Model-reduced gradient-based history matching

MODEL-REDUCED GRADIENT-BASED HISTORY MATCHING

PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. ir. K.C.A.M. Luyben,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op

maandag 4 juli 2011 om 10:00 uur

door

Małgorzata Paulina KALETA

Master of Science in Applied Mathematics
geboren te Kalisz, Polen.

Dit proefschrift is goedgekeurd door de promotoren:
Prof. dr. ir. A. W. Heemink
Prof. dr. ir. J. D. Jansen

Samenstelling promotiecommisie:

Rector Magnificus,	voorzitter
Prof. dr. ir. A. W. Heemink,	Technische Universiteit Delft, promotor
Prof. dr. ir. J. D. Jansen,	Technische Universiteit Delft, promotor
Prof. dr. N. K. Nichols,	University of Reading, United Kingdom
Prof. dr. W. R. Rossen,	Technische Universiteit Delft
Prof. dr. ir. P. M. J. Van den Hof,	Technische Universiteit Delft
dr. R. G. Hanea,	Technische Universiteit Delft, TNO
dr. P. J. Van den Hoek,	Shell Global Solutions International

This research was carried out within the context of the ”Integrated Systems Approach to Petroleum Production” (ISAPP) knowledge center. ISAPP is a long-term co-operation of the Dutch Organization for Applied Scientific Research (TNO), Shell International Exploration and Production (SIEP) and Delft University of Technology (TU Delft) to increase hydrocarbon recovery through the application of innovative reservoir development and management technologies.

ISBN: 978-90-8570-774-5

Printed in The Netherlands by Wöhrmann Print Service.

Acknowledgments

This thesis is the result of supports and contributions from many people who deserve a special mention.

In the first place, I would like to thank my supervisors Arnold Heemink and Jan Dirk Jansen for the help and guidance they gave me during my PhD research; it was a great pleasure to work together. I would like to thank my co-promoter Remus Hanea for his day-to-day support and for his friendship. My gratitude to Paul van den Hoek and Hassan Mahani for their help with the fracture simulator.

Working on my PhD project was a great experience thanks to the people surrounding me; for this reason, I would like to thank all my colleagues from the department of Applied Mathematics, the department of Geotechnology and ISAPP group for the constructive discussions, the friendly atmosphere and all their help. I am especially grateful to Marielba for her comments on this thesis and Dorothée for her help with the Dutch translation of the thesis summary. Moreover, I am indebted to my colleagues at Shell, who helped me with the last arrangements.

I would like to thank all my friends, especially Justyna and Ognen for the great time we have spent together. Thanks to them, I felt in Delft like at home.

An important thank go to my family, my parents and my sister Ewa, for their endless support and encouragement in those difficult but beautiful years of my life.

Last but not least, special thanks to my husband Sergio for his support, encouragement, love and for making my life complete.

Gosia Kaleta

Delft, June 2011

Contents

1 CHAPTER

Introduction

Fossil fuels have been and will remain for decades the main energy source. Since the worlds's energy demand increases every year, the oil & gas industry makes a continuous effort to improve fossil fuel recovery. This chapter first provides an overview of the processes that take place nowadays during the exploration and production phases of hydrocarbon recovery, and finishes with the research objective and the outline of this thesis.

1.1 Importance of fossil fuels

Fossil fuels consist of coal and liquid or gaseous hydrocarbons such as crude oil and natural gas and are millions of years old. Fossil fuels were initially used as a source of light and heating, and for military purposes. The first reference to oil pits is from the fifth century BC, written by the Greek historian Herodotus. The possibility of using natural gas as fuel was discovered by the ancient Chinese, while the use of coal was mentioned by the Greek scientist Theophrastus, in a geological treatise, where he classified rocks based on their behavior when heated. The end of the 18th century was the beginning of the industrial revolution, which was a major turning point in human history. Many important inventions took place, which resulted in changes in agriculture, manufacturing, mining, transport, cultural conditions, and almost every aspect of daily life. The introduction of steam power, fueled primarily by coal, and powered machinery underpinned the dramatic increases in production capacity, while the development of all-metal machine tools made the development and production of manufacturing machines used in other industries easier. The greatest technical discoveries of that period were made by: Thomas Edison who invented the electric light bulb, Robert Bunsen who invented the Bunsen burner, Alphonse de Rochas who patented the four stroke internal-combustion engine, Henry Ford who built the first motor car, the Wright Brothers who built the first successful airplane. Petroleum fossil fuels started to be burned in internal combustion engines to provide power. Since the industrial revolution, fossil fuels became important stock, and many oil & gas companies were created and new fields were discovered on different continents.

