NTNU – Trondheim
Norwegian University of
Science and Technology

Mathematical Optimization in Reservoir Management

Bahador Najafiazar

Petroleum Engineering
Submission date: July 2014
Supervisor: Jon Kleppe, IPT
Co-supervisor: Bjarne Foss, ITK

Norwegian University of Science and Technology
Department of Petroleum Engineering and Applied Geophysics

Mathematical Optimization in Reservoir Management

BAHADOR NAJAFIAZAR

MASTER THESIS

SUBMITTED FOR THE PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

Department of Petroleum Engineering and Applied Geophysics
Faculty of Engineering Science and Technology
Norwegina University of Science and Technology

July 8, 2014

Summary

Getting the most out of a hydrocarbon reservoir is not a trivial task. It takes plenty of interwoven decisions to make. There are many forms of tools that support engineers to make correct decisions. The simplest ones would only display measurements in a suitable way, and appoint the rest of the decision making process to human knowledge and experience. Complex decision support tools may implement model-based estimation and optimization. This work targets methods for optimization-based decision support.

The objective of this study is to formulate, implement and test promising methods of hydrocarbon production optimization through various test cases. To do this, a various optimizations algorithm were applied to the simulated reservoir models using the *Matlab Reservoir Simulation Toolbox* (MRST).

Preface

This Master's Thesis is submitted to the Norwegian University of Science and Technology (NTNU), Trondheim, Norway, for partial fulfillment of the requirements for the degree of Master of Science. The study has been conducted in collaboration between the department of Petroleum Engineering and Applied Geophysics, and the department of Engineering Cybernetics, under the supervision of professor Jon Kleppe and co-supervision of professor Bjarne A. Foss and Dr. Stein Krogstad.

I owe my deepest gratitude to my supervisors Professor Jon Kleppe, Professor Bjarne A. Foss and Dr. Stein Krogstad for the opportunity given to me as their student. Their valueable comments and interesting discussions as well as their patience made it possible to complete this project.

Also, I would like to thank Mansoureh Jesmani for her friendly feedback, encouragement and support during the course of the project.

Bahador Najafiazar
July 8, 2014

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Energy is inevitable for human life. The sustainability of modern societies depends crucially on a secure and accessible supply of energy. The non-stop use of fossil fuels is set to face numerous challenges, one of which is the seemingly inexorable depletion of fossil fuel reserves. Oil remains the world's leading fuel, accounting for 33.1% of global energy consumption in 2012 (BP2, 2013), as shown in Figure 1.1.

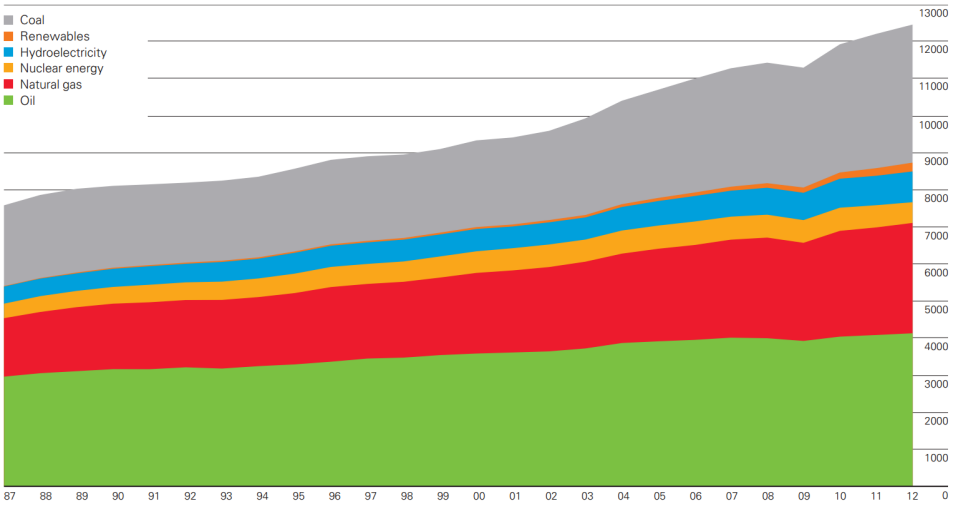

Figure 1.1: World consumption of primary energy, million tons of oil equivalent(BP2, 2013).

Most of the existing major oilfields are already at a mature stage, and the number of new significant discoveries per year is decreasing (Brouwer, 2004). Smaller fields are still regularly found, but at the current oil price it is often not economical to exploit them. As a direct result it becomes more and more difficult to maintain economic reserves at a desirable level.

In order to satisfy the growing worldwide demand for oil and gas, it is becoming increasingly important to produce existing fields as efficiently as possible, while simultaneously decreasing development and operating costs. Optimal control theory is one possible approach that can be deployed to address these difficult issues Sarma et al. (2005). The main benefit of the use of optimal control theory is its efficiency, which makes it suitable for application to real reservoirs simulated using large models, in contrast to many existing techniques.

1.1 Report Structure

The rest of this report is organized as follows:

Chapter 2 presents background material for subsequent chapters. The chapter gives an introduction to various solution techniques for optimization problems. It provides a comparison between local and gloabal optimization algorithms as well as a brief description of linear and nonlinear programming. The chapter also demonstrates the several necessary steps for derivation of a mathematical model for two-phase flow through porous media. The model is simplified by making a number of assumptions and then discretized to become for numerical schemes. Further, it introduces the MATLAB-based reservoir simulator which was implemented in this study. It gives descriptions on the reservoir model and optimization algorithms implemented into the toolbox. At the end of this chapter, based on the author's experience with the simulator, some of its positive and negative aspects are discussed.

Chapter 3 performs a literature review and presents state-of-the-art on methods that include uncertainty, in particular robust optimization and closed loop techniques.

Chapter 4 formulates, implements and tests promising methods on a simple example as well as a more elaborate example, namely the five-spot model and the egg model. The results are discussed and analyzed.

Chapter 5 offers concluding remarks and recommendations for further work.

Chapter 2

Basic Theory and Background

2.1 An Overview of Optimization Techniques

This section aimes at giving the reader a basic overview of the various solution techniques for typical optimization problems.

Before starting this review, it is useful to introduce the standard form of the general optimization problem which will be solved using the techniques presented in the upcoming sections. The standard form for a single-objective, non-linear, constrained optimization problem is (Nocedal and Wright, 2006)

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^n} \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & c_i(\mathbf{x}) = 0, \quad i \in \mathcal{E} \\ & c_i(\mathbf{x}) \geq 0, \quad i \in \mathcal{I} \end{aligned} \tag{2.1}$$

where f is the *objective function*, while $c_i, i \in \mathcal{E}$ are the *equality constraints* and and $c_i, i \in \mathcal{I}$ are the *inequality constraints*. The vector $\mathbf{x}$ has $n \geq 1$ components (sometimes referred to as *design variables*) and f and the functions c_i are all smooth, real-valued functions on a subset of $\mathbb{R}^n$, and $\mathcal{E}$ and $\mathcal{I}$ are two finite sets of indices.

In the general case, the objective and constraint functions can be linear or nonlinear and can be explicit or implicit functions. Implicit functions commonly appear when, for example, a numerical simulation (e.g., a finite element simulation) is used to evaluate a response function (e.g., a pressure value). Also, the variables vector $\mathbf{x}$ need not be continuous.

Optimization problems can have some or all of the variables $\mathbf{x}$ restricted to integer or discrete values (Venter, 2010). These types of problems are referred to as integer or discrete optimization problems. Generally, the local algorithms have difficulty in solving optimization problems with integer and/or discrete variables, while several global algorithms are well adapted to this class of problems.

Optimization techniques are algorithms used to find the solution to the problem specified in (2.1). Their mission is to find the combination of design variable $\mathbf{x}$ values that provide the best objective function value, while satisfying all the equality and inequality constraints. Many problems have more than one optimum (referred to as local or relative optima). Some algorithms aim at finding local optima, while others seek to find the optimal solution among all possible solutions (not just those in a particular neighborhood of values). The latter are called Global optimization techniques (Venter, 2010).

There are many ways to classify the available optimization techniques. One way is to divide the whole range of methods into Local and Global techniques. The rest of this chapter covers a comparison of local and globoal algorithms followed by a brief introduction to two broad classes of optimization problems, *linear programming* (LP) and *nonlinear programming* (NLP).

2.1.1 Local Optimization Algorithms

Most local optimization algorithms are gradient-based (Venter, 2010). As suggested by the name, gradient-based optimization techniques exploit gradient information to find the optimum solution of (2.1). Gradient-based algorithms are used for solving a broad range of optimization problems in engineering. These techniques have gained popularity because of their efficiency (in terms of the number of function evaluations required to find the optimum), ability to solve problems with large numbers of design variables, and typically little problem-specific parameter tuning requirement. These algorithms, however, have several drawbacks which include locating only local optimua (and not gloabal ones), having difficulty solving discrete optimization problems, complexity of algorithms making them difficult to implement efficiently, and susceptiblity to numerical noise.

Gradient-based algorithms typically implement a two-step process to find the optimum. This process can be explained by means of a physical example (Vanderplaats, 2007). Consider a blindfolded boy on a hill. He wishes to find the highest point on the hill (the objective function), while staying inside the two fences (the constraints). Here, the design variables are the x and y coordinates of the boy. Now, because of the blindfold, he cannot just look up the hill and go straight to the

"optimum" point. He may take a small step in the x direction and a small step in the y direction. From this, he can sense the slope of the hill and then search in the upward direction. In a mathematical sense, what he has done is calculate the direction of steepest ascent by calculating the gradient by finite difference methods. He can then start walking in this direction until no more progress is made, which may include reaching a fence. At this point the boy can again take two small steps to determine a new direction that will take him uphill, while staying inside the fences, and continue the process until he reaches the top of the hill.

The mathematical summarization of this two-step iterative process of finding the optimum can be:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k, \tag{2.2}$$

where the first step is to use gradient information for finding a *search direction* $\mathbf{p}_k$ in which to move. The second step is to move in this direction until no more progress can be made. The size of this progression is determined by the *step-length* α_k. This algorithm is known as the *Line Search Method*. There are also gradient-based algorithms that do not rely on a line search, such as the *Trust Region Method*.

Choosing the Search Direction

Line search methods can select among several choices for search directions $\mathbf{p}_k$ (Nocedal and Wright, 2006). The *steepest descent method* is a line search method that moves along $\mathbf{p}_k = -\nabla f_k$ at every step. One advantage of this direction is that it requires calculation of the gradient ∇f_k but not the second derivative. However, it can be agonizingly slow on complex problems. Another important search direction is the *Newton direction*. It is derived from the second order Taylor series approximation to $f(\mathbf{x}_k + \mathbf{p})$, by simply setting its derivative with repect to $\mathbf{p}$ equal to zero. As a result we get

$$\mathbf{p}_k = -(\nabla^2 f_k)^{-1} \nabla f_k. \tag{2.3}$$

Methods that use the Newton direction have a fast rate of local convergence, typically quadratic. The main drawback of the Newton direction is the need for the Hessian $\nabla^2 f(\mathbf{x})$. Explicit computation of this matrix of second derivatives can sometimes be a cumbersome, error-prone, and expensive process. In order to avoid such difficulties, *Quasi-Newton* search directions have been developed. They provide an attractive alternative to Newton's method in that they do not require computation of the Hessian and yet still attain a superlinear rate of convergence. In place of the true Hessian $\nabla^2 f_k$, they use an approximation $\mathbf{B}_k$, which is updated after each step to take account of the additional knowledge gained during the step.

The updates make use of the fact that changes in the gradient provide information about the second derivative of f along the search direction.

Calculating Derivatives

Many practical applications require the optimization of functions whose derivatives are not readily available (Nocedal and Wright, 2006). Problems of this kind can be solved, in principle, by approximating the gradient (and possibly the Hessian) using finite differences. Finite difference gradients provide a flexible means of estimating the gradient information. However, when used, they typically dominate the total computing time required to complete an optimization study (Venter, 2010). When the designer has access to the function source code, *automatic differentiation* (see e.g., (Griewank and Walther, 2008)) can be used to obtain the required gradient information. Automatic differentiation is the generic name for techniques that use the computational representation of a function to produce analytic values for the derivatives. These techniques have the benefit of providing gradient information accurate to working precision. To the contrary, finite difference calculations provide only an approximation to the gradient, with the accuracy depending on the selected step size.

The KKT Conditions

When gradient information is available, the *Karush-Kuhn-Tucker conditions* or *KKT conditions* can be used to determine if a constrained local optimum has been found. The KKT conditions provide the first order necessary conditions for a local optimum and can be summarized as:

1. $\mathbf{x}^*$, the local solution of (2.1), must be feasible.
2. The gradient of the Lagrangian function for the general problem (2.1) must vanish at this local solution $\mathbf{x}^*$:
$$\nabla f(\mathbf{x}^*) - \sum_{i \in \mathcal{E} \cup \mathcal{I}} \lambda_i^* \nabla c_i(\mathbf{x}^*) = 0, \tag{2.4}$$
where λ_i are the Lagrange multipliers (Nocedal and Wright, 2006) and we have $\lambda_{i \in \mathcal{I}}^* \geq 0$, whereas $\lambda_{i \in \mathcal{E}}^*$ are unrestricted in sign.
3. For all $i \in \mathcal{E} \cup \mathcal{I}$ we have $\lambda_i^* c_i(\mathbf{x}^*) = 0$.

The KKT conditions are suitable for indicating whether a local optimum has been found. However, they cannot identify global optima.

2.1.2 Global Optimization Algorithms

Many problems have multiple optima, with a simple univariate function as shown in Figure 2.1. In this figure, point A represents a local (or relative) minimum, while point B represents a global (or absolute) minimum in the feasible region. The local algorithms discussed earlier will converge on any of these points, depending on which one is encountered first.

Superficially, global optimization is just a stronger version of local optimization, whose great usefulness in practice is undisputed (Neumaier, 2004). Instead of searching for a locally unimprovable feasible point, one wants the globally best point in the feasible region (point B in Figure 2.1). In many practical applications, such as problems in the petroleum industry, finding the globally best point is desirable but not essential, since any sufficiently good feasible point is useful and usually an improvement over what is available without optimization. For such problems, there is little harm in doing an incomplete search[1]; and indeed, this is all that can be achieved for many large-scale problems or for problems where function values (and perhaps derivatives) are a available only through a black box routine that does not provide global information. Global optimization is much more difficult than convex programming or finding local minimizers of nonlinear programs, since the gap between the necessary (Karush-Kuhn-Tucker) conditions

[1] An **incomplete** method uses clever intuitive heuristics for searching but has no safeguards if the search gets stuck in a local minimum.

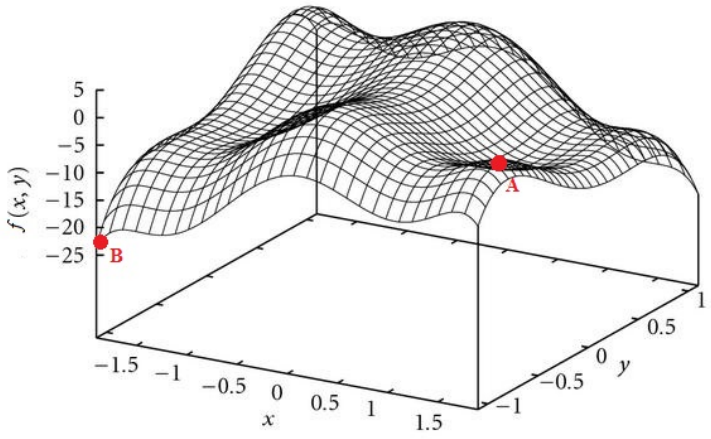

Figure 2.1: A function with several local optima (such as point A) and a global optimum (point B) (Passaro and Starita, 2008).

for optimality and known sufficient conditions for global optimality is tremendous.

Evolutionary optimization algorithms are a large branch of global optimization techniques and have gained popularity in the last couple of decades. In contrast to the local techniques, where a single design point is updated (typically using gradient information) at each iteration, these algorithms do not require any gradient information and typically implement a set of design points (generally referred to as a *population*) to find the global optimum. These methods are typically inspired by some natural phenomena. Their advantages include being extremely robust, having an increased chance of finding a global or near global optimum, being easy to implement, and being well suited for discrete optimization problems. The big drawbacks associated with evolutionary algorithms are high computational cost, poor constraint-handling capabilities, problem-specific parameter tuning and limited problem size.

Currently, two of the most popular evolutionary algorithms are the more established Genetic Algoithm (GA) (Holland, 1975), which was inspired by Darwin's principle of survival of the fittest, and Particle Swarm Optimization (PSO) (Kennedy and Eberhart, 1995), which is based on a simplified social model.

The interested reader is referred to the excellent survey of global optimization algorithms provided by Neumaier (2004) for more detailed information on evolutionary as well as deterministic algorithms.

2.1.3 Linear Programming

The development of linear programming (LP) has been ranked among the most important scientific advances of the mid-20th century (Hillier and Lieberman, 2010). Its impact since just 1950 has been extraordinary. Today it is a standard tool that has saved many thousands or millions of dollars for most companies or businesses of even moderate size in the various industrialized countries of the world.

A linear program is class of problems with a linear objective function and linear equality and/or inequality constraints. The feasible region is a polytope[2]. The contours of the linear objective function are planar, and the solution is at the intersection of contours with a subset of the linear constraints.

The standard form (Nocedal and Wright, 2006) of a linear program is usually stated

[2] A convex, connected set with flat polygonal faces.

as follows:

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^n} \quad & \mathbf{c}^\top \mathbf{x} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} = \mathbf{b}, \\ & \mathbf{x} \geq 0, \end{aligned} \tag{2.5}$$

where $\mathbf{c}$ and $\mathbf{x}$ are vectors in $\mathbb{R}^n$, $\mathbf{b}$ is a vector in $\mathbb{R}^m$, and $\mathbf{A}$ is an $m \times n$ matrix. Convexity of (2.5) ensures that the Karush-Kuhn-Tucker (KKT) conditions are sufficient for a global minimum.

There are two classes of algorithms which have proven to be efficient for LP problems, namely the simplex and the interior-point methods.

The Simplex Method

The simplex method has a number of variants, but the one briefly described here is known as the *revised simplex method*. The principle of this method is to move from one vertex of the polytope to an adjacent one for which the basis[3] differs in exactly one component. On most steps (but not all), the value of the objective function $\mathbf{c}^\top \mathbf{x}$ decreases. Another type of step occurs when the problem is unbounded, where one can move infinitely far without ever reaching a vertex. A major issue at each simplex iteration is to decide which variable to remove from the basis and which one to bring in from outside the basis, and different strategies exist (see (Dantzig and Thapa, 2003) for a comprehensive discussion on simplex algorithms).

Interior-Point Methods

The *primal-dual methods*, a subclass of interior-point methods, have distinguished themselves as the most efficient practical approaches, and proved to be strong competitors to the simplex method on large problems (Nocedal and Wright, 2006).

Consider the linear programming problem in standard form (2.5). The KKT conditions for the optimal solution of this problem are

$$\mathbf{A}^\top \boldsymbol{\lambda} + \mathbf{s} = \mathbf{c}, \tag{2.6a}$$

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \tag{2.6b}$$

$$x_i s_i = 0, \quad i = 1, 2, \ldots, n, \tag{2.6c}$$

$$(\mathbf{x}, \mathbf{s}) \geq 0, \tag{2.6d}$$

[3] A set $\mathcal{B}$ or basis is defined for the problem at hand as a subset of the index set $\{1, 2, \ldots, n\}$ such that $\mathcal{B}$ contains exactly m indices, and $i \notin \mathcal{B} \Rightarrow x_i = 0$.

where $\boldsymbol{\lambda} \in \mathbb{R}^m$ and $\mathbf{s} \in \mathbb{R}^n$ are the Lagrangian multipliers for the equality and inequality constraints, respectively.

