+1}_{g,N}}{\partial p^n_{wf,N_w}} & \frac{\partial f^{n+1}_{wf,1}}{\partial p^n_{wf,N_w}} & \cdots & \frac{\partial f^{n+1}_{wf,N_w}}{\partial p^n_{wf,N_w}} \end{pmatrix}, \tag{3.20}$$

and

$$\nabla_{y^n}\beta = \begin{pmatrix} \frac{\partial\beta}{\partial p_1^n}, & \frac{\partial\beta}{\partial S_{o,1}^n}, & \frac{\partial\beta}{\partial x_1^n}, & \cdots, & \frac{\partial\beta}{\partial x_N^n}, & \frac{\partial\beta}{\partial p_{wf,1}^n}, & \cdots, & \frac{\partial\beta}{\partial p_{wf,N_w}^n} \end{pmatrix}^T \tag{3.21}$$

Note that the coefficient matrix, $\nabla_{y^n}(f^n)^T$, in the adjoint system is simply the transpose of the Jacobian matrix evaluated at y^n which must be saved from the forward simulation run. The details on the evaluation of the terms in Eq. 3.19 and Eq. 3.20 can be found in [26, 51]. The augmented Lagrangian function β consists of two parts: the NPV and the augmented Lagrangian terms of constraints. The derivative of β with respect to primary variables can be written as

$$\nabla_{y^n}\beta = \nabla_{y^n}J_L - \sum_{i=1}^{n_e}\left[\left(\lambda_{e,i} + \frac{s_{e,i}}{\mu}e_i(y^n,U^n)\right)\nabla_{y^n}e_i(y^n,U^n)\right] - \sum_{i=1}^{n_i}\nabla_{y^n}\psi_i(y^n,U^n), \tag{3.22}$$

where

$$\nabla_{y^n}\psi_i(y^n,U^n) = \begin{cases} 0, & c_i(y^n,U^n) \le -\mu\lambda_{c,i}/s_{c,i}, \\ [\lambda_{c,i} + \frac{s_{c,i}}{\mu}c_i(y^n,U^n)]\nabla_{y^n}c_i(y^n,U^n), & \text{otherwise.} \end{cases} \tag{3.23}$$

In order to obtain the derivative of β with respect to the primary vector at the time step n, we need to calculate the derivatives of NPV and constraints with respect to y^n. For three-phase flow, the life-cycle NPV is given by

$$J_L(y,u) = \sum_{n=1}^{N_L}\Big[\sum_{i=1}^{N_p}\left(r_o^n q_{o,i}^n - r_w^n q_{w,i}^n + r_g^n q_{g,i}^n\right) - \sum_{i=1}^{N_{wi}} r_{wi}^n q_{wi,i}^n - \sum_{i=1}^{N_{gi}} r_{gi}^n q_{gi,i}^n\Big]\frac{\Delta t^n}{(1+b)^{t^n/365}}. \tag{3.24}$$

The derivative of life-cycle NPV with respect to y^n should be a summation of the derivatives of flow rates with respect to y^n over all the time steps. Similarly, the physical constraints, such as the field injection rate constraint, producing water oil ratio constraint or production well liquid rate, can be written as the functions of flow rates. For example, the water oil ratio (WOR) of the ith production well at time step n is given by

$$\text{WOR}_i^n = \frac{q_{w,i}^n}{q_{o,i}^n}. \tag{3.25}$$

Therefore, to calculate the derivatives of flow rates with respect to the primary vector y^n, it is essential to obtain $\nabla_{y^n}\beta$. We will provide the derivatives of flow rates with respect to the reservoir primary variables in Appendix A. We solve Eq. 3.18 with the condition $\lambda^{N_L+1} = 0$ backward in time (for $n = N_L, N_L - 1, ..., 1$) for the adjoint variables, λ^n, $n = N_L, ..., 1$. The matrices in Eq. 3.18 are independent of the adjoint variables, which indicates that the computational cost of solving the linear adjoint system is cheaper than that of solving the forward simulation equations which are nonlinear.

Once the adjoint solutions, λ^n, $n = 1, 2, ..., N_L$ are available, we can calculate the total differential of Φ with

$$d\Phi = \sum_{n=1}^{N_L} \left\{ (\lambda^n)^T [\nabla_{U^n}(f^n)^T]^T + [\nabla_{U^n}\beta]^T \right\} dU^n, \tag{3.26}$$

which is obtained by using Eq. 3.17 in Eq. 3.16. The gradient of the functional Φ with respect to U^n is

$$\nabla_{U^n}\Phi = [\nabla_{U^n}(f^n)^T]\lambda^n + \nabla_{U^n}\beta, \tag{3.27}$$

where $\nabla_{U^n}\beta$ represents the explicit partial derivative of the augmented Lagrangian function with respect to control U^n, while $\nabla_{U^n}\Phi$ represents the final gradient of the augmented Lagrangian function with respect to control U^n. Note that β is a function of the flow rates and we treat the flow rates as the functions of the primary variables.

If a well flowing BHP, denoted by p_{wf}, is specified, phase flow rates of that well are computed by Peaceman's Equation [34]. Throughout, we let B_m and μ_m ($m = o, w, g$) denote, respectively, the phase formation volume factor and the phase viscosity. The phase flow rates of a production well l at a completion of layer k (the wellbore gridblock with $z-$direction gridblock index equal to k) are

$$q^n_{o,il,jl,k} = \mathrm{WI}_{il,jl,k} \left(\frac{k_{ro}}{B_o \mu_o} \right)^n_{il,jl,k} (p^n_{il,jl,k} - p^n_{wf,il,jl,k}), \tag{3.28}$$

$$q^n_{w,il,jl,k} = \mathrm{WI}_{il,jl,k} \left(\frac{k_{rw}}{B_w \mu_w} \right)^n_{il,jl,k} (p^n_{il,jl,k} - p^n_{wf,il,jl,k}), \tag{3.29}$$

and

$$q^n_{g,il,jl,k} = \mathrm{WI}_{il,jl,k} \left(\frac{k_{rg}}{B_g \mu_g} + R_s \frac{k_{ro}}{B_o \mu_o} \right)^n_{il,jl,k} (p^n_{il,jl,k} - p^n_{wf,il,jl,k}), \tag{3.30}$$

where (il,jl) indicates the location of the producer well l; $\mathrm{WI}_{il,jl,k}$ denotes the well index term which represents the well geometry. The rates $q^n_{o,il,jl,k}$ and $q^n_{w,il,jl,k}$ are in units of STB/D, and the gas rate $q^n_{g,il,jl,k}$ has units of SCF/D. Similarly, the injection rate, $q^n_{wi,il,jl,k}$, of an injection well l is

$$q^n_{wi,il,jl,k} = \mathrm{WI}_{il,jl,k} \left(\frac{k_{rw}}{B_w \mu_w} \right)^n_{il,jl,k} (p^n_{wf,il,jl,k} - p^n_{il,jl,k}). \tag{3.31}$$

Note that we use a positive value for injection rate in NPV defined in Eq. 3.24. The flow rates in the functional β are actually the prediction data obtained with the forward simulation run and they are not explicit expressions of the control variables. Therefore, $\nabla_{U^n}\beta$ is always equal to 0. With $\nabla_{U^n}\beta = 0$, the gradient of functional with respect to U^n (Eq. 3.27) is rewritten as

$$\nabla_{U^n}\Phi = [\nabla_{U^n}(f^n)^T]\lambda^n. \tag{3.32}$$

Again, as the flow rates in flow equations are not explicit expressions for control variables, the derivatives of flow equations with respect to control variables are 0. The derivatives of well equations with respect to the controls are always -1. For example, if oil rate is specified in lth well, i.e. $U_l^n = q_{o,l}^n$, the well equation is

$$f_{wf,l}^n = \sum_k \mathrm{WI}_{il,jl,k} \left(\frac{k_{ro}}{B_o \mu_o} \right)_{il,jl,k}^n (p_{il,jl,k}^n - p_{wf,il,jl,k}^n) - q_{o,l}^n = 0. \tag{3.33}$$

The derivative of the well equation with respect to $q_{o,l}^n$ is -1. If the bottom hole pressure is specified in the lth well, i.e. $U_l^n = p_{wf,l}^n$, the well equation is

$$f_{wf,l}^n = p_{wf,il,jl}^n - p_{wf,l}^n = 0. \tag{3.34}$$

The derivative of the well equation with respect to $p_{wf,l}^n$ is -1.

As each control step may contain several simulation time steps, the control variables at each simulation time step can be treated as functions of the control variables at the corresponding control step. According to chain rule, the gradient of Φ with respect to the control variable for a specified well at the control step should be a summation of partial derivatives of Φ with respect to the control variables evaluated at the simulation time steps within the control step interval. From Eq. 3.10, we have

$$u^i = U^{k_1+...+k_{i-1}} = ... = U^{k_1+...+k_i}, \tag{3.35}$$

so

$$\nabla_{u^i} \Phi = \sum_{n=k_1+...+k_{i-1}}^{k_1+...+k_i} \nabla_{u^i} (U^n)^T \nabla_{U^n} \Phi. \tag{3.36}$$

From Eq. 3.35, we deduce that $\nabla_{u^i}(U^n)^T$ is equal to an identity matrix. Therefore,

$$\nabla_{u^i} \Phi = \sum_{n=k_1+...+k_{i-1}}^{k_1+...+k_i} \nabla_{U^n} \Phi = \sum_{n=k_1+...+k_{i-1}}^{k_1+...+k_i} \left\{ [\nabla_{U^n} (f^n)^T] \lambda^n \right\}. \tag{3.37}$$

CHAPTER 4

PRODUCTION OPTIMIZATION FOR A SINGLE RESERVOIR MODEL

We provide three cases in this chapter to study the performance of the augmented Lagrangian method for a single reservoir model. In the first case, we optimize NPV and test the choice of scaling factors, initial values for Lagrange multipliers and the penalty parameter, with a 2-dimensional reservoir. In the second case, we optimize NPV with two horizontal wells (one injector and one producer) in a waterflooding project and consider the optimization under an equality or inequality constraints. As the in-house simulator, CLASS (CHEVRON LIMITED APPLICATIONS SIMULATION SYSTEM), does not have a horizontal well model, we apply a vertical well for each segment in the horizontal well. We also compare the results obtained with the gradient-projection trust-region method to the results obtained from the ensemble-based optimization (EnOpt) algorithm for this case. In the third case, we optimize NPV for the Brugge field case which was a benchmark case set up by TNO (an independent research organization) in 2008 in order to test different optimization algorithms [35]. We compare the generalized reduced gradient (GRG) algorithm applied in simulator Eclipse 300 with the augmented Lagrangian method. The first two cases are done with the simulator CLASS, as the constraints are nonlinear functions of control variables. The third case is done with the simulator Eclipse 300, as the Brugge reservoir model has faults, CLASS is not able to handle this model.

4.1 Example 1: A 2D Fluvial Reservoir

In this example, we consider the optimization of waterflooding of a two-dimensional synthetic reservoir with a simulation model defined on a 25 × 25 × 1 grid. The grid block size is defined as $\Delta x = \Delta y = 200$ ft and the thickness of the reservoir is 20 ft. The porosity for the reservoir is homogeneous and fixed equal to 0.2. Fig. 4.1 shows the log-permeability distribution and the well locations. The reservoir is under waterflooding with 4 five-spot patterns as shown in Figs. 4.1, where the injectors INJ2, INJ4, INJ6 and INJ8 and the producers Pro2 and Pro3 are located in the high permeability region. As a result, water breakthrough would tend to occur earlier in Pro2 and Pro3 than in the other two producers. The reservoir has two phases, oil and water, and the water-oil relative permeability curves are shown in Fig. 4.2. The related reservoir properties are summarized in Table 4.1.

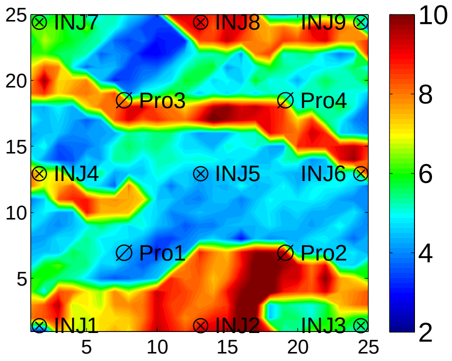


Figure 4.1: Log-permeability distribution.

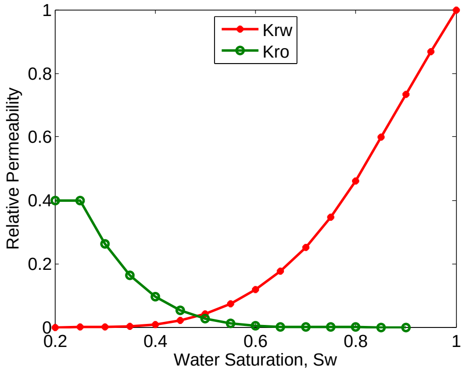


Figure 4.2: The water and oil relative permeability curve.

Table 4.1: Reservoir properties

Grid block size	200 ft
Thickness	20 ft
ρ_{osc}	56 lb/ft^3
ρ_{wsc}	62.4 lb/ft^3
μ_o	1.4 cp at 3601.5 psi
μ_w	0.5 cp
B_o	1.22 RB/STB at 3601.5 psi
B_w	1.0042 RB/STB
Rock Compressibility	3×10^{-5} psi^{-1}
Top Depth	4800 ft

All the producers are at BHP control with an upper bound of 6000 psi and a lower bound of 2000 psi. Each injector is under injection rate control with an upper bound of 1500 STB/D and a lower bound of 0. The anticipated total reservoir life is 1800 days and the control step size is set to 60 days so we have 30 control steps. The total number of controls is $13 \times 30 = 390$. In order to discuss the effect of scaling factors, we impose an average water/oil ratio (WOR) constraint over each control step for each producer and a field injection rate constraint. The WOR constraint (WOR ≤ 9.0) is a nonlinear function of the control variables through the reservoir simulator. The maximum allowable field injection rate is set to 5000 STB/D. Each control step has 4 WOR constraints plus 1 field injection rate constraint so the total number of inequality constraints is $5 \times 30 = 150$. The oil and water prices are set to $r_o = \$50$ /STB, $r_w = \$5.56$ /STB, $b = 10\%$ and $r_{w,inj} = \$0$. As there is no equality constraint, the augmented Lagrangian function can be rewritten as

$$\beta[u, \lambda_c, \mu] = J_L - \sum_{i=1}^{150} \psi_i, \tag{4.1}$$

where ψ_i is defined by Eq. 2.11 and the inequality c_i for field injection rate constraint is defined by

$$c_i = \sum_{j=1}^{9} q_{inj,j}^k - 5000 \leq 0, \quad k = 1, 2, ..., 30, \tag{4.2}$$

for the kth control step and the inequality for WOR constraint is defined by

$$c_i = q_{w,j}^k / q_{o,j}^k - 9 \leq 0, \quad k = 1, 2, ..., 30, \tag{4.3}$$

for the jth producer at the kth control step.

The initial guesses for all the producer BHP controls are equal to their lower bound of 2000 psi. The initial guesses for all the injection well rate controls are 560 STB/D. The initial total injection rate for the field is $560 \times 9 = 5040$ STB/D, which is a small violation of 40 STB/D over the constraint value. The purpose is to test whether our proposed estimate equation defined in Eq. 2.16 for the initial Lagrange multipliers λ's is suitable.

The comparison of the final NPVs is shown in Table 4.2. As WOR is comparatively a small value, a violation of the WOR constraint is negligible at early iterations of the optimization process unless we increase its weight by using a large scaling parameter or use an excessively small value of the penalty parameter. However, if we use an excessively small penalty parameter, the penalty term on field water injection rate is a large negative value, which makes the other parts in augmented Lagrangian function negligible. Rescaling reduces the number of iterations necessary to obtain convergence, which can be seen from Table 4.2. The entries from the 2nd through the 7th row show the results for the case we do not use the scaling factors or $s_{c,i} = 1$ for $i = 1, 2, ..., 150$, and the entries of the remaining rows show the results for the case we use the scaling factors set equal to $1/C_i^2$ for $i = 1, 2, ..., 150$. Overall, by using scaling factors, we are able to obtain higher NPVs with fewer simulation runs. For the case without the scaling factors, the code fails to converge when the initial penalty parameter is less than 10^{-4} due to some overflow problems in the adjoint gradient calculation. Both the 2nd row and the 8th row show the optimization procedure does not converge. As we set the maximum allowable number of outer-loop iterations to 10, the optimization iterations of the cases corresponding to the 2nd row and the 8th row are stopped. The constraints are not satisfied after 10 outer-loop iterations. Due to the updates (reduction) of penalty parameter (only the penalty parameter is updated at each outer-loop iteration for the cases corresponding to the 2nd row and the 8th row in Table 4.2), the augmented Lagrangian function has big weights on the terms, ψ's, and then the augmented Lagrangian function reduces dramatically. If the violation of constraint is significantly large, the penalty parameter is always decreased and the augmented Lagrangian method is actually the pure penalty method, which is very likely to have the ill conditioning and numerical problems for small values of the penalty parameter. As shown in Fig. 4.3(a), the augmented Lagrangian function starts oscillating after 110 simulation runs because the constraints are not satisfied in the inner-loop iterations.

Table 4.2: The performance of different parameters. N_μ: Number of μ updated; N_λ: Number of λ updated.

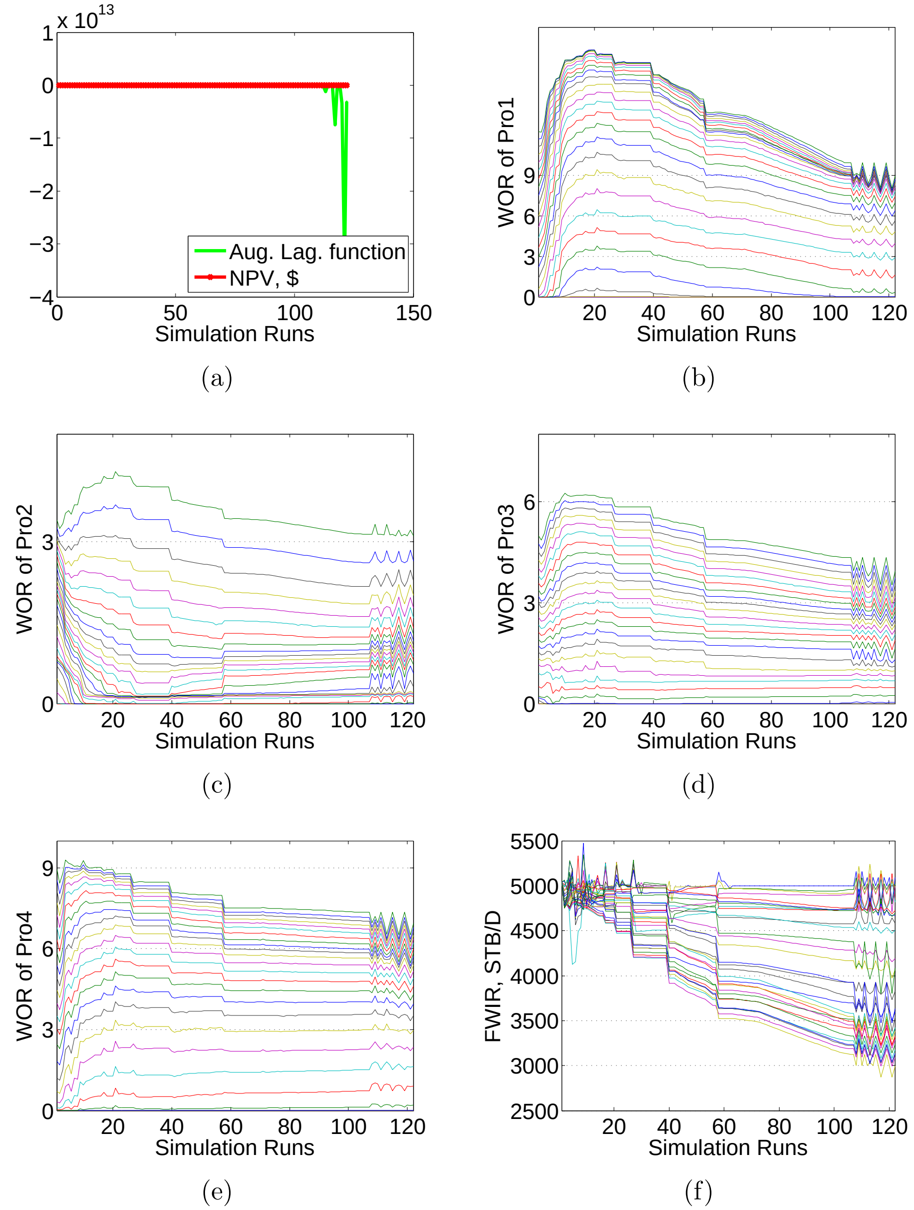
$s_{c,i}$	μ^0	$\lambda_{c,i}^0$	Simulations	Final NPV, $	Outer-Loop	N_μ	N_λ	Converge
1	10^{-1}	0	123	1.94×10^8	10	9	0	No
1	10^{-1}	$\max[0, \frac{s_{c,i}c_i^0}{\mu^0}]$	85	1.89×10^8	9	8	0	Yes
1	10^{-3}	0	44	1.87×10^8	5	4	0	Yes
1	10^{-3}	$\max[0, \frac{s_{c,i}c_i^0}{\mu^0}]$	31	1.85×10^8	5	4	0	Yes
1	10^{-4}	0	28	1.85×10^8	4	3	0	Yes
1	10^{-4}	$\max[0, \frac{s_{c,i}c_i^0}{\mu^0}]$	29	1.83×10^8	4	3	0	Yes
$1/C_i^2$	10^{-6}	0	74	1.93×10^8	10	9	0	No
$1/C_i^2$	10^{-6}	$\max[0, \frac{s_{c,i}c_i^0}{\mu^0}]$	74	1.96×10^8	8	7	0	Yes
$1/C_i^2$	10^{-7}	0	26	1.98×10^8	5	3	1	Yes
$1/C_i^2$	10^{-7}	$\max[0, \frac{s_{c,i}c_i^0}{\mu^0}]$	41	1.91×10^8	5	3	1	Yes
$1/C_i^2$	10^{-8}	0	26	1.97×10^8	3	1	1	Yes
$1/C_i^2$	10^{-8}	$\max[0, \frac{s_{c,i}c_i^0}{\mu^0}]$	17	1.81×10^8	1	0	0	Yes

Another comparison is done for initial Lagrange multipliers. With an initial penalty parameter $\mu^0 = 10^{-1}$ and the initial Lagrange multipliers $\lambda_{c,i}^0 = 0$, the optimization requires a total of 123 simulation runs to obtain the optimal NPV of \$$1.94\times10^8$. A careful check shows that the augmented Lagrangian function diverges during late iteration and the FWIR does not satisfy the constraint value of 5000 STB/D. Also, we found the penalty parameter is updated at every outer-loop iteration, which means the optimization with $\mu^0 = 10^{-1}$ and $\lambda_{c,i}^0 = 0$ is purely the penalty method which can sometimes lead to divergence problem. Another trial using the same μ^0 and $\lambda_{c,i}^0 = \max[0, \frac{s_{c,i}c_i^0}{\mu^0}]$ obtains a lower NPV of \$$1.89\times10^8$ with 85 simulation runs. For the cases (the 8th and 9th rows) that we use $s_{c,i} = 1/C_i^2$ and $\mu^0 = 10^{-6}$, using the initial Lagrange multipliers estimated from the violation at the first iteration guarantee the optimization to converge and even achieve 1.6% higher NPV than using the initial Lagrange multiplier set equal to 0. However, in other cases using $s_{c,i} = 1/C_i^2$ and $\mu^0 = 10^{-7}$ (or $s_{c,i} = 1/C_i^2$ and $\mu^0 = 10^{-8}$), using the initial Lagrange multiplier set equal to 0 obtains a higher NPV with fewer simulation runs than using the Lagrange multipliers estimated from Eq. 2.22. The comparison indicates that estimating initial Lagrangian multipliers using $\lambda_{c,i}^0 = \max[0, \frac{s_{c,i} c_i^0}{\mu^0}]$ does not offer clear advantages.

4.1.1 Optimization results without scaling factors

In this subsection, we provide the optimization results obtained with the scaling factors set equal to 1, i.e. we do not consider rescaling the constraints in the optimization problem.

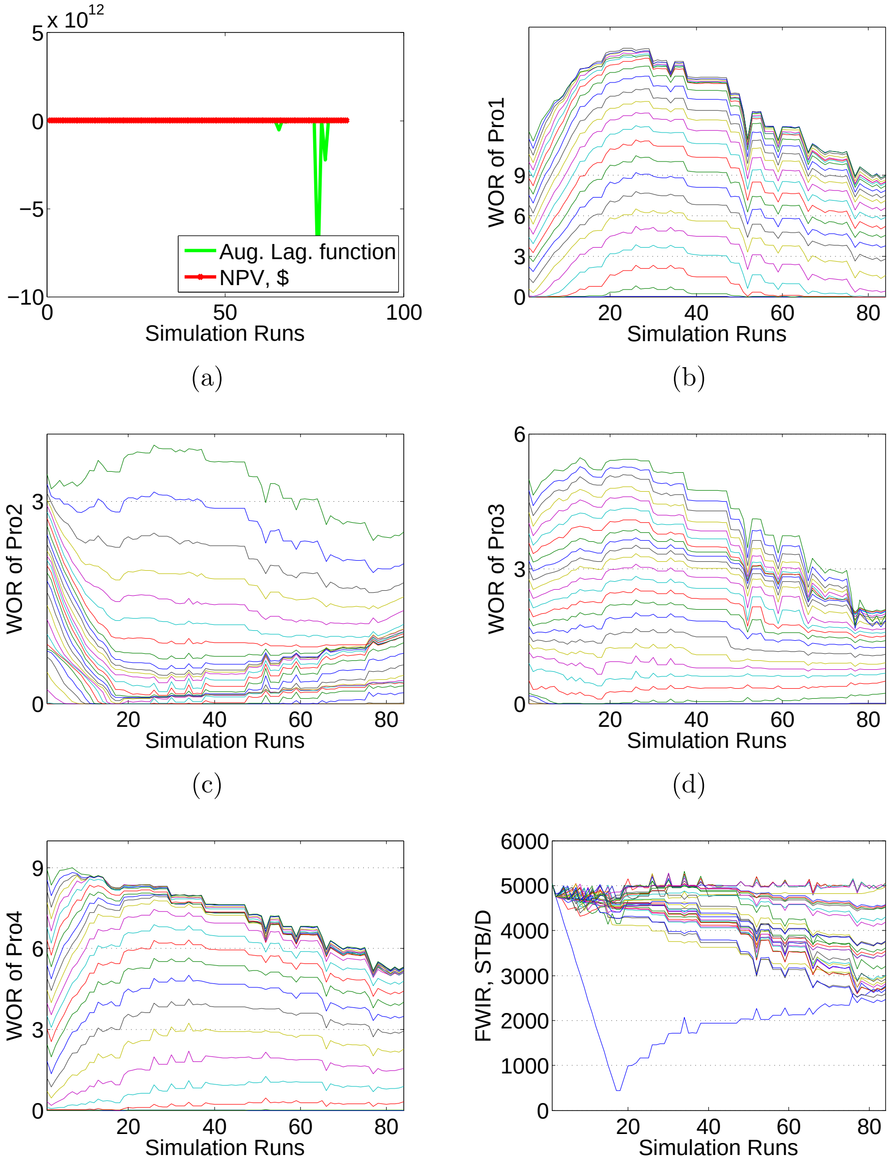
Fig. 4.3 shows the optimization results obtained with $\mu^0 = 10^{-1}$ and $\lambda_{c,i}^0 = 0$. Fig. 4.3(a) shows how the augmented Lagrangian function and NPV change as a function of the number of simulation runs during optimization. The scale of the augmented Lagrangian function is much larger than NPV during late iterations, so the NPV curve appears as a line in Fig. 4.3(a). From Fig. 4.3(a), we can see that the augmented Lagrangian function diverges during late iterations. Fig. 4.3(b) through Fig. 4.3(e) show how WOR of each producer at each control step changes as a function of the number of simulation runs. Each curve corresponds to the WOR of the specified producer at a specified control step versus simulation runs. Note that the WOR curves during late iterations are oscillating. Fig. 4.3(f) shows how FWIR at each control step changes as a function of the number of simulation runs and each curve corresponds to the FWIR at one specified control step. The highest FWIR at the last simulation run is 5092 STB/D, which does not satisfy the constraint value 5000 STB/D. The algorithm keeps reducing the penalty parameter during the outer-loop iterations, which results in decreasing the augmented Lagrangian function when the constraints are not satisfied; see Fig. 4.3(a).



(a) (b) (c) (d) (e) (f)

Figure 4.3: The optimization results with $\mu^0 = 10^{-1}$ and $\lambda^0_{c,i} = 0$.

Fig. 4.4 shows the optimization results using the initial penalty parameter $\mu^0 = 10^{-1}$ and estimate initial Lagrange multiplier using Eq. 2.22, i.e. $\lambda^0_{c,i} = \max[0, \frac{s_{c,i}c^0_i}{\mu^0}]$. Fig. 4.4(a) shows how the augmented Lagrangian function and NPV change as a function of the number of simulation runs during optimization. The augmented Lagrangian function converges within 85 simulation runs. Fig. 4.4(b) through Fig. 4.4(e) show how WOR of each producer at each control step changes as a function of the number of simulation runs. Note that the WOR constraints at many control steps for Pro1 are not satisfied before the 78th simulation runs, due to the weight on the penalty term of WOR constraints during the early iterations is too small. Fig. 4.4(f) shows how FWIR at each control step changes as a function of the number of simulation runs. The highest FWIR at the last simulation run is 5044 STB/D, which is within the tolerance (1% of the constraint value 5000 STB/D).



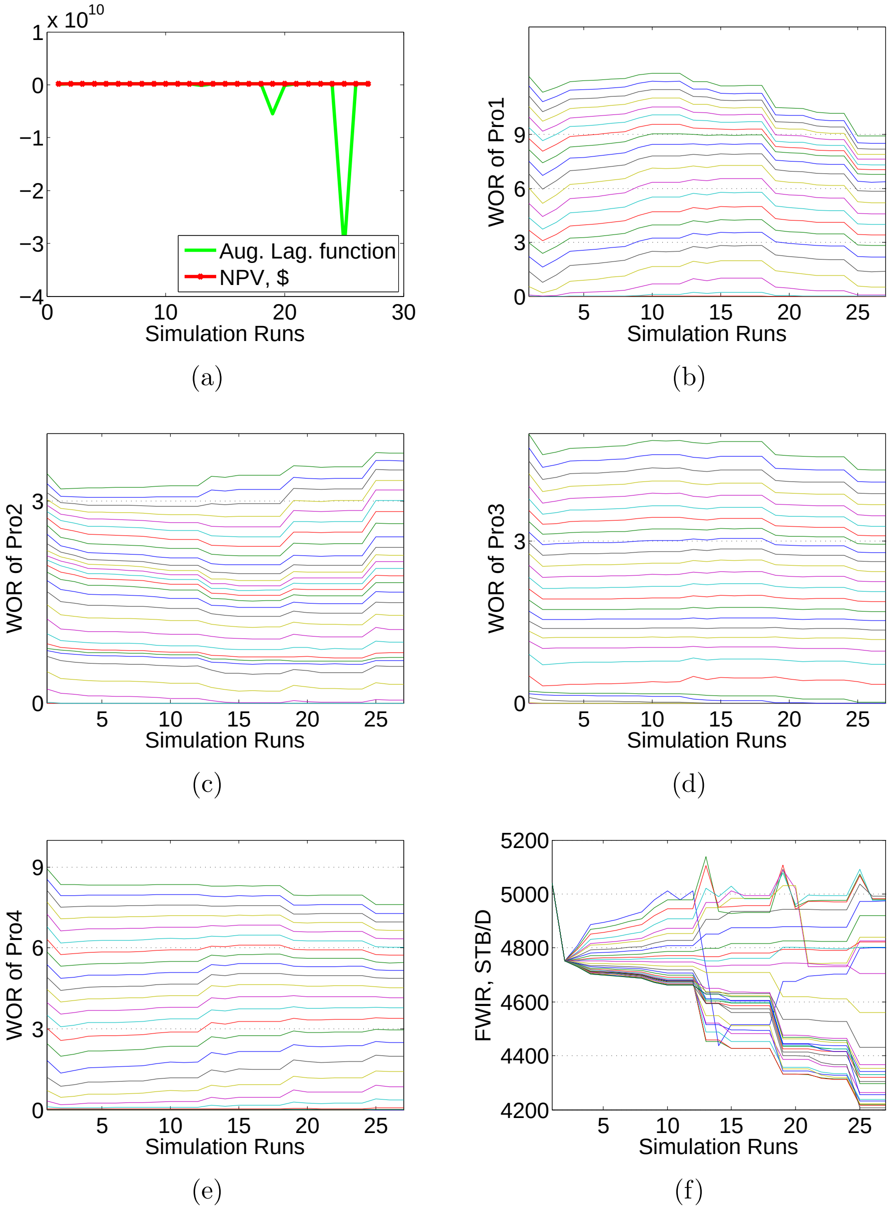
(a) (b)

(c) (d)

(e) (f)

Figure 4.4: The optimization results with $\mu^0 = 10^{-1}$ and $\lambda^0_{c,i} = \max[0, \frac{s_{c,i}c^0_i}{\mu^0}]$.

We also provide the optimization results with $\mu^0 = 10^{-4}$ and $\lambda_{c,i}^0 = 0$ in Fig. 4.5 and results with $\mu^0 = 10^{-4}$ and $\lambda_{c,i}^0 = \max[0, \frac{s_{c,i}c_i^0}{\mu^0}]$ in Fig. 4.6. In these two cases, it can be seen that the augmented Lagrangian function takes less than 30 simulation runs to converge. The augmented Lagrangian function decreases sometimes along the simulation number as we update the Lagrangian multipliers and the penalty parameters. At the end of optimization, all the constraints are satisfied within certain tolerance.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 4.5: The optimization results with $\mu^0 = 10^{-4}$ and λ's equal to 0.

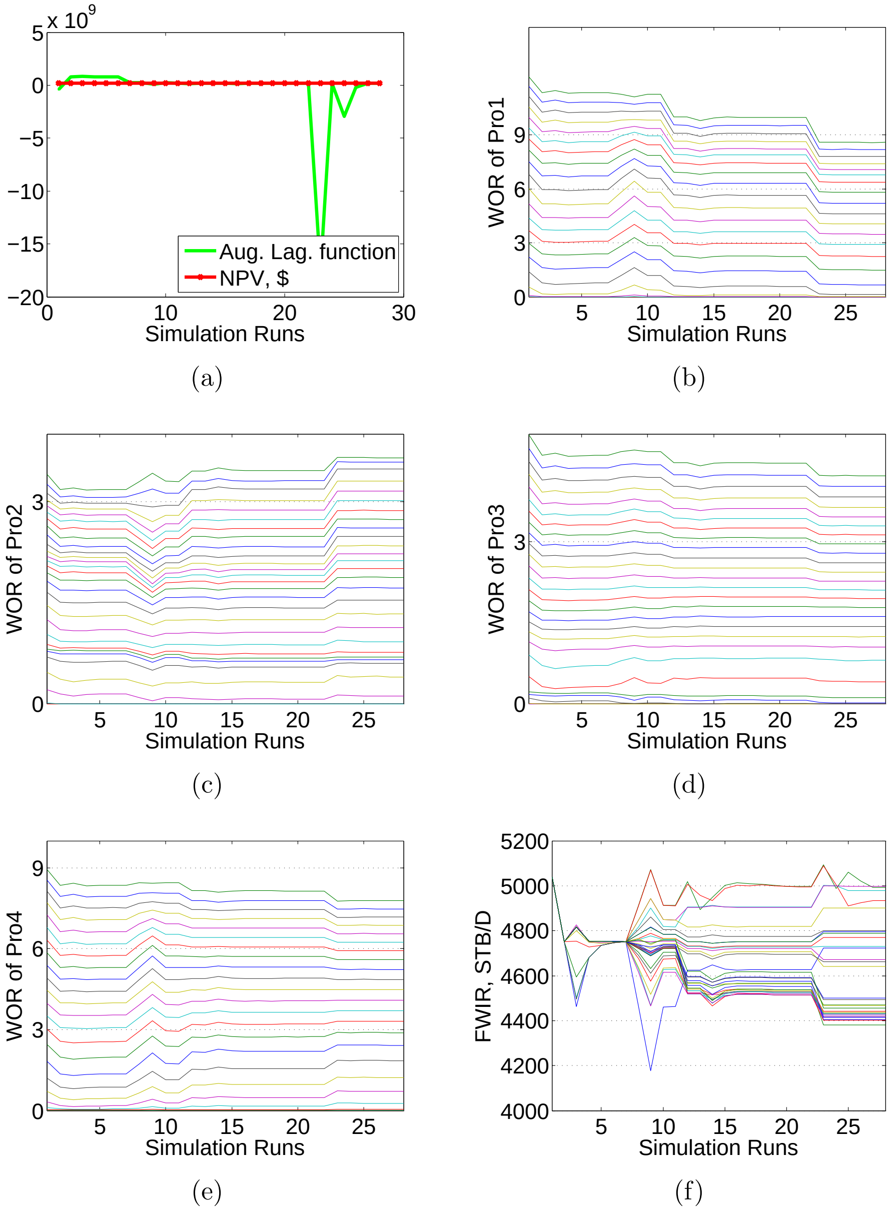
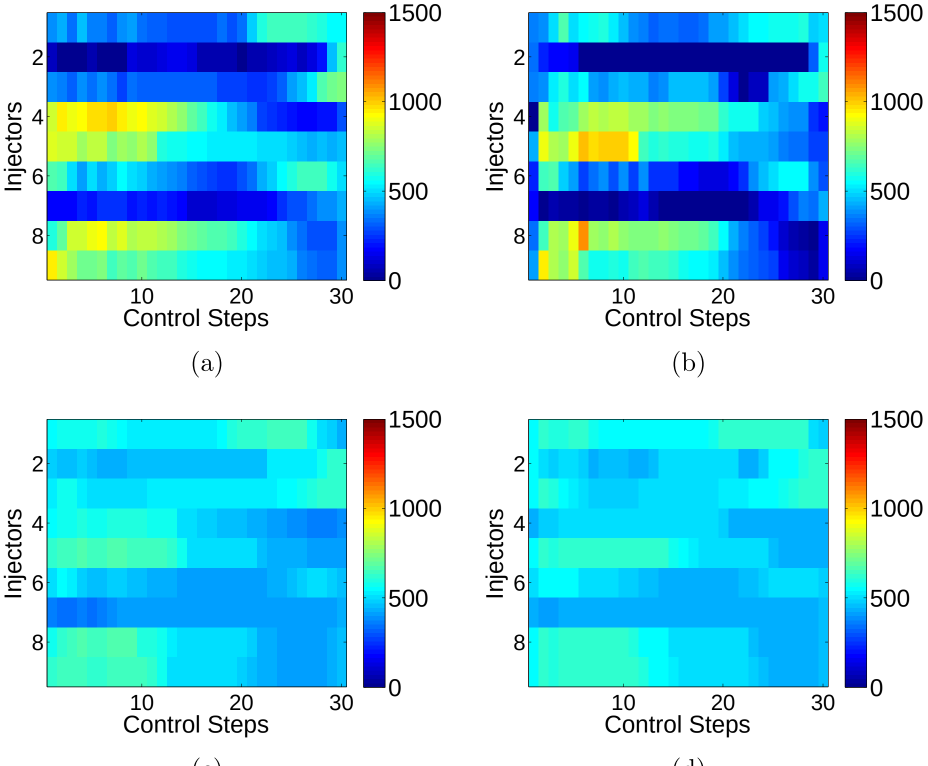


Figure 4.6: The optimization results with $\mu^0 = 10^{-4}$ and $\lambda^0_{c,i} = \max[0, \frac{s_{c,i}c_i^0}{\mu^0}]$.