For decades the oil & gas industry developed technologies to produce petroleum in a safe and efficient manner. The industry also improved the infrastructure to transport it and the processes to refine it into more useful petroleum products, such as gasoline, diesel fuel, heating oil, kerosene, and liquefied petroleum gas. Nowadays, fossil fuels also provide lubricants, asphalt base, paraffin waxes; and it is a feedstock in petrochemical processes that manufacture such products as plastics, detergents, solvents and fibers like nylon and polyesters.

Fossil fuels are the main energy source and together with nuclear energy and renewable energy satisfy the world's demand. Renewable energy uses geothermal, solar, wind or water resources and combustible renewable & waste. The latter comprises solid and liquid biomass (such as wood, vegetal waste, ethanol, animal materials) and additionally biogas, industrial waste as well as wastes produced by the residential, commercial, and public service sectors. In Fig. 1.1, the historical use and the projection of energy use by its type is presented, while in Fig. 1.2 the historical use and the projection of the use of energy is compared in three regions: The United States, China and India, and the rest of the world.

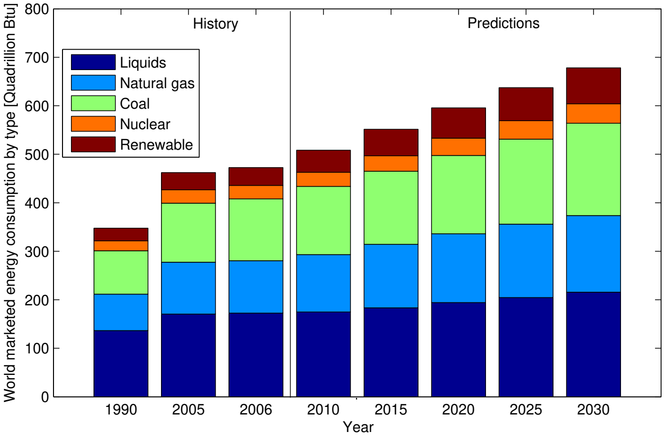


Figure 1.1: Marketed energy use by type (according to IEO (2009)).

Fig. 1.1 shows that the world market energy consumption increases each year and is expected to grow, while Fig. 1.2 shows that higher energy consumption is predicted for developing countries as China and India. This increase is expected to be satisfied both by the renewable energy and fossil fuels. It is more difficult nowadays for oil & gas companies to meet those expectations because many discovered fields are already at a mature state, while new fields are much more difficult to produce. Only a certain portion of the petroleum can be produced from reservoirs and around $10-15\%$ of every produced barrel of oil is used in the process of extracting it from the ground and refining it into a customer product. Even more is used when harder to produce oil and gas is extracted (petroleum more difficult to

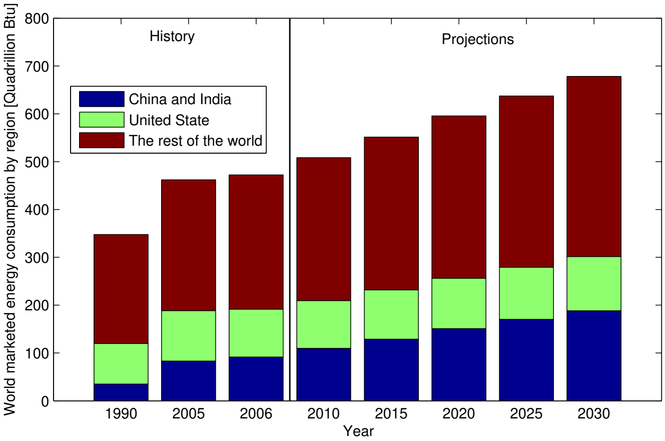


Figure 1.2: Marketed energy use by regions (according to IEO (2009)).

produce such as heavier crude or petroleum buried below difficult terrains, such as kilometers below the sea's surface or in arctic regions). Therefore, oil & gas companies must maximize the production from the existing fields and work on technologies to reduce the energy consumption during hydrocarbon production and processing.