Primal-dual methods find solutins $(\mathbf{x}^*, \boldsymbol{\lambda}^*, \mathbf{s}^*)$ of this system by applying variants of Newton's method[4] to the three equalities in (2.6) and modifying the search directions and step lengths so that the inequalities $(\mathbf{x}, \mathbf{s}) \geq 0$ are satisfied *strictly* at every iteration[5]. The equations (2.6a), (2.6b), (2.6c) are linear or only mildly nonlinear and so are not difficult to solve by themselves. However, the problem becomes much more difficult when we add the nonnegativity requirement (2.6d), which gives rise to all the complications in the design and analysis of interior-point methods (see (Boyd and Vandenberghe, 2009) for an extensive discussion on interior-point methods).

2.1.4 Nonlinear Programming

The general constrained optimization problem (2.1) is called a nonlinear programming when the objective function f or the constraints c_i are not linear. As LP problems are well studied over several decades, only some of the subclasses of NLPs are equally well treated. Hence, there are no efficient methods for solving general NLPs. Even simple looking ones with as few as ten variables can be extremely challenging, while problems with a few hundred variables can be intractable.

NLP problems can be divided into two main classes, convex and non-convex problems. A convex optimization problem is one in which the objective function and the feasible set are convex. The reason for dividing the NLP problems into these groups is that for the convex ones, all local optimal solutions are also global optimal solutions. For the nonconvex problems however, this is not true, since there is no way to characterize the global solution using only local information. It may be hard to verify convexity for NLPs. In such cases it is necessary to assume that they are non-convex.

The most popular solution methods for NLPs are interior-point (or barrier) methods and Sequential quadratic programming (SQP) algorithms. These methods are generally considered the most powerful algorithms for large-scale nonlinear programming. The principal ideas of interior-point methods and SQP are briefly described below.

[4] Choosing a Newton direction (2.3).

[5] This is property is the origin of the term interior-point.

Sequential Quadratic Programming

Sequential Quadratic Programming (SQP) is an efficient method for solving nonlinear constrained optimization. The approach can be used both in line search and trust-region frameworks, and is appropriate for both small and large problems. The idea behind the SQP approach is to approximate the NLP problem (2.1) by a convex quadratic programming (QP) subproblem at the current iterate $\mathbf{x}_k$ as follows:

$$\min_{\mathbf{p}} \quad \tfrac{1}{2}\mathbf{p}^\top \nabla^2_{xx}\mathcal{L}(\mathbf{x}_k, \boldsymbol{\lambda}_k)\mathbf{p} + \nabla f(\mathbf{x}_k)^\top \mathbf{p} \tag{2.7a}$$

$$\text{s.t.} \quad \nabla c_i(\mathbf{x}_k)^\top \mathbf{p} + c_i(\mathbf{x}_k) = 0, \quad i \in \mathcal{E} \tag{2.7b}$$

$$\nabla c_i(\mathbf{x}_k)^\top \mathbf{p} + c_i(\mathbf{x}_k) \geq 0, \quad i \in \mathcal{I} \tag{2.7c}$$

The solution of this QP problem, using for instance an active-set method, provides a search direction $\mathbf{p}$. The active-set method has several similarities to the simplex method, which in turn is an active set method for LP problems. In essence, the simplex method starts by making a guess of the optimal active set, then repeatedly use dual information to drop one constraint from the current estimate of active constraints, and add a new one, until optimality is detected. Active-set methods for convex QPs differ from the simplex method in that the iterates are not necessarily vertices of the feasible polytope. They finn a step from one iterate to the next by solving a quadratic subproblem in which some of the inequality constraints, and all equality constraints are imposed as equalities i.e. the active-set. The algorithm terminates when the solution does not change from one iteration to the next, and the Lagrangian multiplier for all the active inequalities are zero or negative.

Interior-Point Methods

Some of the key ideas, such as primal-dual steps, carry directly over from LP. However, the treatment of non-convexity, the strategy for updating the barrier parameter in the presence of nonlinearities, and the need to ensure progress toward the solution, are several of the important new challenges that arise. These include the treatment of nonconvexity, the strategy for updating the barrier parameter in the presence of nonlinearities, and the need to ensure progress toward the solution (Nocedal and Wright, 2006; Boyd and Vandenberghe, 2009).

2.1.5 Production Optimization

The goal of production optimization problem is finding the right combination of well settings, i.e., the BHP and the flow rates, to maximize an economic objective function, namely the *net present value* (NPV).

The optimization problem can be formulated as follows:

$$\begin{aligned} & u^* = \arg\max_u J \\ & s.t.\ f(x_{k+1}, x_k, u_k) = 0, \quad k = 1, \ldots, K_T \\ & g(u_k) \leq 0, \quad x_0 = \hat{x_0} \end{aligned} \tag{2.8}$$

where J denotes the objective function and f the reservoir model. The initial conditions are represented by $\hat{x_0}$, and g are the constraints corresponding to the lower and upper bounds on the injection rates and the bhp. The optimal control actions are denoted by u^*.

The discrete form of NPV is widely used and is defined by

$$J = \sum_{k=1}^{K_T} \frac{r_o q_{o,k} - r_w q_{w,k} - r_i q_{i,k}}{(1+b)^{\frac{t_k}{\tau_t}}} \Delta t_k \tag{2.9}$$

where r_o denotes the fixed oil price, and r_w and r_i are the water production and the water injection costs, respectively, all of which are assumed to be constant. To account for depreciation, the discount rate, b, is added for a certain reference time, τ_t. The final time step is K_T, and Δt_k denotes the time interval at the k^{th} time step. The oil production, water production and water injection rates are denoted by $q_{o,k}$, $q_{w,k}$, and $q_{i,k}$, respectively, at the time step k.

A variety of methods are available in the literature for solving problem (2.53). If the gradient can be computed efficiently, gradient-based optimization becomes promisingly efficient to deal with large-scale systems, e.g. production optimization in oil reservoirs. The adjoint method has been used in many papers for similar problems, see for example Jansen (2011). Of the few existing methods for calculating gradients, adjoint techniques are the most efficient, especially for a large number of controls, as the algorithm is independent of the number of controls (Sarma et al., 2005).

In this study, the adjoint method developed into the MATLAB Reservoir Simulation Toolbox (MRST) is used to determine the gradient of the objective function with respect to the control settings. In the rest of this chapter, we first introduce MRST briefly. Next, we present a brief derivation of the reservoir model equations

and the adjoint model implemented into the toolbox (See Krogstad and Gulbransen (2011) for further details).

2.1.6 The Implemented Optimization Algorithm

The *OptimizeObjective* function under the *adjoint* module of MRST uses an aggressive line search based on the given gradient. The algorithm handles box-constraints and linear equality and (probably not) inequality constraints. It performs on an iterative scheme, applying the constraints to the gradient until convergence is achieved.

The algorithm makes use of the *gradient projection method* which is explained in the following section.

The Gradient Projection Method

The gradient projection method is most efficient when the constraints are simple in form–in particular, when there are only bounds on the variables (Nocedal and Wright, 2006). A general form of this type of constraints is

$$\mathbf{l} \leq \mathbf{x} \leq \mathbf{u}, \tag{2.10}$$

where $\mathbf{l}$ and $\mathbf{u}$ are vectors of lower and upper bounds on the components of $\mathbf{x}$. The feasible region defined by 2.71 is sometimes called a "box" because of its rectangular shape.

Each iteration of the gradient projection algorithm consists of two stages. In the first stage, we search along the steepest descent direction from the current point $\mathbf{x}$. Whenever a bound is encountered, the search direction is "bent" so that it stays feasible. This is illustrated in Figure 2.5.

We search along the resulting piecewise-linear path trying to locate the first local minimizer of the objective function f, which we denote by $\mathbf{x}^c$ and refer to as the *Cauchy point*. The working set[6] is now defined to be the set of bound constraints that are active at the Cauchy point, denoted by $\mathcal{A}(\mathbf{x}^c)$. In the second stage of each gradient projection iteration, we search on the face of the feasible box on which the Cauchy point lies. To do this, we solve a subproblem in which the active components x_i for $i \in \mathcal{A}(\mathbf{x}^c)$ are fixed at the values x_i^c.

[6]Primal active-set methods find a step from one iterate to the next by solving a quadratic subproblem in which some of the inequality constraints, and all the equality constraints, are imposed as equalities. This subset is referred to as the *working set* and is denoted at the kth iterate $\mathbf{x}_k$ by $\mathcal{W}_k$.

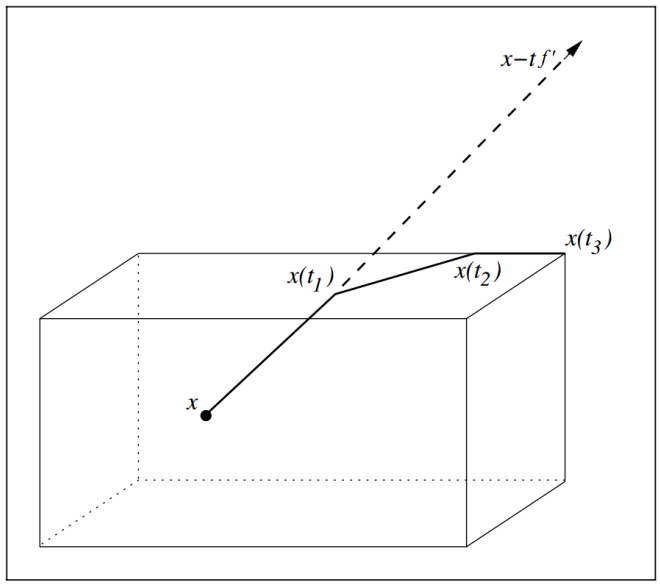

Figure 2.2: The gradient projection method (Nocedal and Wright, 2006).

Line Search Method

The gradient projection method projects the gradient according to constraints iteratively until a certain tollerance is met or a certain number of iterations is met (where it returnes failure). The constraints are applied in the following order:

1. box constraints,
2. linear inequality constraints,
3. linear equality constraints.

The resulting norm of the projected gradient is used as stopping criteria.

Next, a line search is performed along the projected gradient. The step length α in (2.2) for this line search is multiplied by three values $[0\ 0.5\ 1]$ to get $[\alpha_1\ \alpha_2\ \alpha_3]$. Now, based on a comparison of the objective function values for the three new step lengths, a decision is made:

(a) If the projected gradient is equal for α_2 and α_3, then we are on the boundary, done.

(b) If the value of the objective function follows $f(\alpha_1) \leq f(\alpha_2) \leq f(\alpha_3)$, then multiply the current step size by 2 and go to (a).

(c) If $f(\alpha_1) \geq f(\alpha_2))$, then multiply the current step size by 0.5 and go to (a).

(d) Find the minimum on the approximated quadratic curve through $(\alpha_1, \alpha_2, \alpha_3)$, done.

2.2 Reservoir Modeling

The primary objective of a reservoir study is to predict future performance of a reservoir and find ways and means of increasing ultimate recovery. Classical reservoir engineering deals with the reservoir on a gross average basis (tank model) and cannot account adequately for the variations in reservoir and fluid parameters in space and time (Aziz and Settari, 1979). Reservoir simulation allows a more detailed study of the reservoir by dividing the reservoir into a number of blocks (sometimes several thousand) and applying fundamental equations for flow in porous media to each block.

The physical system to be modelled must be expressed in terms of appropriate mathematical equations. This process almost always involves assumptions. The assumptions are necessary from a practical standpoint in order to make the problem tractable.

In this work, as an example of a reservoir model we consider the flow of oil and water through a heterogeneous porous medium. We make the strongly simplifying assumptions that the reservoir is horizontal and of constant height, and that gravity and capillary forces can be neglected. We also assume the reservoir is in its secondary recovery phase where the pressures are above the bubble point pressure of the oil phase. Therefore, two-phase immiscible flow, that is, no mass transfer between the two liquid phases, is a fair assumption. We focus on water-flooding cases for two-phase (oil and water) reservoirs. Further, we assume incompressible fluids and rocks, no-flow boundaries, and isothermal conditions.

To implement the reservoir model, we do not develop an oil reservoir simulator from scratch. Rather, we use an open source MATLAB toolbox described in section 2.2.5. The assumptions stated earlier are covered in the toolbox.

2.2.1 Mass Balance

Let $\Omega \subset \mathbb{R}^d (d \leq 3)$ be a porous medium domain of volume V with boundary $\partial\Omega$ of area s, and let $\mathbf{n}$ be the outward pointing unit normal on the boundary as illustrated in Figure 2.3.

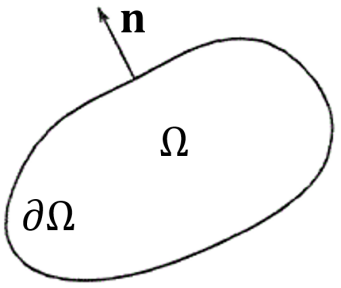

Figure 2.3: Porous medium Ω in two dimensional space $(d = 2)$.

Now, conservation of some quantity c can be formulated as

$$\frac{\partial}{\partial t} \int_\Omega c \, dV + \oint_{\partial\Omega} \mathbf{F} \cdot \mathbf{n} \, ds = \int_\Omega q \, dV, \tag{2.11}$$

Where $\mathbf{F}$ is the mass flux and q is the source/sink term. Equation (2.11) implies that the rate of change inside Ω (the term on the left-hand side) is equal to the rate of mass entering or leaving through the boundary $\partial\Omega$ and the rate of mass contributed by sources or sinks (the terms on the right-hand side). Using Gauss's divergence theorem we get

$$\int_\Omega (\frac{\partial c}{\partial t} + \nabla \cdot \mathbf{F}) \, dV = \int_\Omega q \, dV, \tag{2.12}$$

from which we can obtain the continuity equation

$$\frac{\partial c}{\partial t} + \nabla \cdot \mathbf{F} = q. \tag{2.13}$$

2.2.2 Immiscible Two-Phase Flow Formulations

In reservoir simulation, we are primarily concerned with modelling the displacement, within a porous medium, of oil by either water or gas. While the displacing fluid may be immiscible with the fluid being displaced, the displacement does not

take place as a piston-like process with a sharp interface between the two fluids (Peaceman, 1977). Rather, simultaneous flow of the two immiscible fluids takes place within the porous medium.

In considering this simultaneous flow we assume no mass transfer between the two fluids. One of the fluids wets the porous medium more than the other; we refer to this as the wetting phase fluid, and we refer to the other as the nonwetting phase fluid. In a water-oil system, water is most often the wetting phase.

Except in cases of steam flooding or in-situ combustion we can assume that reservoir flow is isothermal, which implies that we may disregard the energy balance equation (Jansen et al., 2008). Moreover, the movement of fluids is usually so slow that we can disregard inertial effects, and that instead of the momentum balance equation we may use an empirical relationship between pressure drop and flow velocity known as Darcy's law.

We now write the continuity equation (2.13) for the case of a two-phase (oil-water) flow. Here, the conserved quantity c is the mass of each phase $\rho_\alpha \phi S_\alpha$, where ρ is fluid density, ϕ is porosity, S is fluid saturation of the pore space $(0 \leq S \leq 1)$, and the subscript $\alpha \in \{o, w\}$ indicates the oil and water phases, respectively. The density and the porosity are independent of pressure as we assume that the rock and the fluid are incompressible. Substituting for mass flux $\mathbf{F}_\alpha = \rho_\alpha \mathbf{v}_\alpha$ in (2.13), where $\mathbf{v}$ is the (superficial) fluid velocity results in

$$\frac{\partial(\rho_\alpha \phi S_\alpha)}{\partial t} + \nabla \cdot (\rho_\alpha \mathbf{v}_\alpha) = \tilde{q}_\alpha. \tag{2.14}$$

Darcy's law can be expressed as

$$\mathbf{v}_\alpha = -\frac{k_{r\alpha}}{\mu_\alpha} \mathbf{K}(\nabla p_\alpha - \rho_\alpha g \nabla d), \tag{2.15}$$

where $\mathbf{K}$ is the permeability tensor, μ fluid viscosity, k_r relative permeability, p pressure, g acceleration of gravity and d depth. The ratio $k_{r\alpha}/\mu_\alpha$ is called the *phase mobility* λ_α. Employing the no gravity assumption, Darcy's law will simplify to

$$\mathbf{v}_\alpha = -\lambda_\alpha \mathbf{K} \nabla p_\alpha. \tag{2.16}$$

The permeability tensor $\mathbf{K}$, whose elements have units of surface area, represents how easily the fluids flow through the rock in different directions. Usually the orientation of the coordinate system can be aligned with the geological layering in the reservoir such that $\mathbf{K}$ is a diagonal matrix:

$$\mathbf{K} = diag(k_x, k_y, k_z), \tag{2.17}$$

where k_x, k_y, and k_z are directional permeabilities in the x, y and z coordinate directions. The dimensionless relative permeabilities $k_{r\alpha}$ are functions of S_α, and are reduction factors that represent the increase in flow resistance caused by multi-phase effects. The resistance to concurrent flow of oil and water is generally much higher than the sum of the resistances to flow of the individual phases, and the relative permeabilities are therefore a major source of nonlinearity in the multi-phase equations. Relative permeability data are obtained from laboratory experiments using small portions of rock which do not generally represent the rock properties of the whole reservoir. Hence, uncertainties are unavoidable. In this work we assume the relative permeability follows the *Corey* model (Aziz and Settari, 1979)

$$S_{N,\alpha} = \frac{S - S_{\alpha r}}{1 - S_{wr} - S_{or}}, \tag{2.18a}$$

$$k_{r\alpha} = k^0_{r\alpha}(S_{N,\alpha})^{n_\alpha}, \tag{2.18b}$$

where $S_{N,\alpha}$ is the normalized water saturation, S_{wr} and S_{or} are the residual water and oil saturations, $k^0_{r\alpha}$ is the endpoint relative permeability for phase α, and n_α is the empirical coeffiecient for phase α. Typically, relative permeability is considered a quadratic function of water saturation as shown in Figure 2.4.

Substituting (2.16) into (2.14) and noting that the porosity and density are independent of pressure we get

$$\phi \frac{\partial S_\alpha}{\partial t} + \nabla \cdot \mathbf{v}_\alpha = \frac{\tilde{q}_\alpha}{\rho_\alpha}. \tag{2.19}$$

Equation (2.19) (one for each phase) contain four unknowns, p_w, p_o, S_w and S_o, two of which can be eliminated with aid of the relationships

$$S_w + S_o = 1, \tag{2.20a}$$

$$p_o - p_w = p_c(S_w), \tag{2.20b}$$

where $p_c(S_w)$ is the oil-water capillary pressure which is another source of nonlinearity in the flow equations. However, the no capillary assumption simplifies (2.20b) to $p_o = p_w = p$. It is common to choose the water saturation S_w as the primary unknown variable and define it as a variable S. Now, the primary variables are the pressure p and water saturation S.