Fig. 4.7 shows the estimated optimal injection well controls for different initial penalty parameter and Lagrange multipliers. The x-axis corresponds to the control steps and the y-axis corresponds to index of the injectors. It can be seen that the injection rates in Fig. 4.7(a) and Fig. 4.7(b) are similar, and the same observation in Fig. 4.7(c) and Fig. 4.7(d). It seems that the initial value of the Lagrangian multipliers does not affect the optimal control that much compared to the initial value of penalty parameter. In the case we use $\mu^0 = 10^{-1}$, INJ2 and INJ7 use low injection rate for most of the control steps; however, in the case we use $\mu^0 = 10^{-4}$, the optimal injection rates for each injector are fairly close to each other.



(a)

(b)

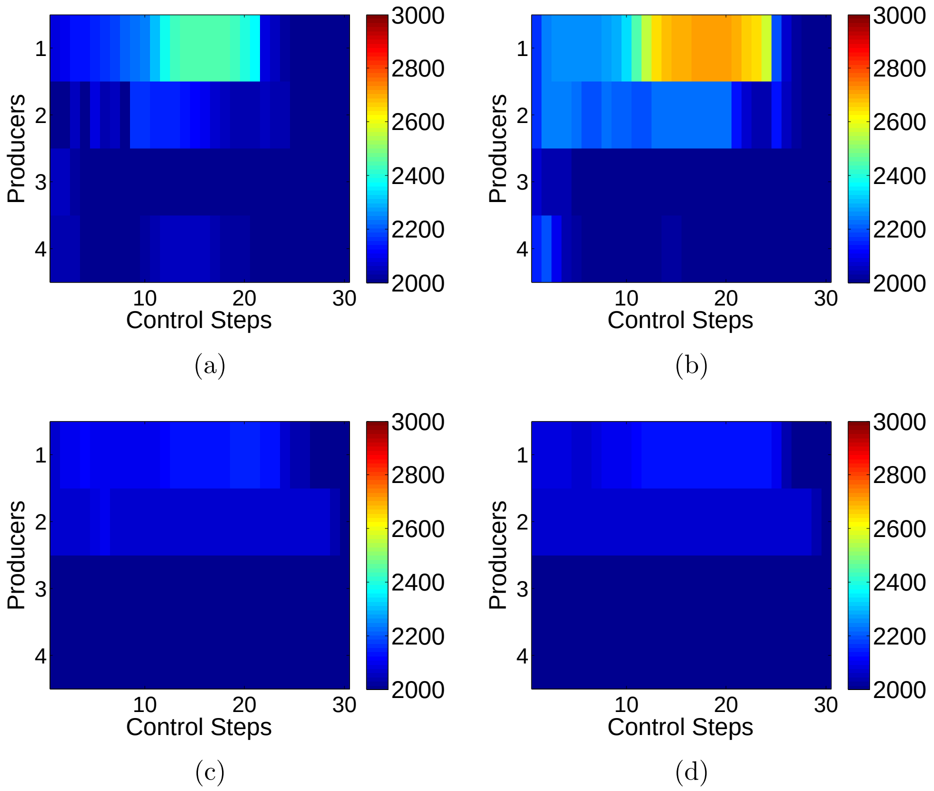
(c)

(d)

Figure 4.7: The estimated optimal injection well controls. (a) $\mu^0 = 10^{-1}$, $\lambda^0 = 0$; (b) $\mu^0 = 10^{-1}$, $\lambda^0_{c,i} = \max[0, \frac{s_{c,i}c_i^0}{\mu^0}]$; (c) $\mu^0 = 10^{-4}$, $\lambda^0 = 0$; (d) $\mu^0 = 10^{-4}$, $\lambda^0_{c,i} = \max[0, \frac{s_{c,i}c_i^0}{\mu^0}]$.

Fig. 4.8 shows the estimated optimal BHP well controls for different initial penalty parameter and Lagrange multipliers. The x-axis corresponds to the control steps and the y-axis corresponds to index of the producers. In the case we use $\mu^0 = 10^{-1}$, Pro1 uses high BHP control between the 13th and the 22nd control steps; however, in the case we use $\mu^0 = 10^{-4}$, all the producers use similar BHP controls, which are close to the lower bound of BHP control.

Fig. 4.9 shows the remaining oil saturation distribution after optimization for different initial penalty parameter and Lagrange multipliers. The red color represents



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

Figure 4.8: The estimated optimal optimal BHP well controls. (a) $\mu^0 = 10^{-1}$, $\lambda^0 = 0$; (b) $\mu^0 = 10^{-1}$, $\lambda^0_{c,i} = \max[0, \frac{s_{c,i}c_i^0}{\mu^0}]$; (c) $\mu^0 = 10^{-4}$, $\lambda^0 = 0$; (d) $\mu^0 = 10^{-4}$, $\lambda^0_{c,i} = \max[0, \frac{s_{c,i}c_i^0}{\mu^0}]$.

high oil saturation while the blue color represents high water saturation. Overall, they are fairly similar to each other. However, it can be seen that the remaining oil in the case with $\mu^0 = 10^{-1}$ (Fig. 4.9(a)) is comparatively less than other cases, which is consistent with the NPV shown in Table 4.2.

Fig. 4.10 shows the comparison of NPV for the cases with different μ^0 and λ^0. The blue curve corresponds to the NPV obtained with $\mu^0 = 10^{-1}$ and $\lambda^0 = 0$. As this choice of μ^0 and λ^0 makes the terms, ψ's in Eq. 4.1, comparatively negligible during early iterations, the optimization honors the NPV part during these iterations and hence the NPV increases fairly fast to \$2.0×10^8. The optimization starts to honor the terms, ψ's, after updating the Lagrangian multipliers or penalty parameter. The NPV decreases gradually but the oscillation happens to NPV during the late iteration. As

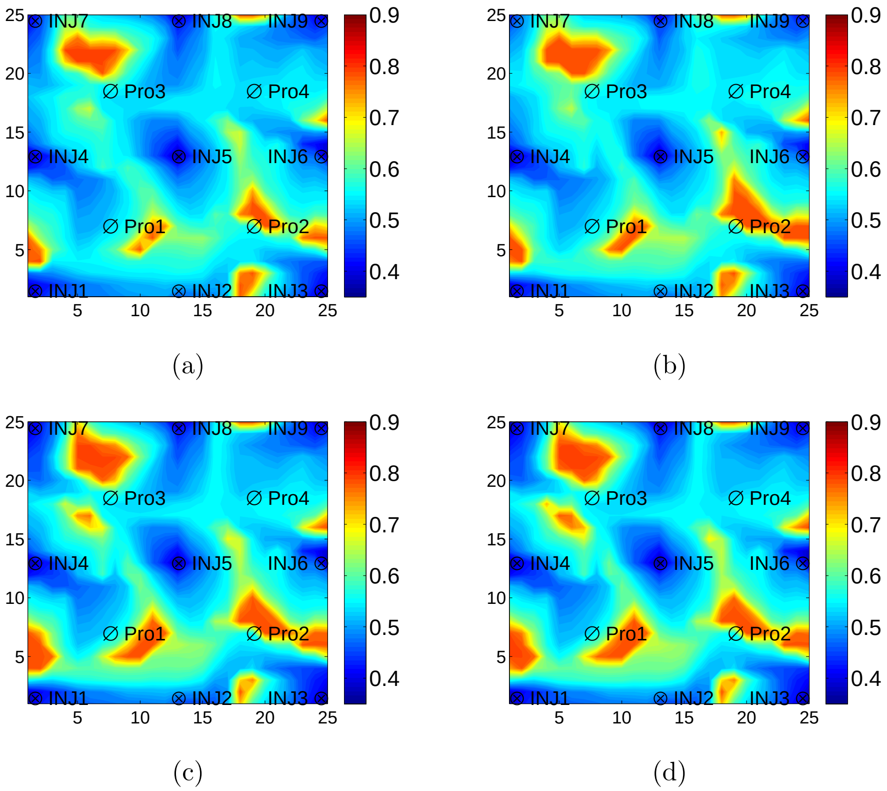


Figure 4.9: Remaining oil saturation distribution after optimization. (a) $\mu^0 = 10^{-1}$, $\lambda^0 = 0$; (b) $\mu^0 = 10^{-1}$, $\lambda^0_{c,i} = \max[0, \frac{s_{c,i}c_i^0}{\mu^0}]$; (c) $\mu^0 = 10^{-4}$, $\lambda^0 = 0$; (d) $\mu^0 = 10^{-4}$, $\lambda^0_{c,i} = \max[0, \frac{s_{c,i}c_i^0}{\mu^0}]$.

explained above, the augmented Lagrangian method this case ($\mu^0 = 10^{-1}$ and $\lambda^0 = 0$) is the pure penalty method and numerical difficulty (e.g. divergence) happens when the penalty parameter μ is too small. The black curve corresponds to the NPV obtained with $\mu^0 = 10^{-1}$ and $\lambda^0_{c,i} = \max[0, \frac{s_{c,i}c_i^0}{\mu^0}]$. As λ^0 is not zero in the first outer-loop, the optimization has a larger weight on the terms, ψ's, and NPV does not increase as fast as in the case of $\mu^0 = 10^{-1}$ and $\lambda^0 = 0$. The NPV in the case ($\mu^0 = 10^{-1}$ and $\lambda^0_{c,i} = \max[0, \frac{s_{c,i}c_i^0}{\mu^0}]$) finally converges to \$1.89×10^8, which shows the augmented Lagrangian method is more stable than the pure penalty method. Although the NPV in the case ($\mu^0 = 10^{-1}$ and $\lambda^0_{c,i} = \max[0, \frac{s_{c,i}c_i^0}{\mu^0}]$) seems oscillating at end, a close check shows the algorithm does not accept the control variables at last two iterations so the augmented Lagrangian function is flat at end. If we use a too small μ^0, the NPVs (green and red curves) have limited increase from the initial points due to the fact that the terms, ψ's, in the augmented Lagrangian function are heavily weighted. This phenomenon indicates it is better to start with a large initial penalty parameter.

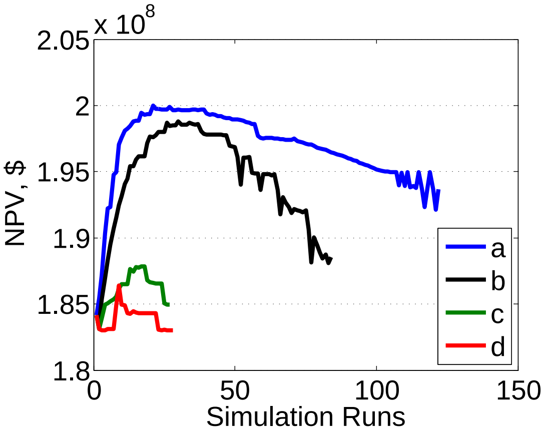


Figure 4.10: NPV comparison for different μ^0 and λ^0. (a) $\mu^0 = 10^{-1}$, $\lambda^0 = 0$; (b) $\mu^0 = 10^{-1}$, $\lambda^0_{c,i} = \max[0, \frac{s_{c,i}c_i^0}{\mu^0}]$; (c) $\mu^0 = 10^{-4}$, $\lambda^0 = 0$; (d) $\mu^0 = 10^{-4}$, $\lambda^0_{c,i} = \max[0, \frac{s_{c,i}c_i^0}{\mu^0}]$.

4.1.2 Optimization results with scaling factors

In this subsection, we provide the optimization results obtained with the scaling factors set equal to $1/C_i^2$, $i = 1, 2, ..., 150$. Fig. 4.11 shows the optimization results obtained with $\mu^0 = 10^{-7}$ and $\lambda^0_{c,i} = 0$. Fig. 4.11(a) shows how the augmented Lagrangian function and NPV change as a function of the number of simulation runs during optimization. The augmented Lagrangian function converges within 26 simulation runs and the final NPV is $\$1.98\times10^8$ which is the highest value in all the optimization cases for this example; see Table 4.2. Note that there is a significant decrease of the augmented Lagrangian function at the 24th simulation run, which is the result of reducing the penalty parameter. Fig. 4.11(b) through Fig. 4.11(e) show how WOR of each producer at each control step changes as a function of the number of simulation runs. Each curve corresponds to the WOR of the specified producer at a specified control step versus simulation runs. Compared to the WOR constraints in Fig. 4.5(b) through Fig. 4.5(e) , the WOR constraints when using the scaling factor during the early iterations are satisfied very well, which means the penalty terms of WOR constraints are allocated a reasonable weight during these iterations. Fig. 4.11(f) shows how FWIR at each control step changes as a function of the number of simulation runs and each curve corresponds to one FWIR at one specified control step. The FWIR constraints are satisfied in the end of optimization.

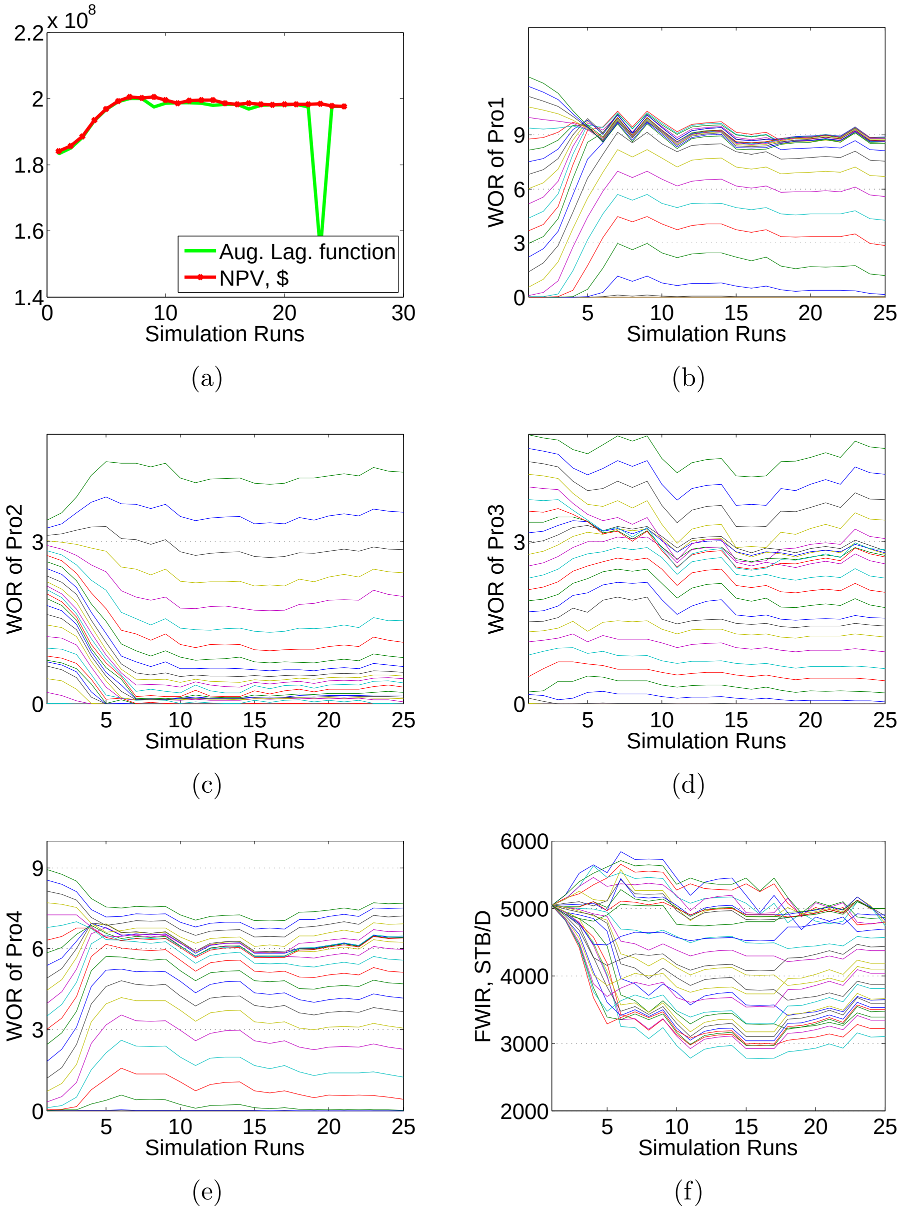
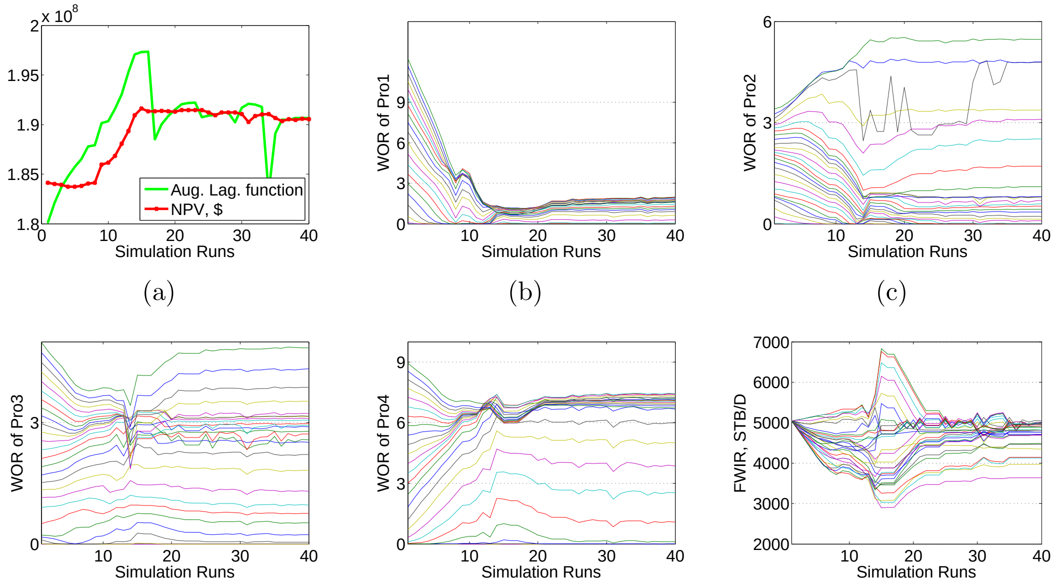


Figure 4.11: The optimization results with $\mu^0 = 10^{-7}$ and $\lambda_{c,i}^0 = 0$.

Fig. 4.12 shows the optimization results when using $\lambda_{c,i}^0 = \max[0, \frac{s_{c,i}c_i^0}{\mu^0}]$ with the same initial penalty parameter $\mu^0 = 10^{-7}$. Fig. 4.12(a) shows how the augmented Lagrangian function and NPV change as a function of the number of simulation runs during optimization. As shown in Eq. 2.11, the augmented Lagrangian function is equal to NPV minus the ψ terms which may take on either positive or negative values. Therefore, the augmented Lagrangian function (green curve) is sometimes below the NPV (red curve) or above the NPV. The augmented Lagrangian function converges within 41 simulation runs. Fig. 4.12(b) through Fig. 4.12(e) show how WOR of each producer at each control step changes as a function of the number of simulation runs. Fig. 4.12(f) shows the FWIR of each control step changes as a function of the number of simulation runs.

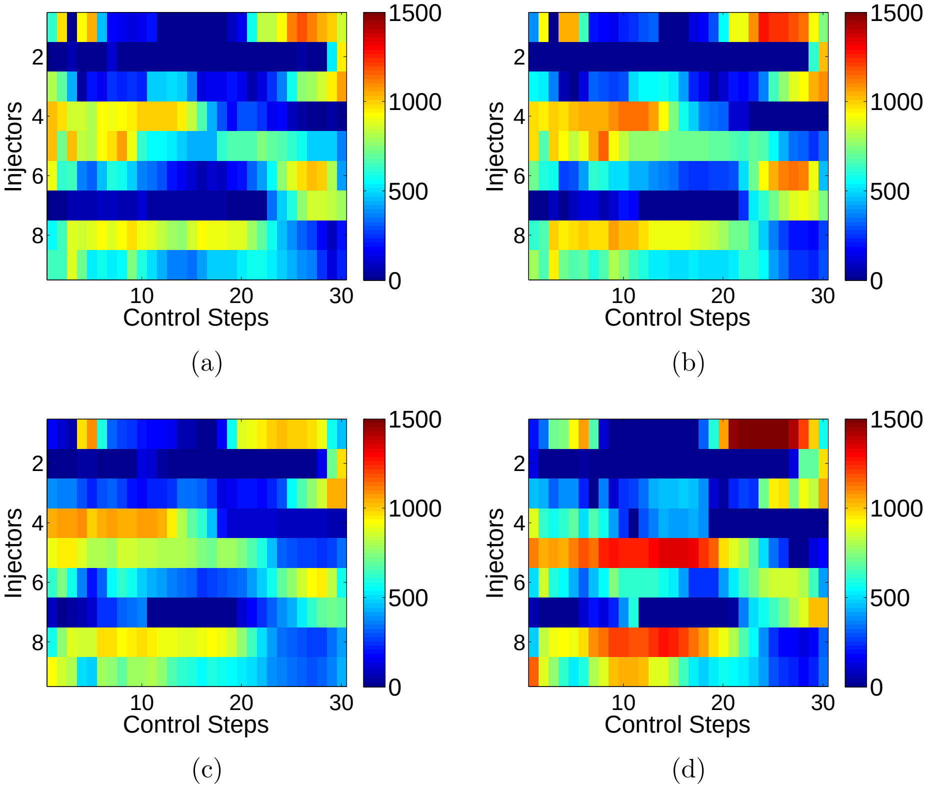


(a) (b) (c)

(d) (e) (f)

Figure 4.12: The optimization results with $\mu^0 = 10^{-7}$ and $\lambda_{c,i}^0 = \max[0, \frac{s_{c,i}c_i^0}{\mu^0}]$.

Fig. 4.13 shows the estimated optimal injection well controls for different initial penalty parameter and Lagrange multipliers. These four subfigures show qualitatively similar injection rate controls. Compared to Fig. 4.7, INJ1, INJ2 and INJ3 use lower injection rates at most of the control steps due to that fact that these three injectors are connected to Pro2 by a high permeability channel. Using high injection rates would result in early water breakthrough in Pro2. Fig. 4.14 shows the estimated optimal BHP well controls for different initial penalty parameter and Lagrange multipliers. In the case we use $\mu^0 = 10^{-6}$, Pro1 uses high BHP control between the 10th and the 20th control steps; however, in the case we use $\mu^0 = 10^{-7}$, Pro1 uses lower BHP controls within the corresponding control steps than for the case with $\mu^0 = 10^{-6}$.



(a)

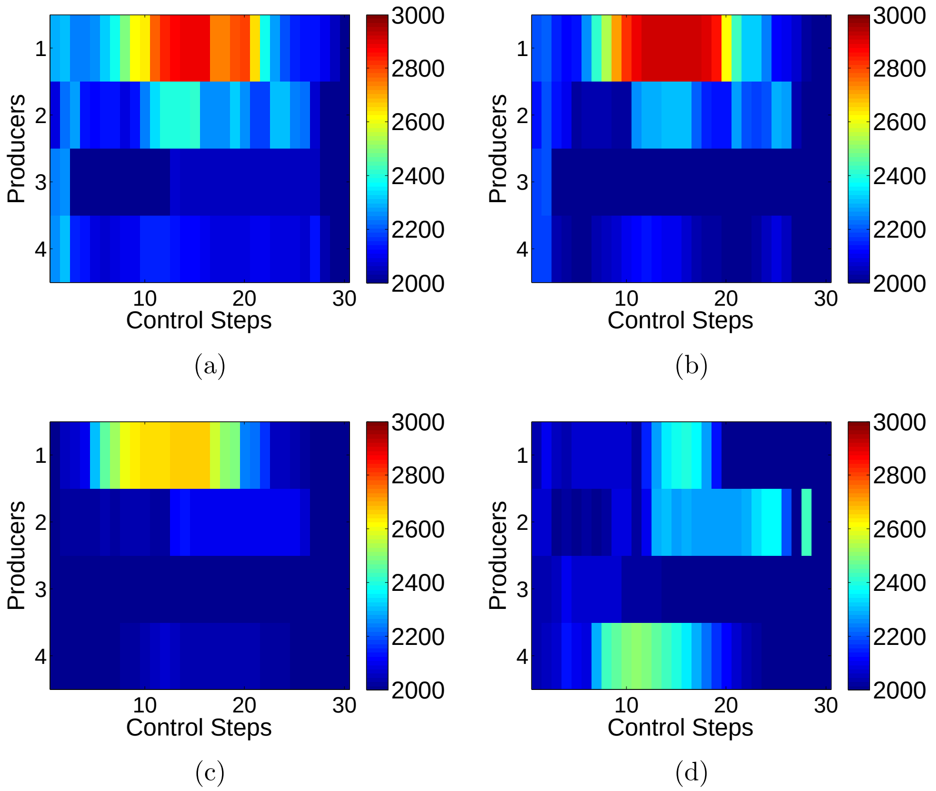
(b)

(c)

(d)

Figure 4.13: The estimated optimal injection well controls. (a) $\mu^0 = 10^{-6}$, $\lambda^0 = 0$; (b) $\mu^0 = 10^{-6}$, $\lambda^0_{c,i} = \max[0, \frac{s_{c,i}c^0_i}{\mu^0}]$; (c) $\mu^0 = 10^{-7}$, $\lambda^0 = 0$; (d) $\mu^0 = 10^{-7}$, $\lambda^0_{c,i} = \max[0, \frac{s_{c,i}c^0_i}{\mu^0}]$.

Fig. 4.15 shows the remaining oil saturation distribution after optimization for



(a)

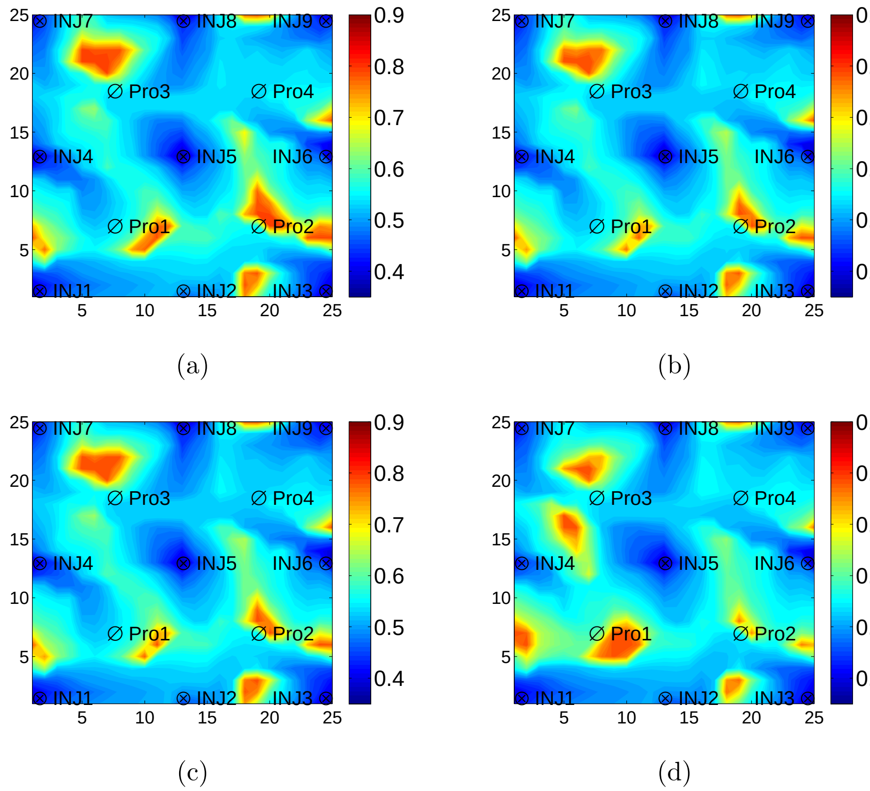
(b)

(c)

(d)

Figure 4.14: The estimated optimal optimal BHP well controls. (a) $\mu^0 = 10^{-6}$, $\lambda^0 = 0$; (b) $\mu^0 = 10^{-6}$, $\lambda^0_{c,i} = \max[0, \frac{s_{c,i}c^0_i}{\mu^0}]$; (c) $\mu^0 = 10^{-7}$, $\lambda^0 = 0$; (d) $\mu^0 = 10^{-7}$, $\lambda^0_{c,i} = \max[0, \frac{s_{c,i}c^0_i}{\mu^0}]$.

different initial penalty parameter and Lagrange multiplier values. Overall, they are fairly similar to each other. However, it can be seen that the remaining oil saturation in the case with $\mu^0 = 10^{-7}$ and $\lambda^0_{c,i} = \max[0, \frac{s_{c,i}c^0_i}{\mu^0}]$ (Fig. 4.15(d)) is comparatively higher than the other three cases, which corresponds to a relatively low NPV of \$1.91×10^8 (see Table 4.2).



(a)

(b)

(c)

(d)

Figure 4.15: Remaining oil saturation distribution after optimization. (a) $\mu^0 = 10^{-6}$, $\lambda^0 = 0$; (b) $\mu^0 = 10^{-6}$, $\lambda^0_{c,i} = \max[0, \frac{s_{c,i}c_i^0}{\mu^0}]$; (c) $\mu^0 = 10^{-7}$, $\lambda^0 = 0$; (d) $\mu^0 = 10^{-7}$, $\lambda^0_{c,i} = \max[0, \frac{s_{c,i}c_i^0}{\mu^0}]$.

4.2 Example 2: Waterflooding with Two Horizontal Wells

This synthetic water flooding example is similar to that used in [6, 40]. The reservoir model is based on a $45 \times 45 \times 1$ grid and the gridblock dimension is equal to 32.8 ft in all directions. The model consists of a horizontal smart water injector and a horizontal smart producer. The horizontal injector and producer are, respectively, along the west and the east side of the reservoir. Similar to Sarma et al. [40], we use 45 vertical pseudo-wells to simulate a smart horizontal well with 45 segments, as our in-house simulator is not able to set up smart horizontal wells. There are two phases, water and oil with a unit mobility ratio. The porosity is homogenous throughout the reservoir and the permeability field is shown in Fig. 4.16 with two high permeability channels indicated by the dark red color. The blue regions represent low permeability. The initial reservoir pressure is 5800 psi and the connate water saturation and residual oil saturation are both 0.1. Other parameters are the same as in [6, 40]: the oil price is at \$12.72/BBL, water injection costs are \$0, water production costs are \$3.18/BBL, and the annual discount rate is 0. In order to compare the result with that in [40], we set each injector segment under rate control with $0 \leq q_{winj} \leq 300$ STB/D and set each producer segment under BHP control with $3500 \leq p_{wf} \leq 6000$ psi. The optimization essentially results in a redistribution of this water among the injector segments. The anticipated total project life is 950 days and the control time step size is set to 190 days so we have 5 control steps. The total number of control variables is $(45 + 45) \times 5 = 450$. In the following subsections, we will discuss: 1) the optimization results for the situation that there is an additional equality constraint for the total injection rate, i.e., FWIR=2700 STB/D; 2) the optimization results for the situation that we have an additional inequality constraint, FWIR≤2700 STB/D; 3)the optimization results with only bound constraints and compare to the results obtained with an ensemble-based gradient-free algorithm, EnOpt.

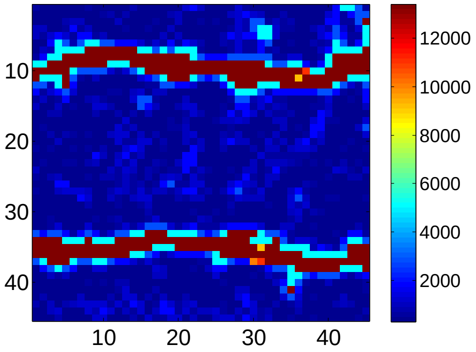


Figure 4.16: Log-permeability distribution.

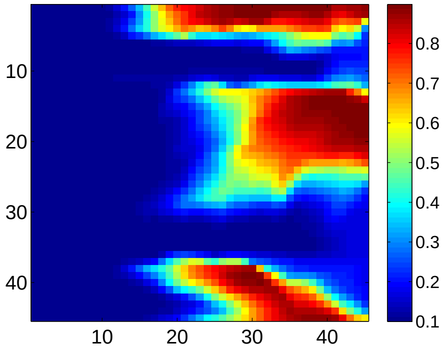


Figure 4.17: Remaining oil saturation distribution with reactive control.

4.2.1 Optimization with bound and equality constraints

In this subsection, we impose an equality constraint on the field water injection rate, FWIR=2700 STB/D, for the maximization problem. With 5 control steps, the total number of constraints is 5. We have no inequality constraints so the augmented Lagrangian function can be rewritten as

$$\beta_L[u, \lambda_e, \mu] = J_L - \sum_{i=1}^{5} [\lambda_{e,i} e_i(y, u) + \frac{s_{e,i}}{2\mu} (e_i(y, u))^2], \tag{4.4}$$

where e_i is defined by

$$e_i = \sum_{j=1}^{45} q^i_{inj,j} - 2700 = 0, \quad i = 1, ..., 5, \tag{4.5}$$

for the jth smart injection segment at the ith control step. In this case, although the total injection rate constraint is the only equality constraint for each control step and there is no scaling problem, we still use the scaling factor, $s_{e,i} = 1/2700^2$.

To show how production optimization improves the recovery, we first set up a reference case, which is run with reactive controls. A production segment is shut in when the water oil ratio reaches its economic limit, i.e., when $r_w q^n_{w,j} \geq r_o q^n_{o,j}$. For this example, the economic limit is reached when WOR $\geq$ 4.0. The injection rate for the jth segment with the reactive controls is set as

$$q_{inj,j} = \frac{1}{k_j} \frac{2700}{\sum_{l=1}^{45} 1/k_l}, \tag{4.6}$$

where k_j is the permeability value of the well gridblock at the jth injection well segment. Therefore, the injection rates for the well segments are distributed inversely proportional to the well segment gridblock permeability. The bottomhole pressure (BHP) of production segments is set to 5500 psi for each control step. Typically, reactive control requires to produce oil under the lowest allowable BHP; however, due to the fact that setting producer BHP to 3500 psi leads to divergence in simulation run, we set producer BHP to 5500 psi. Fig. 4.18 shows a comparison of average reservoir pressure for the reactive control case and the optimization case. The blue curve corresponds to the reactive case. The reservoir pressure keeps almost constant from day 1 to day 570 and then builds up from day 571 to day 950. As water injection rate is fixed during production life and some producer segments are shut-in after day 571, the reservoir pressure increases gradually after day 571. The red curve corresponds to the optimization case. The reservoir pressure keeps constant for each control step, which means this well control scenario basically balances the total injection rate and total production rate during each control step (i.e., it reaches steady-state). Fig. 4.17 shows the final oil saturation distribution with reactive control, in which the blue color represents high water saturation and red color represents high oil saturation. It can be seen that the area close to injection well and the area corresponding to the high permeability channels are well swept by the injected water. However, water breaks through early in the production segments located in the two high permeability channels and the non-channel area near the producing well is hardly touched by the injected water, which results in a low areal sweep and poor oil recovery.

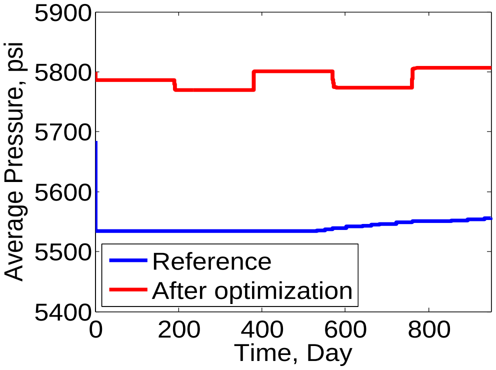


Figure 4.18: Average reservoir pressure for optimized case.

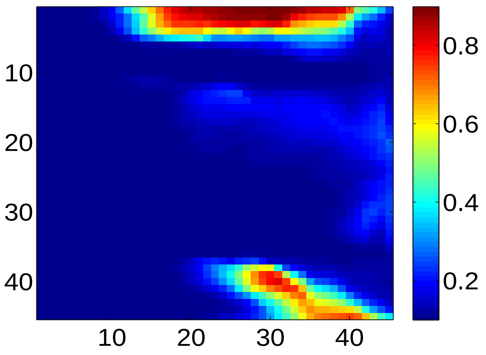


Figure 4.19: Final oil saturation for optimized case.

The final oil saturation distribution obtained with the final optimal controls using the augmented Lagrangian method is shown in Fig. 4.19. Our initial guess for the controls is that each injection segment is allocated 60 STB/D of water with a total injection rate of 2700 STB/D, and the producer has a constant BHP of 5780 psi for the expected reservoir life. Compared to the reference case, the final optimal controls yield a much better sweep efficiency and oil recovery. The low permeability region between the channels is much better swept. The average reservoir pressure for the case with the final optimal controls is shown in Fig. 4.18 as the red solid line. It shows that average reservoir pressure during the expected reservoir life basically remains constant, which indicates that voidage ratio is almost one and the total injection balances the total production for the reservoir life.

Fig. 4.20 shows the final controls for the injectors. The y-axis of this plot corresponds to the 45 injector segments and x-axis corresponds to the 5 control steps. The white region represents zero injection rate, which occurs around the north high permeability channel (segment 10) for the first 4 control steps. The injection rate for the well segments around the lower high permeability channel (segment 36) is low only for the first control step. The injection rates for these segments remain high for control steps 2 to 4. The injection rates among all the 45 segments are almost identical for the fifth control step. Fig. 4.21 shows the final BHP controls for the producer segments. As in the Fig. 4.20, y-axis of the plot corresponds to the 45 producer segments and x-axis corresponds to the 5 control steps. White color means a high BHP value about 5900 psia, which is higher than the average reservoir pressure (5800 psia). As the well gridblock pressure is almost at the average reservoir pressure, the producer segments that have a BHP value of 5900 psia (white color) indicates that segment is shut-in. It can be seen from the plot that many producer segments between the two high permeability channels (segment 9 and segment 37) have high BHP (shut-in in white color) for all control steps. The fact that the injection segments corresponding to the south channel have high injection rates and no production helps the redistribution of the injected water, i.e. sweep oil from south to north. It is surprising to see so many control segments remain shut-in for most part of the reservoir life, especially some segments corresponding to the low permeability segments. This may be the reason that some oil is not swept by the injected water at the north and south bound area of the reservoir.

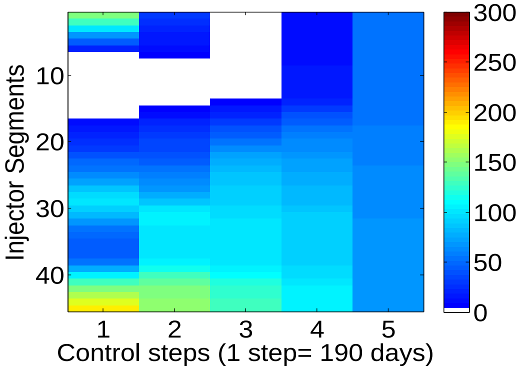


Figure 4.20: Injection rate controls after optimization.