1.2 Exploration and production

Petroleum (or crude oil) is found in geologic formations beneath the Earth's surface. It is trapped in the pores of the rock. It can be found by interpreting geological and seismic data. Seismic data are measurements of wave signals that have traveled from a source at the surface, through the formation. They are reflected from a subsurface layer and are recorded by receivers at the surface. The waves carry information about the properties of the formation through which they travel, and are used to map the formation structure and property fields. When a potential oil or gas field is found, exploration wells are drilled to investigate the presence of oil or gas. In 1931 Conrad and Marcel Schlumberger successfully identified presence of oil in a formation by measuring resistivity, which was a beginning of well-logging measurements, which are nowadays commonly taken in exploration wells. After drilling, the field is investigated to find out if the hydrocarbons can be recovered in an economically and an environmentally acceptable way. This phase is called the exploration and appraisal phase of reservoir development. Several properties of the rock, such as porosity and permeability, are very important for petroleum extraction. Porosity is the fraction of the rock that is occupied by fluids and gases, while permeability describes how easily they can flow through the rock. If the pores are connected such that the fluids and gases can move through them, then the rock is called permeable. The properties of the rock can largely influence the dynamic processes in the reservoir. The rock can be greatly heterogeneous, i.e. through some parts of the reservoir the fluids and gases can flow easily and through others they cannot. This information strongly influences decisions about the position of the wells as well as the operation strategies during recovery. The wells should be distributed and operated in a way that the total amount of the produced hydrocarbons is maximized.

The recovery process is divided into three phases (Ewing (1983)): primary recovery, secondary recovery, and tertiary recovery. The primary recovery phase takes place during the first stage of the reservoir life. Initially, the natural pressure in the reservoir is higher than the pressure in the wells, which causes natural flow to the surface. In time the reservoir pressure declines and at some point it is not sufficient to drive the gas and oil to the surface. This phase usually takes several years and yields around 20-30% recovery.

The secondary recovery phase takes place under flooding of the reservoir with water or gas. The injection of water or gas via injection wells re-pressurizes the reservoir and sweeps the remaining petroleum toward the production wells. Usually, the injection wells are surrounded by production wells to which the displaced oil is directed. Secondary recovery can improve the recovery factor of oil and gas; however, still 50% or more stays in the reservoir.

In the tertiary recovery phase or enhanced oil recovery phase gas, chemicals or steam are injected to alter the fluids' properties and wash the rock. For example, during thermal recovery steam is injected to change the viscosity of the oil and to make it more fluid. In chemical recovery, polymers or surfactants are used to increase the viscosity of the injected water, or modify the wetting behavior of the rock.

1.3 Closed-loop reservoir management

The initial geological model is a physics-based model of the subsurface reservoir based on knowledge from geology, exploration wells, and seismic data. Numerical geological models are typically very detailed (millions of ”voxels” or grid blocks) and therefore they have to be upscaled. The flow in porous media and into the wells is described by partial differential equations which are based on the mass balance law and Darcy's law combined with initial and boundary conditions. Those equations are a complex set of coupled equations, which need to be solved numerically. Numerical methods are applied to calculate the model variables (which are pressures, and saturations of phases or accumulations of components) at discretized time and space domain.