Defining the *total velocity* as $\mathbf{v} = \mathbf{v}_o + \mathbf{v}_w$ and applying it to (2.16) and (2.19) results in

$$\mathbf{v} = -\lambda_t \mathbf{K} \nabla p, \quad \text{within } \Omega \tag{2.21a}$$

$$\nabla \cdot \mathbf{v} = q, \quad \text{within } \Omega \tag{2.21b}$$

$$\mathbf{v} \cdot \mathbf{n} = 0, \quad \text{on } \partial\Omega \tag{2.21c}$$

where q is the total volumetric rate and λ_t is the total mobility defined as

$$q = q_o + q_w = \frac{\tilde{q}_o}{\rho_o} + \frac{\tilde{q}_w}{\rho_w} \tag{2.22}$$

$$\lambda_t(S) = \lambda_o + \lambda_w = \frac{k_{ro}}{\mu_o} + \frac{k_{rw}}{\mu_w}. \tag{2.23}$$

We refer to the combination of (2.21a) and (2.21b) with respective boundary condition (2.21c) as the *pressure equations*, which is an elliptic PDE in this case. Now, we obtain water velocity from (2.21a) as follows

$$\mathbf{v}_w = -\lambda_w \mathbf{K}\nabla p = -\frac{\lambda_w}{\lambda_t}\mathbf{v} = -\frac{\lambda_w}{\lambda_o + \lambda_w}\mathbf{v}. \tag{2.24}$$

Referring to $\frac{\lambda_w}{\lambda_o+\lambda_w}$ as the *water fractional flow*, denoting it by $f_w(S)$, and substituting (2.24) for the water phase of (2.19) we obtain

$$\phi\frac{\partial S}{\partial t} + \nabla \cdot f_w(S)\mathbf{v} = q_w. \tag{2.25}$$

We refer to (2.25) as the *saturation equation* which is a hyperbolic PDE in this case.

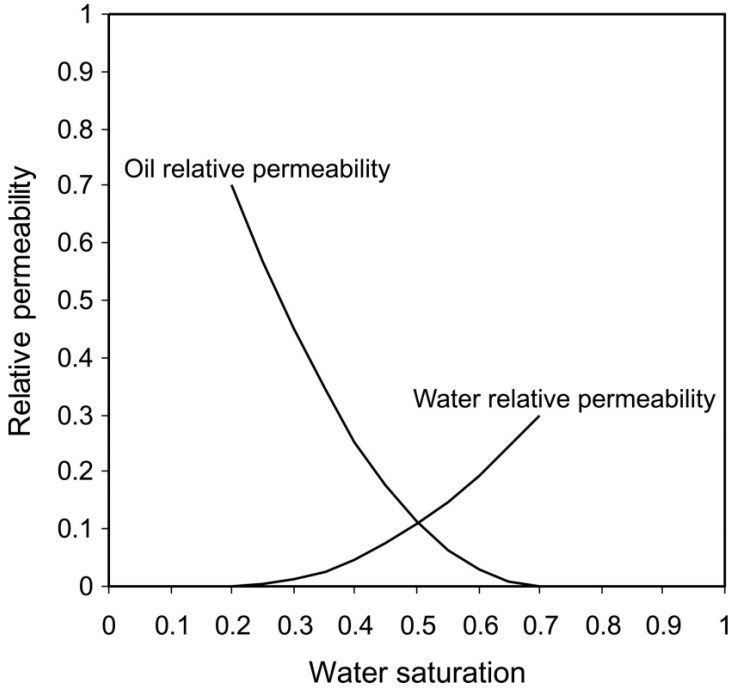

Figure 2.4: Typical relative permeability curves.

2.2.3 Discritization Method

The pressure and saturation equations (2.21) and (2.25) are referred to as the state equations. It is impossible to find an analytical solution of the state equations for typical reservoir models. Hence, one approximates the solutions by some numerical method. In reservoir simulation literature typical methods of choice are the finite difference, finite-volume, and finite-element methods. We do not differentiate the finite-difference and finite-volume method, in the sense that the finite-volume method is a conservative finite-difference scheme that treats the grid cells as control volumes (Aarnes et al., 2007).

However, before proceeding we need to select how to solve the state equations using a sequential time step scheme. In the reservoir simulation literature both implicit and explicit methods are applied. The implicit method is unconditionally stable while the explicit is potentially more efficient but restricted by numerical stability conditions. In addition to the most common fully implicit scheme, the solution strategy which can be used in reservoir management is the IMPES (IMplicit Pressure and Explicit Saturation) method. However, in this work we use the other way around, i.e., explicit-pressure and implicit-saturation. First, the strategy computes relative permeabilities using the initial water saturation. Second, the pressure equation is solved using the initial water saturation and the secondary variables values. Third, with the obtained pressure solution, the velocity is computed and is used to solve the saturation equation. This procedure is repeated until the final time is reached.

We begin with the pressure equations (2.21). In this work we use a cell-centred finite-volume method, which is known as the *two-point flux-approximation* (TPFA) scheme. We discretize the domain Ω into a number of grid blocks (k) such that $\Omega_i \in \Omega$ and $i = 1, 2, \ldots, k$. After some rearrangement, (2.21) can be written as

$$\nabla \cdot (-\lambda_t \mathbf{K} \nabla p) = q. \tag{2.26}$$

The left-hand side of (2.26), after discretization, is called the transmissibilities. (2.26) is a linear equation, $\mathbf{A}(S)\mathbf{p} = \mathbf{q}$, the left-hand side of which is represented by $\mathbf{A}$, a symmetric matrix whose elements are given by

$$a_{ik} = \begin{cases} \sum_j t_{ij} & \text{if } k = i, \\ -t_{ik} & \text{if } k \neq i. \end{cases} \tag{2.27}$$

In a Cartesian grid, matrix $\mathbf{A}$ is a tridiagonal matrix for 1D, pentadigonal for 2D, and heptadiagonal for 3D cases. Note that the discretization above is a spatial discretization of the pressure equation. We need to discretize the equation in time

as well. To perform temporal discretization for the pressure equation, we use a finite difference operator and end up with

$$\mathbf{A}(\mathbf{S}^{n-1})\mathbf{p}^n = \mathbf{B}\mathbf{u}^n. \tag{2.28}$$

Here, we have set the right-hand side as a function of a control input vector $\mathbf{u}^n$ for time step n, which can either be well rates or well pressures (bottom hole pressure/BHP). Also, matrix $\mathbf{B}$ is the arrangement matrix of the control inputs.

Now, we intend to discretize the saturation equation (2.25). We apply the finite difference operator to obtain

$$\frac{\phi_i}{\Delta t}(\mathbf{S}_i^{n+1} - \mathbf{S}_i^n) + \frac{1}{|\Omega_i|}\sum_{j\neq i} R_{ij}(\mathbf{S}^{n+1}) = \frac{q_{w,i}(\mathbf{S}_i^n)}{\rho_w}. \tag{2.29}$$

The porosity in Ω_i is denoted by ϕ_i and R_{ij} is the approximation of the velocity at the edge γ_{ij}, which is

$$R_{ij} \approx \int_{\gamma_{ij}} (f_w(S)_{ij}\mathbf{v}_{ij}) \cdot \mathbf{n}_{ij}\, ds \tag{2.30}$$

where $\mathbf{n}_{ij}$ is the normal vector. The water fractional flow at the edge is approximated by using upstream weighting, such that

$$f_w(S)_{ij} = \begin{cases} f_w(S_i) & \text{if } \mathbf{v}\cdot\mathbf{n}_{ij} \geq 0, \\ f_w(S_j) & \text{if } \mathbf{v}\cdot\mathbf{n}_{ij} < 0, \end{cases} \tag{2.31}$$

This gives the following discrete form of the saturation equation

$$\mathbf{S}^n = \mathbf{S}^{n-1} + \Delta t^n \mathbf{D}_{PV}^{-1}(\mathbf{M}(\mathbf{v}^n) f_w(\mathbf{S}^n) + \mathbf{q}(\mathbf{v}^n)_+). \tag{2.32}$$

Here, Δt^n is the time step and $\mathbf{D}_{PV}$ is the diagonal matrix containing the grid block pore volumes. The matrix $\mathbf{M}(\mathbf{v}^n)$ is the sparse flux matrix based on the upstream weighted discretization scheme, and $\mathbf{q}(\mathbf{v}^n)_+$ is the vector of positive sources (in this setting, water injection rates) (Suwartadi et al., 2010). We note that the matrix $\mathbf{M}$ and vector $\mathbf{q}$ are linear functions of $\mathbf{v}^n$, where $\mathbf{v}^n = \mathbf{T}(\mathbf{S}^{n-1})\mathbf{p}^n$ and $\mathbf{T}(\mathbf{S}^{n-1})$ is a matrix containing the transmissibilities and well indices based on $\mathbf{S}^{n-1}$. We refer to (2.32) as the discretized saturation equation.

The discrete state equations (2.28) and (2.32) can be written in an implicit form $\mathbf{F}(\tilde{\mathbf{x}}, \tilde{\mathbf{u}}) = 0$ as

$$\mathbf{F}(\tilde{\mathbf{x}}, \tilde{\mathbf{u}}) = \begin{bmatrix} \mathbf{F}^0(\mathbf{p}^1, \mathbf{S}^0, \mathbf{S}^1, \mathbf{u}^1) \\ \vdots \\ \mathbf{F}^{N-1}(\mathbf{p}^N, \mathbf{S}^{N-1}, \mathbf{S}^N, \mathbf{u}^N) \end{bmatrix} \begin{aligned} \mathbf{x}^{nT} &= (\mathbf{p}^{nT}, \mathbf{S}^{nT}), \quad n = 1, \ldots, N, \\ \tilde{\mathbf{x}}^T &= (\mathbf{x}^{1T}, \ldots, \mathbf{x}^{NT}), \\ \tilde{\mathbf{u}^T} &= (\mathbf{u}^{1T}, \ldots, \mathbf{u}^{NT}). \end{aligned} \tag{2.33}$$

The state vectors and control input vectors are stacked for all time instances from $n = 1, \ldots, N$.

2.2.4 Boundary Conditions

Due to no-flow boundary conditions, the driving forces of the reservoir models are at the wells. Injector wells inject water while producer wells produce both oil and water. At the wells we can either fix the rate or bottom-hole pressure (BHP). The former amounts to the Neumann boundary condition while the latter gives the so-called Dirichlet boundary condition.

In this study, the well rate is implemented by using the Peaceman well model (Peaceman, 1983), that is

$$q_t = \lambda_t WI(p_{wf} - p_{gf}) \tag{2.34}$$

where *WI* denotes the well index, p_{wf} is the BHP, and p_{gf} is the well-block pressure. The well index is described by the following equation

$$WI = 2\pi \frac{dz \sqrt[3]{k_x k_y k_z}}{V_{gb}(\ln(\frac{r_o}{r_w}) + S)} \tag{2.35}$$

where dz is the well-segment length, V_{gb} the volume of the well grid block, r_w the radius of the well, k_x, k_y, k_z are the permeabilities in the x, y, z directions, respectively, and r_o is the effective well radius which is expressed as

$$r_o = 0.28 \frac{\left[\sqrt{\frac{k_x}{k_y}}(\Delta y)^2 + \sqrt{\frac{k_y}{k_x}}(\Delta x)^2\right]^{\frac{1}{2}}}{\sqrt[4]{\frac{k_x}{k_y}} + \sqrt[4]{\frac{k_y}{k_x}}} \tag{2.36}$$

The grid block length in the x and y direction are denoted by Δx and Δy , respectively.

2.2.5 The Implemented Reservoir Model

In order to build the case study models into the simulator, the following assumptions were considered:

1. two-phase, incompressible flow,

2. no gravity,

3. zero capillary pressure,

4. No-flow at the outer boundary of the reservoir.

Assumption 1 neglects the presence of gas in the system. The pressure and saturation equations for the model are given by:

$$\nabla.\vec{v} = q, \quad \vec{v} = -\boldsymbol{\lambda}(s_w)\mathbf{K}\nabla p, \tag{2.37}$$

$$\phi\frac{\partial s_w}{\partial t} + \nabla.[f_w(s_w)\vec{v}] = q_w, \tag{2.38}$$

where $\vec{v}$ denotes the Darcy velocity, $\lambda = \lambda_o + \lambda_w$ is total mobility, s_w is water saturation, $\mathbf{K}$ is the permeability tensor, p is pressure, q is the fluid source term , ϕ is effective porosity, $f_w = \lambda_w/\lambda$ is fractional flow of water, and q_w is the water source term.

Let $\mathbf{v}$ denote the outward face fluxes and well-perforation rates, $\mathbf{s}$ saturation cell, $\mathbf{p}$ pressure cell, and π face pressures and well pressures. The system corresponding to the pressure equation (2.55) at time t^n is written in the form

$$\begin{bmatrix} \mathbf{B}(\mathbf{s}^{n-1}) & \mathbf{C} & \mathbf{D} \\ \mathbf{C}^\top & \mathbf{0} & \mathbf{0} \\ \mathbf{D}^\top & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{v}^n \\ -\mathbf{p}^n \\ \boldsymbol{\pi}^n \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{v}_\Gamma^n \end{bmatrix} \tag{2.39}$$

where $\mathbf{B}(\mathbf{s}^{n-1})$ and $\mathbf{C}$ are block diagonal matrices. The block $\mathbf{B}$ corresponding to a cell i is $[\lambda(\mathbf{s}_i^{n-1})\mathbf{T}_i]^{-1}$ with an extended diagonal entry $[\lambda(\mathbf{s}_i^{n-1})\mathbf{WI}_i]^{-1}$ for every well perforating cell i. Here, $\mathbf{T}_i$ denotes the transmissibility matrix, while $\mathbf{WI}_i$ is the well productivity index. Each block in matrix $\mathbf{C}$ has an $n_i \times 1$ vector of ones, where n_i is the number of faces plus the number of well perforations in cell i. Further, each column of the matrix $\mathbf{D}$ corresponds to a unique face or well face, and has unit entries in the positions of the face/well-face in the cell wise oredering. The vector $\mathbf{v}_\Gamma^n$ has a zero entry corresponding to every face. Splitting the vector $\boldsymbol{\pi}^{n\top} = \begin{bmatrix} \hat{\boldsymbol{\pi}}^{n\top} & \boldsymbol{\pi}_D^{n\top} \end{bmatrix}$ and the matrix $\mathbf{D} = \begin{bmatrix} \hat{\mathbf{D}} & \mathbf{D}_D^n \end{bmatrix}$ the system becomes

$$\begin{bmatrix} \mathbf{B}(\mathbf{s}^{n-1}) & \mathbf{C} & \hat{\mathbf{D}} \\ \mathbf{C}^\top & \mathbf{0} & \mathbf{0} \\ \hat{\mathbf{D}}^\top & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{v}^n \\ -\mathbf{p}^n \\ \hat{\boldsymbol{\pi}}^n \end{bmatrix} = \begin{bmatrix} \mathbf{A}_D^n\mathbf{u} + \mathbf{b}_D^n \\ \mathbf{0} \\ \mathbf{A}_N^n\mathbf{u} + \mathbf{b}_N^n \end{bmatrix} \tag{2.40}$$

where $\mathbf{A}_D^n$ and $\mathbf{A}_N^n$ are sparse matrices and $\mathbf{u}$ denotes a control input vector of piecewise constant pressures or rates for a given set of wells at timestep t^n. The

subscript D corresponds to bhp-controlled wells (Dirichlet boundary conditions) while the subscript N corresponds to rate-controlled wells (Neumann boundary conditions). The vectors $\mathbf{b}_D^n$ and $\mathbf{b}_N^n$ are all zeros except at entries corresponding to non-controllable boundary conditions (see Lie et al. (2012) for more details).

Next, using the standard upstream weighted implicit finite-volume method, the saturation equation 2.56 may be written in the following form

$$\mathbf{s}^n = \mathbf{s}^{n-1} + \delta t^n \mathbf{D}_{PV}^{-1}[\mathbf{A}(\mathbf{v}^n) f(\mathbf{s}^n) + \mathbf{q}(\mathbf{v}^n)_+] \tag{2.41}$$

where δt^n is the timestep, $\mathbf{D}_{PV}$ is the diagonal matrix containing the cell pore volumes, $\mathbf{A}(\mathbf{v}^n)$ is the sparse flux matrix, and $\mathbf{q}(\mathbf{v}^n)_+$ is the vector of positive cell sources (injection rates).

2.2.6 Application of the Adjoint Method

MRST implements an adjoint model consisting of a multi-scale pressure solver and a saturation solver which works on flow-adapted grids (Lie et al., 2012). Let $\mathbf{x}^n = (\mathbf{v}^n, \mathbf{p}^n, \pi^n, \mathbf{s}^n)$ and $\mathbf{u}^n$ be the state and control input variables, respectively. Each time step of (2.58)-(2.59) can be re-written in a compact form as

$$F^n(\mathbf{x}^n, \mathbf{x}^{n-1}, \mathbf{u}^n). \tag{2.42}$$

The pressure and saturation equations (2.58)-(2.59) may be written in a compact form as $F(\mathbf{x}, \mathbf{u})$. Let $J(\mathbf{x}, \mathbf{u})$ be an objective function with $\nabla_{\mathbf{u}} J$ being the gradient of J with respect to $\mathbf{u}$. Introducing an auxiliary function J_α as

$$J_\alpha(\mathbf{x}, \mathbf{u}) = J(\mathbf{x}, \mathbf{u}) + \boldsymbol{\alpha}^\top F(\mathbf{x}, \mathbf{u}) \tag{2.43}$$

where $\boldsymbol{\alpha} = \boldsymbol{\alpha}(\boldsymbol{\alpha}_\mathbf{v}, \boldsymbol{\alpha}_\mathbf{p}, \boldsymbol{\alpha}_\mathbf{s}, \boldsymbol{\alpha}_{\hat{\boldsymbol{\pi}}})$ is a vector of Lagrange multipliers. The gradient $\nabla_{\mathbf{u}} J_{\boldsymbol{\alpha}}$ is given by

$$\nabla_{\mathbf{u}} J_{\boldsymbol{\alpha}}^\top = \frac{\partial J}{\partial \mathbf{u}} + \frac{\partial J}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial \mathbf{u}} + \boldsymbol{\alpha}^\top \frac{\partial F}{\partial \mathbf{u}} + \boldsymbol{\alpha}^\top \frac{\partial F}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial \mathbf{u}} + F^\top \frac{\partial \boldsymbol{\alpha}}{\partial \mathbf{u}} \tag{2.44}$$

which reduces to

$$\nabla_{\mathbf{u}} J_{\boldsymbol{\alpha}}^\top = \frac{\partial J}{\partial \mathbf{u}} + \boldsymbol{\alpha}^\top \frac{\partial F}{\partial \mathbf{u}} \tag{2.45}$$

while $\boldsymbol{\alpha}$ satisfies the adjoint equation

$$\frac{\partial F^\top}{\partial \mathbf{x}} \boldsymbol{\alpha} = -\frac{\partial J^\top}{\partial \mathbf{x}}. \tag{2.46}$$

In (2.64), $\frac{\partial F}{\partial \mathbf{x}}$ has a block structure with blocks of $\frac{\partial F^n}{\partial \mathbf{x}^n}$ on the main diagonal and $\frac{\partial F^{n+1}}{\partial \mathbf{x}^n}$ on the lower diagonal. Applying this to (2.64), the adjoint equation for time step n is given by

$$\frac{\partial F^{n\top}}{\partial \mathbf{x}^n}\boldsymbol{\alpha}^n + \frac{\partial F^{n+1\top}}{\partial \mathbf{x}^n}\boldsymbol{\alpha}^{n+1} = -\frac{\partial J^{n\top}}{\partial \mathbf{x}^n}\boldsymbol{\alpha}^n. \tag{2.47}$$

(2.65) is solved backwards in time in order to obtain $\boldsymbol{\alpha}$. Now the gradient with respect to time step n is computed from the follwoing equation

$$\nabla_{\mathbf{u}^n} J_{\boldsymbol{\alpha}} = \frac{\partial J^\top}{\partial \mathbf{u}} + \frac{\partial F^{n\top}}{\partial \mathbf{u}^n}\boldsymbol{\alpha}^n \tag{2.48}$$

Defining

$$g(\mathbf{v}, \mathbf{s}) = \delta t \mathbf{D}_{PV}^{-1}[\mathbf{A}(\mathbf{v})f(s) + \mathbf{q}(\mathbf{v})_+] \tag{2.49}$$

and substituting (2.58)-(2.59) into (2.65), the adjoint equations for time step n are obtained as

$$\left[\mathbf{I} - \frac{\partial g(\mathbf{v}^n, \mathbf{s}^n)}{\partial \mathbf{s}^n)}\right]^\top \boldsymbol{\alpha}_s^n = \boldsymbol{\alpha}_s^{n+1} - \frac{\partial J^{n\top}}{\partial \mathbf{s}^n} - \left[\frac{\partial \mathbf{B}(\mathbf{s}^n) v^{n+1}}{\partial \mathbf{s}^n}\right]^\top \boldsymbol{\alpha}_v^{n+1} \tag{2.50}$$

$$\begin{bmatrix} \mathbf{B}(\mathbf{s}^{n-1}) & \mathbf{C} & \hat{\mathbf{D}} \\ \mathbf{C}^\top & \mathbf{0} & \mathbf{0} \\ \hat{\mathbf{D}}^\top & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha}_v^n \\ \boldsymbol{\alpha}_p^n \\ \boldsymbol{\alpha}_{\hat{\pi}}^n \end{bmatrix} = \begin{bmatrix} -\frac{\partial J^{n\top}}{\partial \mathbf{v}^n} + \frac{\partial g(\mathbf{v}^n, \mathbf{s}^n)^\top}{\partial \mathbf{v}^n}\boldsymbol{\alpha}_s^n \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \tag{2.51}$$

Once the adjoint equations have been solved, the gradient of the objective function J at time step n is given by

$$\nabla_{\mathbf{u}^n} J = \frac{\partial J^\top}{\partial \mathbf{u}^n} - \mathbf{A}_D^{n\top}\boldsymbol{\alpha}_v^n - \mathbf{A}_N^{n\top}\boldsymbol{\alpha}_{\hat{\pi}}^n \tag{2.52}$$

This gradients (2.70) can be used in any gradient based methods, one of which will be discussed in the following chapters.