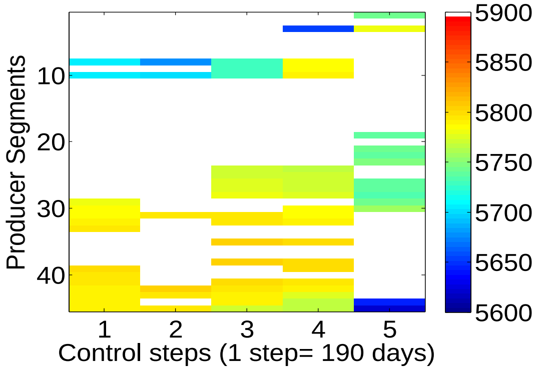


Figure 4.21: Producer BHP controls after optimization.

Fig. 4.22 shows the field water injection rate, the field oil and water production rates for both the reference case and optimized case. The reference case is denoted by lines with open circles and the optimized case is denoted by solid lines. Compared to the reference case in the first three control steps, production optimization gives higher oil production rate (red curves) and lower water production rate (blue curves). The oil and water production rates go very high (shown as spikes) at the beginning of each control step. In the reference case, this is caused by the fact we reduce the production BHP at the beginning of each control step. For the optimized case, the spikes in the production rates at the beginning of some control steps are due to the fact that some segments at some control steps are reopened. By visual inspection, we see that the equality constraints for total injection rates (green curves) are satisfied thoughout the whole reservoir life. Fig. 4.23 shows the cumulative oil and water production for both the reference and optimized cases. At the end of reservoir life of 950 days, the optimized case produces about 290,000 STB more oil than the reference case and about 340,000 STB less water than the reference case. Although the final saturation distribution and NPV in the optimized case are very close to the result obtained by [6, 40], the control strategies are quite different.

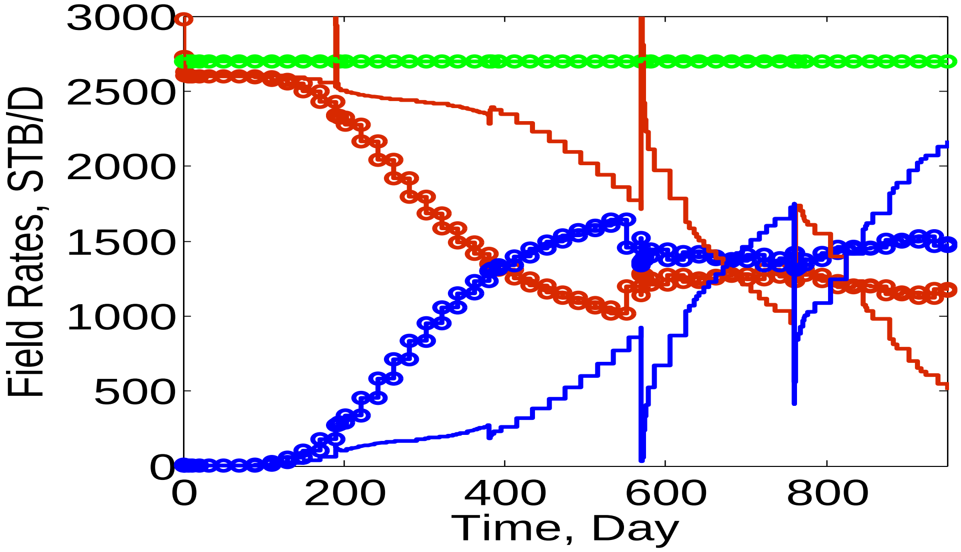


Figure 4.22: Comparison of field rates between optimized and reference cases. Curves in open cycles: Reference case; Curves in solid line: Optimization results. Red: oil production rate; Green: water injection rate; Blue: water production rate.

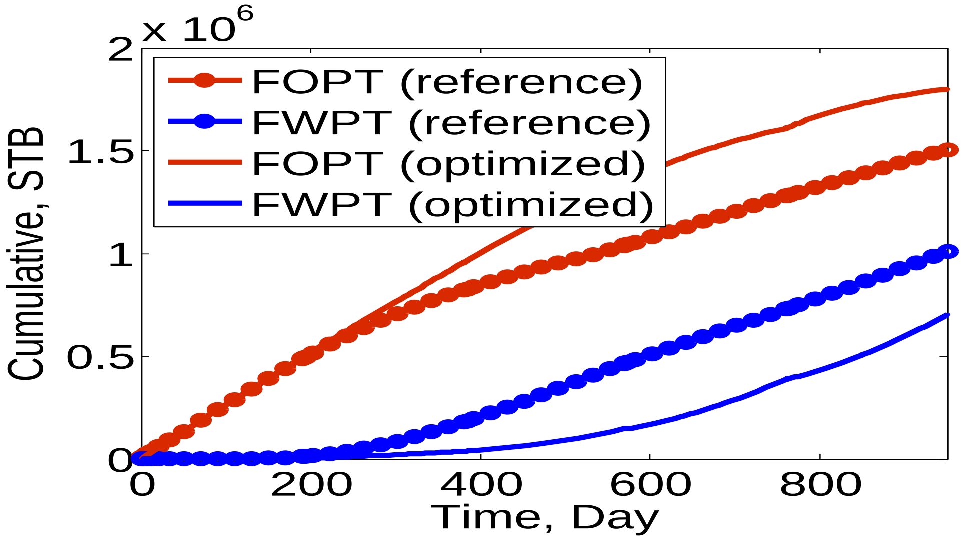


Figure 4.23: Comparison between optimized and reference cases. FOPT: Field oil cumulative production; FWPT: Field water cumulative production.

Fig. 4.24 shows a plot of the augmented Lagrangian function and the NPV versus the number of simulation runs. Starting from a feasible initial guess, the algorithm reaches a final NPV of \$2.067×10^7. The first inner-loop consists of the first

22 simulator runs. At the 13th simulator run, there is a deviation in the augmented Lagrangian function because the violation of the field water injection rate constraint is significantly large, which can be seen in Fig. 4.25 as a spike. The control variables for this iteration are not accepted. Each curve in Fig. 4.25 corresponds to the field injection rate at a certain control step. The plot shows that the field water injection rate for each control step are very close to 2700 STB/D at covergence, which indicates the equality constraints are satisfied.

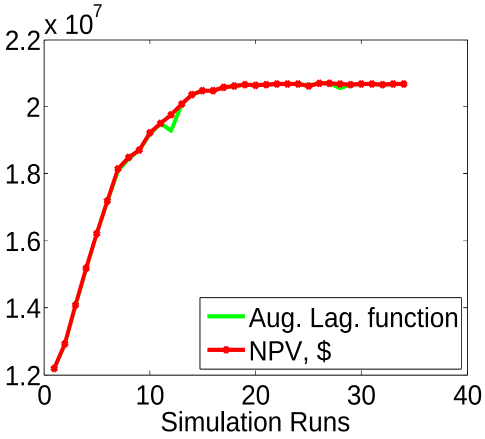


Figure 4.24: Augmented Lagrangian function and NPV versus simulation runs for equality constraint case.

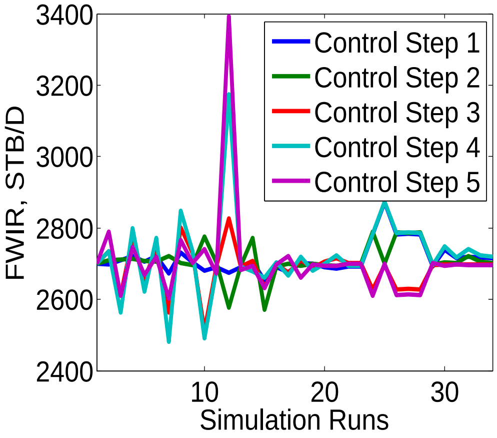


Figure 4.25: Field water injection rate versus simulation runs.

Table 4.3 shows the penalty parameters and the Lagrangian multipliers corresponding to the outer-loop iterations. Note the second row in this table shows the number of inner-loop iterations for each outer-loop iteration. Except for the bound constraints, there is only a single constraint so we simply use λ_i to denote the Lagrangian multiplier for this single equality constraint at the ith control step. We only consider the constraint at each control step so the total number of λ_i's is 5. Note that the updated λ can be negative for the equality constraint.

4.2.2 Optimization with bound and inequality constraints

In this subsection, we impose an inequality constraint on the field injection

Table 4.3: Penalty parameter and Lagrangian multipliers during optimization

Outer-loop iter. No.	1	2	3
Inner-loop iter. Numbers	22	5	7
μ	10^{-8}	10^{-8}	10^{-9}
λ_1	0	-117.7	-117.7
λ_2	0	-15.6	-15.6
λ_3	0	17.9	17.9
λ_4	0	-77.6	-77.6
λ_5	0	8.6	8.6

rate, FWIR≤2700STB/D, for the maximization problem. We have no equality constraints so the augmented Lagrangian function can be rewritten as

$$\beta[u, \lambda_c, \mu] = J_L - \sum_{i=1}^{5} \psi_i, \tag{4.7}$$

where ψ_i is defined by Eq. 2.11 and the inequality constraint is defined by

$$c_i = \sum_{j=1}^{45} q^i_{inj,j} - 2700 \leq 0, \quad i = 1, ..., 5. \tag{4.8}$$

Our initial guess for the controls is that each injection segment is allocated 60.1 STB/D of water with a total injection rate of 2704.5 STB/D, and the producer has a constant BHP of 5780 psi for the expected reservoir life. We use an infeasible point as the initial guess of injection segment rate because we want to test two situations: $\lambda^0_{c,i} = 0$ and $\lambda^0_{c,i} = \max[0, s_{c,i}c^0_i/\mu^0]$.

Table 4.4 lists the optimization results with different initial parameters. As there is only one constraint type and all the constraint values are the same, we fix the scaling factor equal to $1/2700^2$ and only test the effects of different penalty parameter and Lagrange multipliers. Here, we set the maximum number of outer-loops to 10. When using $\lambda^0 = 0$, the optimization with either $\mu^0 = 10^{-6}$ or $\mu^0 = 10^{-8}$ (row 1 and row 5 in Table 4.4) does not converge due to the fact that the violation of the constraints does not satisfy the convergence tolerance after the number of outer-loop iterations reaches the maximum allowable value (10). However, when we use the same initial penalty parameter, the optimization converges to a higher NPV with fewer simulation runs by setting $\lambda^0_{c,i} = \max[0, s_{c,i}c^0_i/\mu^0]$. The least number of simulation runs, where the final NPV is \$2.08×10^7, is obtained by setting $\mu^0 = 10^{-7}$ and $\lambda^0_{c,i} = \max[0, s_{c,i}c^0_i/\mu^0]$. From Table 4.4, it is very clear that choosing $\lambda^0_{c,i} = \max[0, s_{c,i}c^0_i/\mu^0]$ in this two-channel case outperforms choosing $\lambda^0_{c,i} = 0$. Table 4.4 also lists the updated number of μ and λ during outer-loop iteration. The number of μ updated is larger than or equal to the number of λ updated except in a case with $\mu^0 = 10^{-7}$ and $\lambda^0_{c,i} = \max[0, \frac{s_{c,i}c^0_i}{\mu^0}]$.

Table 4.4: The performance of different parameters. N_μ: Number of μ updated; N_λ: Number of λ updated.

μ^0	$\lambda^0_{c,i}$	Simulations	Final NPV, \$	Outer-Loop	N_μ	N_λ	Converge
10^{-6}	0	75	2.09×10^7	10	9	0	No
10^{-6}	$\max[0, \frac{s_{c,i}c^0_i}{\mu^0}]$	44	2.09×10^7	6	4	1	Yes
10^{-7}	0	24	2.08×10^7	3	1	1	Yes
10^{-7}	$\max[0, \frac{s_{c,i}c^0_i}{\mu^0}]$	14	2.08×10^7	2	0	1	Yes
10^{-8}	0	80	2.06×10^7	10	7	2	No
10^{-8}	$\max[0, \frac{s_{c,i}c^0_i}{\mu^0}]$	33	2.07×10^7	4	2	1	Yes

Here, we only show the behaviour of the objective function and the field water injection rate versus simulation runs for the initial parameters $\mu^0 = 10^{-8}$ and $\lambda^0_{c,i} = \max[0, \frac{s_{c,i}c^0_i}{\mu^0}]$. Fig. 4.26 shows the behavior of the objective function. The augmented Lagrangian function and the NPV overlap before the simulation run No. 13. The big decrease of the augmented Lagrangian function at simulation No. 13 is a result of the significant constraint violation (see Fig. 4.27). Another big decrease occurs at simulation No. 31, which is the third simulation run at the fourth outer-loop. The reason is also due to the violation of the FWIR constraint at the 3rd control step.

As shown in Fig. 4.27, all the constraints are satisfied at the end. We also list the values of penalty parameter and Lagrange multipliers at each outer-loop in Table 4.5. The optimization includes four outer-loop iterations, where the penalty parameter is updated at the 3rd and the 4th outer-loop and the Lagrange multipliers are updated at the second outer-loop. Recall that when we update the penalty parameter, we keep the Lagrange multipliers unchanged.

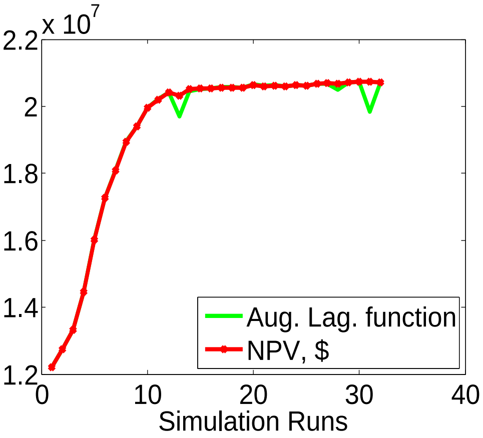


Figure 4.26: Augmented Lagrangian function and NPV versus simulation runs for inequality constraint case; $\mu^0 = 10^{-8}$, $\lambda^0_{c,i} = \max[0, \frac{s_{c,i}c^0_i}{\mu^0}]$.

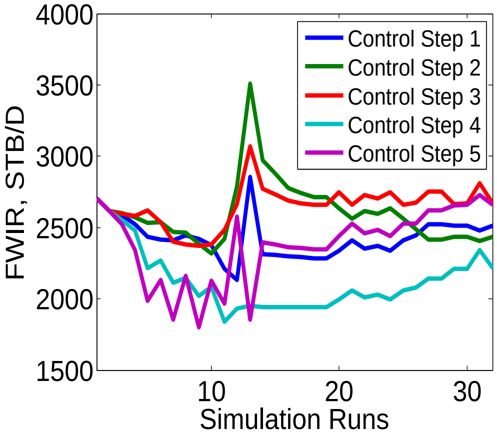


Figure 4.27: Field water injection rate versus simulation runs; $\mu^0 = 10^{-8}$, $\lambda^0_{c,i} = \max[0, \frac{s_{c,i}c^0_i}{\mu^0}]$.

Table 4.5: Penalty parameter and Lagrangian multipliers during optimization ($\mu^0 = 10^{-8}$, $\lambda^0_{c,i} = \max[0, \frac{s_{c,i}c^0_i}{\mu^0}]$)

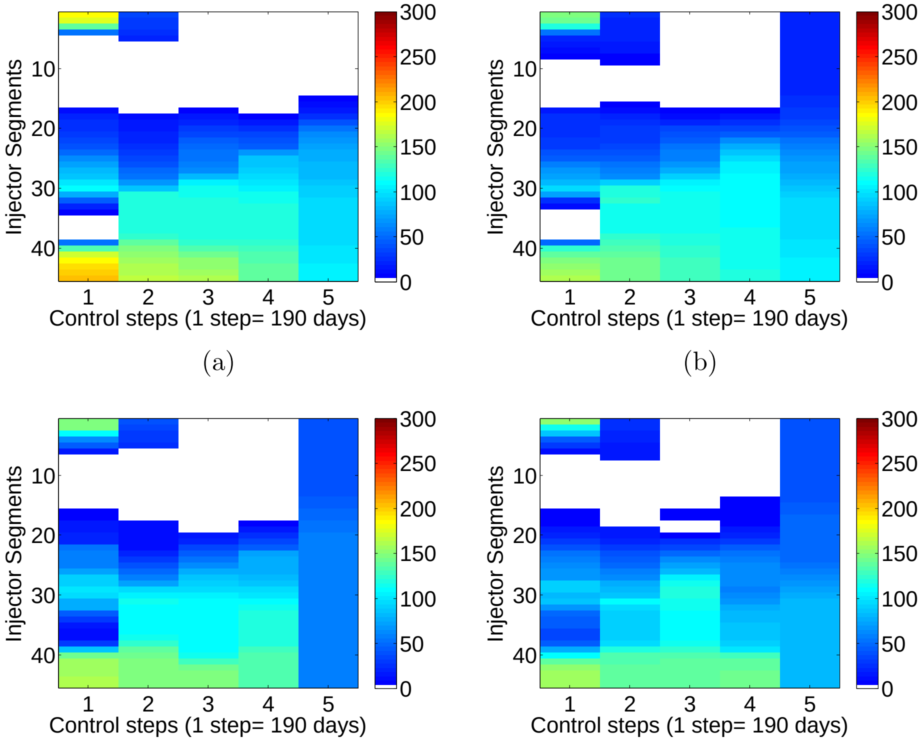
Outer-loop iter. No.	1	2	3	4
Inner-loop iter. Numbers	18	5	4	5
μ	10^{-8}	10^{-8}	10^{-9}	10^{-10}
λ_1	6.17	0	0	0
λ_2	6.17	223.1	223.1	223.1
λ_3	6.17	0	0	0
λ_4	6.17	0	0	0
λ_5	6.17	0	0	0

Fig. 4.28 shows the optimal injection well controls obtained with different ini-

tial parameter settings: (a) $\mu^0 = 10^{-6}$, $\lambda^0_{c,i} = \max[0, \frac{s_{c,i}c^0_i}{\mu^0}]$; (b) $\mu^0 = 10^{-7}$, $\lambda^0_{c,i} = 0$; (c) $\mu^0 = 10^{-7}$, $\lambda^0_{c,i} = \max[0, \frac{s_{c,i}c^0_i}{\mu^0}]$; (d) $\mu^0 = 10^{-8}$, $\lambda^0_{c,i} = \max[0, \frac{s_{c,i}c^0_i}{\mu^0}]$. Although the optimal controls are different, the trend is similar. The injection segments corresponding to the lower, high permeability channel have a high injection rate and the injection segments corresponding to the upper channel are shut in or have very low injection rate for the first several control steps. The purpose is to inject water from lower part and then sweep oil from lower to upper part. Fig. 4.29 shows the optimal producer BHP controls obtained with different initial parameter settings. The optimal well controls are fairly different, although all of them lead to a similar NPV. This indicates the optimal solution for the production optimization problem may not be unique and there are extra degrees of freedom in the control variables which can be used to optimize for a secondary objective function (e.g., short-term NPV). We will discuss the short-term optimization after life-cycle optimization in the next chapter by using the extra degrees of freedom.

4.2.3 Optimization with only bound constraints

The main purpose of this subsection is to compare the gradient-based optimization algorithm with an ensemble-based gradient-free algorithm, EnOpt. The details of the EnOpt algorithm is given in Appendix C. As it is difficult to handle the equality or the inequality constraints with EnOpt, the optimization problem in this subsection is to maximize the life-cycle NPV subject to only the bound constraints on controls. Therefore, the augmented Lagrangian function has only the NPV term, which is maximized with the gradient-projection trust-region method subject to bound constraints. Note that this optimization requires only one outer-loop optimization, as there are no Lagrangian multipliers or penalty parameter to be updated. We tried both the log-transformation and truncation method to handle the bound constraints when using EnOpt; see the discussion of the log-transformation and the truncation method in Chapter 1. As the gradient of NPV with respect to well BHP



(a)

(b)

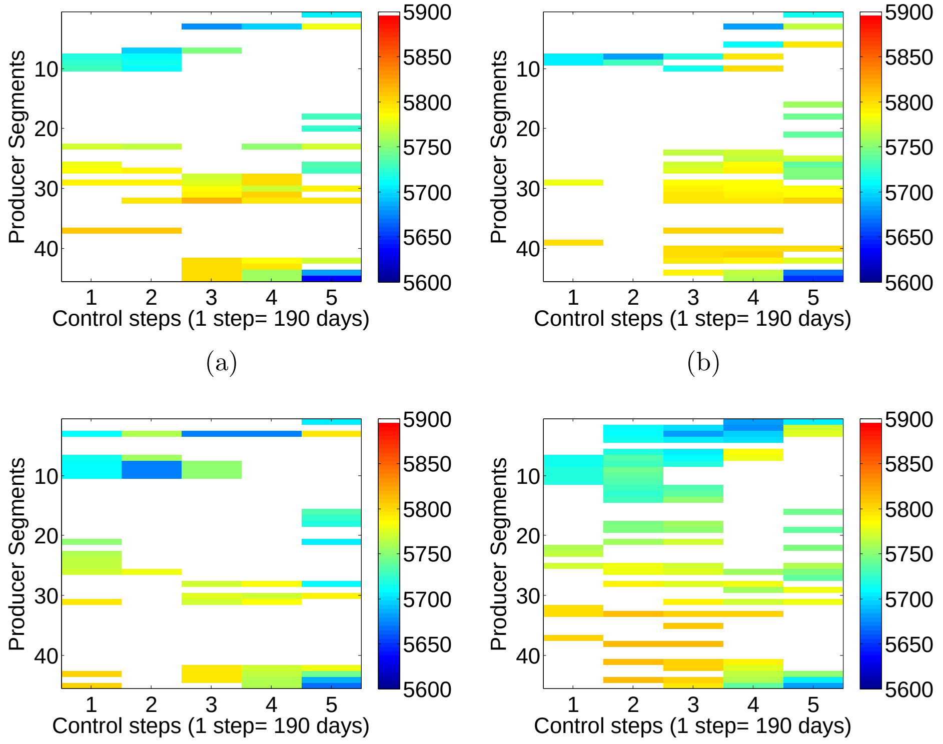
(c)

(d)

Figure 4.28: The estimated optimal injection well controls. (a) $\mu^0 = 10^{-6}$, $\lambda^0_{c,i} = \max[0, \frac{s_{c,i}c_i^0}{\mu^0}]$; (b) $\mu^0 = 10^{-7}$, $\lambda^0_{c,i} = 0$; (c) $\mu^0 = 10^{-7}$, $\lambda^0_{c,i} = \max[0, \frac{s_{c,i}c_i^0}{\mu^0}]$; (d) $\mu^0 = 10^{-8}$, $\lambda^0_{c,i} = \max[0, \frac{s_{c,i}c_i^0}{\mu^0}]$.

controls obtained from Eclipse 300 does not seem to be right for this problem, we use CLASS for the gradient-based production optimization. However, EnOpt is done with Eclipse 100. To make the two results comparable, the NPV at each simulation run in the gradient-based production optimization is obtained by running Eclipse 100 using the controls from each iteration of gradient-based optimization with CLASS. Fig. 4.30(a) shows the NPV versus simulation runs with the simulator CLASS. As shown in Fig. 4.30(a), it only takes 19 simulation runs to obtain the optimal NPV of \$2.10×10^7.

For EnOpt with the log-transformation, the ensemble size is 20 and the initial step size is 1. We calculate the EnOpt gradient by a Gaussian perturbation from $N(0, C_U)$. The correlation length in time in C_U for the control variables is 2 control



(a)

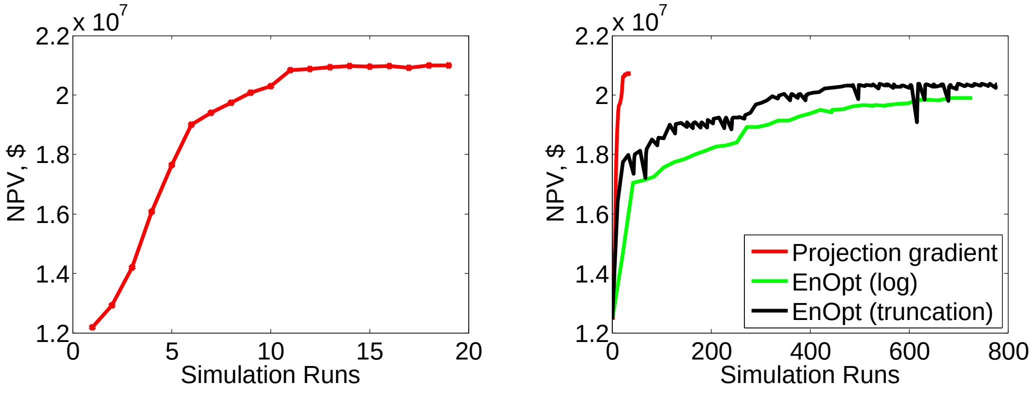
(b)

(c)

(d)

Figure 4.29: The estimated optimal optimal BHP well controls. (a) $\mu^0 = 10^{-6}$, $\lambda^0_{c,i} = \max[0, \frac{s_{c,i}c^0_i}{\mu^0}]$; (b) $\mu^0 = 10^{-7}$, $\lambda^0_{c,i} = 0$; (c) $\mu^0 = 10^{-7}$, $\lambda^0_{c,i} = \max[0, \frac{s_{c,i}c^0_i}{\mu^0}]$; (d) $\mu^0 = 10^{-8}$, $\lambda^0_{c,i} = \max[0, \frac{s_{c,i}c^0_i}{\mu^0}]$.

steps. The variance of the controls is 1. In EnOpt with the truncation, we set the initial step size equal to 50 and the ensemble size is 10. The variance of the controls is 100. Fig. 4.30(b) shows the comparison of NPV as a function of the number of simulation runs during iteration between the augmented Lagrangian method and EnOpt method. The initial guess for these methods is that water injection rate for each injection segment is 60 STB/D and the producer has a constant 5780 psi for all the control steps. The NPV at the initial guess for the controls is about \$1.27 × 10^7. The performance of the algorithms are summarized in Table 4.6. The 38 simulation runs in Table 4.6 required for the method based on an adjoint gradient consists of 19 forward reservoir simulation runs and 19 adjoint solutions for calculating the gradient of NPV. This is the reason that the number of simulation runs in Table 4.6 required for the gradient-projection trust-region method is twice as the number of simulation runs in Fig. 4.30(a).



(a)

(b)

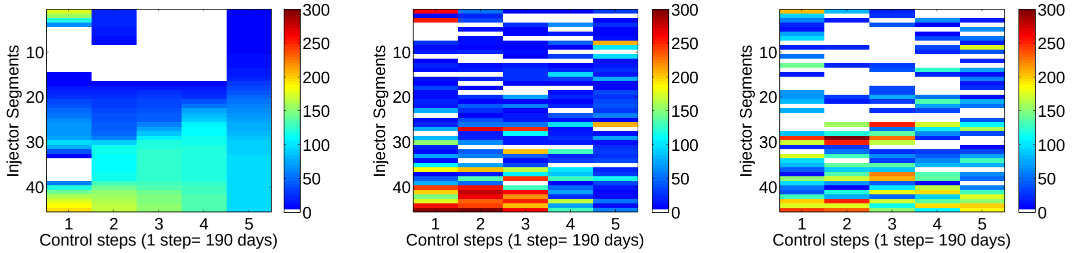
Figure 4.30: The objective function versus simulation runs. (a) Simulation with simulator CLASS; (b) Simulation with simulator Eclipse 100.

Table 4.6: The performance of different optimization algorithms.

Algorithm	No. of simulations	Final NPV,$\times 10^{7}\$$
Adjoint Gradient	38	2.073
EnOpt (log-trans)	727	1.990
EnOpt (truncation)	777	2.036

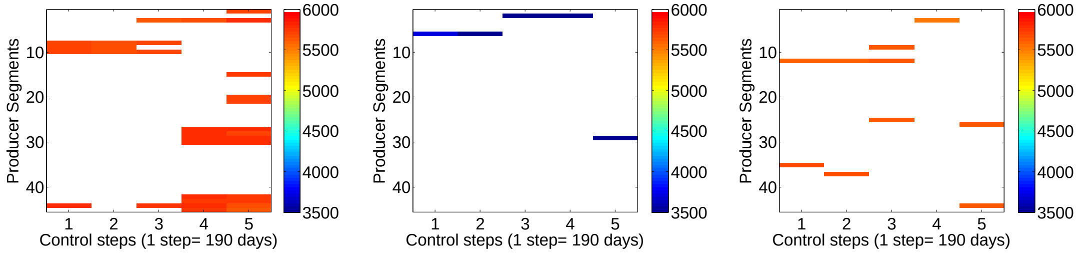
The final controls for producers and injectors obtained with the different optimization algorithms are shown in Figs. 4.31 and 4.32. The final controls are quite different for all the algorithms. Although most of the final producer BHP controls shown in Fig. 4.32(a) are visually higher than the BHP controls in Fig. 4.32(b) and Fig. 4.32(c), the gradient-based optimization keeps more production segments open.

Fig. 4.33 shows the final reservoir oil saturation obtained by reactive control strategy and with the estimate of optimal well controls from the production optimization. The results of Fig. 4.33 show that the estimated optimal controls obtained with all the optimization algorithms result in much better sweep efficiency than is obtained with reactive control. Visually, the oil saturations distributions from the two



(a) Adjoint-gradient (b) EnOpt (log-trans) (c) EnOpt (truncation)

Figure 4.31: The estimated optimal injection well controls.



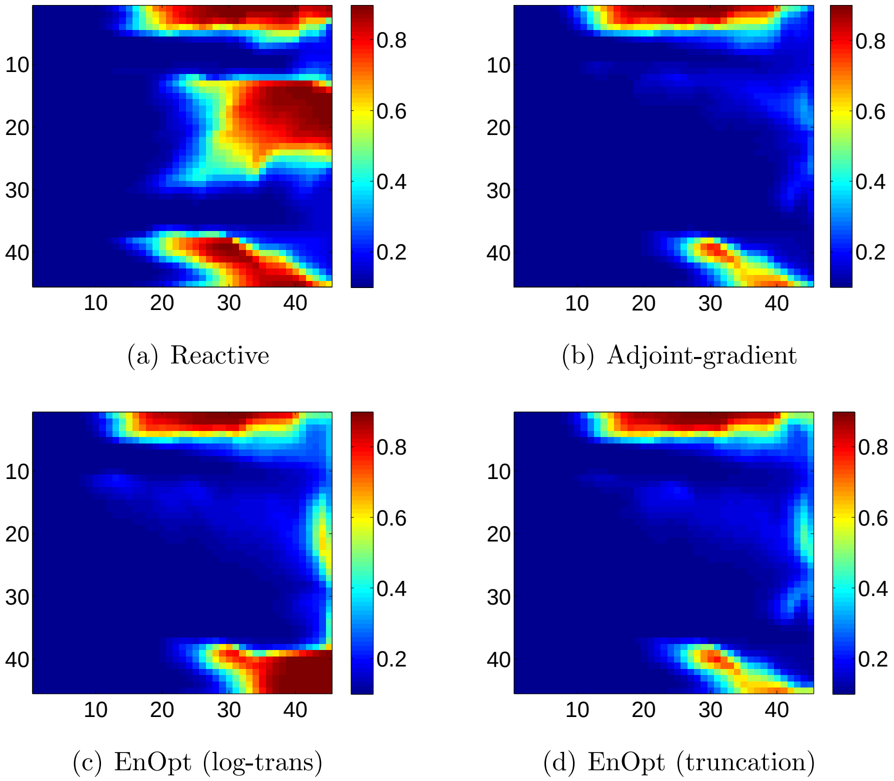
(a) Adjoint-gradient (b) EnOpt (log-trans) (c) EnOpt (truncation)

Figure 4.32: The estimated optimal BHP well controls.

optimization algorithms are fairly similar, but the total water produced with the optimal well controls estimated with EnOpt are higher than the total water production using the optimal controls from the gradient-based optimization algorithm. Thus, EnOpt results in a much lower final NPV value than the gradient-based optimization algorithm.

4.3 Example 3: Brugge Benchmark Case

In this example, we compare the results of production optimization from the augmented Lagrangian multiplier method and the generalized reduced gradient method (an option in the Eclipse 300 for production optimization, see Appendix B) using the Brugge case. A detailed discussion of the Brugge field is given in Peters et al. [35]. Here we provide only a brief summary from that paper. The stratigraphy of the Brugge field is modeled after a typical North Sea Brent field and is an elongated half-dome with one internal fault and one boundary fault. The top structure map



(a) Reactive

(b) Adjoint-gradient

(c) EnOpt (log-trans)

(d) EnOpt (truncation)

Figure 4.33: Remaining oil saturation distribution after optimization.

with well locations is shown in Fig. 4.34. The Brugge reservoir contains 20 "smart" producing wells and 10 "smart" water injection wells. All wells are vertical and each well is equipped with inflow control valves to control flow into three different well segments. The original model was constructed with approximately 20 million gridblocks and then upscaled to a 450,000 gridblock model which formed the true case from which data were generated by TNO [35]. The true reservoir rock property fields were known only to TNO professionals. The truth case was used to construct data such as well logs and facies maps. Using this information, 104 geological realizations upscaled to a 60,000 gridcell model were created and provided to participants.

The sandstone reservoir is bounded with faults and connected to a large aquifer. The reservoir has four geological layers and is further divided into 9 simulation layers. The Schelde formation corresponds to the top two simulation layers, the Waal formation to layer 3 to layer 5, the Maas formation to layer 6 to layer 8 and

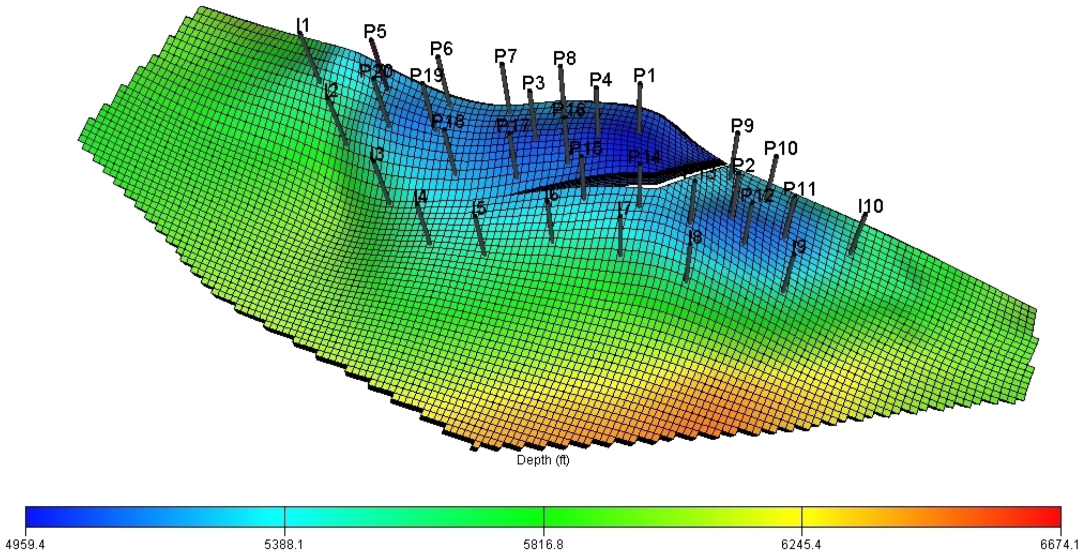


Figure 4.34: The top structure of Brugge field, Example 3.

the Schie formation to the bottom layer. The structure of the field and fault descriptions were the same in all realizations. Each realization consisted of the following properties on each gridblock: net-to-gross (NTG), porosity, initial water saturation and absolute permeability in the x, y and z directions. Different geostatistical modeling methods were used for different realizations. For example, some realizations of properties in the first two layers were created with object-based modeling (channel objects in a shale background) whereas other realizations were created with sequential indicator simulation.

Throughout the reservoir life, only oil and water flow in the reservoir. Relative permeability and capillary pressure curves were provided by TNO [35] but with seven possible values of irreducible water saturation with the value of irreducible water saturation depending on the gridblock value of porosity.

First, participants are required to history match the production history of the first 10 years (the data to be matched are oil rates of the producers and BHPs of all wells). During the first 10 years of production, wells did not have inflow control valves (ICV). The target total liquid rate for each producer during the first 10 years was 2,000 STB/D with the minimum allowable bottomhole pressure equal to 725 psi. Water injection rates during this period were set equal to 4,000 STB/D with a maximum injection pressure of 2,611 psi slightly more than 150 psi above the initial pressure at datum. We were provided with well pressure, production and injection rate data. We assimilated these observed data at six month intervals during the first ten years using EnKF [8]. We simply use the mean model of the ensemble of the updated geological models from EnKF for production optimization. We will expand our discussion on robust optimization with this Brugge case in Chapter 5.

Fig. 4.35 shows the x-direction log-permeability for all the nine simulation layers obtained with the EnKF after assimilating production data of the first 10 years. The first two layers are channelized reservoir and layers 6 to 8 are the relatively homogeneous layers. All the producers are completed in the top 8 layers. Fig. 4.36 shows the oil saturation distribution after 10 years of production. The big blue region is the aquifer and oil region is shown in all layers at the right middle part as yellow to brown colors. Based on the mean model, we optimized the NPV for years 10-30. As ICV's are installed at year 10 to control the production rates and injection rates for the Schelde, the Waal and the Maas formations, respectively, there are a total of 54 production segments and 30 injection segments. Note that Producer 9 has only a single ICV (in the Schelde formation) and Producer 5, 10, 14 and 15 have two ICVs (in the Schelde and Waal formations).

4.3.1 Optimization with bound and inequality constraints

The control variables are production segment liquid rates for producer segments and water injection segment rates for injection segments. We have 84 controls at each control step. During production optimization, the following constraints were imposed: The total liquid production rate for each well should not exceed 3000 STB/D and the BHP for producers should not be less than 725 psi in any segment; the total

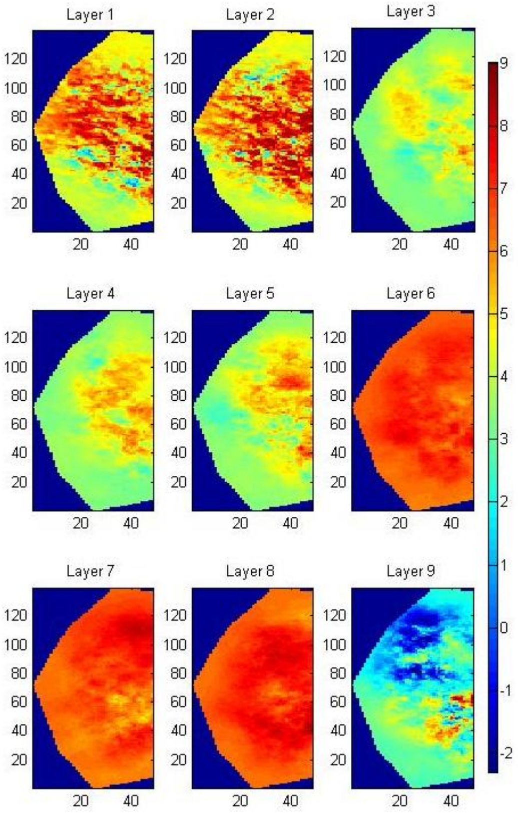


Figure 4.35: $\ln(k_x)$, x-direction log-permeability after history match of period 0-10.