Using a numerical model (or computer model) initial predictions for future reservoir performance are made. If those predictions are economically profitable, the reservoir enters the field development phase. During field development, the main target is to maximize an economic criterion. For this purpose, the numerical models are used, which are expected to mimic the processes occurring in the reservoir during production phase. In the ideal situation when the numerical models fully describe the dynamical behaviour of the reservoir system, the optimal operation strategy can be found. Unfortunately, numerical models can only be a crude approximation of reality. This is due to several reasons. First, not all processes occurring in the reservoir can be modeled in an appropriate way, i.e. many times some simplifications are made to make the problem easier to trace, or there is not sufficient knowledge of the modeled phenomena. This error can to a certain extent be modeled as a model error. Second, many rock properties, such as permeability and porosity, are poorly known. Those errors can be expressed by parameter uncertainty. Moreover, the amount of hydrocarbon in the reservoir is unknown. This can be modeled by the initial-condition error. Ignoring the uncertainty and assuming that those properties are correct may result in wrong predictions and consequently, in wrong decisions. Therefore, when the production process starts, the data from wells are gathered and compared with the data predicted by the numerical models. If the numerical data do not match to the collected data, it is an indication that the numerical model requires a calibration. For the reminder of the thesis the model variables will be referred to as state variables of the system; the rock and fluid properties are referred to as parameters of the model; the setting of the wells, which determine the operation strategies are referred to as the control variables.

The measured data together with numerical simulations can be used in reservoir management for improving the model and increasing the recovery from a field. The concept of closed-loop model-based reservoir management provides a framework for this purpose. The main idea is presented in Fig. 1.3.

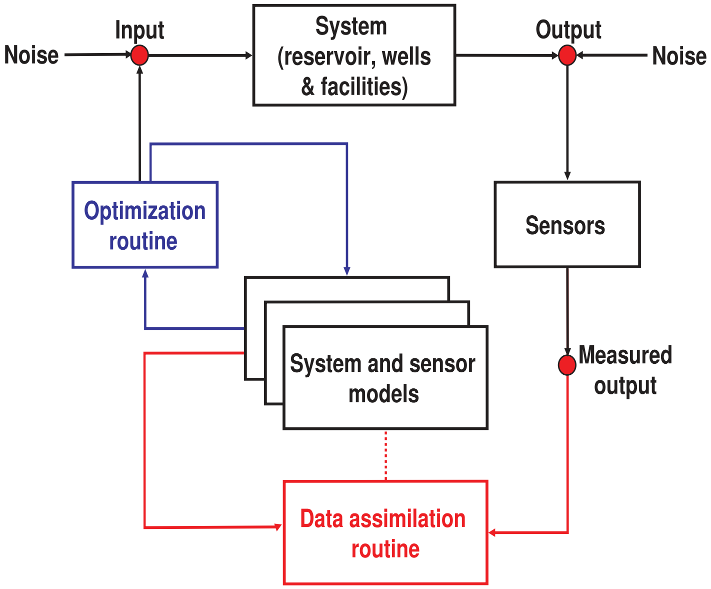


Figure 1.3: Closed-loop model-based reservoir management (after Jansen et al. 2009).

The system box in Fig. 1.3 represents reality: the reservoir with wells, facilities, and sensors. The system is controlled by a number of inputs which are located in the wells and which influence the flooding process. The input to the system is known to a limited extent due, for example, to unknown aquifer support, which is indicated in the figure by noise. The production is monitored by a number of sensors located in the wells, or by additional seismic data gathered all over the reservoir. The measured data is contaminated with noise. This is also indicated in the figure. The box named "system and sensor models" represents the model equations that mimic the real system and are used to simulate its dynamic behavior. The multiple boxes represent the possible geological models of a given reservoir. These models are created after detailed analysis of the available data and then they are upscaled. Fig. 1.3 contains two loops representing two processes. The red loop represents a data-assimilation process while the blue loop represents a production optimization process. Both loops are extensively studied in the petroleum literature. The new aspect of the model-based closed-loop reservoir management framework is the combination of the data-assimilation and production optimization loops. When new data becomes available, the data-assimilation loop is performed providing a better estimate of the model parameters (and states). The assimilation is then followed by an optimization loop that uses the updated model (or models) to provide the updated optimal strategy for future production. The procedure is iterated whenever new data becomes available. See Brouwer et al. (2004), Sarma et al. (2008), Nædval et al. (2006), Wang et al. (2009a), Jansen et al. (2008), Jansen et al. (2009).