The rest of the sections in this chapter includes a definition of the production optimization problem. Further, the simulator implemented in this study is introduced. The simulator is part of the MATLAB Reservoir Simulation Toolbox (MRST). Also, the reservoir model and the adjoint method developed into the simulator will be presented. Finally, some of the positive aspects as well as shortcomings of MRST will be discussed.

2.3 Production Optimization

The goal of production optimization problem is finding the right combination of well settings, i.e., the BHP and the flow rates, to maximize an economic objective function, namely the *net present value* (NPV).

The optimization problem can be formulated as follows:

$$\begin{aligned} & u^* = \arg\max_u J \\ & s.t.\ f(x_{k+1}, x_k, u_k) = 0, \quad k = 1, \ldots, K_T \\ & g(u_k) \leq 0, \quad x_0 = \hat{x_0} \end{aligned} \tag{2.53}$$

where J denotes the objective function and f the reservoir model. The initial conditions are represented by $\hat{x_0}$, and g are the constraints corresponding to the lower and upper bounds on the injection rates and the bhp. The optimal control actions are denoted by u^*.

The discrete form of NPV is widely used and is defined by

$$J = \sum_{k=1}^{K_T} \frac{r_o q_{o,k} - r_w q_{w,k} - r_i q_{i,k}}{(1+b)^{\frac{t_k}{\tau_t}}} \Delta t_k \tag{2.54}$$

where r_o denotes the fixed oil price, and r_w and r_i are the water production and the water injection costs, respectively, all of which are assumed to be constant. To account for depreciation, the discount rate, b, is added for a certain reference time, τ_t. The final time step is K_T, and Δt_k denotes the time interval at the k^{th} time step. The oil production, water production and water injection rates are denoted by $q_{o,k}$, $q_{w,k}$, and $q_{i,k}$, respectively, at the time step k.

A variety of methods are available in the literature for solving problem (2.53). If the gradient can be computed efficiently, gradient-based optimization becomes promisingly efficient to deal with large-scale systems, e.g. production optimization in oil reservoirs. The adjoint method has been used in many papers for similar problems, see for example (Jansen, 2011). Of the few existing methods for calculating gradients, adjoint techniques are the most efficient, especially for a large number of controls, as the algorithm is independent of the number of controls (Sarma et al., 2005).

In this study, the adjoint method developed into the MATLAB Reservoir Simulation Toolbox (MRST) is used to determine the gradient of the objective function with respect to the control settings. In the rest of this chapter, we first introduce

MRST briefly. Next, we present a brief derivation of the reservoir model equations and the adjoint model implemented into the toolbox (See (Krogstad and Gulbransen, 2011) for further details).

2.4 The MATLAB Reservoir Simulation Toolbox (MRST)

MRST is developed by *SINTEF Applied Matemathics* and is a result of their research on the development of new (multiscale) computational methodologies. Version 2013b which was used in this study, is available online for free download under the terms of the GNU General Public License (GPL) (Lie et al., 2012).

The toolbox consists of two main parts: a core offering basic functionality and single and two-phase solvers, and a set of add-on modules offering more advanced models, viewers and solvers.

2.4.1 Core

The toolbox employs the following components for rapid prototyping of solvers for flow and transport:

- *Grids:* a common data structure and interface for all types of grids.
- *Parameters:* a data structure for petrophysical parameters (and a few, very simplified geostatistical routines); common interface for fluid models (there are slight differences in the data structure of the fluid models created with different modules which can be problematic– explained later in the chapter), routines for setting and manipulating boundary conditions, sources/sinks, well models, etc.
- *Reservoir state:* data structure for pressure, fluxes, saturations, ...
- *Solvers:* the toolbox contains several flow and transport solvers (including IMPES and fully implicit).
- *Linear algebra:* MRST relies on MATLAB's builtin linear solvers.
- *Eclipse input:* routines for reading and processing input files created for Schlumberger's Eclipse simulator, supporting grids, petrophysical parameters, fluid models, wells, boundary conditions, simulation setup, etc.

- *Units:* MRST works in strict SI units but supports conversion to/from other unit systems such as field units. Unless reading from an Eclipse input format, the user is responsible for explicit conversion and consistency of units.

2.4.2 Modules

MRST also contains a series of add-on modules, increasing the applicability of the package. They include:

- *Fully implicit solvers:* This module contains a set of fully implicit solvers for a variety of flow problems. The module uses automatic differentiation to calculate Jacobians which makes prototyping of new models faster.
- *IMPES solver:* This module contains an implementation of a pressure/transport solver using an Implicit Pressure, Explicit Saturation (IMPES) strategy for compressible black-oil flow.
- *Deck reader:* The module contains support for input of complete simulation decks in the ECLIPSE format, including input reading, conversion to SI units, and construction of MRST objects for grids, fluids, rock properties, and wells.
- *Adjoint formulations:* This module implements strategies for production optimization based on adjoint formulations. This enables for instance net present value optimization constrained by the bottom hole pressure in wells.

2.5 Reservoir Model

In order to build the case study models into the simulator, the following assumptions were considered:

1. two-phase, incompressible flow,
2. no gravity,
3. zero capillary pressure,
4. No-flow at the outer boundary of the reservoir.

Assumption 1 neglects the presence of gas in the system. The pressure and saturation equations for the model are given by:

$$\nabla.\vec{v} = q, \quad \vec{v} = -\boldsymbol{\lambda}(s_w)\mathbf{K}\nabla p, \tag{2.55}$$

$$\phi\frac{\partial s_w}{\partial t} + \nabla.[f_w(s_w)\vec{v}] = q_w, \tag{2.56}$$

where $\vec{v}$ denotes the Darcy velocity, $\lambda = \lambda_o + \lambda_w$ is total mobility, s_w is water saturation, $\mathbf{K}$ is the permeability tensor, p is pressure, q is the fluid source term , ϕ is effective porosity, $f_w = \lambda_w/\lambda$ is fractional flow of water, and q_w is the water source term.

Let $\mathbf{v}$ denote the outward face fluxes and well-perforation rates, $\mathbf{s}$ saturation cell, $\mathbf{p}$ pressure cell, and π face pressures and well pressures. The system corresponding to the pressure equation (2.55) at time t^n is written in the form

$$\begin{bmatrix} \mathbf{B}(\mathbf{s}^{n-1}) & \mathbf{C} & \mathbf{D} \\ \mathbf{C}^\top & \mathbf{0} & \mathbf{0} \\ \mathbf{D}^\top & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{v}^n \\ -\mathbf{p}^n \\ \boldsymbol{\pi}^n \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{v}_\Gamma^n \end{bmatrix} \tag{2.57}$$

where $\mathbf{B}(\mathbf{s}^{n-1})$ and $\mathbf{C}$ are block diagonal matrices. The block $\mathbf{B}$ corresponding to a cell i is $[\lambda(\mathbf{s}_i^{n-1})\mathbf{T}_i]^{-1}$ with an extended diagonal entry $[\lambda(\mathbf{s}_i^{n-1})\mathbf{WI}_i]^{-1}$ for every well perforating cell i. Here, $\mathbf{T}_i$ denotes the transmissibility matrix, while $\mathbf{WI}_i$ is the well productivity index. Each block in matrix $\mathbf{C}$ has an $n_i \times 1$ vector of ones, where n_i is the number of faces plus the number of well perforations in cell i. Further, each column of the matrix $\mathbf{D}$ corresponds to a unique face or well face, and has unit entries in the positions of the face/well-face in the cell wise oredering. The vector $\mathbf{v}_\Gamma^n$ has a zero entry corresponding to every face. Splitting the vector $\boldsymbol{\pi}^{n\top} = \begin{bmatrix} \hat{\boldsymbol{\pi}}^{n\top} & \boldsymbol{\pi}_D^{n\top} \end{bmatrix}$ and the matrix $\mathbf{D} = \begin{bmatrix} \hat{\mathbf{D}} & \mathbf{D}_D^n \end{bmatrix}$ the system becomes

$$\begin{bmatrix} \mathbf{B}(\mathbf{s}^{n-1}) & \mathbf{C} & \hat{\mathbf{D}} \\ \mathbf{C}^\top & \mathbf{0} & \mathbf{0} \\ \hat{\mathbf{D}}^\top & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{v}^n \\ -\mathbf{p}^n \\ \hat{\boldsymbol{\pi}}^n \end{bmatrix} = \begin{bmatrix} \mathbf{A}_D^n\mathbf{u} + \mathbf{b}_D^n \\ \mathbf{0} \\ \mathbf{A}_N^n\mathbf{u} + \mathbf{b}_N^n \end{bmatrix} \tag{2.58}$$

where $\mathbf{A}_D^n$ and $\mathbf{A}_N^n$ are sparse matrices and $\mathbf{u}$ denotes a control input vector of piecewise constant pressures or rates for a given set of wells at timestep t^n. The subscript D corresponds to bhp-controlled wells (Dirichlet boundary conditions) while the subscript N corresponds to rate-controlled wells (Neumann boundary conditions). The vectors $\mathbf{b}_D^n$ and $\mathbf{b}_N^n$ are all zeros except at entries corresponding to non-controllable boundary conditions (see (Lie et al., 2012) for more details).

Next, using the standard upstream weighted implicit finite-volume method, the saturation equation 2.56 may be written in the following form

$$\mathbf{s}^n = \mathbf{s}^{n-1} + \delta t^n \mathbf{D}_{PV}^{-1}[\mathbf{A}(\mathbf{v}^n)f(\mathbf{s}^n) + \mathbf{q}(\mathbf{v}^n)_+] \tag{2.59}$$

where δt^n is the timestep, $\mathbf{D}_{PV}$ is the diagonal matrix containing the cell pore volumes, $\mathbf{A}(\mathbf{v}^n)$ is the sparse flux matrix, and $\mathbf{q}(\mathbf{v}^n)_+$ is the vector of positive cell sources (injection rates).

2.6 Adjoint Method

MRST implements an adjoint model consisting of a multi-scale pressure solver and a saturation solver which works on flow-adapted grids (Lie et al., 2012). Let $\mathbf{x}^n = (\mathbf{v}^n, \mathbf{p}^n, \pi^n, \mathbf{s}^n)$ and $\mathbf{u}^n$ be the state and control input variables, respectively. Each time step of (2.58)-(2.59) can be re-written in a compact form as

$$F^n(\mathbf{x}^n, \mathbf{x}^{n-1}, \mathbf{u}^n). \tag{2.60}$$

The pressure and saturation equations (2.58)-(2.59) may be written in a compact form as $F(\mathbf{x}, \mathbf{u})$. Let $J(\mathbf{x}, \mathbf{u})$ be an objective function with $\nabla_{\mathbf{u}} J$ being the gradient of J with respect to $\mathbf{u}$. Introducing an auxiliary function J_α as

$$J_\alpha(\mathbf{x}, \mathbf{u}) = J(\mathbf{x}, \mathbf{u}) + \boldsymbol{\alpha}^\top F(\mathbf{x}, \mathbf{u}) \tag{2.61}$$

where $\boldsymbol{\alpha} = \boldsymbol{\alpha}(\boldsymbol{\alpha}_{\mathbf{v}}, \boldsymbol{\alpha}_{\mathbf{p}}, \boldsymbol{\alpha}_{\mathbf{s}}, \boldsymbol{\alpha}_{\hat{\pi}})$ is a vector of Lagrange multipliers. The gradient $\nabla_{\mathbf{u}} J_{\boldsymbol{\alpha}}$ is given by

$$\nabla_{\mathbf{u}} J_{\boldsymbol{\alpha}}^\top = \frac{\partial J}{\partial \mathbf{u}} + \frac{\partial J}{\partial \mathbf{x}}\frac{\partial \mathbf{x}}{\partial \mathbf{u}} + \boldsymbol{\alpha}^\top \frac{\partial F}{\partial \mathbf{u}} + \boldsymbol{\alpha}^\top \frac{\partial F}{\partial \mathbf{x}}\frac{\partial \mathbf{x}}{\partial \mathbf{u}} + F^\top \frac{\partial \boldsymbol{\alpha}}{\partial \mathbf{u}} \tag{2.62}$$

which reduces to

$$\nabla_{\mathbf{u}} J_{\boldsymbol{\alpha}}^\top = \frac{\partial J}{\partial \mathbf{u}} + \boldsymbol{\alpha}^\top \frac{\partial F}{\partial \mathbf{u}} \tag{2.63}$$

while $\boldsymbol{\alpha}$ satisfies the adjoint equation

$$\frac{\partial F^\top}{\partial \mathbf{x}}\boldsymbol{\alpha} = -\frac{\partial J^\top}{\partial \mathbf{x}}. \tag{2.64}$$

In (2.64), $\frac{\partial F}{\partial \mathbf{x}}$ has a block structure with blocks of $\frac{\partial F^n}{\partial \mathbf{x}^n}$ on the main diagonal and $\frac{\partial F^{n+1}}{\partial \mathbf{x}^n}$ on the lower diagonal. Applying this to (2.64), the adjoint equation for time step n is given by

$$\frac{\partial F^{n\top}}{\partial \mathbf{x}^n}\boldsymbol{\alpha}^n + \frac{\partial F^{n+1\top}}{\partial \mathbf{x}^n}\boldsymbol{\alpha}^{n+1} = -\frac{\partial J^{n\top}}{\partial \mathbf{x}^n}\boldsymbol{\alpha}^n. \tag{2.65}$$

(2.65) is solved backwards in time in order to obtain $\boldsymbol{\alpha}$. Now the gradient with respect to time step n is computed from the follwoing equation

$$\nabla_{\mathbf{u}^n} J_{\boldsymbol{\alpha}} = \frac{\partial J^\top}{\partial \mathbf{u}} + \frac{\partial F^{n\top}}{\partial \mathbf{u}^n}\boldsymbol{\alpha}^n \tag{2.66}$$

Defining

$$g(\mathbf{v},\mathbf{s}) = \delta t \mathbf{D}_{PV}^{-1}[\mathbf{A}(\mathbf{v})f(s) + \mathbf{q}(\mathbf{v})_+] \tag{2.67}$$

and substituting (2.58)-(2.59) into (2.65), the adjoint equations for time step n are obtained as

$$\left[\mathbf{I} - \frac{\partial g(\mathbf{v}^n,\mathbf{s}^n)}{\partial \mathbf{s}^n)}\right]^\top \boldsymbol{\alpha}_s^n = \boldsymbol{\alpha}_s^{n+1} - \frac{\partial J^{n\top}}{\partial \mathbf{s}^n} - \left[\frac{\partial \mathbf{B}(\mathbf{s}^n)v^{n+1}}{\partial \mathbf{s}^n}\right]^\top \boldsymbol{\alpha}_v^{n+1} \tag{2.68}$$

$$\begin{bmatrix} \mathbf{B}(\mathbf{s}^{n-1}) & \mathbf{C} & \hat{\mathbf{D}} \\ \mathbf{C}^\top & \mathbf{0} & \mathbf{0} \\ \hat{\mathbf{D}}^\top & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha}_v^n \\ \boldsymbol{\alpha}_p^n \\ \boldsymbol{\alpha}_{\hat{\pi}}^n \end{bmatrix} = \begin{bmatrix} -\frac{\partial J^{n\top}}{\partial \mathbf{v}^n} + \frac{\partial g(\mathbf{v}^n,\mathbf{s}^n)^\top}{\partial \mathbf{v}^n}\boldsymbol{\alpha}_s^n \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \tag{2.69}$$

Once the adjoint equations have been solved, the gradient of the objective function J at time step n is given by

$$\nabla_{\mathbf{u}^n} J = \frac{\partial J^\top}{\partial \mathbf{u}^n} - \mathbf{A}_D^{n\top}\boldsymbol{\alpha}_v^n - \mathbf{A}_N^{n\top}\boldsymbol{\alpha}_{\hat{\pi}}^n \tag{2.70}$$

The gradients (2.70) can be used in any gradient based methods, one of which will be discussed in the following section.

2.7 Optimization Algorithm

The *OptimizeObjective* function under the *adjoint* module of MRST uses an aggressive line search based on the given gradient. The algorithm handles box-constraints and linear equality and (probably not) inequality constraints. It performs on an iterative scheme, applying the constraints to the gradient until convergence is achieved.

The algorithm makes use of the *gradient projection method* which is explained in the following section.

2.7.1 The Gradient Projection Method

The gradient projection method is most efficient when the constraints are simple in form–in particular, when there are only bounds on the variables (Nocedal and Wright, 2006). A general form of this type of constraints is

$$\mathbf{l} \leq \mathbf{x} \leq \mathbf{u}, \tag{2.71}$$

where $\mathbf{l}$ and $\mathbf{u}$ are vectors of lower and upper bounds on the components of $\mathbf{x}$. The feasible region defined by 2.71 is sometimes called a "box" because of its rectangular shape.