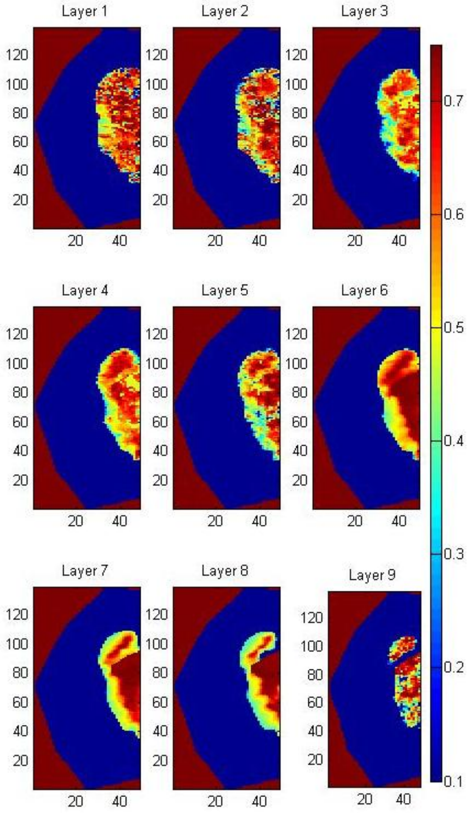


Figure 4.36: Oil Saturation at year 10 before optimization.

water injection rate at each well should not exceed 4000 STB/D and the injection pressure should not exceed 2611 psi in any segment; the liquid production rate for each well segment should be constrained within 0.1 and 3000 STB/D; the injection rate for each well segment should be within 0.1 and 4000 STB/D. The reason why we use 0.1 STB/D as the lower bound for rate control is that Eclipse 300 will shut in the well when its rate is set to zero, and the software does not output the gradient of NPV when a well is shut in. The oil revenue price is \$80/STB and both water production and injection costs are \$5/STB. The annual discount rate is 10%.

In setting up the Brugge case, TNO used a separate well for each segment in the original smart well, so each smart well is actually modeled with three separate wells; to be consistent we do the same, but we still refer to these three separate wells as three smart well segments. The control step for production optimization is 182.5 days (1/2 year). Although Eclipse 300 claims it is able to handle the optimization problem with BHP control variables, we observed in our computational experiments that Eclipse 300 does not always give the correct gradient of NPV with respect to BHP control. Therefore, we choose the controls to be the liquid production rate and the water injection rate for each smart well segment. With 40 control steps for the Brugge test case, we have 3360 control variables for the 20-year production optimization problem. We use two sets of initial guesses for this example: 1). the initial guess for total liquid rate controls is 700 STB/D for each smart production well segment and 1333.3 STB/D for each smart injection well segment; 2). the initial guess for rate controls is 1001 STB/D for each smart production well segment and 1334 STB/D for each smart injection well segment. With the first set of initial guesses, the rate constraints for producers and injectors are all satisfied at the first iteration so the initial values for Lagrangian multipliers estimated with Eq. 2.22 are all equal to zero. With the second set of initial guesses, we are able to obtain nonzero initial values for Lagrangian multipliers estimated from Eq. 2.22. For each smart producer, we have a total liquid rate constraint over the three segments, i.e., $q_{L_{j,1}} + q_{L_{j,2}} + q_{L_{j,3}} \leq 3000$ STB/D for $j = 1, 2, \cdots, 20$ and for each smart injector, we have a total water injection rate constraint over the three segments, i.e., $q_{inj_{j,1}} + q_{inj_{j,2}} + q_{inj_{j,3}} \leq 4000$ STB/D for $j = 1, 2, \cdots, 10$. With 40 control steps, the total number of constraints is 1200. We have no equality constraints so the augmented Lagrangian function can be written as

$$\beta[u, \lambda_c, \mu] = J_L - \sum_{i=1}^{1200} \psi_i, \tag{4.9}$$

where ψ_i is defined by Eq. 2.11. For $k = 1, 2, ..., 40$, c_i is defined as

$$c_i = q^k_{L_{j,1}} + q^k_{L_{j,2}} + q^k_{L_{j,3}} - 3000 \leq 0, \quad i = (k-1) \cdot 20 + j, \tag{4.10}$$

for the jth smart producer at the kth control step where $j = 1, 2, ..., 20$ and

$$c_i = q^k_{inj_{j,1}} + q^k_{inj_{j,2}} + q^k_{inj_{j,3}} - 4000 \leq 0, \quad i = 800 + (k-1) \cdot 10 + j, \tag{4.11}$$

for the jth smart injector at the kth control step where $j = 1, 2, ..., 10$.

If we apply the slack variables to convert these inequality constraints to equality constraints for the augmented Lagrangian method, we would have 3360 + 1200 = 4560 parameters to adjust during optimization with 3360 additional bound constraints on the control variables. However, with the method proposed in the Chapter 2, we have eliminated the slack variables and hence greatly reduced the size of the optimization problem. As Eclipse 300 does not output the gradient of the bottom-hole pressure constraint with respect to the control variable, we enforce constraints on BHP during the forward simulation run, i.e., the simulator will put the well under BHP control if the resulting BHP from rate control is beyond the BHP constraints. For this example, it only happens for a few controls and does not seem to cause a serious problem for the overall production optimization. The NPV's are estimated with the Eclipse 300 package.

Here we compare the results from the generalized reduced gradient (GRG) method (see Appendix B) in Eclipse 300 for production optimization with the results from our augmented Lagrangian multiplier method. As we do not have the true Brugge reservoir model, we optimize NPV with the two methods using the model shown in Fig. 4.35 and Fig. 4.36 as the true model. If all the inequality constraints are inactive, the generalized reduced gradient method deals with the simple bound constraints by limiting the step size. If some inequality constraints become active, the generalized reduced gradient method in Eclipse 300 divides the control variables set into "free" and "dependent" subsets. The gradient of the free control variables are calculated using the backward adjoint run. The gradient of the dependent control variables are obtained with the linearized simulation and active constraint equations.

The search direction can be either steepest ascent direction or conjugate gradient direction. In this study, we use the option of conjugate gradient direction as we observed that the option of steepest ascent direction results in a bit lower optimal NPV than the option of conjugate gradient direction.

For the augmented Lagrangian method, the gradient of the NPV, the first term in Eq. 2.7 is calculated using the adjoint method embedded in Eclipse 300. The gradient of the other two terms in Eq. 2.7 related to the constraints are calculated analytically as all the constraints are linear functions of the rate control.

Figs. 4.37 shows the comparison of objective functions versus simulation runs between the augmented Lagrangian method and the GRG method for the case we try the first set of initial guesses. The GRG applied in Eclipse 300 converges with 20 simulation runs and obtains an NPV of $\$3.81\times10^9$. The augmented Lagrange method shown in Fig. 4.37 obtains a much higher NPV of $\$4.24\times10^9$ than the GRG, although it uses 57 simulation runs. We started with a feasible initial guess, the NPV and the augmented Lagrangian function overlap between simulation run 0 to 5. There are some significant decreases in the augmented Lagrangian function during simulation runs while the NPV increased, due to significant violation of the constraints. The significant decrease in the augmented Lagrangian function at the 14th, 25th, 34th, 38th, 43th and 51st simulation runs is due to the change in the penalty parameter or the Lagrange multipliers in the new outer-loop iteration. While the first inner-loop tried to increase NPV by violating the constraints, the second inner-loop tried to satisfy all the constraints with a slight decrease in NPV. Fig. 4.38 shows the violation of liquid rate constraints versus simulation runs. Each curve corresponds to one c_i defined in Eqs. 4.10 and 4.11. The positive value of c_i indicates the ith constraint is violated; otherwise, the constraint is satisfied. It can be seen from the figure that all constraints are basically satisfied from the 30th simulation run onward. The final estimated NPV using the mean geological model is equal to $\$4.23 \times 10^9$.

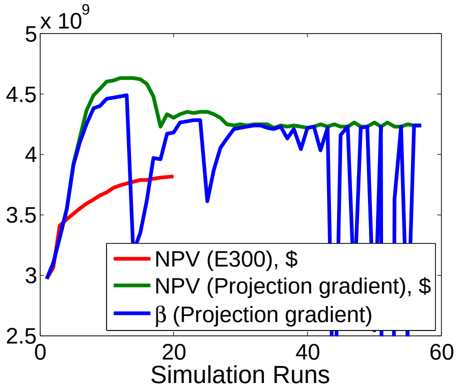


Figure 4.37: Objective functions versus simulation runs; Initial guess 1.

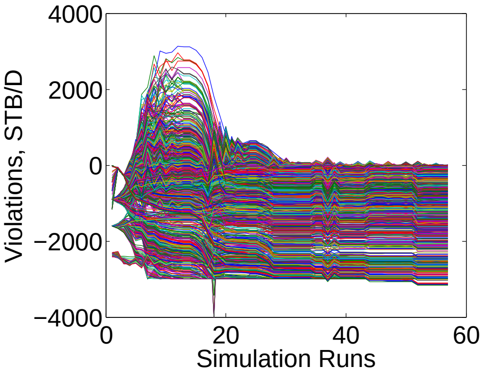


Figure 4.38: Violation versus simulation runs; Initial guess 1.

Table. 4.7 shows the augmented Lagrangian method involves seven outer-loop iterations. The penalty parameter is decreased by 0.1 at each outer-loop iteration except the fourth outer-loop iteration, where the Lagrange multipliers are updated. We only select several non-zero Lagrange multiplier to list in Table. 4.7. In fact, 90 percent of the Lagrange multipliers are still zero at the fourth outer-loop iteration.

Table 4.7: Penalty parameter and Lagrangian multipliers during optimization for $\mu^0 = 10^{-7}$.

Outer-loop iter. No.	1	2	3	4	5	6	7
Inner-loop iterations	13	11	9	4	5	8	7
μ	10^{-7}	10^{-8}	10^{-9}	10^{-9}	10^{-10}	10^{-11}	10^{-12}
$\lambda_{c,1}$	0	0	0	7094	7094	7094	7094
$\lambda_{c,123}$	0	0	0	2527	2527	2527	2527
$\lambda_{c,481}$	0	0	0	5640	5640	5640	5640
$\lambda_{c,871}$	0	0	0	1113	1113	1113	1113

With the first set of initial guesses, the initial point is feasible so that the initial Lagrange multipliers are zero even if we use Eq. 2.22 to estimate the initial Lagrangian multipliers. Therefore, we pick the second set of initial guesses in order to compare the optimization results obtained with $\lambda_{c,i} = 0$ and with $\max[0, \frac{s_{c,i}c_i^0}{\mu^0}]$. Figs. 4.39 shows the comparison of objective functions versus simulation runs between the augmented Lagrangian method and the GRG method for the case we try the second set of initial guesses and initial Lagrange multipliers estimated with Eq. 2.22. The GRG applied in Eclipse 300 converges with 14 simulation runs and obtains an NPV of \$3.74×10^9, which is lower than the NPV obtained with the first set of initial guesses. However, the augmented Lagrangian method shown in Fig. 4.39 obtains the same NPV of \$4.24×10^9 as that obtained with the first set of initial guesses, and it takes only 41 simulation runs to converge. It can be seen from Fig. 4.40 that the all constraints are basically satisfied from the 35th simulation run onward.

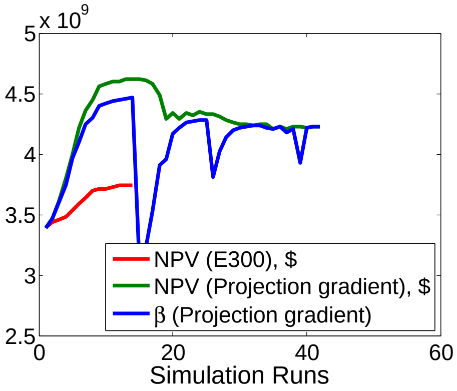


Figure 4.39: Objective functions versus simulation runs; Initial guess 2.

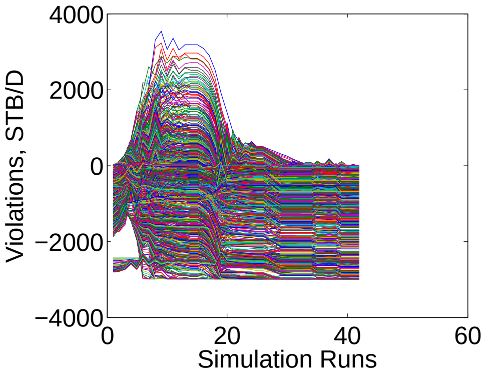


Figure 4.40: Violation versus simulation runs; Initial guess 2.

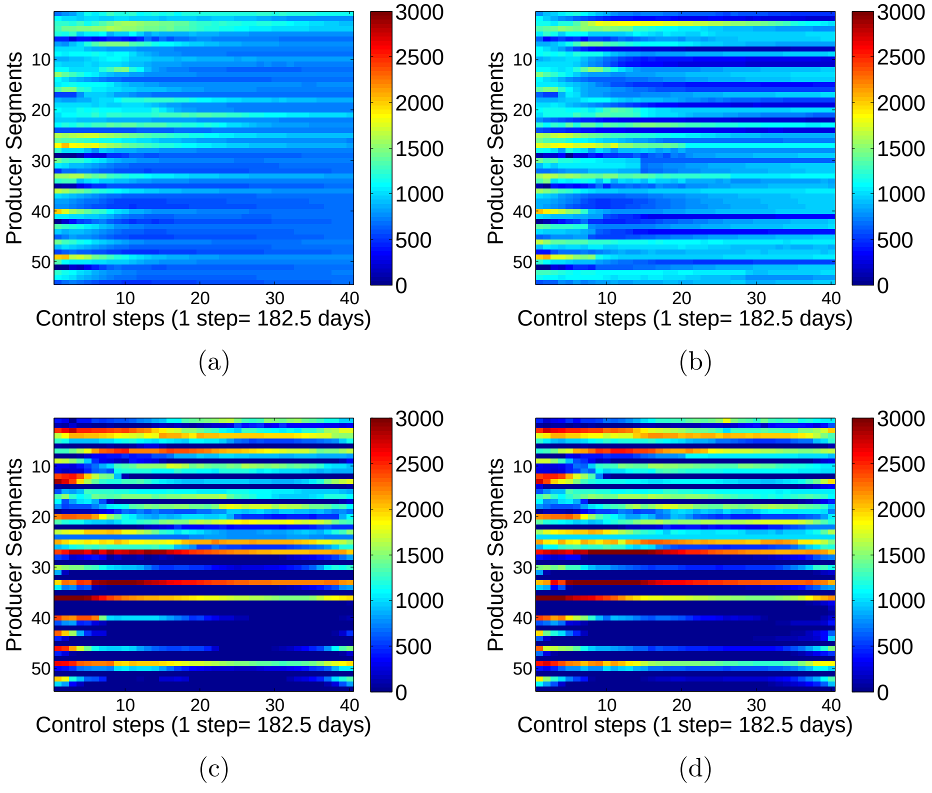
Table 4.8 summarizes the performance for the cases with or without scaling factors and different initial penalty parameter and initial Lagrange multipliers. There are two types of constraint values, C_i=3000 or 4000 STB/D. The first column indicates whether we use the scaling factor; the second column shows the various initial penalty parameter; and the third column indicates how the initial Lagrange multipliers are obtained. They all obtain a final optimal NPV of around \$4.24×10^9 except for the case with $\mu^0 = 10^{-1}$ and $s_{c,i} = 1$, which yields that a final NPV 2.8% less than the highest NPV of \$$4.25 \times 10^9$. It can be seen from Table 4.8, the optimization cost is cheapest when we use $\mu^0 = 10^{-7}$ and $s_{c,i} = 1/C_i^2$. In terms of simulation runs, choosing $\lambda_{c_i}^0 = 0$ requires slightly fewer reservoir simulation runs than are required when we estimate the initial values of Lagrange multipliers from Eq. 2.22. However, the difference in the number of simulation runs and estimated optimal NPV obtained using $\lambda_{c,i}^0 = 0$ and $\lambda_{c,i}^0 = \max[0, \frac{s_{c,i} c_i^0}{\mu^0}]$ is very small so we conclude that the final optimization result is not very sensitive to the choice of the initial Lagrange multipliers. Table 4.8 also lists the updated numbers for μ and λ. For all the cases, the number of μ updated is larger than or equal to the number of λ updated.

Table 4.8: The performance of different parameters (with initial guess 2). N_μ: Number of μ updated; N_λ: Number of λ updated.

$s_{c,i}$	μ^0	$\lambda_{c,i}^0$	Simulations	Final NPV, \$	Outer-Loop	N_μ	N_λ	Converge
1	1	0	45	4.23×10^9	5	3	2	Yes
1	1	$\max[0, \frac{s_{c,i} c_i^0}{\mu^0}]$	50	4.23×10^9	6	3	3	Yes
1	10^{-1}	0	49	4.13×10^9	5	2	2	Yes
1	10^{-1}	$\max[0, \frac{s_{c,i} c_i^0}{\mu^0}]$	50	4.15×10^9	5	2	2	Yes
$1/C_i^2$	10^{-6}	0	54	4.25×10^9	6	3	2	Yes
$1/C_i^2$	10^{-6}	$\max[0, \frac{s_{c,i} c_i^0}{\mu^0}]$	58	4.25×10^9	5	4	0	Yes
$1/C_i^2$	10^{-7}	0	40	4.23×10^9	5	3	1	Yes
$1/C_i^2$	10^{-7}	$\max[0, \frac{s_{c,i} c_i^0}{\mu^0}]$	41	4.23×10^9	5	3	1	Yes

Figs. 4.41 through 4.43 shows the comparison of the estimated optimal liquid rates for production well segments. For the augmented Lagrangian method, the results provided here are obtained with $\mu^0 = 10^{-7}$ and $\lambda_{c,i}^0 = \max[0, \frac{s_{c,i} c_i^0}{\mu^0}]$. The final controls obtained for the two sets of initial guesses are similar for both the augmented Lagrangian method and the GRG method. Fig. 4.42 presents the final optimal controls obtained from the Eclipse 300 optimizer for producers P1, P4 and P12. Fig. 4.43 shows the final optimal controls obtained from the augmented Lagrangian method for these three producers. In both Fig. 4.42 and Fig. 4.43, the panels in the first column show the liquid production rate from each segment and the total well liquid production rate. The column in the middle shows the resulting BHP values for each segment and the right column shows the resulting water-cut values for each segment. Day 1 of the x-axis corresponds to the first day of year 10. As Eclipse 300 does not

provide the gradient when a well is shut in, we use 0.1 STB/D as the lower bound for rate control and Eclipse 300 is able to output the water-cut for a well using 0.1 STB/D liquid production rate. The comparison from the plots shows that the final optimal production strategy from both methods indicates some significant difference. For example, the producer segments of P12 and P4 in Fig. 4.43 are shut-in when water-cut is too high, however, the corresponding well segments in Fig. 4.42 are still open. The third segment of P4 is shut in after year 17.5 but the water-cut is only around 0.85. The optimal control shows a different strategy compared to reactive control. If the reactive control based on the economic limit of the water-cut were used, the well would not be shutin until the water-cut exceeds 0.94 .



(a)

(b)

(c)

(d)

Figure 4.41: The estimated optimal liquid rates for production well segments. (a) E300 optimizer, initial guess 1; (b) E300 optimizer, initial guess 2; (c) Augmented Lagrangian method, initial guess 1; (d) Augmented Lagrangian method, initial guess 2.

Fig. 4.44 shows the comparison of the estimated optimal production liquid

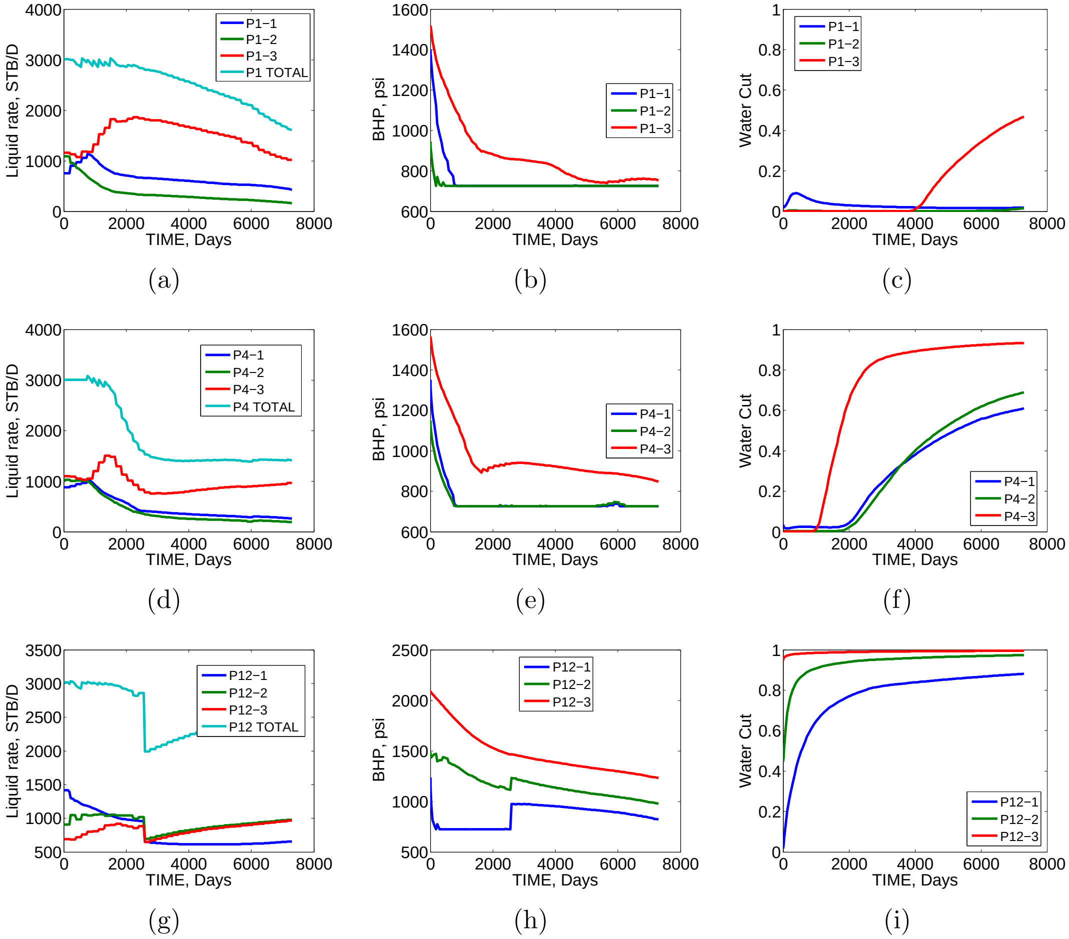
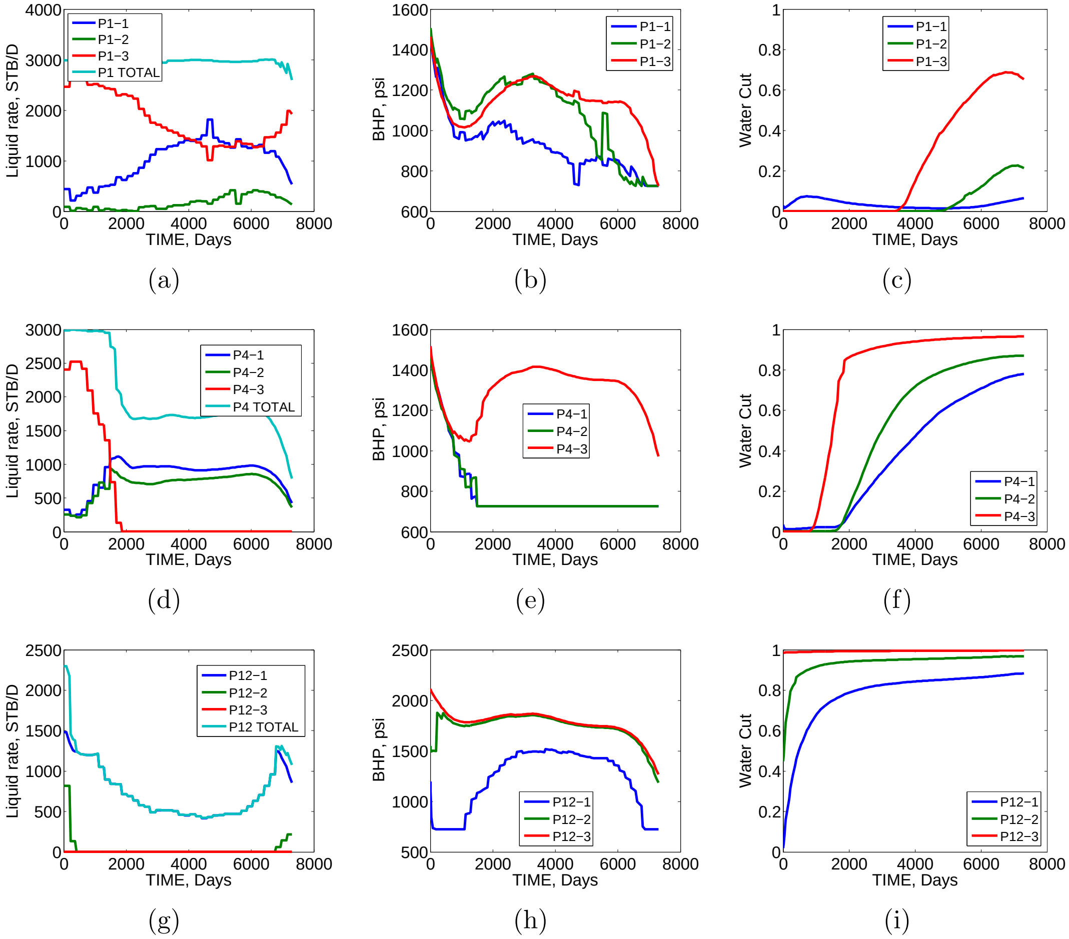


Figure 4.42: Liquid rate controls, BHP behaviors and well segment water-cut for three producers; E300 optimizer; Initial guess 2.

rates of three producers obtained with the augmented Lagrangian method and the Eclipse 300 optimizer. For the producer P1, which is located near the inner fault and far from the injectors, the augmented Lagrangian method uses highest allowable production liquid rate before day 7000 and reduces the liquid rate when the first and second segments switch to BHP controls (shown in Fig. 4.43). It can be seen from Fig. 4.44(a) that the total liquid rate of P1 obtained from E300 optimizer are around the highest allowable value (3000 STB/D) during day 1 to day 1318 and then reduces gradually after day 1318. The oscillation of this total liquid rate happens during day 500 to day 1318, which is a result that the GRG method implemented in Eclipse 300 attempts to avoid violation of the liquid rate constraint (3000 STB/D). However, the



(a) (b) (c)

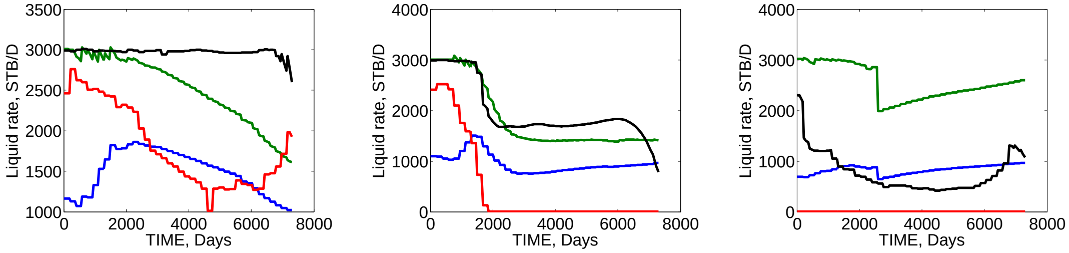
(d) (e) (f)

(g) (h) (i)

Figure 4.43: Liquid rate controls, BHP behaviors and well segment water-cut for three producers; Augmented Lagrangian method; Initial guess 2.

Eclipse 300 optimizer limits line search step size (see discussion in Chapter 1) in order to honor constraints, which may result in a small step size stopping the optimization prematurely. The same oscillation happens for producer P4, shown in Fig. 4.44(b). For the producer P12, which is located far from the inner fault and near the injectors I8 and I9, the augmented Lagrangian method uses much lower total production liquid rate than the GRG method.

Figs. 4.45 shows the estimated optimal rates for injection well segments. Fig. 4.45(a) and (b) show the results obtained with the Eclipse 300 optimizer. Using the first set of initial guesses, only three segments decrease the injection rate after control step 21; however, the injection rate controls are equal to the initial guesses for all the



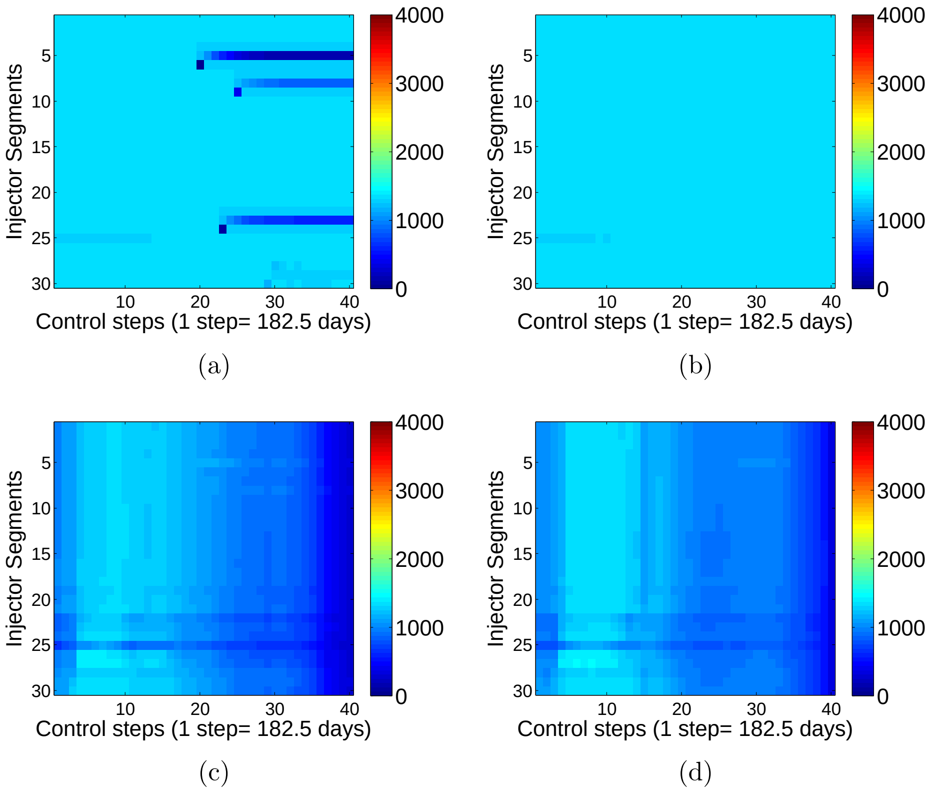
(a) P1 (b) P4 (c) P12

Figure 4.44: Comparison of the estimated optimal production liquid rates (Initial guess 2). Black: Total rate with the augmented Lagrangian method; Red: The 3rd segment rate with the augmented Lagrangian method; Green: Total rate with the E300 optimizer; Blue: The 3rd segment rate with the augmented Lagrangian method.

injection segments if using the second set of initial guesses. Fig. 4.45(c) and (d) show the results obtained with the augmented Lagrangian method. Although there is some small difference for the results obtained with different set of initial guesses, the control strategies are fairly similar and all the injection segments reduce water injection gradually from high injection rates to low injection rates.

Fig. 4.46 show the estimated optimal injection rate controls from the injector I2. In Fig. 4.46(a), the segment I2-3 is shut-in for the control step 20, which is not reasonable. Fig. 4.46(b) shows that all the three injection segments use the same control settings as their initial guesses, which indicates that the objective function is insensitive to these injection segment rates. In Fig. 4.46(c) and (d), the curves representing injection rates for the three segments of I2 are essentially identical, i.e. the three segments of I2 use the same rate controls.

The difference in NPV obtained from the two methods is also reflected in the cumulative oil and water production results shown in Figs. 4.47 and 4.48. The plot shows that the field cumulative oil production is about the same for both methods although the augmented Lagrangian method gives a slightly higher value. However, more water has been injected and produced with the final optimal controls in Eclipse 300. Because of this additional water, the optimization in Eclipse 300 results in a



(a)

(b)

(c)

(d)

Figure 4.45: The estimated optimal rates for injection well segments. (a) E300 optimizer, initial guess 1; (b) E300 optimizer, initial guess 2; (c) Augmented Lagrangian method, initial guess 1; (d) Augmented Lagrangian method, initial guess 2.

much lower value of NPV than is obtained with the augmented Lagrangian method.

4.3.2 Optimization with only bound constraints

For a maximization problem with only bound constraints, it is not necessary to update the Lagrangian multipliers or penalty parameter. The augmented Lagrangian method is then the gradient-projection trust-region method. This section is to compare the performance of the gradient-projection trust-region method and the ensemble-based optimization algorithm [8]. We use both the truncation and the log-transformation to deal with the bound constraints for EnOpt. Chen et al. [10] provided no information on how to initialize the stepsize for each iteration or how to quantify the correlation time length and the variance of control variables. In this

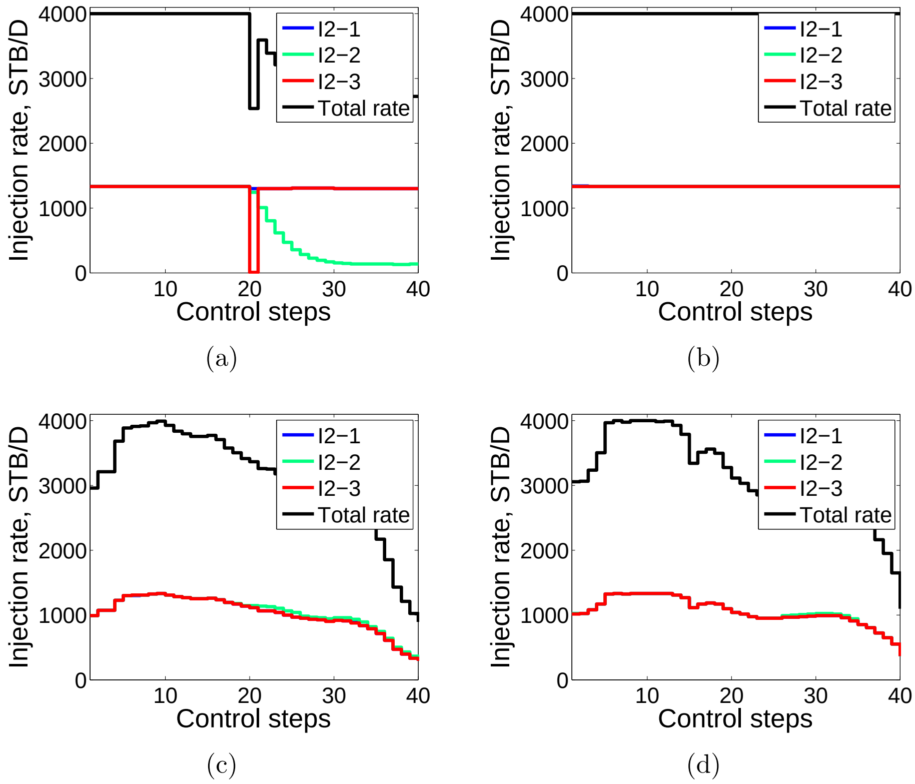


Figure 4.46: The estimated optimal rates for Injector 2. (a) E300 optimizer, initial guess 1; (b) E300 optimizer, initial guess 2; (c) Augmented Lagrangian method, initial guess 1; (d) Augmented Lagrangian method, initial guess 2.

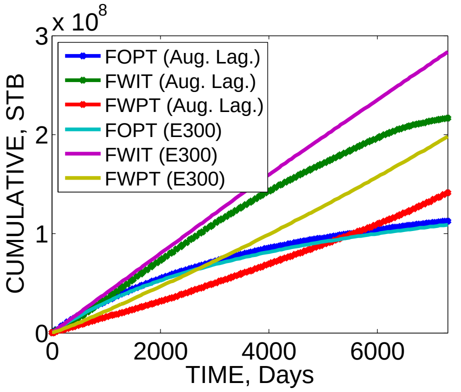


Figure 4.47: Comparison of field cumulatives; Initial guess 1.

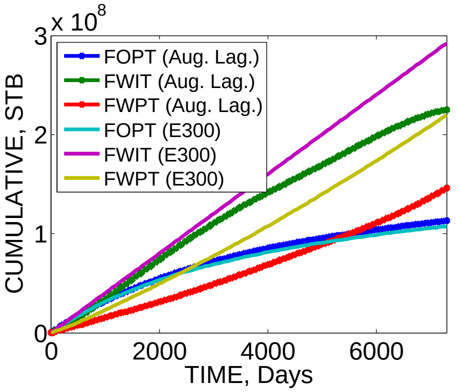


Figure 4.48: Comparison of field cumulatives; Initial guess 2.

example, the control settings for a well are assumed to be correlated over 5 years (10 control steps). The selection of the initial step size and the variance of controls used here are based on a private discussion with Mr. Sy Do, who is a PhD student in Petroleum Engineering at University of Tulsa and has done many experiments on implementing SPSA to this Brugge case. In the EnOpt algorithm using the log-transformation, the variance $\sigma^2 = 1$ and the correlation time length $T = 10$ are used to construct the covariance C_U; the initial step size is set to equal to 1.0. In the EnOpt algorithm using the truncation, the variance $\sigma^2 = 1$ and the correlation time length $T = 10$ are used to construct the covariance C_U; the initial step size is set equal to 1000.0.

Chen et al. [10] provided no information on how to initialize the stepsize α_ℓ^0 for each iteration or perhaps more importantly what to do if d_ℓ is not an uphill direction. Note because Eqs. C.73 and C.74 are based on a series of approximations, the gradient that appears in Eq. C.73 is only an approximation to the true gradient. Thus, there is no way to prove that d_ℓ given in Eq. C.73 is an uphill direction. When Eq. C.73 gives a downhill direction, cutting the stepsizes cannot be expected to finally arrive at a control vector which will increase the value the NPV function, J. Our experience is that Eq. C.73 sometimes gives a downhill direction.

The convergence performance of the gradient-projection trust-region method and the EnOpt algorithm with the same initial guess are summarized in Table 4.9 and Fig. 4.49. In this example, the gradient-projection trust-region method has the fastest convergence rate and highest final NPV compared to EnOpt optimization results. The gradient-projection trust-region method converged to the final NPV of \$$5.161 \times 10^9$ which is 1.8% higher than the NPV obtained with the EnOpt algorithm with truncation option. The 30 simulation runs in Table 4.9 required for the gradient-projection trust-region method based on an adjoint gradient consists of 15 forward reservoir simulation runs and 15 adjoint solutions for calculating the gradient of NPV. Although EnOpt, with truncation or log-transformation, is far slower than the gradient-based optimization in this example, they all converge to a similar NPV (about \$$5 \times 10^9$) within about 840 reservoir simulation runs.

Table 4.9: The performance of different optimization algorithms.