1.3.1 Data assimilation

In data assimilation, one combines the information present in the observations and an existing numerical model, as well as the uncertainties in the observations and the models, to produce more realistic results. Data assimilation is required in several fields including meteorology (Talagrand and Courtier (1987)), oceanography (Bennett (2002)), groundwater flow (Valstar et al. (2004)), and petroleum engineering. In the latter, it is is known as *history matching*, and it typically aims to improve the model parameters. Traditionally, history matching was performed on a campaign basis by manual adjustment of parameters (so called *manual history matching*). Nowadays, the results of semi-automatic procedures performed by computers are analyzed and adjusted by engineers based on their experience and opinions. This technique is known as *the computer-assisted history matching*. The main data assimilation methods developed for reservoir problems are briefly described in the next chapter (see e.g. Oliver and Chen (2011), Oliver et al. (2008)).

1.3.2 Production optimization

During production optimization[1], one tries to identify the position of wells or/and configurations that optimize the production. The optimization objective is usually expressed in terms of *Net Present Value* or *total hydrocarbon recovery*. Since the reservoir is usually very heterogeneous, the fluids flow with different velocities in different parts of the reservoir. During waterflooding, the injected fluid will move much faster through paths which are highly permeable and if the high permeable region is connected with a production well it will reach the production well very fast, leaving the oil located outside this region in the reservoir. Continuing the injection will result in the production of water instead of oil. This is called water breakthrough. To avoid early water breakthrough in the wells, one can try to find the operation strategy that will sweep the oil uniformly towards the production wells. This is typically done by controlling the injection and production settings in the wells. Such a long-term optimization strategy should result in higher recovery factors compared to a reactive production strategy, in which no action is undertaken until significant changes are observed in the production wells. This problem has been investigated by many authors. It has been studied both under the assumption that the model parameters are perfect as well as in the presence of uncertainty. Initially, the production optimization was performed for the optimization of tertiary recovery processes, see Ramirez (1987), later followed by water flooding optimization, see e.g. Asheim (1988), Sudaryanto and Yortsos (2000), Brouwer and Jansen (2004), Sarma et al. (2008), Zandvliet et al. (2007) and Doublet et al. (2009). For an overview see Jansen (2011a).

1.4 Research objective and solution directions

Both processes, history matching and production optimization, are very challenging. Since reservoir models are described by nonlinear system of equations, the optimization is much more difficult than would be in case of a linear system. The difficulties appear in the implementation of some well-known methods as well as in the chance of finding the global optimal solution. Well-known methods to solve those problems are iterative gradient-based optimization techniques, where the gradient is calculated by so-called adjoint-method. The difficulties in the implementation are caused by the complex form of the reservoir model equations, which results in a tremendous implementation effort to obtain the adjoint model and the sensitivities of the model with respect to parameters or controls. Usually, in reservoir models, the Jacobian matrices of the system are available because they are used in a Newton-Raphson time-step iteration during reservoir simulation. Even so, the implementation of the adjoint equations required in gradient calculations is a considerable programming effort which, moreover, requires access to the simulation code. This implies that there is a need for gradient-based, but adjoint-free optimization methods. This requirement becomes even more pressing if reservoir simulation is combined with another simulation, e.g. of geomechanics or rock physics, with a code for which no Jacobians are available.

[1] The term ”production optimization” is normally used for maximization of the instantaneous oil production, not for the optimization of the long term oil production.

1.4.1 Research objective

The considerations exposed above have led to the following research objective of this thesis:

> The objective of this research is to investigate adjoint-free gradient-based methods for history matching of production data, and furthermore, to apply these methods to production optimization of a waterflooded reservoir operated under induced fracturing conditions.

1.4.2 Solution directions

In history matching problems, one deals with a large number of uncertain parameters and very sparse observations. The uncertainty is present in geological parameters, in the initial or boundary conditions, and in the inputs, while the information available from various kinds of data is very limited and noisy. Since in realistic reservoir models the number of grid blocks is very large, usually in the order of 10^4-10^6, the number of states and parameters is very large too. A disadvantage of this situation is that the values of the model parameters (and states) cannot be verified with measurements due to the relatively low information content present in them. This is a conclusion of work done by Van Doren (2010) and it raises a question: can the reservoir dynamics relevant for data assimilation or production optimization be represented by much simpler models than the original one? Recent results on model reduction in fluid flow models indicate that indeed the reservoir dynamics can be accurately described with significantly reduced models, see e.g. Heijn et al. (2004), Markovinović and Jansen (2006), and Cardoso et al. (2009). The proposed approach is inspired by these results.