Each iteration of the gradient projection algorithm consists of two stages. In the first stage, we search along the steepest descent direction from the current point $\mathbf{x}$. Whenever a bound is encountered, the search direction is "bent" so that it stays feasible. This is illustrated in Figure 2.5.

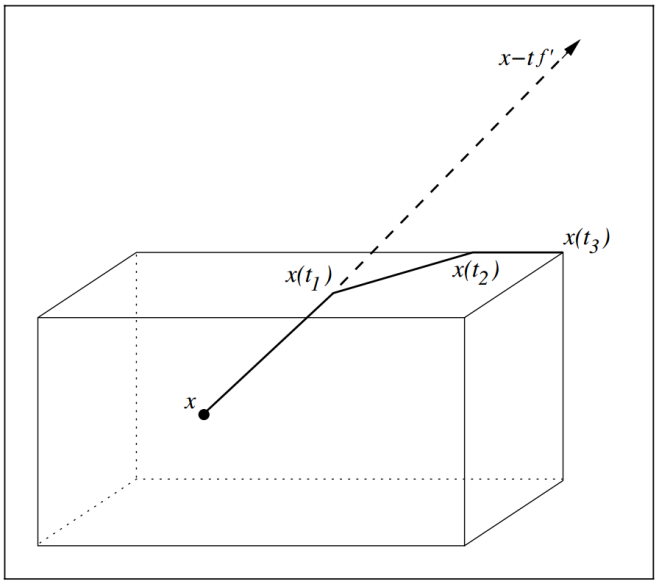

Figure 2.5: The gradient projection method(Nocedal and Wright, 2006).

We search along the resulting piecewise-linear path trying to locate the first local minimizer of the objective function f, which we denote by $\mathbf{x}^c$ and refer to as the *Cauchy point*. The working set[7] is now defined to be the set of bound constraints that are active at the Cauchy point, denoted by $\mathcal{A}(\mathbf{x}^c)$. In the second stage of each gradient projection iteration, we search on the face of the feasible box on which the Cauchy point lies. To do this, we solve a subproblem in which the active components x_i for $i \in \mathcal{A}(\mathbf{x}^c)$ are fixed at the values x_i^c.

[7]Primal active-set methods find a step from one iterate to the next by solving a quadratic subproblem in which some of the inequality constraints, and all the equality constraints, are imposed as equalities. This subset is referred to as the *working set* and is denoted at the kth iterate $\mathbf{x}_k$ by $\mathcal{W}_k$.

2.7.2 Line Search Method

The gradient projection method projects the gradient according to constraints iteratively until a certain tollerance is met or a certain number of iterations is met (where it returnes failure). The constraints are applied in the following order:

1. box constraints,
2. linear inequality constraints,
3. linear equality constraints.

The resulting norm of the projected gradient is used as stopping criteria.

Next, a line search is performed along the projected gradient. The step length α in (2.2) for this line search is multiplied by three values $[0\ 0.5\ 1]$ to get $[\alpha_1\ \alpha_2\ \alpha_3]$. Now, based on a comparison of the objective function values for the three new step lengths, a decision is made:

(a) If the projected gradient is equal for α_2 and α_3, then we are on the boundary, done.

(b) If the value of the objective function follows $f(\alpha_1) \leq f(\alpha_2) \leq f(\alpha_3)$, then multiply the current step size by 2 and go to (a).

(c) If $f(\alpha_1) \geq f(\alpha_2))$, then multiply the current step size by 0.5 and go to (a).

(d) Find the minimum on the approximated quadratic curve through $(\alpha_1, \alpha_2, \alpha_3)$, done.

2.8 Pros and Cons of Using MRST

A wide variety of hydrocarbon reservoir simulators have been developed to date (see e.g., (Christie and Blunt, 2001)). Choosing the right simulator depends chiefly on the problem at hand and the scope of the simulation. Some simulators are developed for rapid modelling of small-scale problems and are good candidates for research purposes. Others have more rigorous routines and algorithms built into them and can be used for larger scale real life problems of the industry.

MRST is a MATLAB-based simulator which is mainly intended as a toolbox for rapid prototyping and demonstration of new simulation methods and modeling

concepts on unstructured grids. Despite this, many of the tools are quite efficient and can be applied to surprisingly large and complex models (mrs, 2013).

MRST was the main tool for performing the case study simulations of this work. Notwithstanding the fact that the toolbox functioned well under the diversified conditions of the two case study problems, it brought about a number of complications and errors throughout the process. The focus of this section is to mention a number of judgments on the toolbox which were formed by the author during the course of the project.

2.8.1 Positive Aspects

Not a Black Box

The main advantage of using MRST over the currently popular hydrocarbon reservoir simulators, such as Eclipse and CMG, is that it is open source. This feature adds a lot of flexibility to the toolbox, by providing the invaluable opportunity to modify the existing routines for specific problems.

The ability to explore the toolbox makes it a potential candidate for academic reservoir simulation courses, provided that students have the necessary MATLAB and linear algebra background.

Ability to Import Data from Eclipse

Schlumberger's Eclipse package is presently one of the most widely-used reservoir simulators. Thus, a lot of models are created with input files suitable for this software. MRST has sensibly exploited this fact by devoting a whole add-on module to direct importing of Eclipse input data files. Although not all the features and keywords of Eclipse are supported by this module, the essential parts of the input files are covered and parsed using MATLAB.

2.8.2 Shortcomings

Insufficient Learning Material

The MRST webpage offers plenty of tutorials and examples for its core and modular environments (mrs, 2013). However, it can be argued that this is not enough.

The tutorials use examples as teaching material rather than a describing general cases. Some of the examples are too specific and the user needs to modify them to work with a different case. Yet, most of the time this requires making changes to lower level functions that constitute the main functions. Since there are no descriptions for such functions, the user has to inverstigate the codes to figure out the algorithms incorporated into them. This is quite a time consuming process.

For example, part of the description of the *optimizeObjective* function (under the *adjoint* module) reads: "A linkage with an external optimizer or use of MATLAB's optimizer toolbox (e.g., fmincon) is recommended." However, it does not been explain how this linkage can be made. Since, the sructure of the arguments of these two functions are quite different, this linkage does not seem straightforward either.

A resolution to this problem could be composing a well-organized manual with explanations to the algorithms in all the functions potential to go through changes by users (see e.g., (Ecl, 2012)). An alternative could be provision of explanations for each part of the codes in functions to facilitate their manipulation.

Inconsistent Data Structures

A major problem with MRST which causes numerious errors is that some of the data structures are not consistent from module to module. Let us go through an example.

Consider a user having an Eclipse input data file who wants to investigate an optimization problem with MRST. He needs to call the *deckformat*, *ad-fi* and *adjoint* modules. The *deckformat* module converts the data from Eclipse format to a format suitable for the *ad-fi* module (which is a module with fully implicit solvers). However, the latter does not include an optimization function similar to *optimizeObjective*, thereby making the user call the *adjoint* module which contains this function. Now, if the input arguments of *optimizeObjective* are created by the *ad-fi* module, running the code will produce an error. The reason is that the structure of variables such as "*fluid*" and "*state*" are different in *ad-fi* and *adjoint* environments.

Chapter 3

Literature Review

Uncertainty is an issue which a lot of fields dealing with modeling and control come across. Handling the uncertainty typically includes the implementation of either or both of the following strategies (van Essen et al., 2009): decreasing the uncertainty by incorporating measurements into the model, and decreasing the sensitivity to the uncertainty.

3.1 Open Loop Reservoir Management

When no production data is available, history matching is no longer a feasible choice for reducing geological uncertainty. However, one can still deal with this situation by reducing the sensitivity to the uncertainty. Yet, this method can be integrated with history matching and result in more accurate outcomes. A few authors have previously addressed the optimization methods that include this strategy.

Yeten et al. (2004) considered the significant effects of uncertainty in geological description of reservoir and reliability of downhole equipment, while presenting a nonlinear CG algorithm for optimization of smart wells[1]. They introduced a probabilistic model to account for hardware failure, and considered 5 geostatistical realizations to include the effects of geologic uncertainty. They applied the method to three scenarios with two-phase flow in a highly heterogeneous North

[1]The term smart (or intelligent) well is referred to nonconventional wells which possess downhole equipment. Completion components such as valves, inflow control devices and sensors are installed on the production tubing to monitor and optimize production.

Sea type reservoir. The results of the decision analysis suggested that optimized smart wells, which have the possibility of equipment failure per se, can still compensate for geological uncertainty by increasing the average net present value and reducing its standard deviation. Moreover, they concluded that for their test cases, the impact of geological uncertainty was dominant over that of equipment reliability (or engineering uncertainty).

3.1.1 Robust Optimization

The method proposed by Yeten et al. (2004) was disputed by van Essen et al. (2009) in that the former had deviated in a number of issues such as the optimization method and the way they had incorporated the realizations in the objective function.

van Essen et al. (2009), on the other hand, proposed a method called *robus optimization (RO)*. Prior to them, an ensemble of geological scenarios would be used to determine the expected revenues for a specified production strategy. However, such ensembles were still not implemented into optimization schemes for oil recovery methods. A suggested approach from the process industry, the RO technique was developed for problems that suffered from a high degree of uncertainty and inadequate measurements. The principle underlying RO is to perform the optimization over a set of realizations.

Within the RO scheme, various objective functions can be used to account for the effect of uncertainty in the set of realizations in different ways (Terwiesch et al., 1998; Ruppen et al., 1995). The most clear-cut approaches consider the RO objective as the expected outcome over the entire set of realizations as shown in equation (3.1).

$$J_{rob}(\mathbf{q}_{1:K}) = \frac{1}{N_r} \sum_{i=1}^{N_r} J(\mathbf{q}_{1:K}, \theta_i) \tag{3.1}$$

where J_{rob} is the robust objective function (usually expected NPV), $\mathbf{q}_{1:K}$ is the vector of conrol inputs (from 1 to K), θ is the vector of uncertain model parameters, and N_r is the number of realizations. The RO problem is defined as:

$$\max_{\mathbf{q}_{1:K}} J_{rob}(\mathbf{q}_{1:K}) \tag{3.2}$$

It follows from (3.1) that a linear operation is required to calculate the expected NPV. Therefore, gradients of (3.1) are calculated by performing a linear operation

on the gradients of each realization as follows:

$$\frac{dJ_{rob}}{d\mathbf{q}_k} = \frac{1}{N_r}\sum_{i=1}^{N_r}\frac{dJ(\mathbf{q}_{1:K},\theta_i)}{d\mathbf{q}_k} \tag{3.3}$$

Yang et al. (2011) presented a SAGD optimization workflow taking into account geological uncertainties. They applied the procedure to a SAGD model with three well pairs and 100 realizations. They concluded that robust optimization using 9 representative realizations was able to account for the uncertainty of 100 realizations. Robustness of such optimization was validated by applying the obtained optimal well location and operating strategy to the full set of 100 realizations. The results showed that the proposed robust optimization procedure was able to find an optimal risk weighted solution that gace good performance for any realization of the uncertainty in the given set. The results also indicated that the optimum solution based on a single realization may lead to worse prediction results for certain realizations. Therefore, risk analysis following nominal optimization should always be performed to assess the risk of applying the nominal optimal solution.

Chen et al. (2012) implemented the augmented Lagrangian method for constrained robust optimization and applied it to estimate optimal well controls for waterflooding projects in order to maximize the NPV of production. They concluded that estimating a good initial value of the penalty parameter and using scaling factors on constraints improve the robustness of the optimization algorithm. They compared two strategies namely "robust long-term alternating with short-term optimization" and "robust sequential short-term optimization" and found out that the former increases the short-term NPV without compromising the life-cycle NPV, while the latter does not.

3.2 Closed Loop Reservoir Management (CLRM)

Closed-loop reservoir management (Brouwer et al., 2004; Sarma et al., 2006; Jansen et al., 2009; Chen et al., 2009; Wang et al., 2009; Peters et al., 2010) is perceived as an important concept to increase recovery from hydrocarbon reservoirs . On a conceptual level, CLRM can be understood as utilizing real-time data from multiple sources and mathematical models to aid long term decision-making and medium term operational decision-making. The objective function is typically net present value, i.e. the net cash flow discounted back to its present value (Foss, 2012).

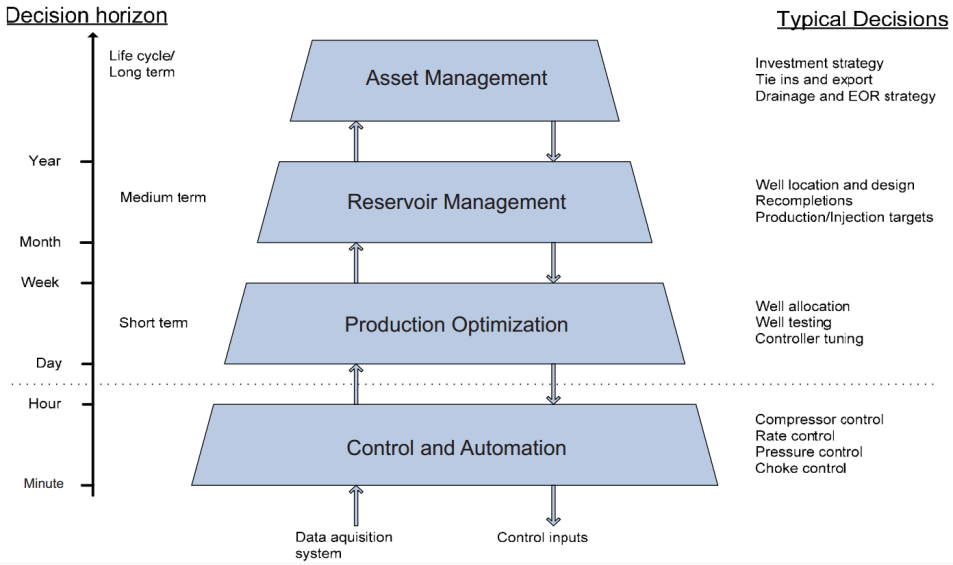

Figure 3.1: Multilevel control hierarchy(Foss and Jensen, 2011).

Closed-loop reservoir management has two major steps: data assimilation and production optimization (Chen et al., 2012). The objectives of the data-assimilation step are to reduce the uncertainty in the reservoir description by assimilating the production data and/or seismic data and to enhance the accuracy of production predictions. Usually, the uncertainty in the reservoir description is modeled by generating a set ofNe plausible realizations of the reservoir. A popular choice for data assimilation is the ensemble Kalman filter (Evensen, 1994, 2007; Naevdal et al., 2005; Aanonsen et al., 2009). In the production-optimization step, one maximizes the hydrocarbon recovery or life-cycle NPV subject to some physical constraints.

Long-term decisions include drainage strategies, technologies, and infrastructure development, while medium-term decisions involve well location, well design, and production and injection rates (Foss and Jensen, 2011). Shorter-term operational decisions, often called closed-loop production optimization, focus on rate allocation between wells. Decisions operating on different time horizons may be organized in a multilevel control hierarchy, as shown in Figure 3.1.

Maximum benefit from the measurement and control equipment of a smart field is expected when used in an integrated monitoring and control approach (Brouwer, 2004), as depicted in Figure 3.2. The figure illustrates an optimizer, a plant (oil reservoir), and state/parameter estimator or observer. Controlling an oil reservoir

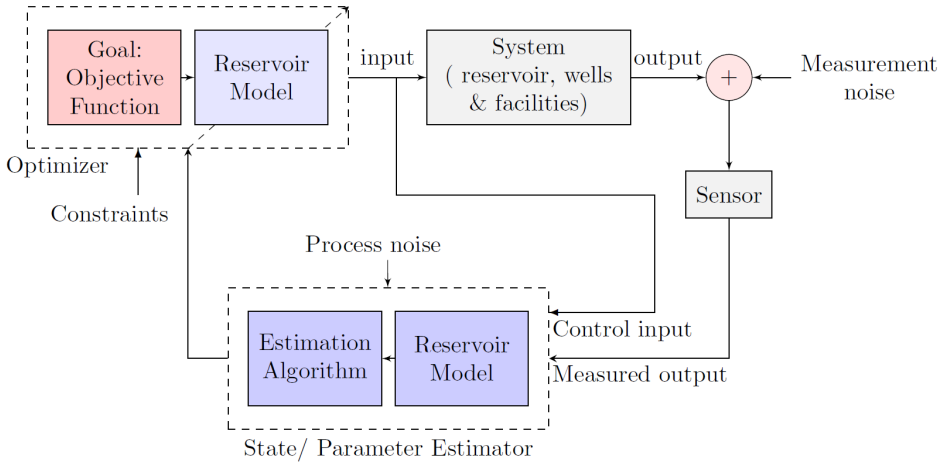

Figure 3.2: Reservoir management depicted as a closed-loop model-based controlled process(Jansen et al., 2005).

is a complex time-consuming process because it cannot be seen and investigated directly, as it lies deep in the Earth. Measurement devices ranging from sensors in wells to seismic data acquisition systems are used to infer reservoir conditions. But, still one is not able to describe the system accurately, since the parameters of the system are known to varying degrees (Jansen et al., 2008): the fluid properties can usually be determined quite well, but the reservoir properties are only really known at the wells. As a consequence, the uncertainties in the model parameters of the subsurface part of the system are very large. In addition, measurement devices also introduce noise. The measurement data are used to close the loop (as feedback) since the observer is connected to the optimizer as it uses updated reservoir models (Suwartadi, 2012). However, reservoir models do still have difficulties fully representing the physics of the system.

As a particular example, let us consider a water flooding process with smart wells as shown in Figure 3.3. The figure illustrates a schematic of a reservoir with a horizontal, smart segmented injector along the left edge, and a horizontal, smart segmented producer along the right edge. Upon injection, the water moves towards the production well where fluids leave reservoir. The water displaces some of the oil on its way. Generally, the oil-water front propagates at different speeds from one place to the other towards the producer, as shown by the irregular shape of the oil-water interface in Figure 3.3. This is, owing to the fact that the reservoir rock properties generally varies in space. The oil-water front illustrated in the figure is the result of a particular injection and production strategy, that is a particular combination of valve-settings. It is possible to control the flow direction, and

thereby the movement of this oil-water front in the reservoir to some degree by manipulating the down-hole valves. The result would be reduced flow through high permeability zones and increased flow outside these zones which will ideally lead to displacement of oil everywhere in the reservoir. The right combination of valve-settings would give the best displacement. Thus, one goal is to find out which combination is the right one. Another goal is to find out what degree of improvement is possible by optimizing the valve-settings. The answer to both of these questions depends on the type of heterogeneity in the reservoir. Another major factor in determining how much improvement is obtained by valve-setting optimization is the physical and economical constraints on the wells and the valves (Brouwer, 2004).