Algorithm	Simulations	Final NPV, $\times 10^9$\$
Adjoint Gradient	30	5.161
EnOpt (truncation)	840	5.068
EnOpt (log-trans)	717	4.983

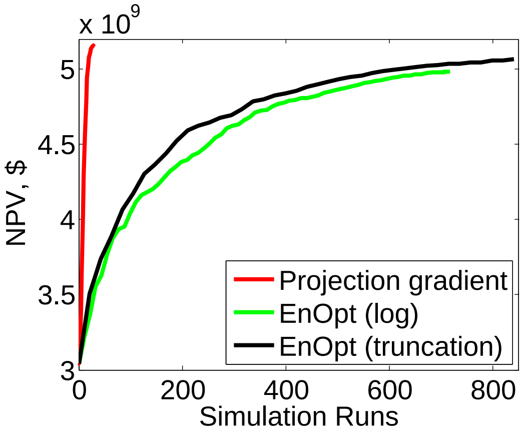
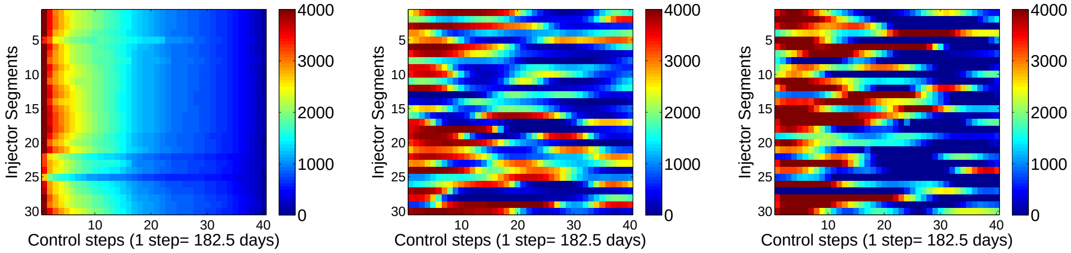


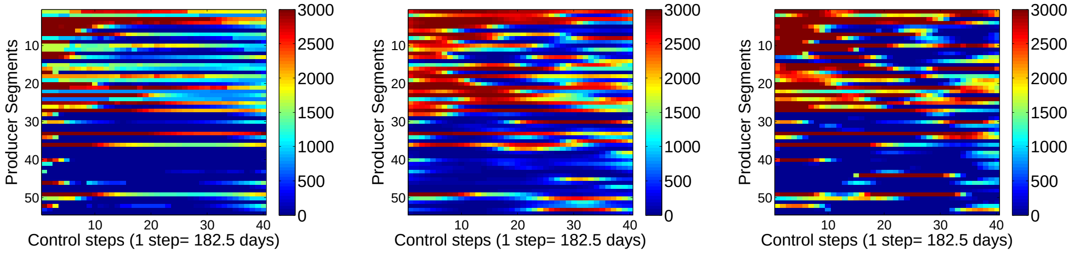
Figure 4.49: NPV versus the number of iterations, Example 3.

The final controls for the producers and injectors obtained by the augmented Lagrangian method and the EnOpt algorithm are shown in Figs. 4.50 and 4.51. Although we apply a smoother for EnOpt algorithm and it usually guarantees the final optimal controls to be smooth in time dimension, the figures show that the final controls obtained by the augmented Lagrangian method without using a smoother are even smoother than the final controls obtained from EnOpt. Another observation is that even through the final value of NPV obtained by the algorithms are not very different (Table 4.9), the estimates of the optimal controls (Figs. 4.50 and 4.51) are substantially different. This suggests that there may exist multiple sets of controls which can achieve essentially the same maximum value of NPV. The injection rate controls in Figs. 4.50(a) show all the injection segments use high injection rates in the beginning and then reduce gradually to the end of production. The controls in Fig. 4.51 show high liquid production rates for the first 26 production segments, which correspond to the first ten producers. From the structural map (Fig. 4.34), these producers are located at the edge of the main fault and are at the farthest locations from the injectors. The wells (the segments after 26) that are close to the injectors produce at much lower liquid rates.



(a) Adjoint-gradient (b) EnOpt (log-trans) (c) EnOpt (truncation)

Figure 4.50: The estimated optimal rates for injection well segments.

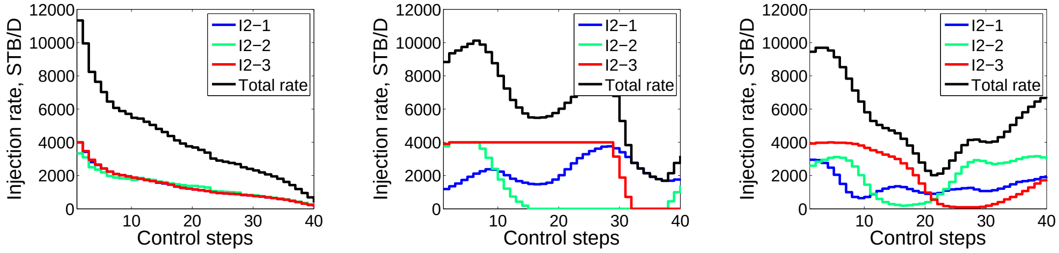


(a) Adjoint-gradient (b) EnOpt (log-trans) (c) EnOpt (truncation)

Figure 4.51: The estimated optimal rates for production well segments.

The estimates of the optimal total water injection rate and segment rates for Injector 2 as a function of time are shown in Fig. 4.52. Fig. 4.52(a) shows the three segment of Injector 2 use almost the same injection rate. The injection rate controls in Figs. 4.52(b) and Figs. 4.52(c) show the injection segments are allowed to inject water in waves, i.e. an injection segment inject alternatively high injection rate and low injection rate.

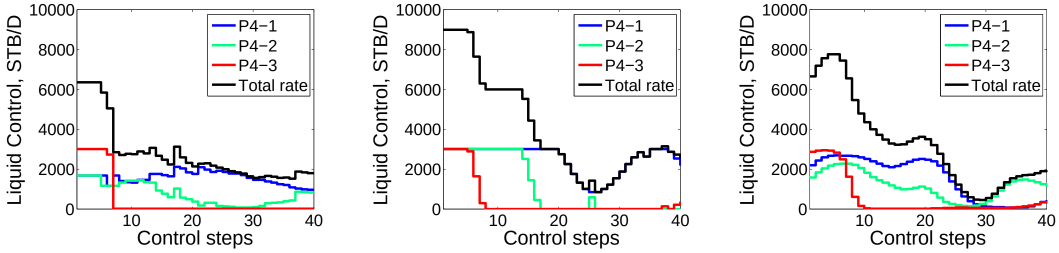
Fig. 4.53 shows the optimal liquid rate controls for the three segments of producer P4 from all the algorithms. We see that the total liquid rates before control step 15 estimated from EnOpt (truncation) and EnOpt (log-transformation) are



(a) Adjoint Gradient (b) EnOpt (truncation) (c) EnOpt (log-trans)

Figure 4.52: The estimated optimal well segment injection rates for Injector 2.

higher than that estimated from the augmented Lagrangian method. Another observation is that the controls in Fig. 4.53(c) are not able to reach the upper limit but the rate controls in Fig. 4.53(a) and (b) reach the upper bounds for a few control steps. Once a converted control variable in log-domain approaches to infinity, the change of control variable in the original domain is not sensitive to the change of converted control variable in the log-domain. As a result, the control variable in original domain is not able to take the value of upper or lower limit.



(a) Adjoint Gradient (b) EnOpt (truncation) (c) EnOpt (log-trans)

Figure 4.53: The estimated optimal well segment liquid production rates for Producer 4.

CHAPTER 5

ROBUST LONG-TERM AND SHORT-TERM OPTIMIZATION

In this chapter, we extend the augmented Lagrangian method to solve the robust production optimization problem in which we maximize the expectation of NPV. Here, we use N_e reservoir models to represent the uncertainty in the reservoir description. The expectation of NPV is thus represented as the average NPV over the N_e reservoir models, i.e., the robust production optimization problem is given by Eq. 1.8 which is repeated here as

$$\max \mathrm{E}[J_L(y,u,m)] = \max \frac{1}{N_e} \sum_{j=1}^{N_e} J_L(y_j,u,m_j), \tag{5.1}$$

subject to the constraints in Eqs. 1.5 through 1.7 applied for each reservoir model. Therefore, the objective function to be maximized is defined as the augmented Lagrangian function consisting of the average NPV over a set of geological realizations and all constraints except the bound constraints. To compute the gradient of the augmented Lagrangian function requires running the simulator and then solving for the adjoint gradient for each geological model. The augmented Lagrangian function is then maximized subject to bound constraints, following the algorithm similar to that outlined in section 2.4 for nominal production optimization. The outer-loop updates the Lagrangian multipliers or penalty parameter while the inner-loop maximizes the augmented Lagrangian function at its fixed Lagrangian multipliers and penalty parameter using a gradient-projection trust-region method. The following two sections describe the robust long-term optimization and the robust short-term after long-term optimization.

5.1 Robust Long-Term Optimization

To calculate the gradient of β_L in Eq. 2.12, we run the simulator with each geological realization m_j, $j = 1, \cdots, N_e$ and apply the adjoint method to compute the gradient of each individual augmented Lagrangian function, which can be formulated as

$$\begin{aligned}\beta_{L,j} = J_L(y_j, u, m_j) - N_e \Bigg[\sum_{i=1}^{n_e} [\lambda_{e,i,j} e_i(y_j, u, m_j) + \frac{s_{e,i}}{2\mu}(e_i(y_j, u, m_j))^2] \\ + \sum_{i=1}^{n_i} \psi_i(y_j, u, m_j) \Bigg],\end{aligned} \tag{5.2}$$

where $y_j = y(u_j, m_j)$ and ψ_i is defined in

$$\psi_i(y_j, u, m_j) = \begin{cases} -\dfrac{\mu}{2s_{c,i}} \lambda_{c,i,j}^2, \quad \text{if} \quad c_i(y_j, u, m_j) \le -\dfrac{\mu \lambda_{c,i,j}}{s_{c,i}}, \\ \lambda_{c,i,j} c_i(y_j, u, m_j) + \dfrac{s_{c,i}}{2\mu}(c_i(y_j, u, m_j))^2, \quad \text{otherwise.} \end{cases} \tag{5.3}$$

Note that the number of the Lagrange multipliers can be extremely large, as the λ's are required for each constraint and each geological model. The objective function, β_L, is the expectation of the augmented Lagrangian function, which can be approximated by

$$\beta_L = \frac{1}{N_e} \sum_{j=1}^{N_e} \beta_{L,j}, \tag{5.4}$$

so the gradient of β_L is

$$\nabla_u \beta_L = \frac{1}{N_e} \sum_{j=1}^{N_e} \nabla_u \beta_{L,j}. \tag{5.5}$$

In the augmented Lagrangian method, we usually need to update the Lagrange multipliers (λ's) or the penalty parameter (μ) a few times before the optimization converges. With fixed value of the Lagrangian multipliers and a fixed value of the penalty parameter, we maximize the augmented Lagrangian function in Eq. 5.4 subject to the bound constraints, i.e., we solve the problem

$$\begin{aligned} &\max\ \beta_L(u), \\ &\text{subject to } u_i^{\text{low}} \le u_i \le u_i^{\text{up}}, i = 1, 2, ..., n_b. \end{aligned} \tag{5.6}$$

The objective function β_L is first approximated by a quadratic model, in which the Hessian matrix is approximated using BFGS method. The quadratic model together with the bound constraints is maximized using the gradient-projection trust-region method. Once the inner-loop iteration converges, we update the Lagrange multipliers λ's or the penalty parameter μ depending on the violation of the constraints evaluated at the end of the inner-loop iteration. For robust production optimization, we define an average constraint violation parameter σ_{c_v} by

$$\sigma_{c_v} = \begin{cases} \sqrt{\dfrac{1}{n_v} \displaystyle\sum_{j=1}^{N_e} \Big[\sum_{i=1}^{n_e} s_{e,i} e_i^2(y_j^\ell, u^\ell, m_j) + \sum_{i=1}^{n_i} s_{c,i} (\max[0, c_i(y_j^\ell, u^\ell, m_j)])^2 \Big]}, & n_v > 0, \\ 0, & n_v = 0, \end{cases} \tag{5.7}$$

where n_v is the total number of violated constraints. Note that e_i and c_i are functions of each geological model m_j. If there is only one reservoir model or say $N_e = 1$, Eq. 5.7 can be rewritten as Eq. 2.14. If σ_{c_v} is less than a tolerance [12, 13], η^ℓ, which is also updated in the outer-loop, we update the Lagrange multipliers λ's and η without changing μ as

$$\lambda_{e,i,j}^{\ell+1} = \lambda_{e,i,j}^{\ell} + \frac{s_{e,i} e_i(y_j^\ell, u^\ell, m_j)}{\mu^\ell}, \quad i = 1, 2, ..., n_e, \tag{5.8}$$

$$\lambda_{c,i,j}^{\ell+1} = \max[0, \lambda_{c,i,j}^{\ell} + \frac{s_{c,i} c_i(y_j^\ell, u^\ell, m_j)}{\mu^\ell}], \quad i = 1, 2, ..., n_i, \tag{5.9}$$

for $j = 1, 2, ..., N_e$ and

$$\eta^{\ell+1} = \eta^\ell (\mu^\ell)^{0.9}, \tag{5.10}$$

where ℓ is the outer-loop iteration index. Similar to discussion in Chapter 2, we may choose to estimate the initial λ's as

$$\lambda_{e,i,j}^{0} = \frac{s_{e,i} e_i(y_j^0, u^0, m_j)}{\mu^0}, \quad i = 1, 2, ..., n_e, \tag{5.11}$$

and

$$\lambda_{c,i,j}^{0} = \max[0, \frac{s_{c,i} c_i(y_j^0, u^0, m_j)}{\mu^0}], \quad i = 1, 2, ..., n_i, \tag{5.12}$$

for $j = 1, 2, ..., N_e$ where $y_j^0 = y(u^0, m_j)$. This requires one simulation run for each m_j to obtain values for $e_i(y_j^0, u^0, m_j)$ and $c_i(y_j^0, u^0, m_j)$ with the initial guess of the control vector u^0.

If σ_{c_v} is larger than η^ℓ, we update μ without changing λ as

$$\mu^{\ell+1} = \tau \mu^\ell, \tag{5.13}$$

and

$$\eta^{\ell+1} = \eta^0 (\mu^{\ell+1})^{0.1}. \tag{5.14}$$

We use $\tau = 0.1$ and $\eta^0 = 0.1$ in the results presented here. Overall, once σ_{c_v} is less than 0.01 or $n_v/[N_e \times (n_e + n_i)] \leq 0.001$, we stop the optimization.

In the example sections of this chapter, we compare the robust long-term optimization results with those obtained with the reactive control method and those obtained with nominal optimization method. Only the two-phase system of oil and water is considered. For reactive control, the production wells are shut-in when the production of that well reaches the specified economic limit. In nominal optimization, the optimized controls are only guaranteed to honor the constraints for a single model; some constraints may be violated for other realizations of the vector of reservoir model parameters.

5.2 Robust Short-Term After Long-Term Optimization

When the number of control variables is large, there may be several sets of controls that give essentially the same optimal value of the objective function [45]. In this case, the optimization problem may have additional degrees of freedom which can be used to achieve a second objective, e.g., to optimize NPV over the short-term without compromising the long-term (life-cycle) NPV of production [45]. Letting u^* denote the vector of optimal controls obtained with robust long-term optimization, we approximate the expectation of the NPV at these life-cycle optimal controls by

$$\mathrm{E}[J_L(y^*, u^*, m)] = \frac{1}{N_e} \sum_{j=1}^{N_e} J_L(y_j^*, u^*, m_j), \tag{5.15}$$

where $y_j^* = y_j(u^*, m_j)$.

Following the long-term optimization, we maximize the expectation of the short-term NPV, where for a given m_j, the short-term NPV, $J_s(y_j, u, m_j)$ is defined by Eq. 1.3, i.e.

$$J_s(y_j, u, m_j) = \sum_{n=1}^{N_s} \left[\sum_{i=1}^{N_p} \left(r_o^n q_{o,i}^n - r_w^n q_{w,i}^n + r_g^n q_{g,i}^n \right) - \sum_{i=1}^{N_{wi}} r_{wi}^n q_{wi,i}^n \right] \frac{\Delta t^n}{(1+b)^{t^n/365}}, \tag{5.16}$$

with the total number of simulation time steps N_L replaced by N_s. N_s is the total number of simulation time steps for the short-term NPV and $N_s < N_L$. The short-term robust optimization problem is defined as

$$\max \mathrm{E}[J_s(y, u, m)] = \max \frac{1}{N_e} \sum_{j=1}^{N_e} J_s(y_j, u, m_j), \tag{5.17}$$

subject to the following constraints:

$$c_{L,j} = J_L(y_j^*, u^*, m_j) - J_L(y_j, u, m_j) \leq 0, \quad j = 1, 2, ..., N_e; \tag{5.18}$$

$$e_i(y_j, u, m_j) = 0, \quad i = 1, 2, ..., n_e, \quad j = 1, 2, ..., N_e; \tag{5.19}$$

$$c_i(y_j, u, m_j) \leq 0, \quad i = 1, 2, ..., n_i, \quad j = 1, 2, ..., N_e; \tag{5.20}$$

$$u_i^{\text{low}} \leq u_i \leq u_i^{\text{up}}, \quad i = 1, 2, ..., n_b, \quad j = 1, 2, ..., N_e. \tag{5.21}$$

Although the short-term NPV depends only on the controls up to time step N_s, the constraints in Eqs. 5.18 to 5.21 are functions of the controls up to simulation time step N_L. The inequality constraints in Eq. 5.18 indicate that we wish to maximize the short-term NPV without decreasing the long-term NPV.

Now we define the augmented Lagrangian function for the short-term optimization problem as

$$\begin{aligned} \beta_s(y, u, \lambda, \mu, m) =& \frac{1}{N_e} \sum_{j=1}^{N_e} [J_s(y_j, u, m_j)] - \sum_{j=1}^{N_e} \Bigg[\sum_{i=1}^{n_e} [\lambda_{e,i,j} e_i(y_j, u, m_j) \\ &+ \frac{s_{e,i}}{2\mu} (e_i(y_j, u, m_j))^2] + \sum_{i=1}^{n_i} \psi_i(y_j, u, m_j) + \psi_L(y_j, u, m_j) \Bigg], \end{aligned} \tag{5.22}$$

where ψ_i is defined in Eq. 5.3 and ψ_L is defined as

$$\psi_L(y_j, u, m_j) = \begin{cases} -\dfrac{\mu}{2s_{L,j}} (\lambda_{L,j})^2, & c_{L,j} < -\dfrac{\mu \lambda_{L,j}}{s_{L,j}}, \\ \lambda_{L,j} c_{L,j} + \dfrac{s_{L,j}}{2\mu} (c_{L,j})^2, & \text{otherwise.} \end{cases} \tag{5.23}$$

Similar to the scaling factor in Eq. 2.3, we choose the scaling factor for the long-term NPV constraint of the jth model as

$$s_{L,j} = \frac{1}{(J_L(y_j^*, u^*, m_j))^2}, \tag{5.24}$$

where $y_j^* = y_j(u^*, m_j)$. To perform the short-term robust optimization after long-term robust optimization, we maximize β_s in Eq. 5.22 subject to the bound constraints of Eq. 5.21. The optimization procedure is the same as the one discussed for life-cycle production optimization. The constraint violation parameter for short-term optimization is defined as

$$\sigma_{c_v} = \begin{cases} \sqrt{\frac{1}{n_v}\sum_{j=1}^{N_e}\left[\sum_{i=1}^{n_e} s_{e,i}e_i^2 + \sum_{i=1}^{n_i} s_{c,i}(\max[0,c_i])^2 + s_{L,j}(\max[0,c_{L,j}])^2\right]}, & n_v > 0, \\ 0, & n_v = 0, \end{cases} \tag{5.25}$$

where $e_i = e_i(y_j^\ell, u^\ell, m_j)$, $c_i = c_i(y_j^\ell, u^\ell, m_j)$ and $c_{L,j} = c_{L,j}(y_j^\ell, u^\ell, m_j)$. As we discussed in the previous section on robust long-term optimization, if σ_{c_v} is within the prescribed tolerance (η^ℓ), we update the Lagrange multipliers without changing μ. The $\lambda_{e,i,j}$, $\lambda_{c,i,j}$ and η are updated using Eqs. 5.8, 5.9 and 5.10, respectively. $\lambda_{L,j}$ is updated as

$$\lambda_{L,j}^{\ell+1} = \max[0, \lambda_{L,j}^{\ell} + \frac{s_{L,j}c_{L,j}(y_j^\ell, u^\ell, m_j)}{\mu^\ell}], \tag{5.26}$$

for $j = 1, 2, ..., N_e$. If σ_{c_v} is larger than the prescribed tolerance (η^ℓ), we update μ and η without changing λ using Eqs. 5.13 and 5.14, respectively.

In the examples, we consider robust optimization where we alternate life-cycle and short-term optimization. In practice, after each short-term optimization, we would use the optimal short-term controls to "produce the reservoir" for the length of the short-term control step and then do another long-term optimization step for the remaining life of the reservoir followed by another short-term optimization for the next short-term control step to determine the well controls for this short-term step. This procedure would be repeated throughout the economic life of the reservoir. In the examples considered here, we do not assign a true reservoir model but simply apply the alternating long-term and short-term optimization steps to see if it is possible to increase the expectation of short-term NPV while maintaining the value of the expected NPV obtained by life-cycle optimization. The NPV obtained by robust long-term alternating with short-term optimization (LASTOpt) are compared with those NPV obtained by robust sequential short-term optimization (SSOpt), NPV results obtained by life-cycle nominal optimization and NPV results obtained by reactive control. In robust SSOpt, we obtain optimal controls for each short-term sequentially in time. After we obtain the final optimal controls for the whole reservoir life, we run the simulator with all reservoir models to calculate the life-cycle NPV for each reservoir model.

When the nominal optimization is done on a single reservoir model, production from the other reservoir models is also calculated by running the simulator using the optimal well controls found by nominal optimization. By simulating reservoir production with the nominal well controls, one can quantify the maximum economic loss that can occur when using nominal optimization. When the reservoir production of the other models is simulated using the well controls from nominal optimization, constraints may be violated. To make a fair comparison with the other optimization procedures, a well is shut in if the water oil ratio (WOR) becomes high enough so that the cost of disposing of the produced water exceeds the revenue that can be obtained by selling the produced oil. To compare the results of nominal optimization with the results from robust optimization, we simulate production from all other reservoir models using the optimal well controls estimated by nominal optimization. In these simulation runs, we force all constraints to be satisfied and shut in a well if its water oil ratio (WOR) exceeds the bound specified for the WOR.

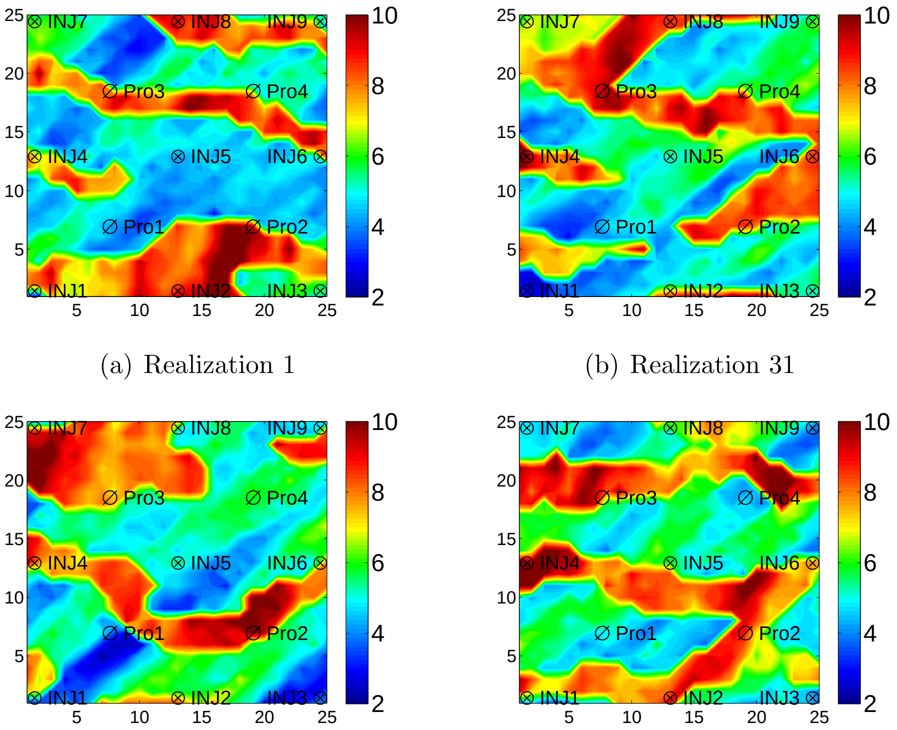
5.3 Example 1: A 2D Fluvial Reservoir

In this subsection, we consider the optimization of waterflooding of a two-dimensional synthetic reservoir with a simulation model defined on a $25 \times 25 \times 1$ grid. For the robust optimization, we generate 100 geological realizations to represent the uncertainty in the reservoir description. The reservoir models are generated using a combination of multi-point and two-point statistics. For each realization, we first generate the channelized structure (all realizations are conditioned to observed facies at the wells) using SNESIM [28]. Within each facies (the channel facies and the nonchannel facies), we generate a log-permeability distribution using a spherical covariance function with the parameters given in Table 5.1. In this table, r_1 and r_2 are, respectively, the correlation length in the principal direction and in the direction orthogonal to the principal direction; α is the angle between the principal direction and the x-direction measured in the clockwise direction. The mean of $\ln k$ in the channel facies is 8.5 and the mean of $\ln k$ in the non-channel facies is 5. The variance for log-permeability is 1. The porosity for the reservoir is homogeneous and fixed equal to 0.2. Fig. 5.1 shows four realizations of the log-permeability distribution. The mean model of log-permeability based on 100 realizations is shown in Fig. 5.2. The reservoir has two phases, oil and water, and the water-oil relative permeability curves are shown in Fig. 5.3. The reservoir is under waterflooding with 4 five-spot patterns as shown in Figs. 5.1 and 5.2. From the mean model of Fig. 5.2, we can deduce that the injectors INJ2, INJ4, INJ6 and INJ8 and the producers Pro2 and Pro3 are located in the high permeability region. As a result, water breakthrough would tend to occur earlier in Pro2 and Pro3 than in other two producers. The related reservoir properties are listed in Table 5.2. We use an in-house simulator (CLASS) as the forward model.

Table 5.1: Geological parameters

$\ln k$	Channel facies	nonchannel facies
mean	8.5	5
variance	1.0	1.0
α	45	45
r_1 (ft)	8000.0	5600.0
r_2 (ft)	2000.0	1000.0

The oil and water prices are set to $r_o = \$50$ /STB, $r_w = \$5.56$ /STB, $b = 10\%$



(a) Realization 1

(b) Realization 31

(c) Realization 51

(d) Realization 71

Figure 5.1: Log-permeability distribution of four realizations.

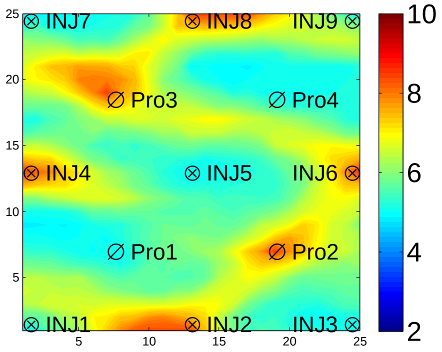


Figure 5.2: Mean of Log-Permeability distribution.

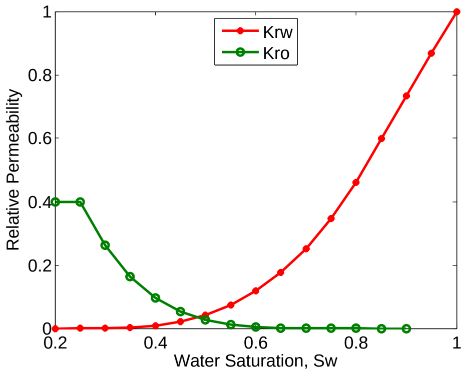


Figure 5.3: The water and oil relative permeability curve.

and $r_{w,inj} = \$0$. We impose a water/oil ratio (WOR) constraint of WOR ≤ 9.0 for each producer. The WOR constraint is a nonlinear function of the control variable through the reservoir simulator. The anticipated total reservoir life is 1800 days and the control step size is set to 60 days so we have 30 control steps. The total number of controls is $13 \times 30 = 390$. All the producers are at BHP control with an upper

Table 5.2: Reservoir properties

Grid block size	200 ft
Thickness	20 ft
ρ_{osc}	56 lb/ft^3
ρ_{wsc}	62.4 lb/ft^3
μ_o	1.4 cp at 3601.5 psi
μ_w	0.5 cp
B_o	1.22 RB/STB at 3601.5 psi
B_w	1.0042 RB/STB
Rock Compressibility	3×10^{-5} psi^{-1}
Top Depth	4800 ft
Porosity	0.2

bound of 6000 psi and a lower bound of 2000 psi. The injector is under injection rate control with an upper bound of 1500 STB/D and a lower bound of 0.

We compare results from five strategies: reactive control, nominal optimization, robust long-term optimization, robust LASTOpt and robust SSOpt. For reactive control, the maximum allowable injection rate and the minimum allowable producer BHP are used and a producer is shut in once it reaches the economic limit (WOR=9.0). For nominal optimization, we maximize NPV for the expected reservoir life using the specified geological realization with the bound and WOR constraints. The optimal controls from nominal optimization are then applied to other realizations and we shut-in the producers when the WOR reaches 9.0 for comparison purposes. For robust long-term optimization, we maximize the average NPV for an ensemble of 100 geological models under the WOR constraint (WOR $\leq$ 9.0) for each producer and the bound constraints on the control variables. For robust LASTOpt, the objective is to maximize the expectation of the short-term NPV with the bound, WOR and the long-term NPV constraints in Eq. 5.18, which ensures that the long-term NPV does not decrease during short-term optimization. For short-term production optimization, we divide the whole reservoir life into five short-term steps so that each short term has six control steps. For robust SSOpt, we sequentially optimize the expectation of the short-term NPV on these five short-term steps. The initial guess for all the producer BHP controls are equal to their lower bound of 2000 psi. The initial guess for all the injection well rate controls is 1000 STB/D.

5.3.1 Results of nominal optimization

To choose a suitable initial value for the penalty parameter, we performed nominal optimization on reservoir realization 1 (Fig. 5.1) with different initial values of the penalty parameter ranging from 10^{-1} to 10^{-8}. As there is only one constraint type and all the constraint values are the same, we fix the scaling factor equal to $1/9^2$ and only test the effects of different penalty parameter and Lagrange multipliers. Table 5.3 summarizes the optimization results after applying different initial penalty parameter and Lagrangian multipliers. The choice of $\mu^0 = 10^{-7}$ and $\lambda^0_{c,i} = 0$ obtains the highest NPV of \$2.01×10^8 with 29 simulation runs. The choice with $\lambda^0_{c,i} = 0$ and $\mu^0 = 10^{-6}$ also achieves better results than other choices in terms of computational cost. It seems that in this example, $\lambda^0_{c,i} = 0$ gives better performance than $\lambda^0_{c,i} = \max[0, \frac{s_{c,i}c^0_i}{\mu^0}]$, which is different than the performance in Example 2 with inequality constraints in Chapter 4.

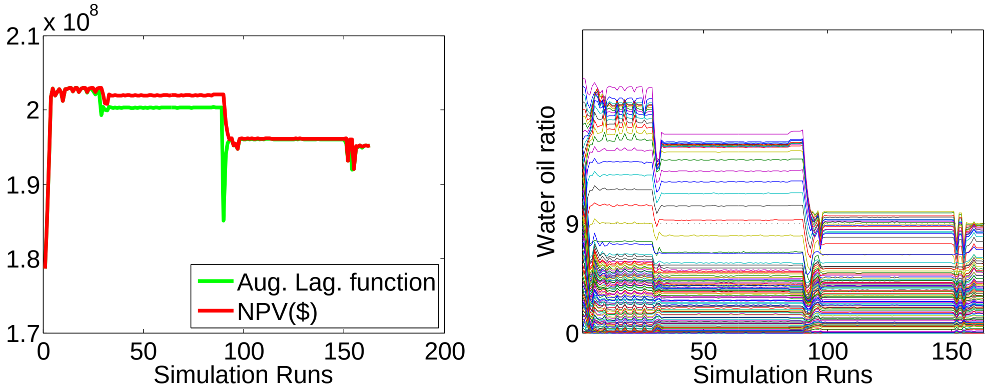
We provide below the optimization results obtained from choice of (a) $\mu^0 = 10^{-1}$ and $\lambda^0_{c,i} = \max[0, \frac{s_{c,i}c^0_i}{\mu^0}]$; (b) $\mu^0 = 10^{-4}$ and $\lambda^0_{c,i} = \max[0, \frac{s_{c,i}c^0_i}{\mu^0}]$; (c) $\mu^0 = 10^{-5}$ and $\lambda^0_{c,i} = \max[0, \frac{s_{c,i}c^0_i}{\mu^0}]$; (d) $\mu^0 = 10^{-7}$ and $\lambda^0_{c,i} = 0$. Figs. 5.4 through 5.7 show how the objective function and WOR constraint change as a function of the number of reservoir simulations for the four different values of the initial penalty parameter and Lagrangian multipliers. The left panels (Fig. 5.4(a), Fig. 5.5(a),Fig. 5.6(a), Fig. 5.7(a)) show how the augmented Lagrangian function and NPV change during optimization and the right panels (Fig. 5.4(b), Fig. 5.5(b),Fig. 5.6(b), Fig. 5.7(b)) show how the WOR of each producer at each control step changes as a function of the number of simulation runs during optimization. Each curve in the right panels (Fig. 5.4(b), Fig. 5.5(b),Fig. 5.6(b), Fig. 5.7(b)) represents the WOR of each producer

Table 5.3: The performance of different parameters for the nominal optimization with realization No. 1. N_μ: Number of μ updated; N_λ: Number of λ updated.

μ^0	$\lambda_{c,i}^0$	Simu-lations	Final NPV, $	Outer-Loop	N_μ	N_λ	Converge
10^{-1}	0	60	1.95×10^8	9	8	0	Yes
10^{-1}	$\max[0, \frac{s_{c,i}c_i^0}{\mu^0}]$	163	1.95×10^8	9	8	0	Yes
10^{-2}	0	60	1.96×10^8	5	4	0	Yes
10^{-2}	$\max[0, \frac{s_{c,i}c_i^0}{\mu^0}]$	76	1.93×10^8	9	7	1	Yes
10^{-3}	0	106	1.96×10^8	7	6	0	Yes
10^{-3}	$\max[0, \frac{s_{c,i}c_i^0}{\mu^0}]$	54	1.95×10^8	8	6	1	Yes
10^{-4}	0	32	1.95×10^8	6	5	0	Yes
10^{-4}	$\max[0, \frac{s_{c,i}c_i^0}{\mu^0}]$	50	1.93×10^8	6	5	0	Yes
10^{-5}	0	104	1.95×10^8	5	4	0	Yes
10^{-5}	$\max[0, \frac{s_{c,i}c_i^0}{\mu^0}]$	30	2.00×10^8	5	4	0	Yes
10^{-6}	0	20	1.99×10^8	3	2	0	Yes
10^{-6}	$\max[0, \frac{s_{c,i}c_i^0}{\mu^0}]$	19	1.91×10^8	1	0	0	Yes
10^{-7}	0	29	2.01×10^8	5	3	1	Yes
10^{-7}	$\max[0, \frac{s_{c,i}c_i^0}{\mu^0}]$	43	1.96×10^8	2	0	1	Yes
10^{-8}	0	70	1.99×10^8	3	1	1	Yes
10^{-8}	$\max[0, \frac{s_{c,i}c_i^0}{\mu^0}]$	19	1.67×10^8	8	7	0	Yes

at each control step during optimization. From Fig. 5.4, we can see that it takes 163 reservoir simulations to obtain convergence. In this case, there are 9 outer-loop iterations and the penalty parameter is reduced to 10^{-9} at the end, which means the initial penalty parameter is too large and it takes many outer-loop iterations to get suitable weights on the augmented Lagrangian terms. For all the cases, the WOR constraints are all satisfied at convergence. This example has only inequality constraints. If the augmented Lagrangian function is less than the corresponding NPV, it suggests that some constraints are violated. It can be clearly seen in Figs. 5.4 through 5.7 that if the augmented Lagrangian function (green curve) is below the NPV (red curve), some WOR curves are above 9.0 at the corresponding simulation number.

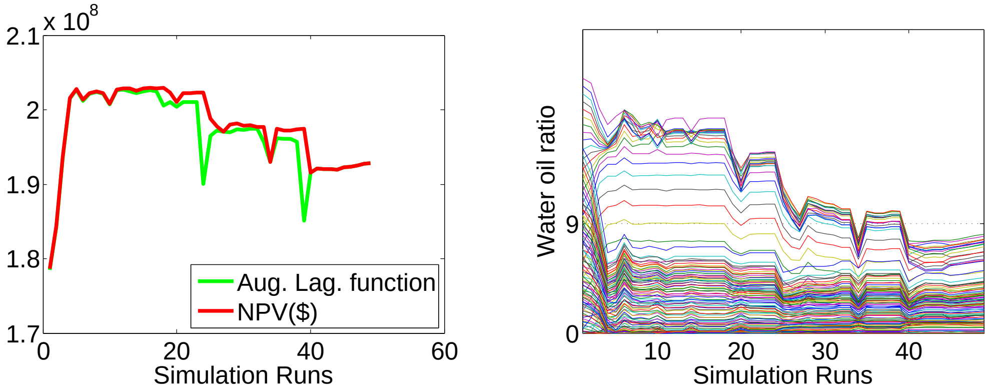
Fig. 5.8 and Fig. 5.9 show the optimal injection rates and producer BHP



(a) Objective function

(b) WOR constraints

Figure 5.4: The optimization results for realization 1 with $\mu^0 = 10^{-1}$ and $\lambda^0_{c,i} = \max[0, \frac{s_{c,i}c^0_i}{\mu^0}]$.



(a) Objective function

(b) WOR constraints

Figure 5.5: The optimization results for realization 1 with $\mu^0 = 10^{-4}$ and $\lambda^0_{c,i} = \max[0, \frac{s_{c,i}c^0_i}{\mu^0}]$.

controls obtained with nominal optimization based on the first realization of the log-permeability field. Although we select different initial values for the penalty parameter and Lagrangian multipliers, the optimal controls look fairly similar in all cases. In reservoir model 1 (Fig. 5.1), there are two high permeability channels: one connects INJ1, INJ2 and Pro2 and another one connects Pro3, Pro4 and INJ6. The optimal injection rate from INJ2 is kept close to zero throughout the reservoir life in order to reduce water production from Pro2. INJ1 also has a very low injection rate from control step 7 to 16. For each case, the water injection rate of INJ4 is

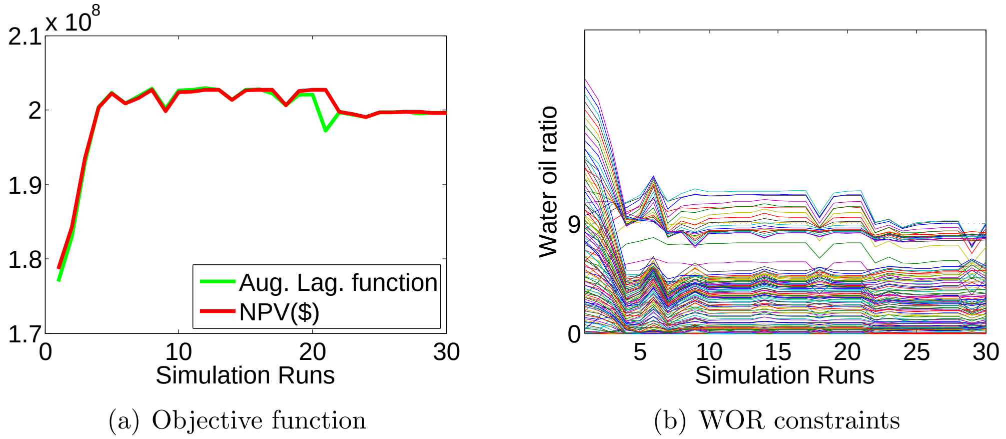


Figure 5.6: The optimization results for realization 1 with $\mu^0 = 10^{-5}$ and $\lambda_{c,i}^0 = \max[0, \frac{s_{c,i}c_i^0}{\mu^0}]$.