We aim to solve history-matching problems (or production-optimization problems) assuming that the simulator does not contain automatic gradient calculations for the desired parameters (or controls), and that the users do not have access to the simulation source code. The first method that satisfies these assumptions is a method based on reduced-order modeling techniques. This approach avoids the implementation of the adjoint method by replacing the original model with a low-order linear approximation. The Jacobians required for linearization of the model are approximated by finite differences. Since it uses a linearized low-order model, an adjoint model is created by the transpose of the Jacobians. To compare the quality of the obtained results, we estimate the parameters for which the classical adjoint-based gradient is indeed available. This allows us to verify the results of the proposed methods with the classical approach. Additionally, for the comparison of the efficiency of the methods, the naive finite-difference gradient-based approach is used.

A second approach satisfying our requirements is the simultaneous perturbation stochastic algorithm (SPSA) proposed by Spall (1998). It is a gradient-based

method where gradients are approximated by a random perturbation of all parameters in once. A detailed description of this method, as well as references to the relevant petroleum engineering papers, are given in this thesis. This approach is very simple to implement because it only requires measuring the response of the objective function to a perturbation of model parameters.
Another method satisfying our requirements is the "ensemble optimization" technique proposed for recovery optimization by Lorentzen et al. (2006), and thereafter refined by Chen et al. (2009). This approach uses an ensemble of control variables and computes the cross-covariance between the control variables and the objective function which can then be used as an approximate gradient. This approach is also simple to implement; however, it is not tested in this thesis.

1.5 Thesis outline

The outline of this thesis is as follows:

- Chapter 2 consists of a brief overview of flow modeling in a porous medium. Additionally, the adjoint model derivation is presented and an overview of a coupling of the reservoir flow model with a geomechanical model describing fracture propagation or shrinkage is presented. The possibility of extending the existing adjoint code to coupled system is also described. The chapter concludes with remarks about the simulation tools and the implementation difficulties associated with the adjoint model.

- Chapter 3 defines the history matching problem, describes the available data, and gives a brief overview of existing methods, paying special attention to their computational complexity and implementation aspects. In addition, an overview of reparameterization techniques is presented.

- Chapter 4 consists of the description of two data-driven projection-based model reduction techniques: proper orthogonal decomposition and balanced proper orthogonal decomposition. The chapter finishes with comments on the connection between them.

- In Chapter 5, the model-reduced gradient-based method is described in detail. The computational complexity of the method is summarized and the implementation complexity is described. In addition, the classical gradient-based adjoint-based method as well as two gradient-based adjoint-free methods, namely, the finite difference method and the simultaneous perturbation stochastic algorithm (SPSA), are presented. This chapter is partly based on Kaleta et al. (2008) and Kaleta et al. (2011).

- Chapter 6 consists of results for two synthetic models, which differ in complexity and dimension. This chapter is to a large extend based on Kaleta et al. (2010) and Kaleta et al. (2011).

- In Chapter 7, the production optimization of a simple reservoir model with induced fracture is presented. Implementation issues associated with the model-reduced approach are discussed. The simultaneous perturbation stochastic algorithm (SPSA) and the finite-difference based method are applied to solve production optimization problem.

- In Chapter 8, conclusions are drawn and recommendations for future work are given.

2 CHAPTER

Reservoir flow model without and under induced fracturing conditions

To predict reservoir performance under different business decisions a simulation model is created. This model describes the physical processes taking place in a reservoir and is solved numerically. Typically, commercial software is used. In this chapter, a numerical model based on simplistic flow equations is presented, and comments on the complexity of the commercial simulators are given. Additionally, an adjoint model for reservoir flow model is derived, which is a very important element to solve parameter estimation and production optimization problems. A coupled system that combines the standard reservoir flow model with a geomechanical model is also described. This coupling is used to model a fracture that can be created around an injection well during the production process. A possibility of completing the adjoint code for this coupled system is described. The chapter concludes with a short discussion of the difficulties in the adjoint implementation and of the simulators used in this thesis.