3.2.1 Reducing the Uncertainty by History Matching

The model of a hydrocarbon reservoir initially incorporates the engineer's best estimate of reservoir description parameters (e.g. porosity and permeability). This data, however, is not representative of the whole reservoir because of the uncertainties in the estimates. Therefore, the data must be adjusted until the performance of the model and the history of the reservoir have a minimized discrepancy. The practice of modifying the reservoir model in expectation of reproducing the past

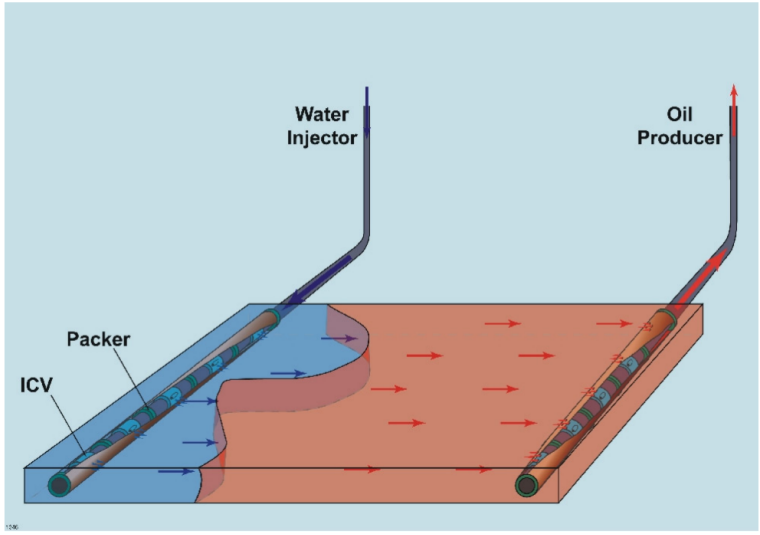

Figure 3.3: Schematic of horizontal reservoir with two horizontal, segmented smart wells (Brouwer, 2004).

behavior of a reservoir is called *history matching*.

Reservoir history matching is a complicated inverse problem. Conventionally, this is done manually (see Crichlow, 1977, chap. 9). However, obtaining a decent match by manually varying the reservoir description parameters is excruciatingly cumbersome. Therefore, mathematical approaches based on Bayesian statistics have been developed to systematically evaluate and characterize uncertainty. Oliver et al. (1996) introduced a method into reservoir characterization which is now commonly referred to az the *randomized maximum likelihood (RML)* method. Many studies have used this technique to produce an estimate of the probability density function (PDF) for reservoir models, given production or seismic data (see e.g., Zhang et al., 2005).

Because of their time-saving characteristic over the traditional trail-and-error methods, computer aided history matching had been increasingly used by the oil and gas industry (Tavassoli et al., 2004). Gradient based optimization techniques play a crucial role in this process.The process of using gradient based optimization techniques to perform history matching is usually referred to as *automatic history matching*. It involves solution of a least-squares problem where the objective function represents the discrepancy between the model and the observed (measured) behavior of the reservoir. Many studies have implemented the available solution methods for least-square problems such as Gauss-Newton and/or Levenberg-Marquardt algorithms (see e.g., Landa and Horne, 1997; Li et al., 2003).

Many other methods have also been used for history matching such as stochastic modeling techniques (Jimenez et al., 1997; Calatayud et al., 1994; Tyler et al., 1993), optimal theory (Chavent et al., 1975; Wasserman and Emanuel, 1976), sensitivity analysis techniques (Watson, 1989; Dogru and Seinfeld, 1981) and gradual deformation method (Gallo and Ravalec-Dupin, 2000).

Chapter 4

Case Study and Analysis

Water flooding scenarios where assessed for two test cases. The problems were of different scales. The first test case was a small *five spot* model, while the second one was a relatively large *egg* model.

In this study, three different approaches were applied to each of the two models: A nominal control, a reactive control and and an optimal control strategy. Also, open and closed loop control were applied separately, and the outcomes were compared.

4.1 Case Study 1 – The Five-Spot Model

The first case is a simple five spot model that effectively showcases the applicability of adjoint-based optimization to well control. The schematic of the reservoir and well configuration is shown in Figure 4.1. The model consists of one water injector in the middle of the reservoir and four producers, one at each corner. Thus there are two drive mechanisms: depletion drive and water injection. The reservoir covers an area of 21000×21000 m^2 and has a thickness of 10 m and is modeled by a $21 \times 21 \times 1$ horizontal 2D grid. The fluid system is an essentially incompressible, two-phase, oil-water system, with a mobility ratio close to 2, connate water saturation of 0.2 and residual oil saturation of 0.2. Figure 4.2 shows the heterogeneous permeability field with the highest and lowest permeability around wells P4 and P2, respectively.

This contrast in permeability near the production wells is around a factor of 30-40, and it is this heterogeneity that makes the optimization results interesting.

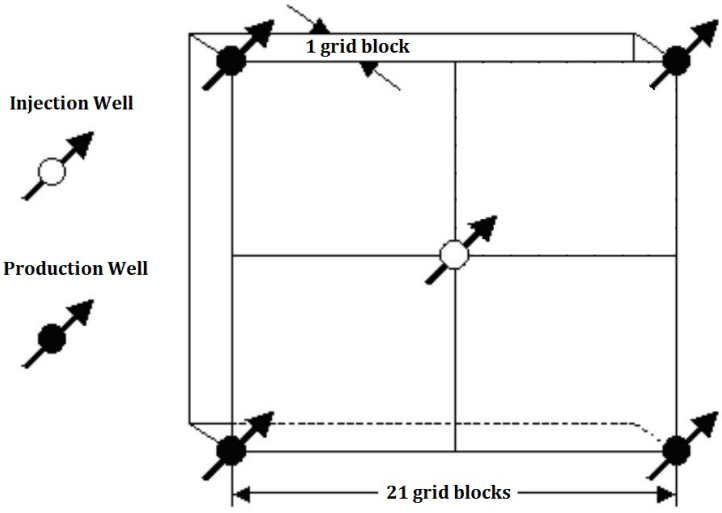

Figure 4.1: 5-Spot Model: Schematic of reservoir and wells.

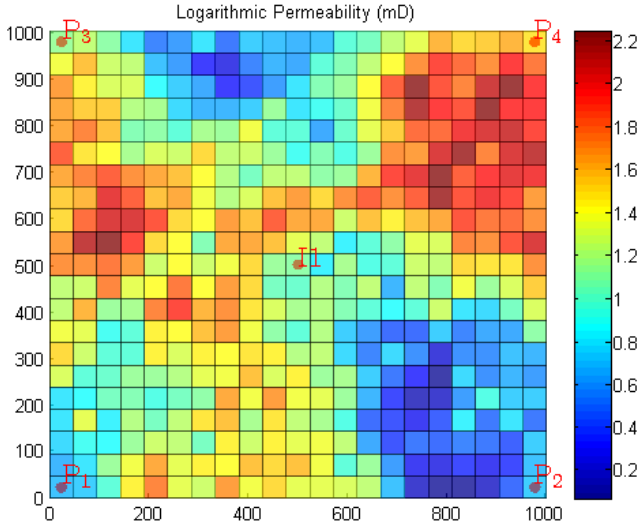

Figure 4.2: 5-Spot Model: Permeability field.

For the purpose of this case study, the base case is a constant rate/constant BHP production strategy. On the part of optimization, the injector segment is placed under rate control, and the producer segments are under BHP control. There is a total injection constraint of 9000 Bbl/day (STBD). Further, there are also bounds on the BHPs of the producers, which could for example correspond to bubble point pressures or fracture pressures. The model is produced for 2500 days . This time period is divided into ten control steps of 250 days each. Thus the total number of controls is equal to 5(well segments)$\times$10(time steps) = 50. All constraints in this problem are linear with respect to the controls.

4.1.1 Comparison of Constant Control Scheme to Nominal Optimization

In order to appreciate the benefit of any optimization process, it is usual to compare the optimization results against a base or reference case (Sarma et al., 2005). In the case of production optimization, such a base case would be a reasonable production strategy that an engineer might devise given a simulation model and a set of constraints. It is, however, very difficult (and often nonintuitive) to understand the implications of varying well controls on the optimization process. It is thus usual for engineers to specify constant production/injection rates or BHPs until some detrimental reservoir response such as water breakthrough is observed.

The objective of the optimization process is to maximize NPV. The NPV discount factor is set to a reasonable value of 8%. The oil price is conservatively set at 100 \$/m^3, water injection costs at 10 \$/m^3, and water production costs at 10 \$/m^3. It should be noted that it is relatively easy to vary these cost/prices with time and even to implement uncertainty models for them (Sarma et al., 2005).

Starting from 80% oil saturation throughout the reservoir, Figures 4.3 and 4.4 demonstrate the final oil saturations for the uncontrolled and optimized case after 2500 days. It is clear that the optimization leads to a good improvement in the sweep efficiency, leading to the increase in NPV of around 10% (Figure 4.5).

The reasons behind the better sweep in the optimized case can be easily explained by analizing the optimized trajectories of the controls – BHPs of the producers– as seen in Figure 4.6. Areas with higher permeability force water to move very quickly toward the producers, resulting in early breakthrough and thus poor sweep. It is obvious from the figure that the producer P4, which is completed in the highest permeability area, has the highest bottom hole pressure for the whole time, preventing early breakthrough. For well P3, which is completed in an area with higher than average permeability, it can be observed that the optimization process

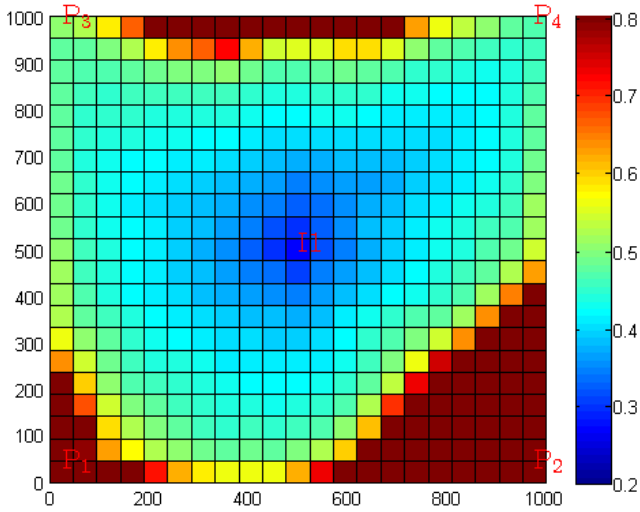

Figure 4.3: 5-spot Model: Final oil saturation map, reference case.

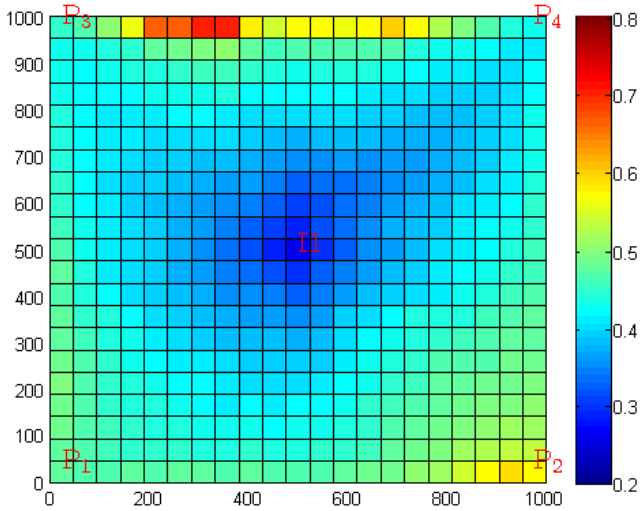

Figure 4.4: 5-spot Model: Final oil saturation map, optimized case.

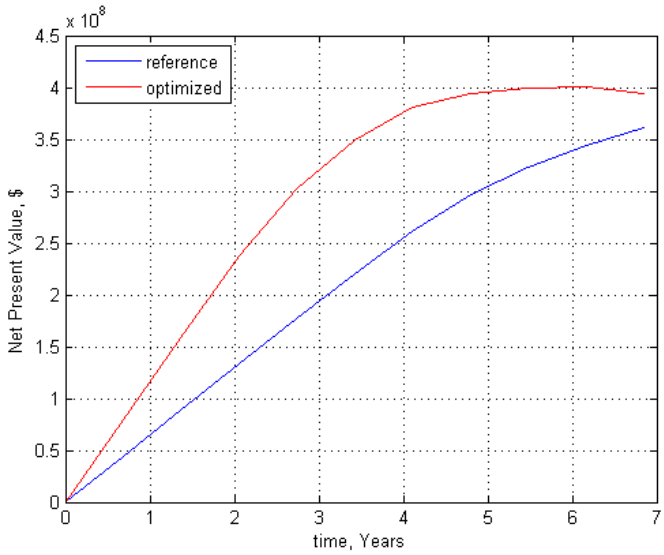

Figure 4.5: 5-spot Model: Comparison of NPV's with 8% discount rate.

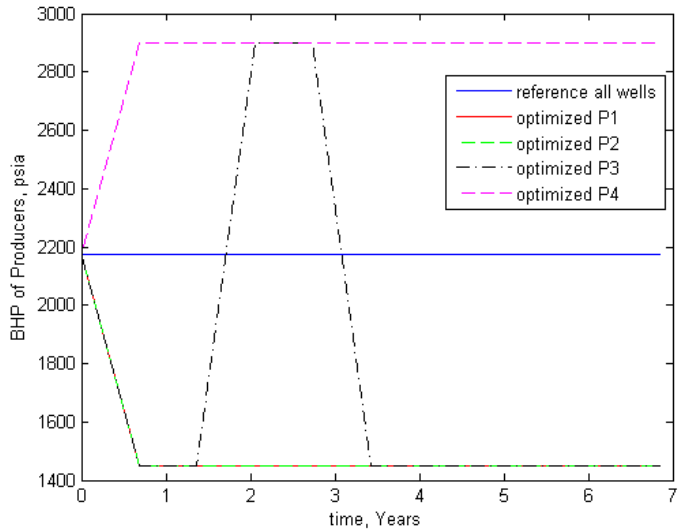

Figure 4.6: 5-spot Model: Comparison of BHP's of producers.

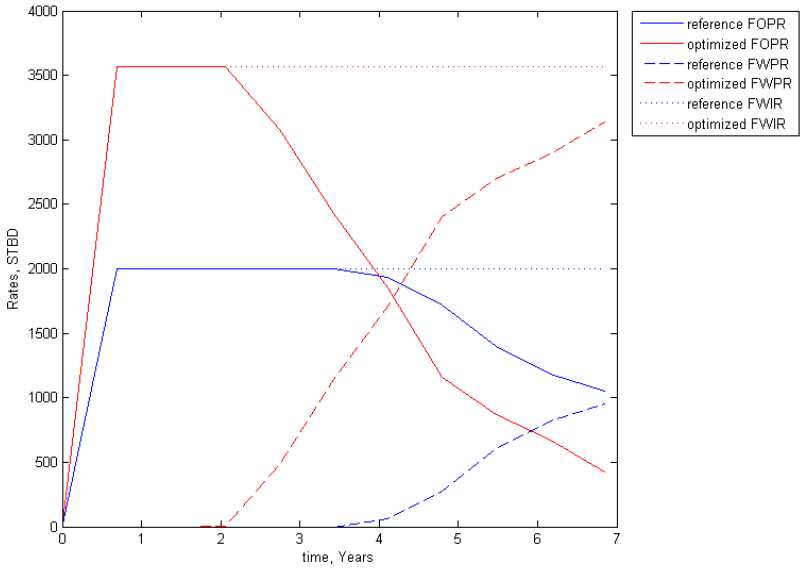

Figure 4.7: 5-spot Model: Comparison of total production rates.

tries to make a balance between a high BHP (late water breakthrough) and a low BHP (high production). Producers P1 and P2, which are around the lowest permeable areas, are allowed to produce with their maximum allowed potential (lowest BHP for the whole time).

Figure 4.7 demonestrates the field water injection rate, the oil production rate and the water production rate for the base case and optimized case. In the optimized case, for most of the time, the oil production rate is significantly higher than the base case. Water production is also higher in the optimized case (due to a higher injection rate), however the overall trade-off between oil and water production results in a higher NPV for the optimized case.

Figure 4.8 shows that there is a noticeable increase in cumulative oil production (23%) attributed to higher water injection rate and better water sweep. The Optimization process required 3 iterations of the optimization algorithm; the total number of simulations required for the optimization was 16.

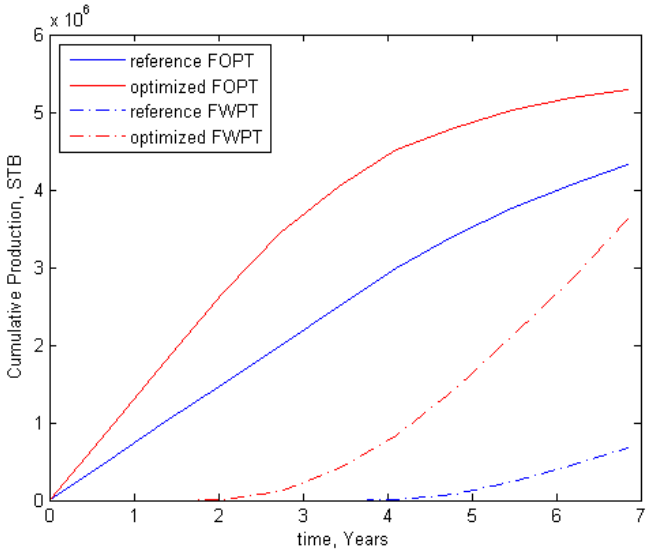

Figure 4.8: 5-spot Model: Comparison of cumulatives.

4.2 Case Study 2 – The Egg Model

The "Egg Model" is a synthetic reservoir model consisting of an ensemble of 101 relatively small three-dimensional realizations of a channelized reservoir produced under water flooding conditions with eight water injectors and four producers (Jansen et al., 2013). It has been in numerious publications to demonestrate a variety of aspects related to computer-assisted flooding optimization and history matching.

Table 4.1: Geological and fluid properties, Egg model.

Property	Value	Unit
ϕ	0.2	–
ρ_o (at 1 bar)	900	$\mathrm{kg/m^3}$
ρ_w (at 1 bar)	1000	$\mathrm{kg/m^3}$
c_o	10^{-5}	$\mathrm{bar^{-1}}$
c_w	10^{-5}	$\mathrm{bar^{-1}}$
μ_o	5×10^{-3}	Pa.s
μ_w	10^{-3}	Pa.s
p_{cow}	0	bar

The original stochastic model has $60 \times 60 \times 7 = 25,200$ grid cells of which 18,533

cells are active. The inactive cells are all at the outside of the model, leaving an egg-shaped model of active cells. Because the model has no aquifer and no gas cap, primary production is almost negligible. The production mechanism is water flooding with the aid of eight injection wells and four production wells, as shown in Figure 4.9.

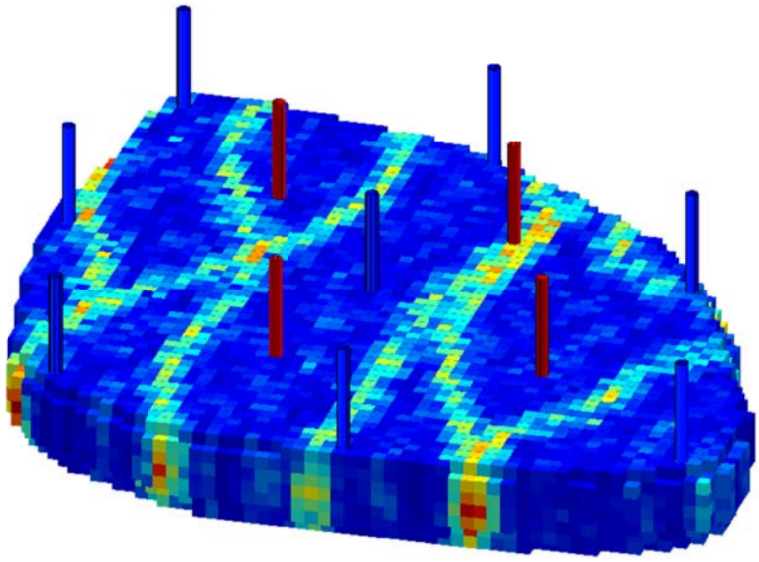

Figure 4.9: Egg Model: The position of the injectors (blue) and producers (red)(Jansen et al., 2013).