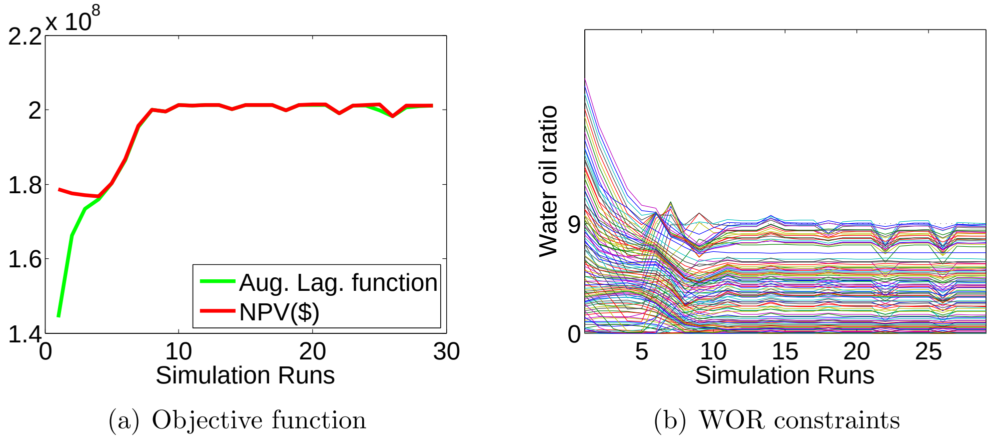
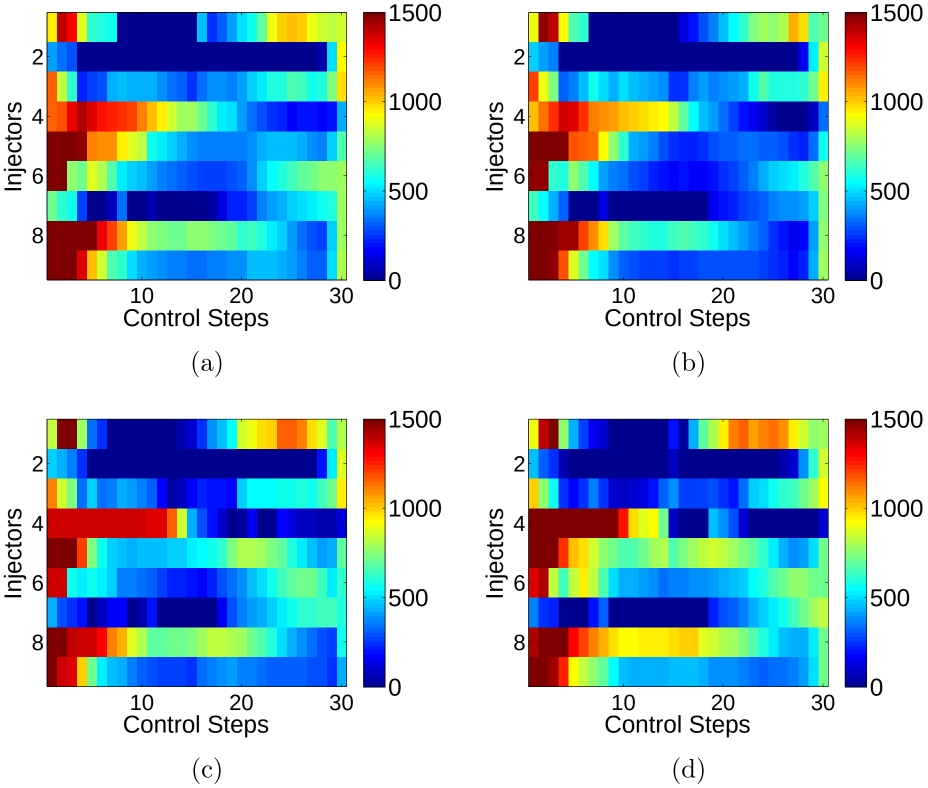


Figure 5.7: The optimization results for realization 1 with $\mu^0 = 10^{-7}$ and $\lambda_{c,i}^0 = 0$.

over 800 STB/D for the first 15 control steps, as INJ4 provides pressure support for Pro1 and Pro3 and is not connected to any producer by a high permeability channel. Similar to the role of INJ4, INJ8 injects more than 1200 STB/D for 6 control steps and then continues to inject at a significant rate for 18 additional control steps. In Fig. 5.9, all four cases have a similar control strategy for Pro1, i.e. Pro1 uses high BHP control between control step 8 and step 16. Note that INJ4 has a high rate and the permeability connectivity is fairly high between Pro1 and INJ4, so Pro1 has to use high BHP control to reduce water production; as can be seen from Fig. 5.10, the remaining oil saturation is low between Pro1 and INJ4. Pro2 is under lower bound

BHP control for a short period and then changes to a fairly high BHP control, which decreases the liquid production rate and delays the time of water breakthrough. Pro3 and Pro4 operate under the lower bound BHP controls for all control steps.



(a)

(b)

(c)

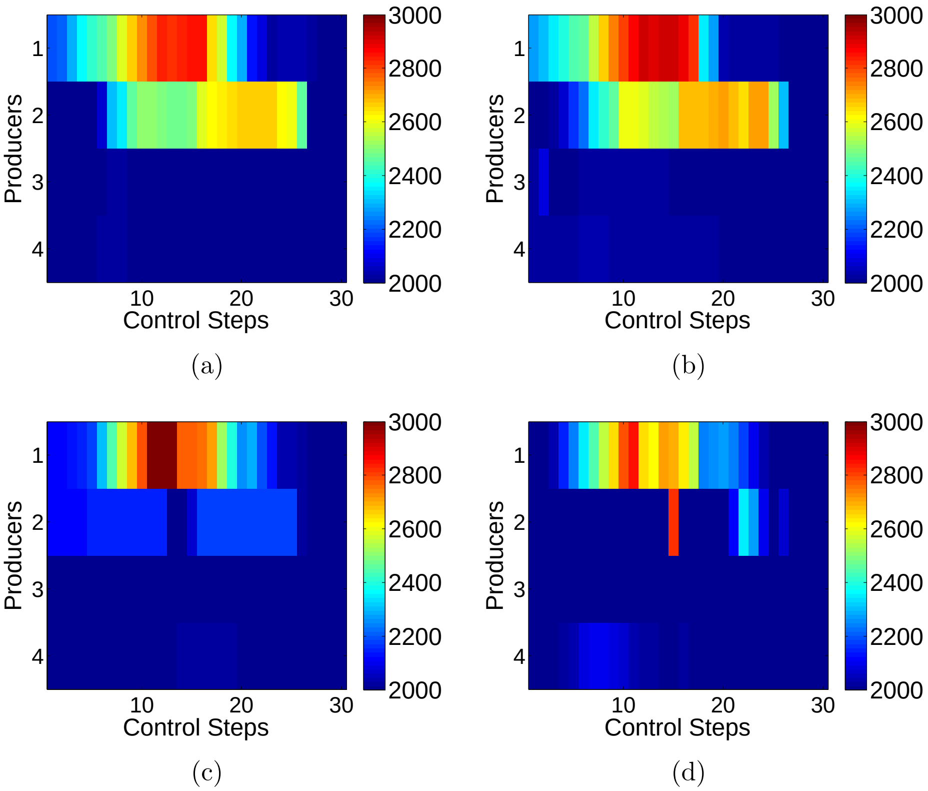
(d)

Figure 5.8: The optimal injection rate controls for realization No. 1. (a) $\mu^0 = 10^{-1}$, $\lambda^0_{c,i} = \max[0, \frac{s_{c,i}c^0_i}{\mu^0}]$; (b) $\mu^0 = 10^{-4}$, $\lambda^0_{c,i} = \max[0, \frac{s_{c,i}c^0_i}{\mu^0}]$; (c) $\mu^0 = 10^{-5}$, $\lambda^0_{c,i} = \max[0, \frac{s_{c,i}c^0_i}{\mu^0}]$; (d) $\mu^0 = 10^{-7}$, $\lambda^0_{c,i} = 0$.

Fig. 5.10 shows the comparison of remaining oil saturation at the day 1800 obtained for the four cases. The blue color represents high water saturation and the red color represents high oil saturation. It can be seen that the optimal controls result in a good areal sweep efficiency although there are a few sporadic spots not swept by the injected water. The remaining oil saturation in Fig. 5.10(c) and Fig. 5.10(d) looks lower than that in Fig. 5.10(a) and Fig. 5.10(b).

5.3.2 Results of robust long-term Optimization

For robust long-term optimization, we maximize the expectation of NPV based

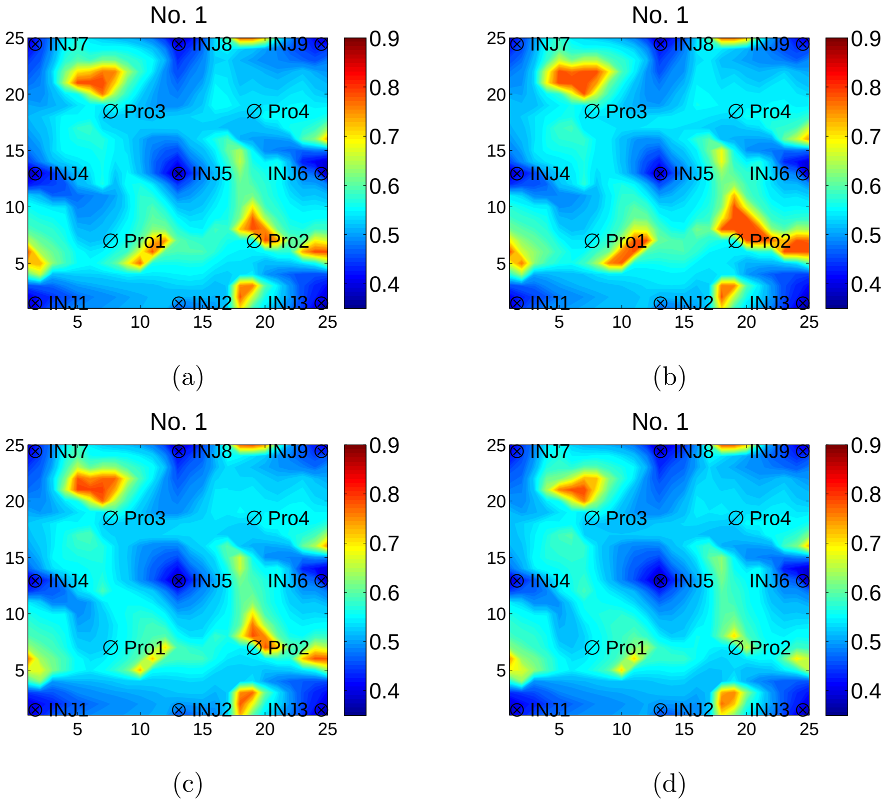


(a) (b)

(c) (d)

Figure 5.9: The optimal producer BHP controls for realization No. 1. (a) $\mu^0 = 10^{-1}$, $\lambda^0_{c,i} = \max[0, \frac{s_{c,i}c^0_i}{\mu^0}]$; (b) $\mu^0 = 10^{-4}$, $\lambda^0_{c,i} = \max[0, \frac{s_{c,i}c^0_i}{\mu^0}]$; (c) $\mu^0 = 10^{-5}$, $\lambda^0_{c,i} = \max[0, \frac{s_{c,i}c^0_i}{\mu^0}]$; (d) $\mu^0 = 10^{-7}$, $\lambda^0_{c,i} = 0$.

on 100 realizations with WOR constraints (WOR ≤ 9.0) and bound constraints. The total number of inequality constraints is $N_e \times n_i = 100 \times 120 = 12000$. From the optimization results listed in Table 5.3, we may choose either $\mu^0 = 10^{-5}$, $\lambda^0_{c,i} = \max[0, \frac{s_{c,i}c^0_i}{\mu^0}]$ or $\mu^0 = 10^{-7}$, $\lambda^0_{c,i} = 0$ for robust optimization because these initial parameters give fairly good nominal optimization results. However, to check this choice, we did perform robust life-cycle optimization with different initial μ^0 and $\lambda^0_{c,i}$ and summarized the final results in Table 5.4. As we optimize the robust cases for 100 geological models, the number of simulations is significantly larger than the nominal optimization. From Table 5.4, the final expectations of NPV are very close for each trial and they seem not sensitive to the choice of the initial penalty parameter. However, we still notice that the number of simulation runs in the cases, $\mu^0 = 10^{-4}$



(a)

(b)

(c)

(d)

Figure 5.10: Remaining oil for realization No. 1 saturation after optimization. (a) $\mu^0 = 10^{-1}$, $\lambda^0_{c,i} = \max[0, \frac{s_{c,i}c_i^0}{\mu^0}]$; (b) $\mu^0 = 10^{-4}$, $\lambda^0_{c,i} = \max[0, \frac{s_{c,i}c_i^0}{\mu^0}]$; (c) $\mu^0 = 10^{-5}$, $\lambda^0_{c,i} = \max[0, \frac{s_{c,i}c_i^0}{\mu^0}]$; (d) $\mu^0 = 10^{-7}$, $\lambda^0_{c,i} = 0$.

and $\mu^0 = 10^{-5}$, is less than the other two cases, $\mu^0 = 10^{-6}$ and $\mu^0 = 10^{-7}$.

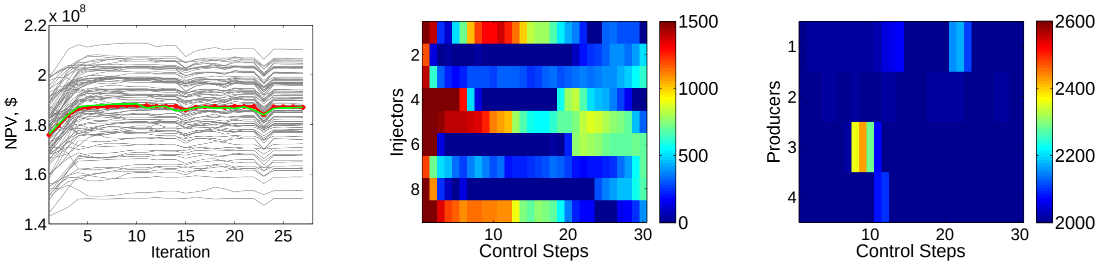
We provide the robust life-cycle optimization results obtained with $\mu^0 = 10^{-5}$ and $\lambda^0_{c,i} = \max[0, \frac{s_{c,i}c_i^0}{\mu^0}]$ as the initial values of the penalty parameter and the Lagrange multipliers. Fig. 5.11 shows results of the robust long-term optimization. Fig. 5.11(a) shows the NPV and augmented Lagrangian function as a function of iteration number for each realization. Each iteration involves 100 simulation runs as we have 100 geological models. The red curve is the expectation of NPV. The green curve is the augmented Lagrangian function. The gray curves are the NPVs obtained from different reservoir realizations. There are five outer-loop iterations and a total of 27 inner-loop iterations. In the first outer-loop (including 9 inner-loop iterations), the expectation of NPV increases 6.3% from \$1.76 × 10^8 to over \$1.87 × 10^8. In the final

Table 5.4: The performance of different parameters for the robust life-cycle optimization with 100 geological realizations. N_μ: Number of μ updated; N_λ: Number of λ updated.

μ^0	$\lambda^0_{c,i}$	Simulations	Final NPV, \$	Outer-Loop	N_μ	N_λ	Converge
10^{-4}	0	2800	1.86×10^8	6	5	0	Yes
10^{-4}	$\max[0, \frac{s_{c,i}c^0_i}{\mu^0}]$	2400	1.86×10^8	5	4	0	Yes
10^{-5}	0	3000	1.87×10^8	5	4	0	Yes
10^{-5}	$\max[0, \frac{s_{c,i}c^0_i}{\mu^0}]$	2700	1.87×10^8	4	3	0	Yes
10^{-6}	0	3100	1.87×10^8	4	3	0	Yes
10^{-6}	$\max[0, \frac{s_{c,i}c^0_i}{\mu^0}]$	4800	1.87×10^8	5	3	1	Yes
10^{-7}	0	3300	1.86×10^8	5	3	1	Yes
10^{-7}	$\max[0, \frac{s_{c,i}c^0_i}{\mu^0}]$	3800	1.86×10^8	2	1	0	Yes

outer-loop, the augmented Lagrangian function is less than the expectation of NPV because some WOR constraints (WOR ≤ 9) are slightly violated at some control steps; see Fig. 5.12. However, the total number of violated constraints is less than 0.1% of the total number of constraints so that the optimization converges. Figs. 5.11(b) and (c) show the optimal injection rate and the producer BHP controls, which are quite different from those obtained with the nominal optimization. The optimal injection rates indicate that we need to inject over 1200 STB/D for INJ5 until control step 9 as INJ5 is located at the center and it offers pressure support and displace water towards all the producers; see Fig. 5.2. In addition, INJ1, INJ4 and INJ9 have higher injection rates during early time than all other injectors except INJ5.

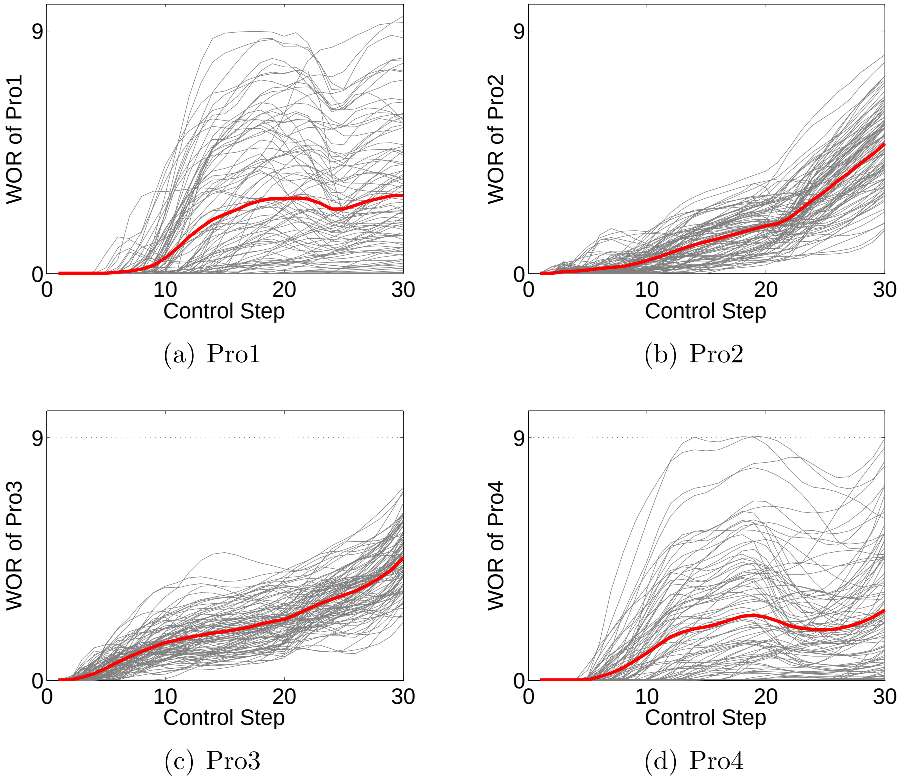
Fig. 5.12 shows the WOR of the 4 producers as a function of the control step obtained using the final optimal controls from the robust long-term optimization. The red curves represent the mean of the ensemble of WOR values. Only a few realizations slightly violated WOR ≤ 9.0 constraints at a few control steps. From the results of Fig. 5.12, we can see that the water breakthrough time for Pro1 and Pro4 ranges from the fifth control step to the final control step. None of the producers is shut-in



(a) (b) (c)

Figure 5.11: Robust long-term optimization with $\mu^0 = 10^{-5}$ and $\lambda_{c,i}^0 = \max[0, \frac{s_{c,i}c_i^0}{\mu^0}]$: (a) Long-term NPV versus iteration numbers (1 iteration=100 simulation runs); red curve: mean of NPV; green curve: augmented Lagrangian function; gray curves: ensemble; (b) Optimized injection rate controls; (c) Optimized producer BHP controls.

at the end of the optimization and this indicates that robust optimization tries to reduce water injection while keeping the producers open in order to satisfy the water oil ratio constraint.



(a) Pro1 (b) Pro2

(c) Pro3 (d) Pro4

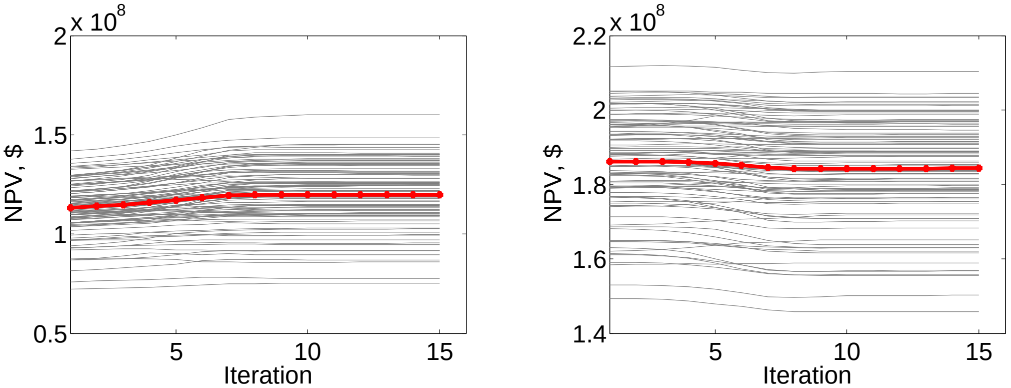
Figure 5.12: The final WOR curves for robust long-term optimization with $\mu^0 = 10^{-5}$ and $\lambda_{c,i}^0 = \max[0, \frac{s_{c,i}c_i^0}{\mu^0}]$.

5.3.3 Results of one robust short-term after long-term optimization

For robust short-term after long-term optimization, we first consider the first-short term optimization step to demonstrate the optimization results. The first short-term optimization step consists of the first six control steps. In doing the robust short-term optimization, the long-term result is used as an inequality constraint; see Eq. 5.18. We use the optimal controls from the robust long-term optimization as our initial guess.

Fig. 5.13 shows the results of the first robust short-term optimization. In Fig. 5.13(a), the red curve is the expectation of the short-term NPV. The gray curves are the short-term NPV for different reservoir realizations. Although not all individual NPVs increase with iteration, the expectation of the short-term NPV increases from $\$1.13 \times 10^8$ to $\$1.19 \times 10^8$, i.e., the short-term optimization results in 5.3% short-term NPV increase on average. In Fig. 5.13(b), the gray curves are the long-term NPV for different reservoir realizations and the red curve is the expectation of the long-term NPV. As shown in Fig. 5.13(b), the mean of long-term NPV remains almost constant during the iterations, which indicates that we have sufficient degrees of freedom to increase the short-term NPV without sacrificing the previously optimized life-cycle NPV.

Fig. 5.14(a) shows the estimated optimal injection rates of the 9 injectors at the end of the robust short-term optimization step. Compared to the optimal water injection rate from life-cycle optimization (Fig. 5.11(b)), all injectors have a higher injection rate during the first six control steps. Fig. 5.14(b) shows the BHP controls of producers after the first short-term optimization step and all the producers are under the lower bound BHP control during the first six control steps. Compared to the robust life-cycle optimization results(Fig. 5.11(c)), the BHP controls with the robust short-term optimization are very similar. Fig. 5.15 shows the final WOR for each producer obtained from the robust short-term optimization. From the results of

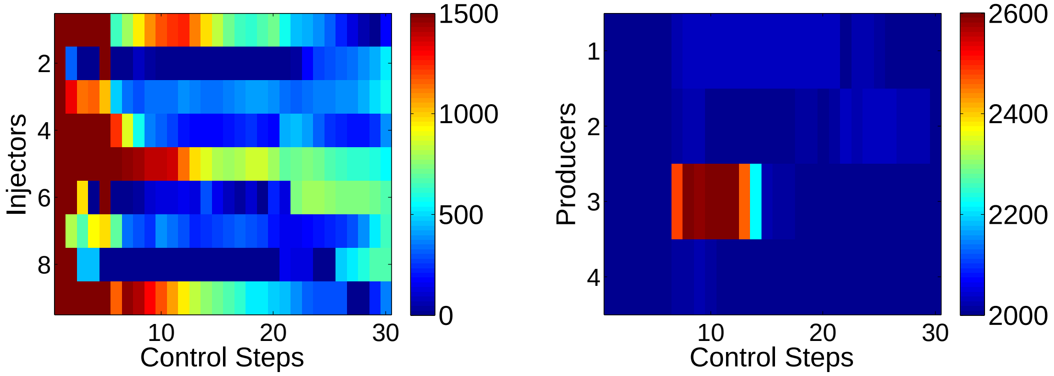


(a)

(b)

Figure 5.13: A single robust short-term after long-term optimization with $\mu^0 = 10^{-5}$ and $\lambda^0_{c,i} = \max[0, \frac{s_{c,i}c^0_i}{\mu^0}]$: (a) Short-term NPV versus iteration numbers (1 iteration=100 simulation runs); red curve: mean of short-term NPV; gray curves: ensemble; (b) Long-term NPV versus iteration numbers.

Fig. 5.15, we can see that one or two realizations slightly violate the WOR constraint at some control steps. All realizations predict similar water breakthrough times for Pro2 and Pro3. However, there is a large spread in the breakthrough time for Pro1 and Pro4.



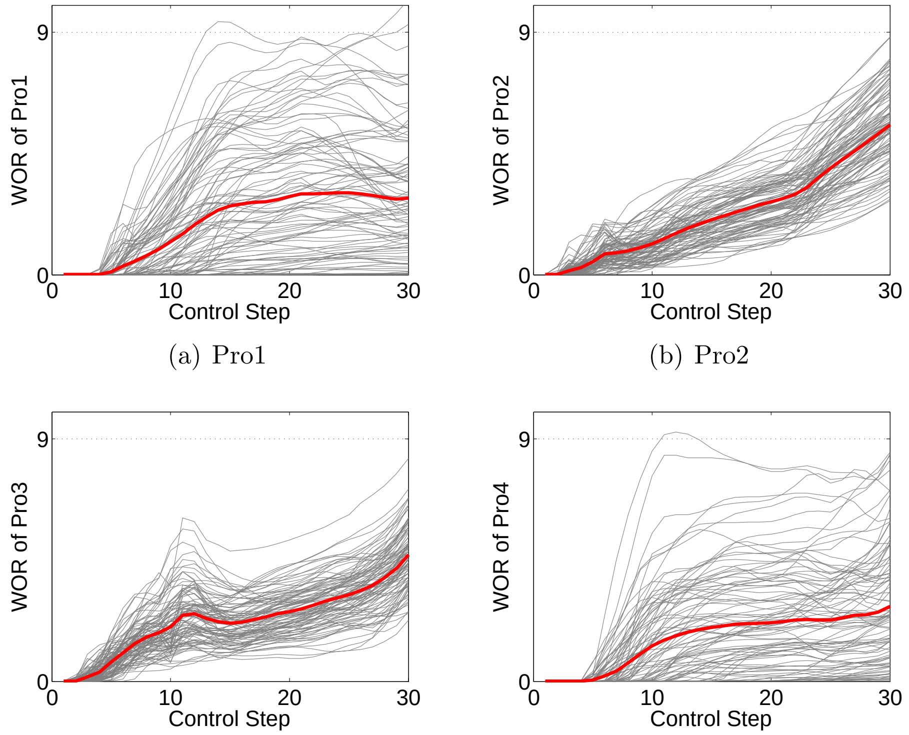
(a) Injection rate controls

(b) Producer BHP controls

Figure 5.14: Optimal controls after one robust short-term optimization.

5.3.4 Comparison of optimization strategies

Table 5.5 compares statistics for NPV results obtained from reactive control strategy, nominal optimization, robust long-term optimization, a single robust



(a) Pro1

(b) Pro2

(c) Pro3

(d) Pro4

Figure 5.15: The final WOR curves after one robust short-term optimization.

short-term after long-term optimization, robust LASTOpt and robust SSOpt. For reactive control, the maximum allowable injection rate and the minimum allowable producer BHP are used and a producer is shut in once it reaches the economic limit (WOR=9.0). For comparison purposes, we shut-in a producer once its WOR reaches 9.0 when we apply the nominal optimal controls to all other the reservoir realizations. In this synthetic case, the reactive control strategy obtains an expectation of NPV equal to \$1.52 $\times 10^8$, which is the minimum value among all the optimization strategies. For each nominal optimization, we have an expectation of NPV. For example, nominal optimization with realization No. 1 achieves an expectation of NPV equal to \$1.75 $\times 10^8$ and nominal optimization with realization No. 7 achieves an expectation of NPV equal to \$1.77 $\times 10^8$. Although the expectation of NPV obtained from nominal optimization with these two realizations are fairly close, the standard deviation obtained from realization No. 7 is much higher than that obtained from realization

No. 1. Robust long-term optimization obtains the highest expected NPV, which is 3.3% higher than the average of nominal optimization. A single robust short-term after long-term optimization obtains 0.5% lower expected NPV than robust long-term optimization does, because the short-term optimization aims at increasing the short-term NPV while maintaining the long-term NPV close to the long-term optimization results. However, robust long-term alternating with short-term optimization yields the same expectation of NPV as robust long-term optimization, which means we maintain the expected long-term NPV after we optimize through the five short-term steps. For the STD comparison, robust LASTOpt obtains a 7.6% higher STD than robust long-term optimization. Robust SSOpt does not yield an expectation of NPV as high as robust long-term optimization. Another important observation is that reactive control and nominal optimization result in the largest standard deviation in NPV, which means that making decisions based on results obtained with these optimization scenarios involves greater uncertainty and greater financial risk.

Table 5.5: Comparison of statistics of NPV ($\times 10^8$\$)

Case	Mean	STD	Minimum	Maximum
Reactive control	1.52	0.186	1.08	1.87
Nominal with realization No. 1	1.75	0.123	1.43	1.99
Nominal with realization No. 7	1.77	0.158	1.34	2.19
Robust long-term	1.87	0.130	1.49	2.12
A single robust short-term after long-term	1.85	0.136	1.46	2.10
Robust LASTOpt	1.87	0.141	1.47	2.12
Robust SSOpt	1.82	0.144	1.42	2.10

Fig. 5.16 shows the cumulative distribution function (CDF) of NPV (based on the 100 reservoir models) for reactive control (black curve), robust life-cycle optimization (red curve), a single robust short-term after long-term optimization (yellow curve), robust LASTOpt (blue curve) and robust SSOpt (green curve), nominal optimization (gray curves). Note that we have 20 CDF curves for nominal optimization.

Each CDF curve for nominal optimization is obtained by estimating optimal controls by maximizing NPV based on a single reservoir model and then these optimal controls are applied to the other 99 reservoir models. Robust long-term optimization and robust LASTOpt yield higher expected NPVs than are obtained with reactive control, nominal optimization and robust SSOpt.

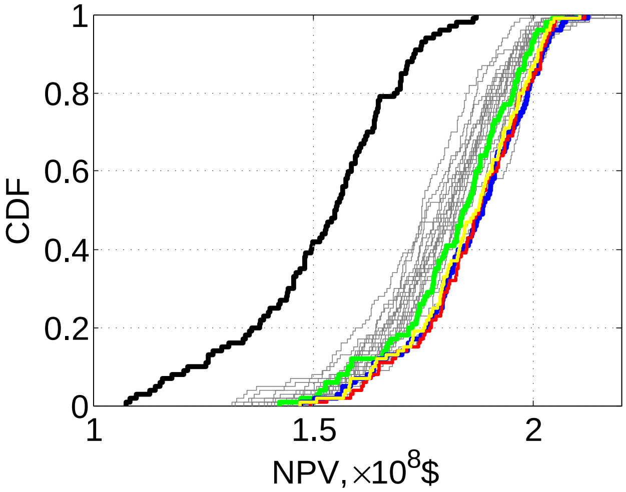
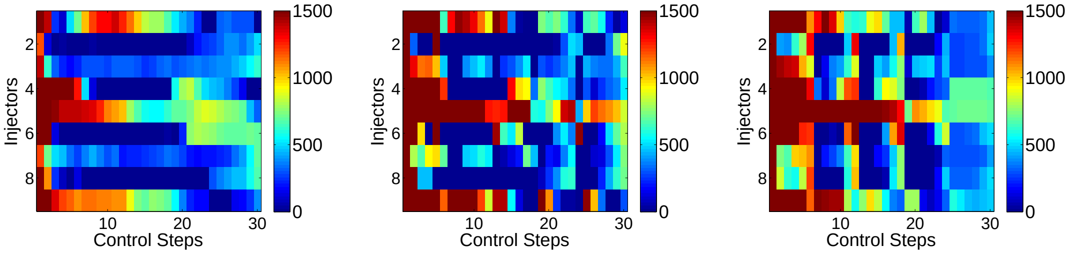


Figure 5.16: Cumulative density function of NPV with different strategies. Black: Reactive control; Red: Robust life-cycle optimization; Yellow: A single robust short-term after long-term optimization; Blue: Robust LASTOpt; Green: Robust SSOpt; Gray: Nominal optimization.

Fig. 5.17 compares the optimal injection rates from robust long-term optimization, robust LASTOpt and robust SSOpt. Overall, we see that robust LASTOpt and robust SSOpt yield higher estimates of the optimal water injection rates, especially for the first short-term step. Fig. 5.18 compares the optimal BHP controls from these three different robust optimization procedures. From Fig. 5.18(b), the BHP controls are always set to the lower limit in robust long-term alternating with short-term optimization (LASTOpt), which is different from the optimal BHP controls obtained from one robust short-term after long-term optimization (Fig. 5.14(b)).

Figs. 5.19 through 5.21 compare the oil saturation distribution of three realizations and average oil saturation of 100 realizations at day 360 for robust long-term

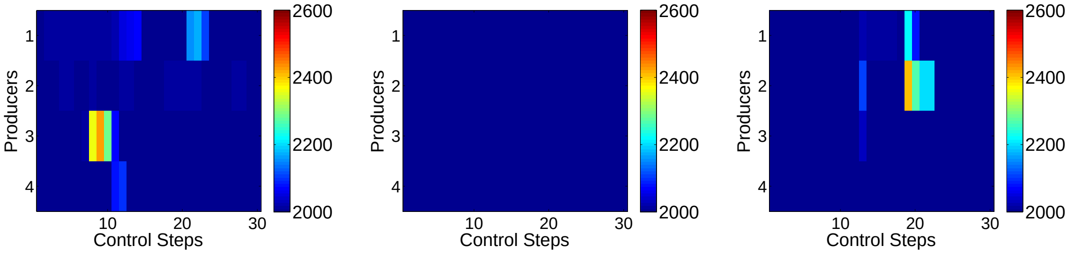


(a) Long-term opt.

(b) LASTOpt.

(c) SSOpt

Figure 5.17: The optimal injection rates from different robust strategies.



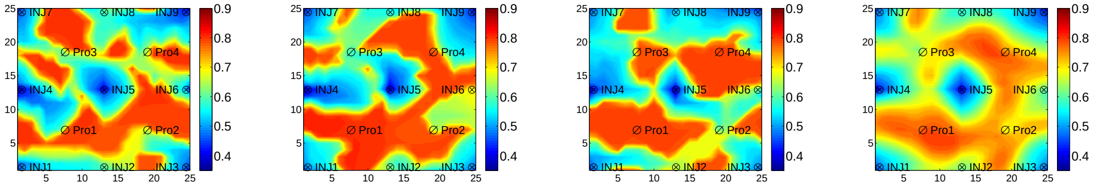
(a) Long-term opt.

(b) LASTOpt.

(c) SSOpt

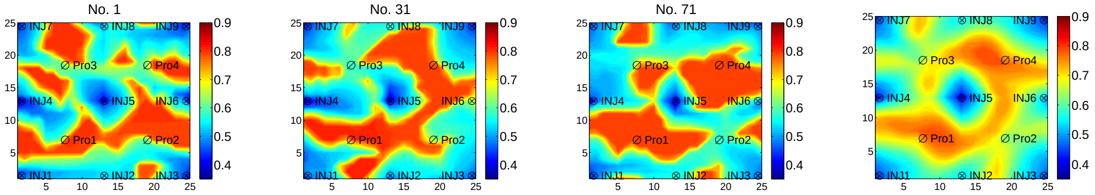
Figure 5.18: The optimal BHP controls from different robust strategies.

optimization, robust LASTOpt and robust SSOpt. Each short term step has six control steps. The blue color represents high water saturation and red represents high oil saturation. From Figs. 5.20 and 5.21, it can be seen that robust LASTOpt and robust SSOpt result in less remaining oil in the reservoir after the first short-term period (360 days) than that when only robust life-cycle optimization applied (Fig. 5.19), which indicates that short-term optimization yields a better sweep efficiency in the first 360 days (or first six control steps). Comparing Figs. 5.22, 5.23 and 5.24, the final oil saturation at day 1800 for all of these three robust optimization strategies are quite similar.



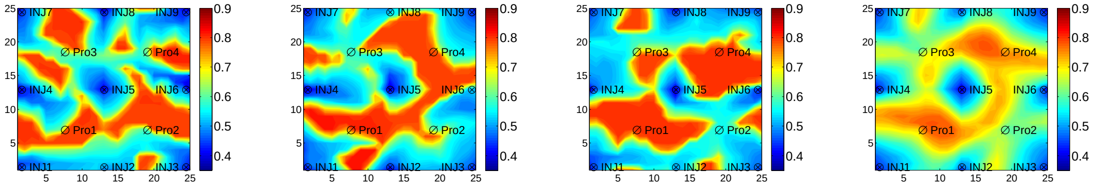
(a) Realization 11 (b) Realization 31 (c) Realization 71 (d) Mean

Figure 5.19: The oil saturation at day 360 obtained with applying optimal controls from robust long-term optimization.



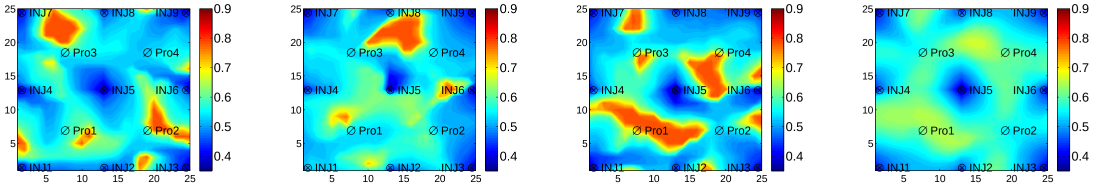
(a) Realization 1 (b) Realization 31 (c) Realization 71 (d) Mean

Figure 5.20: The oil saturation at day 360 obtained with applying optimal controls from robust LASTOpt.



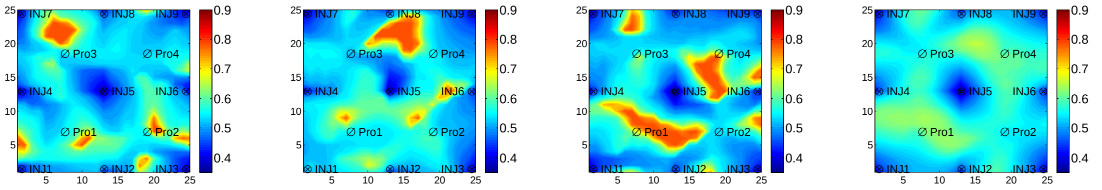
(a) Realization 1 (b) Realization 31 (c) Realization 71 (d) Mean

Figure 5.21: The oil saturation at day 360 obtained with applying optimal controls from robust SSOpt.



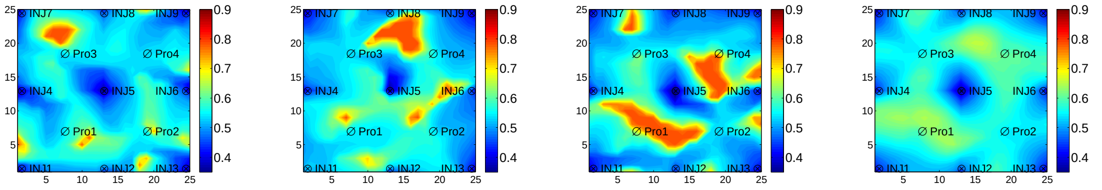
(a) Realization 1 (b) Realization 31 (c) Realization 71 (d) Mean

Figure 5.22: The oil saturation at day 1800 obtained with applying optimal controls from robust long-term optimization.



(a) Realization 1 (b) Realization 31 (c) Realization 71 (d) Mean

Figure 5.23: The oil saturation at day 1800 with applying optimal controls from robust LASTOpt.



(a) Realization 1 (b) Realization 31 (c) Realization 71 (d) Mean

Figure 5.24: The oil saturation at day 1800 with applying optimal controls from robust SSOpt.

5.4 Example 2: Brugge Test Case

The reservoir description of the Brugge reservoir model is the same as the example used in Chapter 3. Note TNO generated different realizations with different geostatistical modeling methods. For example, some realizations of properties in the first two layers were created with object-based modeling, i.e. channel objects in a shale background (Fig. 5.25) whereas other realizations were created with sequential indicator simulation (Fig. 5.26). We do not have the true reservoir model, which is known only by TNO. We consider optimizing the expectation of NPV based on a subset of the given ensemble of reservoir models.

Here we consider a total production life of 20 years and we run the first 10-year production schedule given by TNO with the initial ensemble of reservoir models generated by TNO to obtain the primary simulation variables at year 10. Our objective here is to discuss the results of robust optimization for the second 10 years of

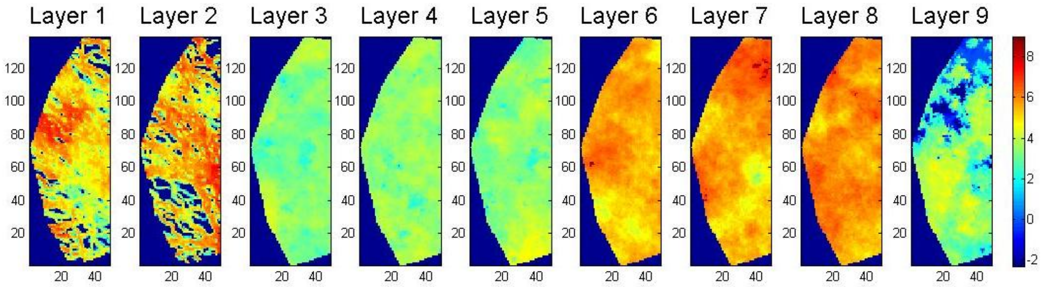


Figure 5.25: $\ln k$ at x direction of Realization No. 34.

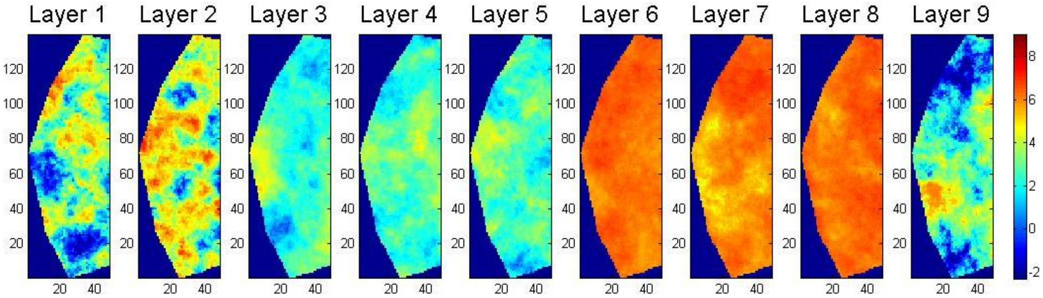
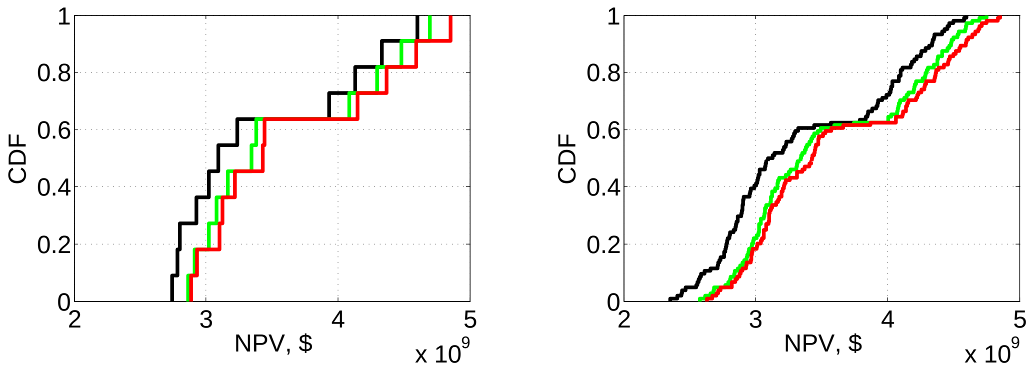


Figure 5.26: $\ln k$ at x direction of Realization No. 16.

production. For the second 10 years of production, each well is equipped with two or three inflow control valves that allow the operator to control the liquid flow rate for each production segment and the water injection rate for each injection segment. Here a well segment rate refers to the rate through one ICV. There are a total of 54 production segments and 30 injection segments. The upper bound constraint on liquid rate is 3000 STB/D for each production segment and 4000 STB/D for each injection segment. The lower bound constraint on rates for all the segments are 0. The control step size is 1/2 year and the number of control variables is 1680. Following the original description in Peters et al. [35], we set the BHP constraint to 725.2 psi for producers and 2611 psi for injectors. The oil and water prices are set to $r_o =$ \$80/STB, $r_w =$ \$5/STB, $r_{w,inj} =$ \$5/STB and annual discount rate $b = 10\%$. For the initial guess, we use 700 STB/D for each production segment and 1333.3 STB/D for each injection segment. We divide the second 10-year production period into four short-term steps, each of which has five control steps. As the computational cost for robust optimization is very expensive, we choose 11 corresponding reservoir models corresponding to P1, P10, P20,..., P90, P99 of the cumulative distribution of NPVs obtained with the initial guess of controls. After we estimate the optimal controls, we rerun the controls for these 11 reservoir models in order to plot the cumulative distribution of NPVs, which is shown in Fig. 5.27(a). We also provide the CDF of NPV based on all 104 realizations (Fig. 5.27(b)) based on these final optimal controls. The black curve is the CDF of NPV obtained with reactive control; the green curve is obtained from robust SSOpt; and the red curve is from robust long-term optimization. Fig. 5.27(b) indicates the expectation of NPV obtained from robust long-term optimization on 104 reservoir models is $\$3.63\times10^9$ which is 2% higher than the expected value obtained from robust SSOpt. However, the standard deviation of NPV obtained from robust long-term optimization based on 104 reservoir models is 4% higher than from robust SSOpt. In this Brugge case, robust short-term after long-term optimization does not improve the short-term NPV so that we do not provide the CDF of NPV from robust short-term after long-term optimization. The possible reason that we cannot increase short-term NPV is that many segment rate controls in the first short term step are converted to BHP controls during simulation, which means the segment rate controls have already reached their true upper limits determined by the corresponding BHP constraints, and then there are not enough degrees of freedom remaining to increase the short-term NPV without sacrificing the long-term NPV.

Table 5.6 summarizes the statistics of life-cycle NPV results based on 11 realizations. Robust life-cycle optimization yields the highest expected NPV ($\$3.65\times10^9$). However, for the Brugge case, the standard deviation of NPV for each strategy is about the same. Table 5.7 shows the statistics of the first short-term NPV obtained with running the estimated optimal controls from robust life-cycle optimization and robust SSOpt. Comparing the expectation of the first short-term NPV (shown in Table 5.7), we found the expected NPV from robust SSOpt is only 1% higher than



(a) CDF based on 11 realizations

(b) CDF based on all 104 realizations

Figure 5.27: Cumulative density function of NPV. Black: reactive control; Green: robust SSOpt; Red: robust long-term optimization.

from robust long-term optimization. This indicates that there is little room to improve the first short-term NPV based on the optimal controls obtained from life-cycle optimization. However, robust long-term optimization achieves a higher life-cycle NPV.

Table 5.6: Comparison of statistics of life-cycle NPV ($\times 10^9$\$) based on 11 reservoir models

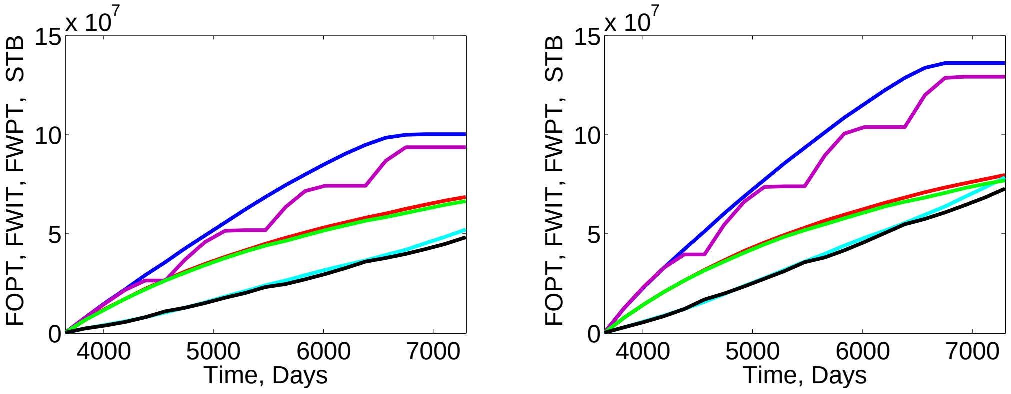
Case	Mean	STD	Minimum	Maximum
Reactive control	3.39	0.66	2.35	4.58
Robust life-cycle optimization	3.65	0.71	2.89	4.85
Robust SSOpt	3.56	0.68	2.86	4.70

Table 5.7: Comparison of statistics of first short-term NPV ($\times 10^9$\$) based on 11 reservoir models

Case	Mean	STD	Minimum	Maximum
Robust life-cycle optimization	1.93	0.50	1.47	2.74
Robust SSOpt	1.95	0.50	1.48	2.76

Fig. 5.28 shows the comparison of field cumulative oil production, water production and water injection with robust long-term optimization and robust sequential

short-term optimization. Day 3649 in Fig. 5.28 corresponds to the starting day of year 11. Fig. 5.28(a) shows field cumulative production or injection based on the realization No. 16, which corresponds to P30 of the cumulative distribution of NPVs; Fig. 5.28(b) shows the average field cumulative production or water injection over all the 104 realizations. The final NPVs obtained with realization No. 16 are, respectively, \$$3.22\times10^9$ for robust long-term optimization and \$$3.17\times10^9$ for robust sequential short-term optimization. It can be clearly seen that for the first short term (day 0 - day 912.5 or corresponding to day 3649 - day 4561.5 in Fig. 5.28(a) and (b)), robust sequential short-term optimization leads to less water injection (purple curve) than robust long-term optimization (blue curve) but achieves almost the same cumulative oil production and cumulative water production. As a result, for the first short-term step, robust sequential short-term optimization obtains a higher NPV than robust long-term optimization. However, at the end of the production life, robust long-term optimization results in higher cumulative oil production (red curve) than robust sequential short-term optimization and the oil revenue compensates the cost of the additional water injected and produced in the robust long-term result.

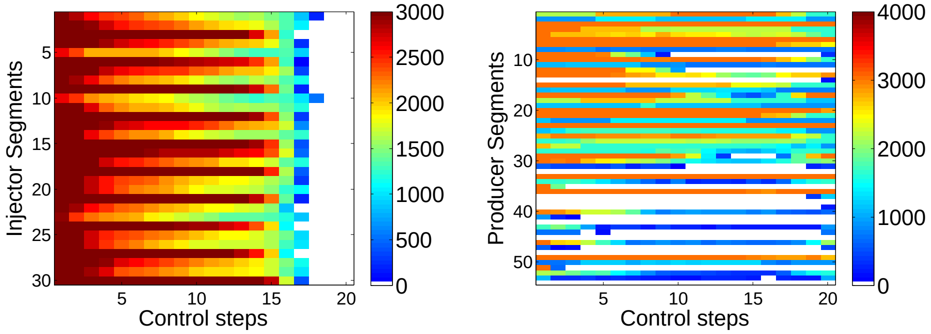


(a) Realization No. 16

(b) Average over 104 realizations

Figure 5.28: Field cumulative oil production, water production and water injection with optimal controls. Blue: field water injection (life-cycle optimization); Purple: field water injection (sequential short-term); Red: field oil production (life-cycle optimization); Green: field oil production (sequential short-term); Light blue: field water production (life-cycle optimization); Black: field water production (sequential short-term).

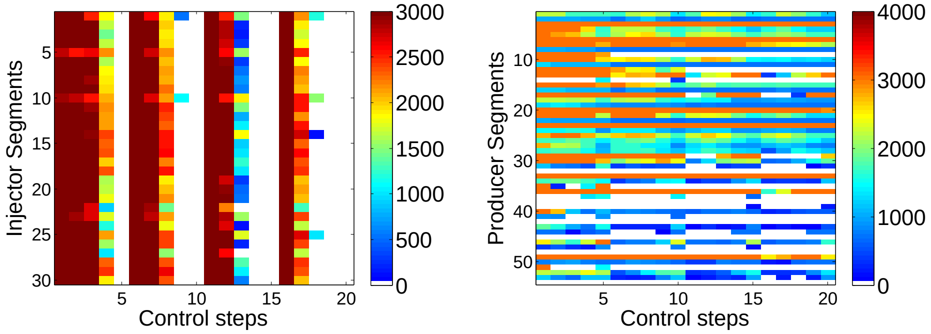
Fig. 5.29 and Fig. 5.30 compare the optimal controls from robust long-term optimization and robust sequential short-term optimization. White color represents a zero segment rate. There is a significant difference between the two sets of injection rates. The optimal injection rates from robust SSOpt require alternating periods of high injection rates with zero rates, whereas with robust life-cycle optimization, the injection rate for each segment is equal or close to its upper bound at early control steps and then gradually decreases until it becomes zero at the last few control steps. Both strategies show some "Bang-Bang" control behavior [50] on the injection rate control. As we use injection rate and liquid production rate as well controls for this waterflooding example with only upper and lower bound for the control variables, the optimization problem may satisfy the "Bang-Bang" control conditions.



(a) Injection segment rate

(b) Producer segment rate

Figure 5.29: Optimal controls from robust long-term optimization.



(a) Injection segment rates

(b) Production segment rates

Figure 5.30: Optimal controls from robust SSOpt.

CHAPTER 6

DISCUSSION AND CONCLUSIONS

The main contribution of this work was the development of practical optimization methods that can efficiently deal with large-scale production optimization problems with bound, linear and nonlinear constraints. We tried to solve three challenging problems arising in the field of production optimization: production optimization with a single reservoir model, production optimization with uncertain description of reservoir models and dual-objective production optimization (i.e. maximizing both the short-term NPV and the long-term NPV.

Following the idea from the LANCELOT method [13, 31], we implemented the augmented Lagrangian method to deal with equality and inequality constraints and the gradient-projection method to deal with the bound constraints. The augmented Lagrangian function incorporates the equality and inequality constraints with Lagrangian multipliers and penalty parameters into the objective function, NPV. At each outer-loop iteration, we update either the Lagrangian multipliers or penalty parameter depending on constraint violation. To maximize the augmented Lagrangian function at its fixed Lagrangian multipliers and penalty parameter, we approximate the function using a quadratic model, in which the Hessian matrix is approximated using the BFGS method. This forms the inner-loop iteration. The bound constraints are enforced using the gradient-projection method during the optimization of the approximate quadratic function. The major advantages are that it requires only one backward adjoint run for the gradient calculation and the constraints are satisfied at convergence. In LANCELOT, the gradient of each constraint term in the augmented Lagrangian function needs to be specified for its input deck, which is not practical for problems with many constraints as each gradient of the specified term requires one backward adjoint run for gradient calculation. Unlike LANCELOT, we calculate the gradient of the augmented Lagrangian function with only one backward adjoint run, which only requires to run one forward simulation and save the Jacobian matrix evaluated at the primary variables.

Although the augmented Lagrangian method offers a nice way to incorporate all equality and inequality constraints into the objective function, some constraints may be insensitive to the change in controls during optimization due to the scale difference between the constraints and between constraints and NPV. We apply a scaling factor for each constraint. The initial values of the Lagrangian multipliers and penalty parameter and the choice of the scaling factor may effect the efficiency of the optimization procedure. Therefore, we use three synthetic cases in Chapter 4 and one case in Chapter 5 to discuss the influence of the penalty parameter, Lagrange multipliers and scaling factors. We observed from the results of Example 1 in Chapter 4 that the convergence rates can be slowed appreciably by a poor scaling of the constraints. By introducing the scaling factor, we are able to rescale the various type of constraints to comparable levels. Example 1 in Chapter 4 shows we obtain higher NPVs with fewer simulation runs with proper rescaling. For Lagrange multipliers, we test two initial sets of the values, one is $\lambda^0 = 0$ and another one is to evaluate the initial Lagrange multipliers from the violation values based on the first simulation run. All examples except one (Example 2 in Chapter 4) demonstrate that using $\lambda^0_{c,i} = \max[0, \frac{s_{c,i}c^0_i}{\mu^0}]$ is not better than simply using $\lambda^0_{c,i} = 0$. However, the results in Example 2 of Chapter 4 show the choice of $\lambda^0_{c,i} = 0$ leads to divergence problem but the choice of $\lambda^0_{c,i} = \max[0, \frac{s_{c,i}c^0_i}{\mu^0}]$ gives higher NPVs with fewer simulation runs. Therefore, there is no definite conclusion on which method of choosing the initial set of Lagrange multipliers is better and it might be case-dependent. For the initial penalty parameters, if μ^0 is too large, the penalty term in the augmented Lagrangian function does not have enough weight during early iterations, which may cause large constraint violations while getting a higher NPV during early iterations. But at late iterations, the algorithm tries to satisfy the constraint as the penalty term becomes more important. However, these constraints cannot be satisfied without reducing the NPV. This makes the algorithm quite inefficient. But if μ^0 is too small, the heavy weight on the penalty term would limit the increase of NPV. Here, we suggest to use 0.1–10 per cent of the order of the NPV based on the initial simulation run. Estimating a good initial value of the penalty parameter and using proper scaling factors on constraints improve the robustness and efficiency of the optimization algorithm.

As there is large uncertainty on the reservoir description for green fields, which have not been fully developed, we apply the robust production optimization, where the reservoir uncertainty is described with a set of geological models which may be obtained from the data assimilation step of closed-loop reservoir management. In the robust long-term optimization problem, the augmented Lagrangian function is simply the average life-cycle NPV over the set of geological models with all equality and inequality constraints from each geological model incorporated with Lagrangian multipliers and penalty terms. The gradient of the augmented Lagrangian function is a summation over the gradients of the individual augmented Lagrangian function from each geological model. Experiments in Chapter 5 show that the field can not always achieve the optimal NPV using the optimal well controls obtained based on a single, but uncertain reservoir model, whereas the application of robust optimization significantly reduces this risk.

Another major challenge is the multi-objective optimization problem arising from the business need of reservoir development for life-cycle production optimization and the objective of field operators for short-term NPV optimization. The requirement of short-term optimization is to maximize the short-term NPV without compromising the life-cycle NPV. Robust sequential short-term optimization (i.e. the optimization of short-term NPV is done one short-term step following another one) generally results in a lower expected value of NPV over the reservoir life than is achieved with robust life-cycle optimization. For the short-term NPV optimization problem, we propose a two-stage production optimization procedure. We first solve the life-cycle constrained optimization problem. Then, we optimize the short-term NPV subject to the constraint that the long-term NPV is no less than the optimal NPV obtained by life-cycle production optimization. This long-term NPV constraint is incorporated into the short-term optimization together with the physical constraints using the augmented Lagrangian method.

Example 1 in Chapter 5 shows the expected value of short-term NPV can be increased with a negligible decrease in the expectation of life-cycle NPV if sufficient degrees of freedom exist after life-cycle optimization. We compare results from five strategies: reactive control, nominal optimization, robust long-term optimization, robust LASTOpt (long-term alternating with short-term optimization) and robust SSOpt (sequential short-term optimization). In the examples, the reactive control strategy obtains the lowest expectation of NPV among all the optimization strategies. When we apply estimated optimal controls from the nominal optimization based on a randomly selected reservoir model to all the reservoir models, there is a large variance in the final NPV from these different models, which indicates that making decisions based on results obtained with the nominal optimization scenarios involves large uncertainty and financial risk. Robust long-term optimization obtains the highest expected NPV. Robust LASTOpt yields the same (or very close) expectation of NPV as robust long-term optimization, which can achieve the operator's short-term goal of maximizing the short-term NPV while maintaining the long-term NPV. The robust SSOpt does not yield an expectation of the life-cycle NPV as high as the robust long-term optimization.

BIBLIOGRAPHY

[1] A.H. Alhuthali, D. Oyerinde, and A. Datta-Gupta. Optimal waterflood management using rate control,SPE 102478. In *Proceedings of the SPE Annual Technical Conference and Exhibition*, 2006.

[2] H. Asheim. Maximization of water sweep efficiency by controlling production and injection rates, SPE 18365. In *Proceedings of the SPE European Petroleum Conference*, 1988.

[3] K. Aziz and A. Settari. *Petroleum Reservoir Simulation*. Elsevier Applied Science Publishers, London, 1979.

[4] W. Bangerth, H. Klie, M.F. Wheeler, P.L. Stoffa, and M.K. Sen. On optimization algorithm for the reservoir oil well placement problem. *Computational Geosciences*, 10:303–319, 2006.

[5] A. Ben-Tal and A. Nemirovski. Robust optimization – methodology and applications. *Mathematical Programming*, 92(3):453–480, 2002.

[6] D.R. Brouwer and J.D. Jansen. Dynamic optimization of water flooding with smart wells using optimial control theory. *SPE Journal*, 9(4):391–402, 2004.

[7] D.R. Brouwer, G. Nævdal, J.D. Jansen, E.H. Vefring, and C.P.J.W. van Kruijsdijk. Improved reservoir management through optimal control and continuous model updating, SPE 90149. In *Proceedings of the SPE Annual Technical Conference and Exhibition, Houston, Texas, 26-29 September*, 2004.

[8] C. Chen, G. Li, and A.C. Reynolds. Closed-loop reservoir management on the Brugge test case. *Computational Geosciences*, 14(4):691–703, 2010.

[9] Y. Chen and D.S. Oliver. Ensemble-based closed-loop optimization applied to brugge field. *SPE Reservoir Evaluation & Engineering*, 13(1):56–71, 2010.

[10] Y. Chen, D.S. Oliver, and D. Zhang. Efficient ensemble-based closed-loop production optimization. *SPE Journal*, 14(4):634–645, 2009.

[11] A.R. Conn, N.I.M. Gould, and P.L. Toint. Testing a class of methods for solving minimization problems with simple bounds on the variables. *Mathematics of Computation*, 50(182):399–430, 1988.

[12] A.R. Conn, N.I.M. Gould, and P.L. Toint. A globally convergent augmented langrangian algorithm for optimization with general constraints and simple bounds. *SIAM Journal on Numerical Analysis*, 28(2):545–572, 1991.

[13] A.R. Conn, N.I.M. Gould, and P.L. Toint. *LANCELOT: A Fortran Package for Large-Scale Nonlinear Optimization (Release A).* Springer-Verlag, New York, 1992.

[14] A.R. Conn, L.N. Vicente, and C. Visweswariah. Two-step algorithms for nonlinear optimization with structured applications. *SIAM Journal on Optimization*, 9:924–947, 1999.

[15] P. de Montleau, A. Cominelli, K. Neylong, D. Rowan, I. Pallister, O. Tesaker, and I. Nygard. Production optimization under constraints using adjoint gradients. *Proceedings of the 10th European Conference on the Mathematics of Oil Recovery*, 2006.

[16] Y. Dong and D.S. Oliver. Quantitative use of 4D seismic data for reservoir characterization, SPE 84571. In *Proceedings of the SPE Annual Technical Conference and Exhibition*, 2003.

[17] D.C. Doublet, R. Martinsen, S.I. Aanonsen, and X.C. Tai. Efficient optimization of production from smart wells based on augmented Lagranfian method. In

Proceedings of the 10th European Conference on the Mathematical Oil Recovery - Amsterdam, 4-7 September, 2006.

[18] G. Evensen. Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *Journal of Geophysical Research*, 99(C5):10143–10162, 1994.

[19] G. Evensen. *Data Assimilation: The Ensemble Kalman Filter*. Springer, Berlin, 2007.

[20] F. Forouzanfar, G. Li, and A.C. Reynolds. A two-stage well placement optimization method based on adjoint gradient, SPE-135304. In *Proceedings of the SPE Annual Technical Conference and Exhibition*, 2010.

[21] G. Gao. *Data Integration and Uncertainty Evaluation for Large Scale Automatic History Matching Problems*. Ph.D. thesis, The University of Tulsa, Tulsa, Oklahoma, 2005.

[22] G. Gao, G. Li., and A.C. Reynolds. A stochastic algorithm for automatic history matching. *SPE Journal*, 12(2):196–208, 2007.

[23] J.D. Jansen, S.D. Douma, D.R. Brouwer, P.M.J. Van den Hof, and A.W. Heemink. Closed-loop reservoir management, SPE-119098. In *Proceedings of the SPE Reservoir Simulation Symposium, 2-4 February*, 2009.

[24] J.F.B.M. Kraaijevanger, P.J.P. Egberts, J.R. Valstar, and H.W. Buurman. Optimal waterflood design using the adjoint method, SPE 105764. In *Proceedings of the SPE Reservoir Simulation Symposium*, 2007.

[25] G. Li and A.C. Reynolds. Uncertainty quantification of reservoir performance predictions using a stochastic optimization algorithm (doi: 10.1007/s10596-010-9214-2). *Computational Geosciences*, 2010.

[26] R. Li. *Conditioning Geostatistical Models to Three-Dimensional Three-Phase Flow Production Data by Automatic History Matching.* Ph.D. thesis, The University of Tulsa, Tulsa, Oklahoma, 2001.

[27] R. Li, A.C. Reynolds, and D.S. Oliver. Sensitivity coefficients for three-phase flow history matching. *J. Canadian Pet. Tech.*, 42(4):70–77, 2003.

[28] Y. Liu. Using the SNESIM program for multiple-point statistical simulation. *Computers & Geosciences*, 32(10):1544–1563, 2006.

[29] R.J. Lorentzen, A.M. Berg, G. Nævdal, and E.H. Vefring. A new approach for dynamic optimization of waterflooding problems, SPE 99690. In *Proceedings of the SPE Intelligent Energy Conference and Exhibition*, 2006.

[30] D.G. Luenberger. *Linear and Nonlinear Programming.* Addison-Wesley Publishing Company, 1984.

[31] J. Nocedal and S.J. Wright. *Numerical Optimization.* Springer, New York, 1999.

[32] J. Nwaozo. Dynamic optimization of a water flood reservoir. Master's thesis, University of Oklahoma, Norman, Oklahoma, 2006.

[33] D.S. Oliver, N. He., and A.C. Reynolds. Conditioning permeability fields to pressure data. In *Proccedings of the European Conference for the Mathematics of Oil Recovery*, 1996.

[34] D.W. Peaceman. Interpretation of well-block pressures in numerical reservoir simulation with non-square grid blocks and anisotropic permeability. *SPE Journal*, 23(6):531–543, 1983.

[35] L. Peters, R.J. Arts, G.K. Brouwer, C.R. Geel, S. Cullick, R.J. Lorentzen, Y. Chen, K.N.B. Dunlop, F.C. Vossepoel, R. Xu, P. Sarma, A.H. Alhuthali,

and A.C. Reynolds. Results of the Brugge benchmark study for flooding optimisation and history matching. *SPE Reservoir Evaluation & Engineering*, 13(3): 391–405, 2010.

[36] M.J.D Powell. The newuoa software for unconstrained optimization without derivatives. In P. Pardalos, G. Pillo, and M. Roma, editors, *Large-Scale Nonlinear Optimization*, volume 83 of *Nonconvex Optimization and Its Applications*, pages 255–297. Springer US, 2006. ISBN 978-0-387-30065-8.

[37] S.S. Rao. *Engineering Optimization: Theory and Practice, Third Edition.* John Wiley and Sons Inc., New York, 1996.

[38] A.C. Reynolds, M. Zafari, and G. Li. Iterative forms of the ensemble Kalman filter. *Proceedings of the 10th European Conference on the Mathematics of Oil Recovery*, 2006.

[39] P. Sarma, L.J. Durlofsky, and K. Aziz. Implementation of adjoint solution for optimal control of smart wells, SPE 92864. In *Proceedings of the SPE Reservoir Simulation Symposium*, 2005.

[40] P. Sarma, W.H. Chen, L.J. Durlofsky, and K. Aziz. Production optimization with adjoint models under nonlinear control-state path inequality constraints, SPE 99959. In *Proceedings of the SPE/EAGE Annual Conference*, 2006.

[41] J.C. Spall. Implementation of the simultaneous perturbation algorithm for stochastic optimization. *IEEE Transactions on Aerospace and Electronic Systems*, 34(3):817–823, 1998.

[42] B. Sudaryanto and Y.C. Yortsos. Optimization of fluid front dynamics in porous media using rate control. I. equal mobility fluids. *Physics of Fluids*, 12(7):1656–1670, 2000.

[43] B. Sudaryanto and Y.C. Yortsos. Optimization of displacement in porous media using rate control. In *Proceedings of the SPE Annual Technical Conference and Exhibition*, 2001.

[44] P.L. Toint. Non-monotone trust-region algorithms for nonlinear optimization subject to convex constraints. *Mathematical programming*, 77:69–94, 1997.

[45] G. van Essen, P. Van den Hof, and J.D. Jansen. Hierarchical long-term and short-term production optimization (doi:10.2118/124332-pa, online). *SPE Journal*, 2010.

[46] G.M. van Essen, M.J. Zandvliet, P.M.J. Van den Hof, O.H. Bosgra, and J.D. Jansen. Robust waterflooding optimization of multiple geological scenarios. *SPE Journal*, 14(1):202–210, 2009.

[47] C. Wang, G. Li, and A.C. Reynolds. Optimal well placement for production optimization. *SPE Journal*, 14(3):506–523, 2009.

[48] M. Zafari and A.C. Reynolds. Assessing the uncertainty in reservoir description and performance predictions with the ensemble Kalman filter. *SPE Journal*, 12 (3):382–391, 2007.

[49] I.S. Zakirov, S.I. Aanonsen, E.S. Zakirov, and B.M. Palatnik. Optimizating reservoir performance by automatic allocation of well rates. In *Proceedings of the 5th European Conference on the Mathematical Oil Recovery - Leoben, Austria, 3-5 September*, 1996.

[50] M.J. Zandvliet, O.H. Bosgra, J.D. Jasen, P.M.J. Van den Hof, and J.F.B.M. Kraaijevvanger. Bang-bang control and singular arcs in reservoir flooding. *Journal of Petroleum Science and Engineering*, 58:186–200, 2007.

[51] F. Zhang. *Automatic History Matching of Production Data for Large Scale Problems*. Ph.D. thesis, The University of Tulsa, Tulsa, Oklahoma, 2002.

[52] F. Zhang and A.C. Reynolds. Optimization algorithms for automatic history matching of production data. In *Proceedings of the 8th European Conference on the Mathematics of Oil Recovery*, 2002.

[53] K. Zhang, G. Li, A.C. Reynolds, L. Zhang, and J. Yao. Optimal well placement using and adjoint gradient. *Journal of Petroleum Science and Engineering*, 73: 220–226, 2010.

[54] H. Zhao, C. Chen, S. Do, G. Li, and A.C. Reynolds. Maximization of a dynamic quadratic interpolation model for production optimization, SPE 141317. In *Proccedings of the SPE Reservoir Simulation Symposium, The Woodlands, Texas, USA, 21-23 February*, 2011.

APPENDIX A

DERIVATIVES OF β WITH RESPECT TO PRIMARY VARIABLES

A.1 Derivatives of Flow Rates at the Production Well l

The flow equations have been given in Eqs. 3.28 through 3.31. Throughout, we let k denote the layer index. For the derivative of flow rates with respect to the gridblock pressure of the producer l, $p^n_{il,jl,k}$, we need to consider two situations. If $S^n_{g,il,jl,k} = 0$, then the derivatives of flow rates with respect to the gridblock pressure of a producer well l at a completion in layer k, $p^n_{il,jl,k}$, are

$$\begin{aligned}\frac{\partial q^n_{o,il,jl,k}}{\partial p^n_{il,jl,k}} =&\mathrm{WI}_{il,jl,k}\Big[-\frac{k^n_{ro,il,jl,k}}{(B^n_{o,il,jl,k}\mu^n_{o,il,jl,k})^2}(\mu^n_{o,il,jl,k}\frac{\partial B^n_{o,il,jl,k}}{\partial p^n_{il,jl,k}} + B^n_{o,il,jl,k}\frac{\partial \mu^n_{o,il,jl,k}}{\partial p^n_{il,jl,k}})\\ &(p^n_{il,jl,k} - p^n_{wf,il,jl,k}) + (\frac{k_{ro}}{B_o\mu_o})^n_{il,jl,k}\Big],\end{aligned} \tag{A.1}$$

$$\begin{aligned}\frac{\partial q^n_{w,il,jl,k}}{\partial p^n_{il,jl,k}} =&\mathrm{WI}_{il,jl,k}\Big[-\frac{k^n_{rw,il,jl,k}}{(B^n_{w,il,jl,k}\mu^n_{w,il,jl,k})^2}(\mu^n_{w,il,jl,k}\frac{\partial B^n_{w,il,jl,k}}{\partial p^n_{il,jl,k}} + B^n_{w,il,jl,k}\frac{\partial \mu^n_{w,il,jl,k}}{\partial p^n_{il,jl,k}})\\ &(p^n_{il,jl,k} - p^n_{wf,il,jl,k}) + (\frac{k_{rw}}{B_w\mu_w})^n_{il,jl,k}\Big],\end{aligned} \tag{A.2}$$

and

$$\begin{aligned}\frac{\partial q^n_{g,il,jl,k}}{\partial p^n_{il,jl,k}} =&\mathrm{WI}_{il,jl,k}\Big\{\big[-\frac{k^n_{rg,il,jl,k}}{(B^n_{g,il,jl,k}\mu^n_{g,il,jl,k})^2}(\mu^n_{g,il,jl,k}\frac{\partial B^n_{g,il,jl,k}}{\partial p^n_{il,jl,k}} + B^n_{g,il,jl,k}\frac{\partial \mu^n_{g,il,jl,k}}{\partial p^n_{il,jl,k}})\\ &-\frac{(k_{ro}R_s)^n_{il,jl,k}}{(B^n_{o,il,jl,k}\mu^n_{o,il,jl,k})^2}(\mu^n_{o,il,jl,k}\frac{\partial B^n_{o,il,jl,k}}{\partial p^n_{il,jl,k}} + B^n_{o,il,jl,k}\frac{\partial \mu^n_{o,il,jl,k}}{\partial p^n_{il,jl,k}})\big]\\ &(p^n_{il,jl,k} - p^n_{wf,il,jl,k}) + (\frac{k_{rg}}{B_g\mu_g} + R_s\frac{k_{ro}}{B_o\mu_o})^n_{il,jl,k}\Big\}.\end{aligned} \tag{A.3}$$

If $S^n_{g,il,jl,k} > 0$, Eq. A.2 keeps the same equation and the derivatives of oil and gas rates with respect to the gridblock pressure are

$$\begin{aligned}\frac{\partial q^n_{o,il,jl,k}}{\partial p^n_{il,jl,k}} =& \mathrm{WI}_{il,jl,k}\Big\{ -\frac{k^n_{ro,il,jl,k}}{(B^n_{o,il,jl,k}\mu^n_{o,il,jl,k})^2}\big[\mu^n_{o,il,jl,k}\frac{\partial B^n_{o,il,jl,k}}{\partial p^n_{il,jl,k}} + B^n_{o,il,jl,k}\frac{\partial \mu^n_{o,il,jl,k}}{\partial p^n_{il,jl,k}} \\ &+ (\mu^n_{o,il,jl,k}\frac{\partial B^n_{o,il,jl,k}}{\partial R^n_{s,il,jl,k}} + B^n_{o,il,jl,k}\frac{\partial \mu^n_{o,il,jl,k}}{\partial R^n_{s,il,jl,k}})\frac{\partial R^n_{s,il,jl,k}}{\partial p^n_{il,jl,k}}\big] \\ &(p^n_{il,jl,k} - p^n_{wf,il,jl,k}) + (\frac{k_{ro}}{B_o\mu_o})^n_{il,jl,k}\Big\},\end{aligned} \tag{A.4}$$

and

$$\begin{aligned}\frac{\partial q^n_{g,il,jl,k}}{\partial p^n_{il,jl,k}} =& \mathrm{WI}_{il,jl,k}\Big\{\big[-\frac{k^n_{rg,il,jl,k}}{(B^n_{g,il,jl,k}\mu^n_{g,il,jl,k})^2}(\mu^n_{g,il,jl,k}\frac{\partial B^n_{g,il,jl,k}}{\partial p^n_{il,jl,k}} + B^n_{g,il,jl,k}\frac{\partial \mu^n_{g,il,jl,k}}{\partial p^n_{il,jl,k}}) \\ &+ \big((\frac{k_{ro}}{B_o\mu_o})^n_{il,jl,k} - \frac{(k_{ro}R_s)^n_{il,jl,k}}{(B^n_{o,il,jl,k}\mu^n_{o,il,jl,k})^2}(\mu^n_{o,il,jl,k}\frac{\partial B^n_{o,il,jl,k}}{\partial R^n_{s,il,jl,k}} + B^n_{o,il,jl,k}\frac{\partial \mu^n_{o,il,jl,k}}{\partial R^n_{s,il,jl,k}})\big) \\ &\frac{\partial R^n_{s,il,jl,k}}{\partial p^n_{il,jl,k}}\big](p^n_{il,jl,k} - p^n_{wf,il,jl,k}) + (\frac{k_{rg}}{B_g\mu_g} + R_s\frac{k_{ro}}{B_o\mu_o})^n_{il,jl,k}\Big\}.\end{aligned} \tag{A.5}$$