The fluid system is a two-phase oil-water system with low compressibility and close to 5 mobility ratio. The relative permeabilities are the usual Corey type curves.

The injectors are under rate control and the producers are under BHP control. The reference case is again a constant rate/BHP case, with the injection rate set at 500 STBD for each injector and all producer BHPs at 395 bara ($\cong$ 5800 psia). The injection rate is constrained at a maximum rate of 2500 STBD per injector and the producer BHPs at a minimum of 200 bara ($\cong$ 3000 psia) and a maximum of 395 bara ($\cong$ 5800 psia).

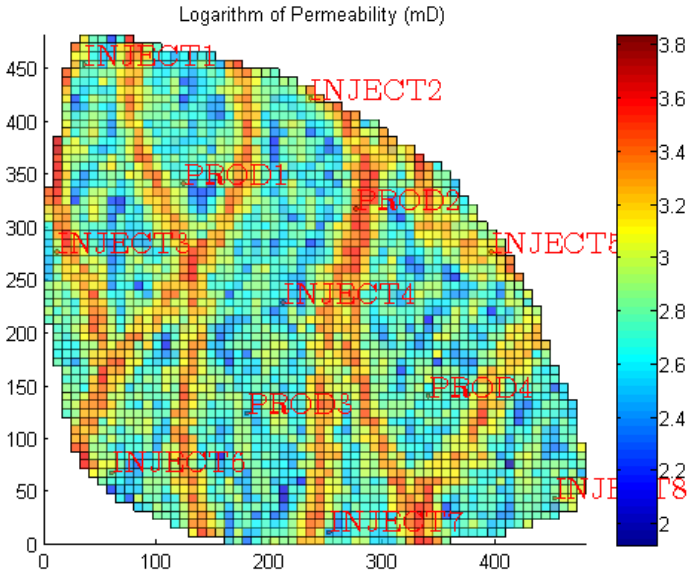

Figure 4.10: Egg Model: Permeability field for the first layer.

4.2.1 Comparison of Constant Control Scheme to Nominal Optimization

In this section we only consider one of the geological realizaitons of the egg model. The objective is to maximize the NPV over a period of around 9.5 years, divided into 10 control periods. The NPV discount factor is set at 8%. The oil price is conservatively set at 100 \$/Bbl, water injection costs at 10 \$/Bbl and water production costs at 10 \$/Bbl.

Figure 4.12 compares the injection, water production and oil production rates of the optimized case to that of the reference case. There is a substantial decrease in injection rate as expected. Due to the high water production costs, water injection is reduced so that water production may be reduced. The main drive mechanism thus changes from water injection in the reference case to depletion drive in the optimized case.

Figure 4.13 compares the cumulative water injection, oil production and water production of the base case and optimized case. It is clear that the significant increase in NPV (Figure 4.14) is due to the huge reduction in water injection and production, while the cumulative oil production is not much reduced.

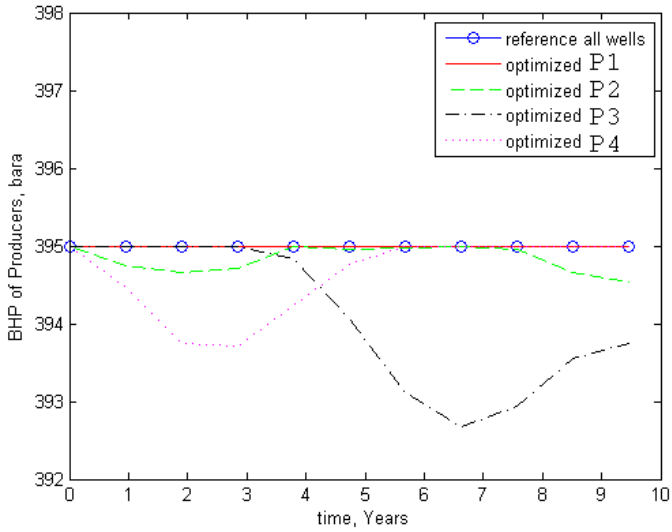

Figure 4.11: Egg Model: Comparison of BHP's of producers.

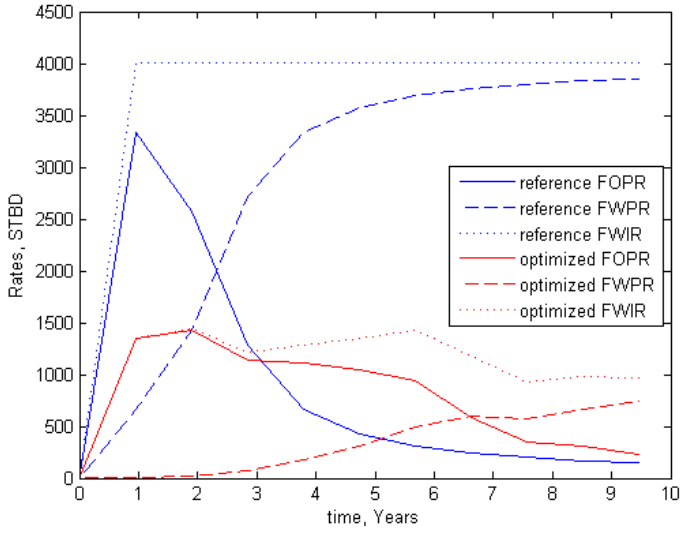

Figure 4.12: Egg Model: Comparison of total production rates.

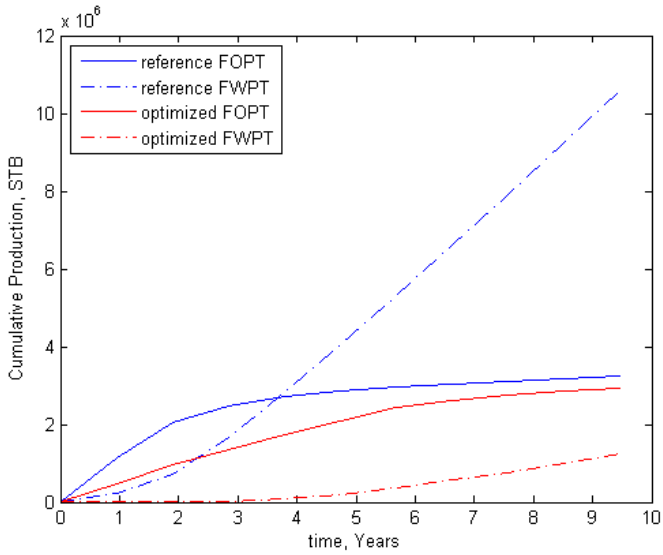

Figure 4.13: Egg Model: Comparison of cumulatives.

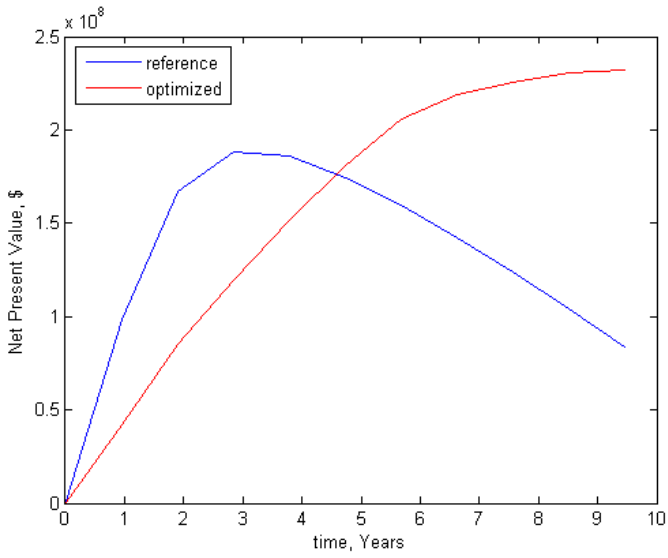

Figure 4.14: Egg Model: Comparison of NPV's with 8% discount rate.

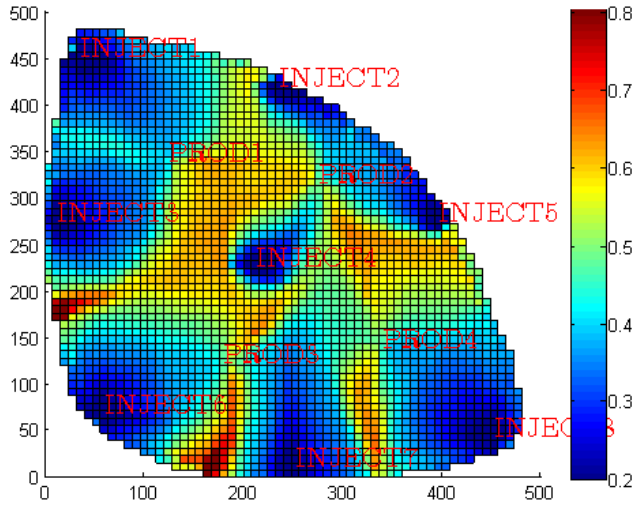

Figure 4.15: Egg Model: Final oil saturation map, reference case.

Another remark on the NPV profiles as shown in Figure 4.14 is that the optimized case increases the overall NPV in the long term which is a significant achievement. However, the NPV profile for the optimized case is lower than that of the reference case for the first half of the whole simulation time. It is important to have high NPVs close to the beginning of the production period. In order to keep the NPV profile relatively high from the beginning, one needs to make a balance between short- and long-term decisions (van Essen et al., 2011).

It is interesting to compare the final oil saturation maps for the reference and optimized case, as seen in Figure 4.15 and 4.16. Unlike the previous case, the optimization results in a decrease of the overall sweep, as the water injected is much less.

The number of iterations for the optimization algorithm required for the above optimization was 5 which corresponds to 35-40 simulation runs. Average run time for one simulation on a 2.4 GHZ machine with 12 GB RAM was around 1.5 minutes and the total optimization time was 1 hour.

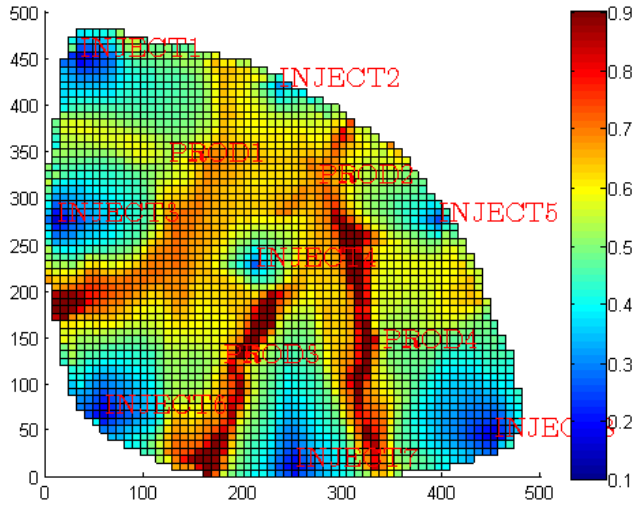

Figure 4.16: Egg Model: Final oil saturation map, optimized case.

4.2.2 Comparison of Nominal Optimization, Robust Optimization and Reactive Control for Multiple Geologiacal Realizations

In this section, all of the 101 realizations embedded within the egg model were used. Three different methods were used to obtain the final NPV of the reservoir for the set of all realizations, namely nominal optimization, robust optimization, and reactive control.

Nominal Optimization (NO). The NO approach uses a single realization. However, as mentioned at the beginning of the section, we have a set of 101 realization available. In this case, the decision of which realization must be used in the NO approach is an arbitrary one, since none of the realizations have priority over others. To get an unbiased result, any decision incoporating potentially biased choices must be avoided. Thus, the NO was applied on each of the 101 realizations. In other words, 101 different optimized stratedies for production from the egg model were generated.

The number of control parameter for the NO procedure for each of the realizations was 120, which is equal to the number of injection and production wells (12) times the number of timesteps (10). Nevertheless, the adjoint method runs only one simulation to obtain the gradients pertaining to all control parameters. As a result, the number of control parameter can not be an issue.

The rock and fluid properties were set the same as the previous section. A single computer required, on average, half an hour for the optimization procedure to converge to the solution for a single realization. The optimal flowrates for the injection and production wells with the NO procedure are illustrated in Figure 4.17. In this figure, the results are based on realization number 1.

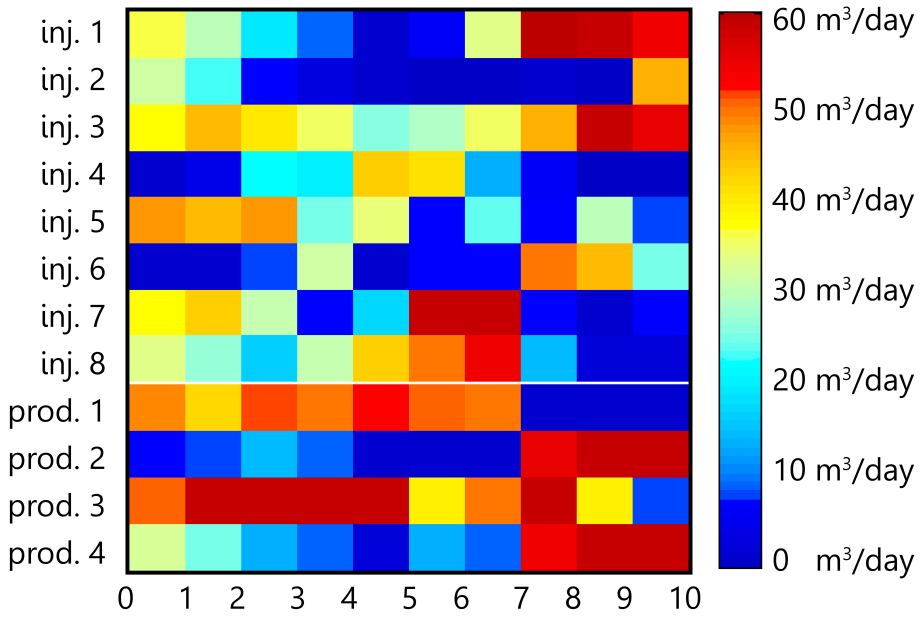

Figure 4.17: Egg Model: The optimal injection and production flow rates resulted form NO, based on realization number 1.

Running the NO algorithm on each of the 101 realizations resulted in 101 different control strategies. Each of these strategies were applied on all of the realizations generating 10201 values for our objective function (the NPV). These values are illustrated in Figure 4.19 in terms of 101 PDF and CDF curves. Each run of the NO algorithm took approximately 3 hours. The optimization procedure for the whole set took 101 times longer, appraximately 2 weeks.

Reactive Control. The idea of reactive control, as described before, is simple. Each production well continues to function until production from this particular well is not cost-effective. This criterion for profitability is determined by a maximum allowed water cut. We consider a water cut of 87% as our threshold.

Since there are 8 injection and 4 production wells in the system, and due to the incompressibility of the model, the flow rate of injection wells must be half of the

value for production wells in order to appreciate the mass balance. Initially, the production wells are fixed at their maximum capacity of 60 m^3/day, and thus the injection wells have a flowrate of 30 m^3/day.

One important note regarding the algorithm used for the reactive approach is that when a production well is shut in (as a result of reaching the threshold water cut), the injection rates are decreased with the same proportion, in order to honor the mass balance.

A disadvantage of the reactive approach is that it usually does not lead to an optimal strategy during the life of the reservoir. On the other hand, since this approach is independent of the reservoir model, when applied to a real field, a wrong representation of the geological realizations does not affect the results.

If the performance of the reactive approach and a model based strategy are simulated on a set of realizations, we can evaluate and compare the performances. In order to validate this assessment, the set of realizations must be a good representation of the true modeling uncertainty. Otherwise, the truth will not be reflected. In case that the implemented set of realizations is not close to the reality, one could not compare the control strategies in terms of their performance.

The reactive approach was applied to each of the 101 components of the set of realizations. Figure 4.18 illustrates the water cut of the four production wells over time as response to the reactive control approach. When the profitability threshold of 87% is reached, the producer stops. For the reactive approach, every year was divided into 16 timesteps in order to get more accurate results (as seen in Figure 4.18. The broad distribution in the time of water breakthrough represents the degree to which the realizations introduce changeability.

Running the reactive conrol algorithm on each of the 101 realizations generated 101 values for our objective function (the NPV as before). These 101 values are illustrated in Figure 4.19 in terms of their corresponding PDF and CDF. The corresponding expected value for NPV and its standard deviation are presend in Table 4.2.

Table 4.2: Results from Reactive control, NO and RO.

	Reactive Control	NO (average)	RO
Expected NPV	268 million NOK	285 million NOK	291 million NOK
Std. Deviation	7.4 million NOK	6.2 million NOK	4.9 million NOK

Robust Optimization (RO). Implementing the RO approach means that the entire set of realizations will be used to determine a single control strategy for maximiz-

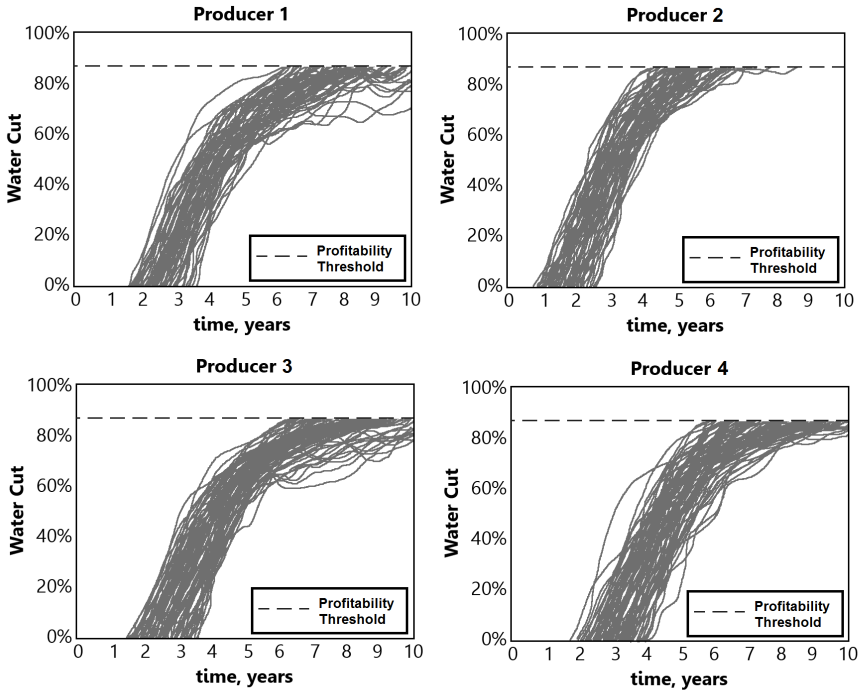

Figure 4.18: Egg Model: Water cut of the four production wells over time as response to the reactive control approach. When the profitability threshold of 87% is reached, the producer stops.

ing the expected NPV over the whole set of realizations.

The same gradient-based optimization procedure and optimization parameters in the NO strategy were used to determine the robust optimal control. Yet, there was a big difference between the two approaches in terms of the computational time. In order to calculate the robust gradient, we need to determine the gradient for each realization separately. Therefore, 101 simulations must run in order to obtain the gradient information for one timestep. In other words, the simulation time for RO is approxiamely 100 times longer the time required for one NO run (i.e., almost 2 weeks).

We applied the resulting RO control strategy to each of the 101 realizations in the set, and determined the value of objective function for each member. The corresponding PDF and CDF are shown in Figure 4.19. The expected NPV and standard deviation are presented in Table 4.2.

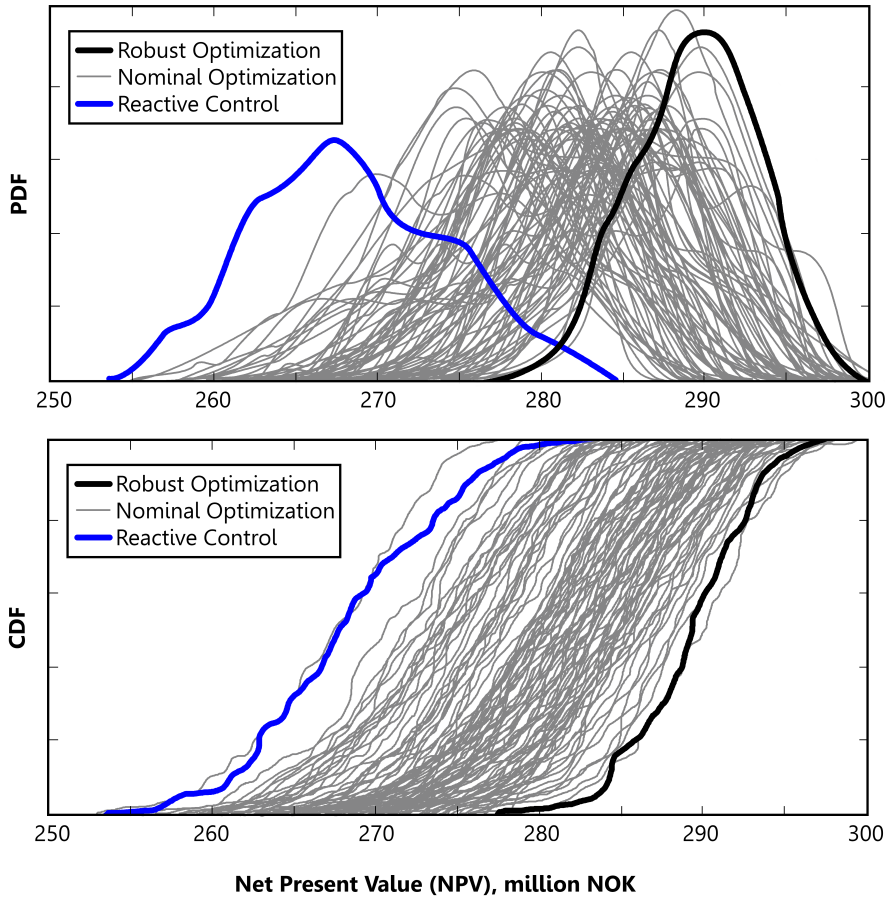

Figure 4.19: Egg Model: PDF and CDF of 101 realizations. Reactive control, RO, and 101 NO strategies are shown.

4.2.3 Closed Loop Robust Optimization

All the strategies applied so far have been open loop, meaning that there was no feedback involved. In this section feedback is introduced to our solution approach. To avoid the state estimation problem, we will assume that the actual states are known. For this reason we will consider the NO control sequence with the highest expected NPV and use its corresponding state trajectories as the optimal states. Then we use an MPC (Model Predictive Control) to make the states follow our desired path.

In this case, at each timestep of the RO scheme, an optimization problem will be solved to minimize the difference between the calculated states and control values with the optimal ones. This means that the total computer time required for this approach will be at least twice the openloop case. It took approximately 5 weeks for the algorithm to run on a 2.4 GHZ machine with 12 GB RAM. Figure 4.20 illustrates the difference between the PDF of RO in open- and closed-loop schemes. Also, Table 4.3 presents the corresponding data.

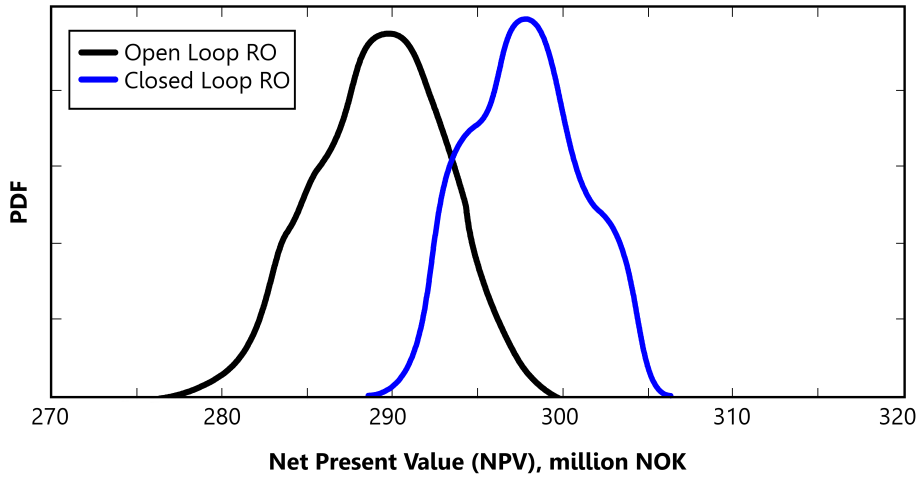

Figure 4.20: Egg Model: PDF and CDF of 101 realizations. Reactive control, RO, and 101 NO strategies are shown.

The results show 2.4% higher expected NPV and 24.5% lower standard deviation for the closed loop case. The reason for higher expected NPV is that each iteration of the MPC tries to bring the controls and states toward the optimal trajectories chosen. Now, the selected optimal trajectory was the NO strategy with the highest expected NPV, and this explains the increase in the expected NPV from open- to

Table 4.3: Comparison of results between Open- and Closed-Loop RO.

	Open Loop RO	Closed Loop RO
Expected NPV	291 million NOK	298 million NOK
Std. Deviation	4.9 million NOK	3.7 million NOK

closed-loop RO. The other role that the feedback plays in this case is decreasing the standard deviation resulted from the extra optimization relative to the open loop RO in each iteration of the MPC.

Chapter 5

Conclusions and Recommendations

The Robust Optimization scheme maximizes the revenues from oil recovery and at the same time takes into account the geological uncertainty. Therefore, RO is an approach that draws the attantion of the E&P industry. When implemented within a closed loop system, it can honor the uncertainties to a greater degree. We examined its expected performance against those of NO and reactive control. To do this, a water flooding model with 101 geological realizations was simulated. The following conclusions can be drawn from the obtained results:

- The particular geological realization chosen to base a Nominal Optimization on, massivly affects the performance of the NO strategy. In the case study, there is no reason to regard one geological realization as more likely to occure. The results obtained from the Nominal Optimizations are very different and consistently worse than the Robust Optimization scheme.
- Testing using a large set of realizations demonstrated that in comparison to NO and recactive control strategies, Robust Optimization virtually enhances the expected NPV. Also, the ranges of NPV outcomes is smaller in the RO approach (reduced variance).
- Since reactive control approach is independent of the reservoir model, when applied to a real field, a wrong representation of the geological realizations does not affect the results. However, model based NO and RO strategies suffer from a wrong representation of geological realizations. On the other hand, the disadvantage of the reactive approach is that it usually does not lead to an optimal strategy during the life of the reservoir.

- Comparison between the results of open loop and closed loop RO shows a higher expected NPV and a lower standard deviation for the closed loop case. The reason for higher expected NPV is that each iteration of the MPC tries to bring the controls and states toward the optimal trajectories chosen. Now, the selected optimal trajectory was the NO strategy with the highest expected NPV, and this explains the increase in the expected NPV from open- to closed-loop RO. The other role that the feedback plays in this case is decreasing the standard deviation resulted from the extra optimization relative to the open loop RO in each iteration of the MPC.

- While applying any of the optimization strategies we had assumed that the set of geological realizations is a good representative of the reservoir system uncertainties. Therefore, the conclusion of enhanced performance of RO versus NO and reactive control holds only if this assumption is right. If otherwise, the truth is not reflected meaning that the implemented set of realizations is not close to the reality, one could not compare the control strategies in terms of their performance.

Further research can be done incorporating a more integrated approach, in which measurements are employed to narrow down the set of geological realizations or to enhance the quality of this set by means of history matching.

Bibliography

, 2012. ECLIPSE Reference Manual. Schlumberger, version 2012.2.

, June 2013. BP Statistical Review of World Energy. British Petroleum (BP), bp.com/statisticalreview.
URL http://bp.com/statisticalreview

, 2013. The matlab reservoir simulation toolbox, version 2013b. URL http://www.sintef.no/MRST/.
URL http://www.sintef.no/MRST/

Aanonsen, S.I. adn Naevdal, G., Oliver, D., Reynolds, A., Valles, B., 2009. The ensemble kalman filter in reservoir engineeringa review. SPE Journal 14, 393–412.

Aarnes, J. E., Gimse, T., Lie, K.-A., 2007. An introduction to the numerics of flow in porous media using Matlab. Springer Verlag, pp. 265–306.

Aziz, K., Settari, A., 1979. Petroleum Reservoir Simulation. Applied Science Publishers LTD, iSBN 0-85334-787-5.

Boyd, S., Vandenberghe, L., 2009. Convex Optimization. Cambridge University Press, iSBN 978-0-521-83378-3.

Brouwer, D., Nvdal, G., Jansen, J., Vefring, E., van Kruijsdijk, C., 26-29 September 2004. Improved reservoir management through optimal control and continuous model updating. In: 13th SPE European Petroleum Conference. No. SPE 90149. Houston, Texas, USA.

Brouwer, D. R., October 2004. Dynamic waterflood optimization with smart wells using optimal control theory. Ph.D. thesis, Delft University of Technology, Delft, Netherlands.

Calatayud, P., Petit, F., Tessier, P., Norris, R., Coffin, P., Alabert, F., 2527 October 1994. Combined stochastic and deterministic reservoir characterization leads to faster history matching of a horizontal well in a complex fluvio-lacustrine environment. In: SPE European Petroleum Conference. No. SPE 28585.

Chavent, G., Dupuy, M., Lemonnier, P., 1975. History matching by use of optimal theory. SPE Journal (SPE 4627), 74–86.

Chen, C., Li, G., Reynolds, A., 2012. Robust constrained optimization of short- and long-term net present value for closed-loop reservoir management. SPE Journal 17, 849–864.

Chen, Y., Oliver, D., Zhang, D., 2009. Efficient ensemble-based closed-loop production optimization. SPE Journal 14, 634–656.

Christie, M., Blunt, M., August 2001. Tenth spe comparative solution project: A comparison of upscaling techniques. SPE Reservoir Evaluation & Engineering, 308–317.

Crichlow, H. B., 1977. Modern Reservoir Engineering– A Simulation Approach. Prentice-Hall, Inc., Englewood Cliffs, New Jersey 07632, iSBN 0-13-597468-2.

Dantzig, G. B., Thapa, M. N., 2003. Linear Programming: 2: Theory and Extensions. Springer, iSBN 0-387-98613-8.

Dogru, A. H., Seinfeld, J. H., 1981. Comparison of sensitivity coefficient calculation methods in automatic history matching. SPE Journal, 551–557.

Evensen, G., 1994. Sequential data assimilation with a nonlinear quasi-geostrophic model using monte carlo methods to forecast error statistics. Journal of Geophysical Research 99, 10143–10162.

Evensen, G., 2007. Data Assimilation: The Ensemble Kalman Filter. Springer Verlag.

Foss, B., 2012. Process control in conventional oil and gas fields–challenges and opportunities. Control Engineering Practice 20, 1058–1064.

Foss, B., Jensen, J. P., March 2011. Performance analysis for closed-loop reservoir management. SPE Journal, 183–190.

Gallo, Y. L., Ravalec-Dupin, M. L., 14 October 2000. History matching geostatistical reservoir models with gradual deformation method. In: SPE Annual Technical Conference and Exhibition. Dallas.

Griewank, A., Walther, A., 2008. Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation, 2nd Edition. SIAM.

Hillier, F. S., Lieberman, G. J., 2010. Introduction to Operations Research, 9th Edition. McGraw-Hill, iSBN 0073376299.

Holland, J. H., 1975. Adaptation in Natural and Artificial Systems. University of Michigan Press.

Jansen, J., 2011. Adjoint-based optimization of multi-phase flow through porous media– a review. Computers and Fluids 46, 40–51.

Jansen, J., Douma, S., Brouwer, D., Van den Hof, P., Bosgra, O., Heemink, A., 24 February 2009. Closed loop reservoir management. In: Paper presented at the SPE Reservoir Simulation Symposium. The Woodlands, Texas, USA.

Jansen, J.-D., Bosgra, O. H., den Hof, P. M. V., 2008. Model-based control of multiphase flow in subsurface oil reservoirs. Journal of Process Control 18, 846–855.

Jansen, J.-D., Brouwer, D., Naevdal, G., van Kruijsdijk, C., 2005. Closed-loop reservoir management. First Break 23, 43–48.

Jansen, J.-D., Fonseca, R. M., Kahrobaei, S., Siraj, M., van Essen, G., den Hof, P. V., November 2013. The egg model. Tech. rep., Department of Geoscience and Engineering at TU Delft and Department of Control Systems at Eindhoven University of Technology.

Jimenez, Z., Azpiritxaga, I., Lozada, T., 8-11 June 1997. Stochastic modeling technique as applied to history matching of eocene-misoa reservoir, lake maracaibo, venezuela. In: SPE Reservoir Simulation Symposium. No. SPE 38020. Dallas.

Kennedy, J., Eberhart, R., 1995. Particle swarm optimization. In: Proceedings of the 1995 IEEE International Conference on Neural Networks. Perth, Australia, pp. 1942–1948.

Krogstad, S., Gulbransen, A., 171 2011. Adjoint multi-scale mixed finite elements. SPE Journal 16 (1), 162.

Landa, J. L., Horne, R. N., 1997. A procedure to integrate well test data, reservoir performance history and 4-d seismic information into a reservoir description. In: SPE Annual Technical Conference and Exhibition. No. SPE 38653. San Antonio, Texas.

Li, R., Reynolds, A., Oliver, D., 2003. History matching of three-phase flow production data. SPE Journal 4, 328–34.

Lie, K.-A., Krogstad, S., Ligaarden, I. S., Natvig, J. R., Nilsen, H. M., Skaflestad, B., 2012. Open source matlab implementation of consistent discretisations on complex grids. Comput. Geosci. 16 (2), 297–322, dOI: 10.1007/s10596-011-9244-4.

Naevdal, G., Johnsen, L., Aanonsen, S., Vefring, E., 2005. Reservoir monitoring and continuous model updating using ensemble kalman filter. SPE Journal 10, 66–74.

Neumaier, A., 2004. Complete search in continuous global optimization and constraint satisfaction. In: Acta Numerica 2004, a. iserles Edition. Cambridge University Press.

Nocedal, J., Wright, S. J., 2006. Numerical Optimization, 2nd Edition. Springer.

Oliver, D., He, N., Reynolds, A., 3-6 September 1996. Conditioning permeability fields to pressure data. In: 5th European Conference on the Mathematics of Oil Recovery. Leoben, Austria.

Passaro, A., Starita, A., 2008. Particle swarm optimization for multimodal functions: A clustering approach. Journal of Artificial Evolution and ApplicationsArticle ID 482032.

Peaceman, D. W., 1977. Fundamentals of Numerical Reservoir Simulation. Developments in Petroleum Science, 6. Elsevier Scientific Publishing Company, iSBN 0-444-41578-5.

Peaceman, D. W., 1983. Interpretation of well-block pressures in numerical reservoir simulation with nonsquare grid blocks and anisotropic permeability. SPE Journal 23, 531–543.

Peters, L., Arts, R., Brouwer, G.K., e. a., 2010. Results of the brugge benchmark study for flooding optimization and history matching. SPE Reservoir Evaluation & Engineering 13, 391–405.

Ruppen, D., Benthack, C., Bonvin, D., 1995. Optimization of batch reactor operation under parametric uncertainty- computational aspects. Journal of Process Control 5 (4), 235–240.

Sarma, P., Aziz, K., Durlofsky, L., 31 January- 2 February 2005. Implementation of adjoint solution for optimal control of smart wells. In: SPE ReservoirSimulation Symposium. SocietyofPetroleumEngineers Inc., Houston, Texas, U.S.A.

Sarma, P., Durlofsky, L., Aziz, K., Chen, W., 2006. Efficient realtime reservoir management using adjoint-based optimal control and model updating. Comput. Geosci. 1, 3–36.

Suwartadi, E., March 2012. Gradient-based methods for production optimization of oil reservoirs. Ph.D. thesis, Norwegian University of Science and Technology, Department of Engineering Cybernetics, Trondheim, Norway.

Suwartadi, E., Krogstad, S., Foss, B., December 2010. A lagrangian-barrier function for adjoint state constraints optimization of oil reservoirs water flooding. In: 49th IEEE Conference on Decision and Control.

Tavassoli, Z., Carter, J. N., King, P. R., September 2004. Errors in history matching. SPE Journal (SPE 86883), 352–361.

Terwiesch, P., Ravemark, D., Schenker, B., Rippin, D. W. T., 1998. Semi-batch process optimization under uncertainty: Theory and experiments. Computers & Chemical Engineering 22 (1-2), 201–213.

Tyler, K., Svanes, T., Omdal, S., 36 October 1993. Faster history matching and uncertainty in predicted production profiles with stochastic modeling. In: SPE Annual Technical Conference and Exhibition. No. SPE 26420. Houston.

van Essen, G., den Hof, P., Jansen, J., 2011. Hierarchical long-term and short-term production optimization. SPE Journal 16 (1).

van Essen, G., Zandvliet, M., den Hof, P. V., Bosgra, O., Jansen, J., 2009. Robust waterflooding optimization of multiple geological scenarios. SPE Journal, 202–210.

Vanderplaats, G. N., 2007. Multidiscipline Design Optimization. Vanderplaats Research and Development, Inc. Colorado Springs.

Venter, G., 2010. Encyclopedia of Aerospace Engineering. John Wiley and Sons, Ltd., Ch. Review of Optimization Techniques.

Wang, C., , Li, G., Reynolds, A. C., September 2009. Production optimization in closed-loop reservoir management. SPE Journal (SPE 109805), 506–623.

Wasserman, M., Emanuel, A. S., 1976. History matching three-dimensional models using optimal control theory. Journal of Petroleum Science & Engineering, 70–77.

Watson, A. T., 1989. Sensitivity analysis of two-phase reservoir history matching. SPE Reservoir Engineering, 319–324.

Yang, C., Card, C., Nghiem, L. X., Fedutenko, E., 2011. Robust optimization of sagd operations under geological uncertainties. In: SPE Reservoir Simulation Symposium. Society of Petroleum Engineers, pp. –.

Yeten, B., Brouwer, D. R., Durlofsky, L. J., Aziz, K., 2004. Decision analysis under uncertainty for smart well deployment. Journal of Petroleum Science & Engineering 43, 183–199.

Zhang, F., Skjervheim, J. A., Reynolds, A., Oliver, D., 2005. Automatic history matching in a bayesian framework: Example applications. SPE Reservoir Evaluation & Engineering 8 (SPE-84461-PA), 214–223.