The derivatives of flow rates with respect to the gridblock oil saturation of a producer well l at a completion in layer k, $S^n_{o,il,jl,k}$, are

$$\frac{\partial q^n_{o,il,jl,k}}{\partial S^n_{o,il,jl,k}} = \mathrm{WI}_{il,jl,k}\frac{1}{(B_o\mu_o)^n_{il,jl,k}}\frac{\partial k^n_{ro,il,jl,k}}{\partial S^n_{o,il,jl,k}}(p^n_{il,jl,k} - p^n_{wf,il,jl,k}), \tag{A.6}$$

$$\frac{\partial q^n_{w,il,jl,k}}{\partial S^n_{o,il,jl,k}} = \mathrm{WI}_{il,jl,k}\frac{1}{(B_w\mu_w)^n_{il,jl,k}}\frac{\partial k^n_{rw,il,jl,k}}{\partial S^n_{o,il,jl,k}}(p^n_{il,jl,k} - p^n_{wf,il,jl,k}), \tag{A.7}$$

and

$$\begin{aligned}\frac{\partial q^n_{g,il,jl,k}}{\partial S^n_{o,il,jl,k}} =& \mathrm{WI}_{il,jl,k}\Big[\frac{1}{(B_g\mu_g)^n_{il,jl,k}}\frac{\partial k^n_{rg,il,jl,k}}{\partial S^n_{o,il,jl,k}} + \frac{R^n_{s,il,jl,k}}{(B_o\mu_o)^n_{il,jl,k}}\frac{\partial k^n_{ro,il,jl,k}}{\partial S^n_{o,il,jl,k}}\Big] \\ &(p^n_{il,jl,k} - p^n_{wf,il,jl,k}).\end{aligned} \tag{A.8}$$

For $S^n_{g,il,jl,k} > 0$, the derivatives of flow rates with respect to the gridblock gas saturation of a producer well l at a completion in layer k, $S^n_{g,il,jl,k}$, are

$$\frac{\partial q^n_{o,il,jl,k}}{\partial S^n_{g,il,jl,k}} = \mathrm{WI}_{il,jl,k}\frac{1}{(B_o\mu_o)^n_{il,jl,k}}\frac{\partial k^n_{ro,il,jl,k}}{\partial S^n_{g,il,jl,k}}(p^n_{il,jl,k} - p^n_{wf,il,jl,k}), \tag{A.9}$$

$$\frac{\partial q^n_{w,il,jl,k}}{\partial S^n_{g,il,jl,k}} = \mathrm{WI}_{il,jl,k} \frac{1}{(B_w\mu_w)^n_{il,jl,k}} \frac{\partial k^n_{rw,il,jl,k}}{\partial S^n_{g,il,jl,k}} (p^n_{il,jl,k} - p^n_{wf,il,jl,k}), \tag{A.10}$$

and

$$\begin{aligned} \frac{\partial q^n_{g,il,jl,k}}{\partial S^n_{g,il,jl,k}} =& \mathrm{WI}_{il,jl,k} \Big[(\frac{1}{(B_g\mu_g)^n_{il,jl,k}} \frac{\partial k^n_{rg,il,jl,k}}{\partial S^n_{g,il,jl,k}} + \frac{R^n_{s,il,jl,k}}{(B_o\mu_o)^n_{il,jl,k}} \frac{\partial k^n_{ro,il,jl,k}}{\partial S^n_{g,il,jl,k}}) \\ & (p^n_{il,jl,k} - p^n_{wf,il,jl,k}) \Big]. \end{aligned} \tag{A.11}$$

For $S^n_{g,il,jl,k} = 0$, the derivatives of flow rates with respect to the gridblock dissolved gas oil ratio, $R^n_{s,il,jl,k}$, are

$$\begin{aligned} \frac{\partial q^n_{o,il,jl,k}}{\partial R^n_{s,il,jl,k}} =& - \mathrm{WI}_{il,jl,k} (p^n_{il,jl,k} - p^n_{wf,il,jl,k}) \\ & \frac{k^n_{ro,il,jl,k}}{(B^n_{o,il,jl,k}\mu^n_{o,il,jl,k})^2} (\mu^n_{o,il,jl,k} \frac{\partial B^n_{o,il,jl,k}}{\partial R^n_{s,il,jl,k}} + B^n_{o,il,jl,k} \frac{\partial \mu^n_{o,il,jl,k}}{\partial R^n_{s,il,jl,k}}), \end{aligned} \tag{A.12}$$

$$\frac{\partial q^n_{w,il,jl,k}}{\partial R^n_{s,il,jl,k}} = 0, \tag{A.13}$$

and

$$\begin{aligned} \frac{\partial q^n_{g,il,jl,k}}{\partial R^n_{s,il,jl,k}} =& \mathrm{WI}_{il,jl,k} (p^n_{il,jl,k} - p^n_{wf,il,jl,k}) \Big[\frac{k^n_{ro,il,jl,k}}{B^n_{o,il,jl,k}\mu^n_{o,il,jl,k}} - \frac{(k_{ro}R_s)^n_{il,jl,k}}{(B^n_{o,il,jl,k}\mu^n_{o,il,jl,k})^2} \\ & (\mu^n_{o,il,jl,k} \frac{\partial B^n_{o,il,jl,k}}{\partial R^n_{s,il,jl,k}} + B^n_{o,il,jl,k} \frac{\partial \mu^n_{o,il,jl,k}}{\partial R^n_{s,il,jl,k}}) \Big]. \end{aligned} \tag{A.14}$$

Assume there are a total of K completions for the producer well l, then the derivatives of flow rates with respect to the well flowing pressure, $p_{wf,il,jl}$, are

$$\frac{\partial q^n_{o,il,jl}}{\partial p^n_{wf,il,jl}} = - \sum_{k=1}^{K} \mathrm{WI}_{il,jl,k} (\frac{k_{ro}}{B_o\mu_o})^n_{il,jl,k}, \tag{A.15}$$

$$\frac{\partial q^n_{w,il,jl}}{\partial p^n_{wf,il,jl}} = - \sum_{k=1}^{K} \mathrm{WI}_{il,jl,k} (\frac{k_{rw}}{B_w\mu_w})^n_{il,jl,k}, \tag{A.16}$$

and

$$\frac{\partial q^n_{g,il,jl}}{\partial p^n_{wf,il,jl}} = -\sum_{k=1}^{K} \mathrm{WI}_{il,jl,k}(\frac{k_{rg}}{B_g\mu_g} + R_s\frac{k_{ro}}{B_o\mu_o})^n_{il,jl,k}. \tag{A.17}$$

A.2 Derivative of Water Rate at the Injection Well l

The derivatives of water rate with respect to the gridblock pressure, $p^n_{il,jl,k}$, is

$$\begin{aligned}\frac{\partial q^n_{wi,il,jl,k}}{\partial p^n_{il,jl,k}} =&\mathrm{WI}_{il,jl,k}[-\frac{k^n_{rw,il,jl,k}}{(B^n_{w,il,jl,k}\mu^n_{w,il,jl,k})^2}(\mu^n_{w,il,jl,k}\frac{\partial B^n_{w,il,jl,k}}{\partial p^n_{il,jl,k}} + B^n_{w,il,jl,k}\frac{\partial \mu^n_{w,il,jl,k}}{\partial p^n_{il,jl,k}})\\ &(p^n_{wf,il,jl,k} - p^n_{il,jl,k}) - (\frac{k_{rw}}{B_w\mu_w})^n_{il,jl,k}],\end{aligned} \tag{A.18}$$

As CLASS specifies $k_{rw} = 1$ for all the simulation steps for the injector, the derivatives of water injection rates with respect to the gridblock oil saturation, $S^n_{o,il,jl,k}$, is

$$\frac{\partial q^n_{wi,il,jl,k}}{\partial S^n_{o,il,jl,k}} = 0, \tag{A.19}$$

and the derivatives of water injection rates with respect to the gridblock gas saturation is also zero.

Assume there are a total of K completions for the injection well l, then the derivative of water rate with respect to the well flowing pressure, $p_{wf,il,jl}$, is

$$\frac{\partial q^n_{wi,il,jl}}{\partial p^n_{wf,il,jl}} = \sum_{k=1}^{K} \mathrm{WI}_{il,jl,k}(\frac{k_{rw}}{B_w\mu_w})^n_{il,jl,k}. \tag{A.20}$$

A.3 Derivative of the NPV with Respect to Primary Variables

As the augmented Lagrangian function β is a combination of the NPV and the terms of constraints, the derivative of β with respect to primary variables can be written as

$$\nabla_{y^n}\beta = \nabla_{y^n}J - \sum_{i=1}^{n_e}\left[\left(\lambda_{e,i} + \frac{s_{e,i}}{\mu}e_i(y^n,U^n)\right)\nabla_{y^n}e_i(y^n,U^n)\right] - \sum_{i=1}^{n_i}\nabla_u\psi_i(y^n,U^n), \tag{A.21}$$

where

$$\nabla_{y^n}\psi_i(y^n,U^n) = \begin{cases} 0, & c_i(y^n,U^n) \le -\mu\lambda_{c,i}/s_{c,i}, \\ [\lambda_{c,i} + \frac{s_{c,i}}{\mu}c_i(y^n,U^n)]\nabla_{y^n}c_i(y^n,U^n), & \text{otherwise.} \end{cases} \tag{A.22}$$

We first provide the derivative of NPV with respect to the reservoir primary variables. The derivative of the NPV with respect to the gridblock pressure of a producer well l at a completion in layer k, $p^n_{il,jl,k}$, is

$$\frac{\partial J}{\partial p^n_{il,jl,k}} = \Big(r^n_o\frac{\partial q^n_{o,il,jl,k}}{\partial p^n_{il,jl,k}} - r^n_w\frac{\partial q^n_{w,il,jl,k}}{\partial p^n_{il,jl,k}} + r^n_g\frac{\partial q^n_{g,il,jl,k}}{\partial p^n_{il,jl,k}}\Big)\frac{\Delta t^n}{(1+b)^{t^n/365}}. \tag{A.23}$$

The derivative of the life-cycle NPV with respect to the gridblock oil saturation of a producer well l at a completion in layer k, $S^n_{o,il,jl,k}$, is

$$\frac{\partial J}{\partial S^n_{o,il,jl,k}} = \Big(r^n_o\frac{\partial q^n_{o,il,jl,k}}{\partial S^n_{o,il,jl,k}} - r^n_w\frac{\partial q^n_{w,il,jl,k}}{\partial S^n_{o,il,jl,k}} + r^n_g\frac{\partial q^n_{g,il,jl,k}}{\partial S^n_{o,il,jl,k}}\Big)\frac{\Delta t^n}{(1+b)^{t^n/365}}. \tag{A.24}$$

If $S^n_{g,il,jl,k} > 0$, the derivative of the life-cycle NPV with respect to the gridblock gas saturation of a producer well l at a completion in layer k, $S^n_{g,il,jl,k}$, is

$$\frac{\partial J}{\partial S^n_{g,il,jl,k}} = \Big(r^n_o\frac{\partial q^n_{o,il,jl,k}}{\partial S^n_{g,il,jl,k}} - r^n_w\frac{\partial q^n_{w,il,jl,k}}{\partial S^n_{g,il,jl,k}} + r^n_g\frac{\partial q^n_{g,il,jl,k}}{\partial S^n_{g,il,jl,k}}\Big)\frac{\Delta t^n}{(1+b)^{t^n/365}}. \tag{A.25}$$

If $S^n_{g,il,jl,k} = 0$, the derivative of the life-cycle NPV with respect to the gridblock dissolved gas oil ratio, $R^n_{s,il,jl,k}$, is

$$\frac{\partial J}{\partial R^n_{s,il,jl,k}} = \Big(r^n_o\frac{\partial q^n_{o,il,jl,k}}{\partial R^n_{s,il,jl,k}} - r^n_w\frac{\partial q^n_{w,il,jl,k}}{\partial R^n_{s,il,jl,k}} + r^n_g\frac{\partial q^n_{g,il,jl,k}}{\partial R^n_{s,il,jl,k}}\Big)\frac{\Delta t^n}{(1+b)^{t^n/365}}. \tag{A.26}$$

The derivative of the life-cycle NPV with respect to the well flowing pressure of the producer l, $p_{wf,il,jl}$, is

$$\frac{\partial J}{\partial p^n_{wf,il,jl}} = \left(r^n_o \frac{\partial q^n_{o,il,jl,k}}{\partial p^n_{wf,il,jl}} - r^n_w \frac{\partial q^n_{w,il,jl,k}}{\partial p^n_{wf,il,jl}} + r^n_g \frac{\partial q^n_{g,il,jl,k}}{\partial p^n_{wf,il,jl}}\right) \frac{\Delta t^n}{(1+b)^{t^n/365}}. \tag{A.27}$$

The derivative of the life-cycle NPV with respect to the gridblock pressure of an injection well l at a completion in layer k, $p^n_{il,jl,k}$, is

$$\frac{\partial J}{\partial p^n_{il,jl,k}} = -\left(r^n_{wi} \frac{\partial q^n_{wi,il,jl,k}}{\partial p^n_{il,jl,k}}\right) \frac{\Delta t^n}{(1+b)^{t^n/365}}. \tag{A.28}$$

The derivative of the life-cycle NPV with respect to the gridblock oil or gas saturation of an injection well l at a completion in layer k are zeros. The derivative of the life-cycle NPV with respect to the well flowing pressure of the injector l, $p_{wf,il,jl}$, is

$$\frac{\partial J}{\partial p^n_{wf,il,jl}} = -\left(r^n_{wi} \frac{\partial q^n_{wi,il,jl,k}}{\partial p^n_{wf,il,jl}}\right) \frac{\Delta t^n}{(1+b)^{t^n/365}}. \tag{A.29}$$

A.4 Derivatives of Constraints with Respect to Primary Variables

The equality an inequality constraints are functions of flow rates. Here, we only provide an example of derivative of the water-oil ratio at a single production well l with respect to primary variables. The derivative of the WOR with respect to the gridblock pressure of a producer well l at a completion in layer k, $p^n_{il,jl,k}$, is

$$\frac{\partial}{\partial p^n_{il,jl,k}} \left(\frac{q^n_{w,il,jl}}{q^n_{o,il,jl}}\right) = \frac{1}{(q^n_{o,il,jl})^2} \left(q^n_{o,il,jl} \frac{\partial q^n_{w,il,jl,k}}{\partial p^n_{il,jl,k}} - q^n_{w,il,jl} \frac{\partial q^n_{o,il,jl,k}}{\partial p^n_{il,jl,k}}\right). \tag{A.30}$$

The derivative of the WOR with respect to the gridblock oil saturation of a producer well l at a completion in layer k, $S^n_{o,il,jl,k}$, is

$$\frac{\partial}{\partial S^n_{o,il,jl,k}}\left(\frac{q^n_{w,il,jl}}{q^n_{o,il,jl}}\right)=\frac{1}{(q^n_{o,il,jl})^2}\left(q^n_{o,il,jl}\frac{\partial q^n_{w,il,jl,k}}{\partial S^n_{o,il,jl,k}}-q^n_{w,il,jl}\frac{\partial q^n_{o,il,jl,k}}{\partial S^n_{o,il,jl,k}}\right). \tag{A.31}$$

If $S^n_{g,il,jl,k}>0$, the derivative of the WOR with respect to the gridblock gas saturation of a producer well l at a completion in layer k, $S^n_{g,il,jl,k}$, is

$$\frac{\partial}{\partial S^n_{g,il,jl,k}}\left(\frac{q^n_{w,il,jl}}{q^n_{o,il,jl}}\right)=\frac{1}{(q^n_{o,il,jl})^2}\left(q^n_{o,il,jl}\frac{\partial q^n_{w,il,jl,k}}{\partial S^n_{g,il,jl,k}}-q^n_{w,il,jl}\frac{\partial q^n_{o,il,jl,k}}{\partial S^n_{g,il,jl,k}}\right). \tag{A.32}$$

If $S^n_{g,il,jl,k}=0$, the derivative of the WOR with respect to the gridblock dissolved gas oil ratio, $R^n_{s,il,jl,k}$, is

$$\frac{\partial}{\partial R^n_{s,il,jl,k}}\left(\frac{q^n_{w,il,jl}}{q^n_{o,il,jl}}\right)=\frac{1}{(q^n_{o,il,jl})^2}\left(q^n_{o,il,jl}\frac{\partial q^n_{w,il,jl,k}}{\partial R^n_{s,il,jl,k}}-q^n_{w,il,jl}\frac{\partial q^n_{o,il,jl,k}}{\partial R^n_{s,il,jl,k}}\right). \tag{A.33}$$

The derivative of the WOR with respect to the well flowing pressure of the producer l, $p_{wf,il,jl}$, is

$$\frac{\partial}{\partial p^n_{wf,il,jl}}\left(\frac{q^n_{w,il,jl}}{q^n_{o,il,jl}}\right)=\frac{1}{(q^n_{o,il,jl})^2}\left(q^n_{o,il,jl}\frac{\partial q^n_{w,il,jl}}{\partial p^n_{wf,il,jl}}-q^n_{w,il,jl}\frac{\partial q^n_{o,il,jl}}{\partial p^n_{wf,il,jl}}\right). \tag{A.34}$$

APPENDIX B

GENERALIZED REDUCED GRADIENT METHOD

In the presence of state-control constraints, the set of control variables is divided into two subsets, free (independent) and dependent control variables. Throughout, let f and d, respectively, denote the free set and the dependent set, i.e., the control vector u can be rewritten as $u = [u_f^T, u_d^T]^T$. The dimension of the dependent control vector is determined by the total number of equality and inequality constraints. If we apply the slack variables to convert these inequality constraints to equality constraints, we need the same number of slack variables as the number of inequality constraints. Namely,

$$N_{u_d} = n_e + n_c = n_e + n_v, \tag{B.35}$$

where N_{u_d} denotes the dimension of the dependent control vector; n_e, n_c and n_v are, respectively, the number of equality constraints, inequality constraints and the slack variables. The above rule indicates that one constraint reduces one degree of freedom assuming the slack vector is a free control vector. The life-cycle NPV defined in Eq. 1.3 can be rewritten as the function of the simulation primary vector y, the free controls and the dependent control vectors:

$$J_L(y, u) = J_L(y, u_f, u_d). \tag{B.36}$$

The reservoir simulator equations are given in Eq. 3.8 as

$$f^{n+1} = f(y^{n+1}, y^n, u^{n+1}) = 0, \tag{B.37}$$

for $n = 0, 1, ..., N_L - 1$. The dimension of y^n is $N_y \times 1$. We combine the simulation equations at all time steps into one overall simulation equation

$$F(y, u) = F(y, u_f, u_d) = 0, \tag{B.38}$$

i.e. we define

$$F(y, u) = \begin{bmatrix} f^1(y^1, y^0, u^1) \\ f^2(y^2, y^1, u^2) \\ \vdots \\ f^{N_L}(y^{N_L}, y^{N_L - 1}, u^{N_L}) \end{bmatrix} = \begin{bmatrix} f^1(y^1, y^0, u_f^1, u_d^1) \\ f^2(y^2, y^1, u_f^2, u_d^2) \\ \vdots \\ f^{N_L}(y^{N_L}, y^{N_L - 1}, u_f^{N_L}, u_d^{N_L}) \end{bmatrix}. \tag{B.39}$$

Similarly, the equality and inequality can be formulated as

$$E(y, u) = \begin{bmatrix} e^1(y^1, u^1) \\ e^2(y^2, u^2) \\ \vdots \\ e^{N_L}(y^{N_L}, u^{N_L}) \end{bmatrix} = E(y, u_f, u_d) = 0, \tag{B.40}$$

$$C(y, u) = \begin{bmatrix} c^1(y^1, u^1) \\ c^2(y^2, u^2) \\ \vdots \\ c^{N_L}(y^{N_L}, u^{N_L}) \end{bmatrix} = C(y, u_f, u_d) \leq 0. \tag{B.41}$$

Note that the bound constraints are not imposed in the equations of inequality constraints. The GRG method applied in Eclipse 300 deals with the simple bound constraints by limiting the step size.

The Lagrangian function L is defined as,

$$\begin{aligned} L &= J_L + \lambda_F^T F + \lambda_E^T E + \lambda_C^T (C - v) \\ &= J_L(y, u_f, u_d) + \lambda_F^T F(y, u_f, u_d) + \lambda_E^T E(y, u_f, u_d) + \lambda_C^T (C(y, u_f, u_d) - v) \end{aligned} \tag{B.42}$$

where v is a column vector of slack variables; λ_F, λ_E and λ_C are adjoint vectors for, respectively, reservoir simulation system, equality constraints and inequality constraints. The dimensions of λ_F, λ_E and λ_C are, respectively, $N_L N_y \times 1, N_L n_e \times 1$ and $N_L n_c \times 1$.

The total derivative of L is

$$\begin{aligned} dL = {} & (J_y + \lambda_F^T F_y + \lambda_E^T E_y + \lambda_C^T C_y) dy + (J_{u_f} + \lambda_F^T F_{u_f} + \lambda_E^T E_{u_f} + \lambda_C^T C_{u_f}) du_f \\ & + (J_{u_d} + \lambda_F^T F_{u_d} + \lambda_E^T E_{u_d} + \lambda_C^T C_{u_d}) du_d - \lambda_C^T dv, \end{aligned} \tag{B.43}$$

where

$$J_y = [(\nabla_{y^1} J)^T, (\nabla_{y^2} J)^T, ..., (\nabla_{y^{N_L}} J)^T]_{1 \times N_L N_y}, \tag{B.44}$$

$$J_{u_d} = [(\nabla_{u_d^1} J)^T, (\nabla_{u_d^2} J)^T, ..., (\nabla_{u_d^{N_L}} J)^T]_{1 \times N_L N_{u_d}}, \tag{B.45}$$

$$F_y = \begin{bmatrix} (\nabla_{y^1}(f^1)^T)^T & & & \\ (\nabla_{y^1}(f^2)^T)^T & (\nabla_{y^2}(f^2)^T)^T & & \\ & \ddots & \ddots & \\ & & (\nabla_{y^{N_L-1}}(f^{N_L})^T)^T & (\nabla_{y^{N_L}}(f^{N_L})^T)^T \end{bmatrix}_{N_L N_y \times N_L N_y}, \tag{B.46}$$

$$E_y = \begin{bmatrix} (\nabla_{y^1}(e^1)^T)^T & & & \\ & (\nabla_{y^2}(e^2)^T)^T & & \\ & & \ddots & \\ & & & (\nabla_{y^{N_L}}(e^{N_L})^T)^T \end{bmatrix}_{N_L n_e \times N_L N_y}, \tag{B.47}$$

$$C_y = \begin{bmatrix} (\nabla_{y^1}(c^1)^T)^T & & & \\ & (\nabla_{y^2}(c^2)^T)^T & & \\ & & \ddots & \\ & & & (\nabla_{y^{N_L}}(c^{N_L})^T)^T \end{bmatrix}_{N_L n_c \times N_L N_y}, \tag{B.48}$$

$$F_{u_f} = \begin{bmatrix} (\nabla_{u_f{}^1}(f^1)^T)^T & & & \\ & (\nabla_{u_f{}^2}(f^2)^T)^T & & \\ & & \ddots & \\ & & & (\nabla_{u_f{}^{N_L}}(f^{N_L})^T)^T \end{bmatrix}_{N_L N_y \times L N_{u_f}}, \tag{B.49}$$

$$E_{u_f} = \begin{bmatrix} (\nabla_{u_f{}^1}(e^1)^T)^T & & & \\ & (\nabla_{u_f{}^2}(e^2)^T)^T & & \\ & & \ddots & \\ & & & (\nabla_{u_f{}^{N_L}}(e^{N_L})^T)^T \end{bmatrix}_{N_L n_e \times N_L N_{u_f}}, \tag{B.50}$$

$$C_{u_f} = \begin{bmatrix} (\nabla_{u_f{}^1}(c^1)^T)^T & & & \\ & (\nabla_{u_f{}^2}(c^2)^T)^T & & \\ & & \ddots & \\ & & & (\nabla_{u_f{}^{N_L}}(c^{N_L})^T)^T \end{bmatrix}_{N_L n_c \times N_L N_{u_f}}, \tag{B.51}$$

$$F_{u_d} = \begin{bmatrix} (\nabla_{u_d{}^1}(f^1)^T)^T & & & \\ & (\nabla_{u_d{}^2}(f^2)^T)^T & & \\ & & \ddots & \\ & & & (\nabla_{u_d{}^{N_L}}(f^{N_L})^T)^T \end{bmatrix}_{N_L N_y \times N_L N_{u_d}}, \tag{B.52}$$

$$E_{u_d} = \begin{bmatrix} (\nabla_{u_d{}^1}(e^1)^T)^T & & & \\ & (\nabla_{u_d{}^2}(e^2)^T)^T & & \\ & & \ddots & \\ & & & (\nabla_{u_d{}^{N_L}}(e^{N_L})^T)^T \end{bmatrix}_{N_L n_e \times N_L N_{u_d}}, \tag{B.53}$$

and

$$C_{u_d} = \begin{bmatrix} (\nabla_{u_d^1}(c^1)^T)^T & & & \\ & (\nabla_{u_d^2}(c^2)^T)^T & & \\ & & \ddots & \\ & & & (\nabla_{u_d^{N_L}}(c^{N_L})^T)^T \end{bmatrix}_{N_L n_c \times N_L N_{u_d}} . \tag{B.54}$$

In Eq. B.43, we set

$$J_y + \lambda_F^T F_y + \lambda_E^T E_y + \lambda_C^T C_y = 0, \tag{B.55}$$

and

$$J_{u_d} + \lambda_F^T F_{u_d} + \lambda_E^T E_{u_d} + \lambda_C^T C_{u_d} = 0. \tag{B.56}$$

After solving Eq. B.55 and Eq. B.56 for the adjoint vectors λ_F, λ_E and λ_C, the gradient with respect to the free control vector and the slack vector can be obtained as

$$\nabla_{u_f} L = J_{u_f} + \lambda_F^T F_{u_f} + \lambda_E^T E_{u_f} + \lambda_C^T C_{u_f} \tag{B.57}$$

and

$$\nabla_v L = -\lambda_C^T. \tag{B.58}$$

To do optimization, we need the gradient of L with respect to dependent control variables, $\nabla_{u_d} L$, as well. As the equality and inequality constraints are nonlinear in nature, the linearized simulation and constraint equations are solved forward in time for the search direction of dependent variables as follows,

$$(\nabla_{y^{l-1}}(f^l)^T)^T \cdot \Delta y^{l-1} + (\nabla_{y^l}(f^l)^T)^T \cdot \Delta y^l + (\nabla_{u_f^l}(f^l)^T)^T \cdot \Delta u_f^l + (\nabla_{u_d^l}(f^l)^T)^T \cdot \Delta u_d^l = 0, \tag{B.59}$$

$$(\nabla_{y^l}(e^l)^T)^T \cdot \Delta y^l + (\nabla_{u_f^l}(e^l)^T)^T \cdot \Delta u_f^l + (\nabla_{u_d^l}(e^l)^T)^T \cdot \Delta u_d^l = 0, \tag{B.60}$$

and

$$(\nabla_{y^l}(c^l)^T)^T \cdot \Delta y^l + (\nabla_{u_f{}^l}(c^l)^T)^T \cdot \Delta u_f^l + (\nabla_{u_d{}^l}(c^l)^T)^T \cdot \Delta u_d^l = \Delta v^l, \tag{B.61}$$

where $\Delta u_f^l = \nabla_{u_f^l} L$ and $\Delta v^l = \nabla_{v^l} L$ and $\Delta y^0 = 0$. Eq. B.59 through Eq. B.61 can be written in a matrix form as

$$\begin{aligned}&\begin{pmatrix} (\nabla_{y^l}(f^l)^T)^T & (\nabla_{u_d{}^l}(f^l)^T)^T & 0 \\ (\nabla_{y^l}(e^l)^T)^T & (\nabla_{u_d{}^l}(e^l)^T)^T & 0 \\ (\nabla_{y^l}(c^l)^T)^T & (\nabla_{u_d{}^l}(c^l)^T)^T & -I \end{pmatrix} \begin{pmatrix} \Delta y^l \\ \Delta u_d^l \\ \Delta v^l \end{pmatrix} \\ &= \begin{pmatrix} -(\nabla_{y^{l-1}}(f^l)^T)^T \cdot \Delta y^{l-1} - (\nabla_{u_f{}^l}(e^l)^T)^T \cdot \Delta u_f^l \\ -(\nabla_{u_f{}^l}(e^l)^T)^T \cdot \Delta u_f^l \\ -(\nabla_{u_f{}^l}(c^l)^T)^T \cdot \Delta u_f^l \end{pmatrix},\end{aligned} \tag{B.62}$$

where I is an identity matrix with the same dimension of Δv^l. In [15, 49], only active constraints are considered so the number of free variables is equal to the number of active constraints. As a result, the dimension of Eq. B.62 is reduced. The choice of free parameters is claimed to be arbitrary [15]. In Rao [37], the dependant variables are directly solved by

$$\begin{aligned}\begin{pmatrix} \Delta y^l \\ \Delta u_d^l \\ \Delta v^l \end{pmatrix} &= \begin{pmatrix} (\nabla_{y^l}(f^l)^T)^T & (\nabla_{u_d{}^l}(f^l)^T)^T & 0 \\ (\nabla_{y^l}(e^l)^T)^T & (\nabla_{u_d{}^l}(e^l)^T)^T & 0 \\ (\nabla_{y^l}(c^l)^T)^T & (\nabla_{u_d{}^l}(c^l)^T)^T & -I \end{pmatrix}^{-1} \\ &\quad \begin{pmatrix} -(\nabla_{y^{l-1}}(f^l)^T)^T \cdot \Delta y^{l-1} - (\nabla_{u_f{}^l}(e^l)^T)^T \cdot \Delta u_f^l \\ -(\nabla_{u_f{}^l}(e^l)^T)^T \cdot \Delta u_f^l \\ -(\nabla_{u_f{}^l}(c^l)^T)^T \cdot \Delta u_f^l \end{pmatrix}.\end{aligned} \tag{B.63}$$

This way may not be applicable in production optimization problem, as Eq. B.63 involves inversion of big matrix which is possibly singular. [15, 24, 49] do not provide details how to solve the linear equations in Eq. B.62. After solving for Δy^l, Δu_d^l, and Δv^l, the new control vector is updated with

$$u^{l+1} = u^l + \alpha d^l, \tag{B.64}$$

where $d^l = [(\Delta u_f^l)^T, (\Delta u_d^l)^T]^T$ is the search direction and α is the step size and u^{l+1} is the updated control vector at the $(l+1)$th iteration. α is determined by a limiting step size method which ensures all the constraints to be satisfied. As the limiting step size method is very inefficient, Zakirov et al. [49] points out it is necessary to loose the constraint within a certain tolerance in order to accelerate the convergence.

In Eclipse reference manual (version 2006.1), it claims that it uses the following convergence criteria: 1) objective function convergence tolerance between optimization iterations is 10^{-6}; 2) minimum RMS gradient of objective function is 10^{-6}; 3) line search step size convergence tolerance is 10^{-6}; 4) convergence tolerance for control parameter between optimization iterations is 10^{-6}. In the examples using Eclipse 300, we use these default settings for the termination criteria.

APPENDIX C

ENSEMBLE-BASED OPTIMIZATION METHOD

Applying ensemble-based optimization (EnOpt) method to maximize the NPV as a function of the well controls for a given reservoir model requires the generation of an ensemble of controls. This is done by specifying a covariance matrix C_U for the control vector and generating an ensemble of N_e perturbed controls

$$\hat{u}_j^\ell = u_{\text{opt}}^\ell + C_U^{1/2} Z_j \tag{C.65}$$

for $j = 1, 2, \cdots N_e$. Here, each vector Z_j is sampled from the Gaussian distribution $N(0, I)$ where I is the $N_u \times N_u$ identity matrix. The entries in C_U are generated by a spherical covariance function for each well given by,

$$C_{i,j} = \begin{cases} \sigma^2 \left[1 - \frac{3}{2} \left(\frac{|i-j|}{T} \right) + \frac{1}{2} \left(\frac{|i-j|}{T} \right)^3 \right], & \text{if } |i-j| < T, \\ 0, & \text{otherwise}, \end{cases} \tag{C.66}$$

where σ is the standard deviation, T is the correlation time lag and i, j are the control step indices.

These perturbations of controls are used to generate an approximate gradient of NPV (J) at the previous estimate of the optimal set of controls (u_{opt}^*). To do so, one approximates the cross-covariance between u and J as

$$C_{U,J}^\ell = \frac{1}{N_e - 1} \sum_{j=1}^{N_e} (\hat{u}_j^\ell - \overline{\hat{u}^\ell})(J(\hat{u}_j^\ell) - \bar{J}^\ell)^T, \tag{C.67}$$

where the two sample means appearing in Eq. C.67 are defined by

$$\overline{\hat{u}^\ell} = \frac{1}{N_e}\sum_{j=1}^{N_e} \hat{u}_j^\ell \quad \text{and} \quad \bar{J}^\ell = \frac{1}{N_e}\sum_{j=1}^{N_e} J(\hat{u}_j^\ell). \tag{C.68}$$

Somewhat similar to the work of Reynolds et al. [38], who showed the ensemble Kalman filter update is similar to one Gauss-Newton iteration using an average sensitivity matrix to update each ensemble member, [9, 10] derived a formula for ensemble-based optimization (EnOpt) by using the approximations of the following two equations:

$$\bar{J}^\ell \approx J(\overline{\hat{u}^\ell}) \approx J(u_{\text{opt}}^\ell), \tag{C.69}$$

$$J(\hat{u}_j^\ell) \approx J(u_{\text{opt}}^\ell) + \left(\nabla_u J(u_{\text{opt}}^\ell)\right)^T (\hat{u}_j^\ell - u_{\text{opt}}^\ell). \tag{C.70}$$

Note Eq. C.70 is simply a first-order Taylor series approximation. Finally, [9, 10] observed that Eq. C.65 implies that the mean of the sample of perturbed controls should be approximately equal to the estimate of the vector of optimal controls at iteration ℓ, i.e.,

$$\overline{\hat{u}^\ell} \approx u_{\text{opt}}^\ell. \tag{C.71}$$

Using the approximations of Eqs. C.68, C.69 and C.70 in Eq. C.67, it follows that

$$C_{U,J}^\ell = \frac{1}{N_e - 1}\sum_{j=1}^{N_e} (\hat{u}_j^\ell - \overline{\hat{u}^\ell})(\hat{u}_j^\ell - \overline{\hat{u}^\ell})^T \nabla_u J(u_{\text{opt}}^\ell) \approx C_U \nabla_u J(u_{\text{opt}}^\ell). \tag{C.72}$$

Thus, the cross-covariance between the controls and the NPV function is approximately equal to C_U times the gradient, i.e., a preconditioned steepest ascent direction or a Newton-type method with C_U used to approximate the inverse Hessian. Instead of using this preconditioned steepest descent direction as the search direction, however, [9, 10] multiplied again by C_U for additional smoothing to obtain the search direction

$$d_\ell = C_U C^\ell_{U,J} \approx C_U^2 \nabla_u J(u^\ell_{\text{opt}}), \tag{C.73}$$

so that the EnOpt algorithm has the form

$$u^{\ell+1}_{\text{opt}} = u^\ell_{\text{opt}} + \alpha_\ell d_\ell = u^\ell_{\text{opt}} + \alpha_\ell C_U C^\ell_{U,J}, \tag{C.74}$$

where α_ℓ is the step size. If $J(u^{\ell+1}_{\text{opt}}) \leq J(u^\ell_{\text{opt}})$, then $u^{\ell+1}_{\text{opt}}$ is not accepted as the new estimate of the vector of optimal controls; instead we decrease the step-size and reapply Eq. C.74.

Chen et al. [10] provided no information on how to initialize the stepsize α^0_ℓ for each iteration or perhaps more importantly what to do if d_ℓ is not an uphill direction. Note because Eqs. C.73 and C.74 are based on a series of approximations, the gradient that appears in Eq. C.73 is only an approximation to the true gradient. Thus, there is no way to prove that d_ℓ given in Eq. C.73 is an uphill direction. When Eq. C.73 gives a downhill direction, cutting the stepsizes cannot be expected to finally arrive at a control vector which will increase the value the NPV function, J. Our experience is that Eq. C.73 sometimes gives a downhill direction.

Because of the comments of the preceding paragraph, our implementation of EnOpt is as follows: First replace Eq. C.74 by

$$u^{\ell+1}_{\text{opt}} = u^\ell_{\text{opt}} + \alpha_\ell \frac{C_U C^\ell_{U,J}}{\parallel C_U C^\ell_{U,J} \parallel_\infty}. \tag{C.75}$$

After normalization of the search direction, we set the initial stepsize α^0_ℓ to be approximately equal to around 1/10 of the length of the smallest interval specifying bounds on a component of the control vector, i.e.,

$$\alpha^0_\ell = \frac{1}{10} \min_i \{u^{\text{up}}_i - u^{\text{low}}_i\}. \tag{C.76}$$

For example, if bounds on wellbore pressure controls in psi are $1500 \leq p_{wf} \leq 3500$, and the bounds on the liquid flow rate in STB/D at each well are $0 \leq q_\ell \leq 10000$, then an appropriate value of α_ℓ^0 would be 200. Unfortunately, this is just a rule of thumb and setting the initial step size inappropriately can severely decrease the computational efficiency of the algorithm as well as result in a suboptimal value of the NPV. Although a good value of α_ℓ^0 can usually be found by experimentation with a couple of test runs, such experimentation is also computationally expensive. The choice of C_U is also somewhat ad hoc and also effects the performance of the algorithm. In the examples presented here, we simply define C_U from a spherical covariance function with specified variance and correlation "length" given in the example section. This covariance function is applied on a well by well basis so there is no correlation between controls at any two different wells. If with $\alpha_\ell = \alpha_\ell^0$, the $u_{\text{opt}}^{\ell+1}$ computed from Eq. C.75 does not increase the NPV, then the step size is cut in half and $u_{\text{opt}}^{\ell+1}$ is recomputed with this reduced step size. This process of cutting the stepsize continues until we have obtained an increase in the NPV or we have performed the maximum number of step size reductions that are allowed, which is 6 in the examples presented here. If with the maximum number of stepsize cuts, a $u_{\text{opt}}^{\ell+1}$ has not been found such that $J(u_{\text{opt}}^{\ell+1}) > J(u_{\text{opt}}^{\ell})$, then we generate a new set of perturbed controls from Eq. C.65 for use in Eq. C.67 until we find an uphill search direction. The algorithm is terminated when 1) the relative increase of the objective function is less than 10^{-4}, i.e.,

$$\frac{J(u_{\text{opt}}^{\ell+1}) - J(u_{\text{opt}}^{\ell})}{J(u_{\text{opt}}^{\ell})} \leq 10^{-4}, \tag{C.77}$$

and 2) the ℓ_2 norm of the relative change in the control vector is less than 10^{-3}, i.e.,

$$\frac{\|u_{\text{opt}}^{\ell+1} - u_{\text{opt}}^{\ell}\|}{\max(\|u_{\text{opt}}^{\ell}\|, 1.0)} \leq 10^{-3}. \tag{C.78}$$