2.1 Reservoir flow model

Reservoir flow simulation typically is performed in two stages: first, one or more geological models are created by geologists and geophysicists which aim to provide a static description of the reservoir; secondly, reservoir simulation models are created by reservoir engineers using information about reservoir geometry and fluids present in the reservoir. Depending on the processes involved in the reservoir recovery, e.g. water flooding, steam injection, polymers or chemical flooding, specialized simulation tools are chosen that model them adequately. These ”dynamic” reservoir models are then used to simulate the reservoir performance. The presented here models are described in more details in Aziz and Settari (1979) and Jansen (2011b).

2.1.1 Reservoir geometry and properties

Typically, the reservoir geometry, i.e. the domain of the model, is three-dimensional. The typical grid to model reservoir geometry is a Cartesian grid, where the grid blocks are defined as small boxes. Some simulators offer other griding options such as radial coordinates, or map coordinates, or direct corner-point coordinates. To guarantee an accurate solution of the simulation the grid blocks should be small enough to accurately represent the reservoir heterogeneities, but large enough to assure that the computational time of the model stays within practical limits. In some simulators to produce more accurate modeling of selected region a multi-level local grid refinement is possible by subdividing a grid block into a number of sub-blocks. This allows the accurate modeling within practical limits. For each grid block of the model the absolute permeabilities, (pressure-dependent) porosities and, for fractured reservoirs optional permeability compaction data need to be specified. The effect of aquifers or faults, if present, needs to be modeled too.
A geometry of the simplistic flow model presented in this thesis is two-dimensional with a Cartesian grid, no-flow boundary conditions, no fractured structure, no aquifers and no faults are present.

2.1.2 Phase behavior

The reservoir is a porous medium which pores are filled with hydrocarbons and water. The hydrocarbons are mixtures of a large number of chemical components, however, a full multi-component modeling is usually prohibitively expensive in terms of computation time. Therefore simplifications are usually made to the reservoir fluid description in order to obtain a computationally acceptable model. A commonly made assumption is to model phases by very limited number of chemical components. The most simple approach is to model a phase by a single chemical component. Usually this assumption is used only for the water phase, but is not valid for the hydrocarbon phases because they can share the same components. Since the heavier and lighter hydrocarbon components do not interact in many reservoir situations, it is common to assume that the hydrocarbon mixture consists of two pseudo-components: ”heavy” and ”light” hydrocarbon. The heavy component is defined as the part of the reservoir fluid that becomes the stock tank oil when taken to surface, and the light component is defined as the part that becomes stock tank gas when taken to surface. This assumption leads to *the black oil description* (Aziz and Settari (1979)). The distribution of the components over the phases are usually specified with the aid of semi-empirical rules. After the phase distribution is fixed, each phase requires specification of different properties, such as, viscosity μ (expressed in $[Pa \cdot s]$), density ρ (expressed in $\left[\frac{kg}{m^3}\right]$) and phase compressibility (expressed in $[1/psi]$). These properties are typically not known for the whole reservoir and they are derived by fitting the parameters to results of well tests and laboratory tests. Since the density and viscosity depend on pressure, temperature or fluids composition, these dependencies need to be modeled too. For liquids (oil and water), the density and viscosity depend mostly only weakly on the pressure, while for vapors (gas), they depend mostly strongly on pressure and relatively weakly on the fluid composition. The viscosities of oil and gas are generally strongly dependent on temperature. The presented here simplistic flow model assumes the presence of oil and water phase and constant viscosity, density and compressibility.

2.1.3 Modeling flow in porous medium

The common model variables are pressure, phase saturation (or component accumulation) and temperature which are called the primary variables or state variables and they are determined at every time step of a simulation. In this thesis we assume isothermal conditions within the reservoir, i.e. constant temperature; therefore temperature is not a primary variable. The derivation of the flow equations in porous media is based on the law of conservation of the mass and Darcy's law (see Aziz and Settari (1979)). In this thesis we assume that there is no flow across the boundaries of the reservoir, i.e. the reservoir is surrounded by impermeable rock, and the only boundary flow occurs through the wells. This results in following partial differential equations for phase $l = \{o, w\}$, where o stands for oil and w for water:

$$\nabla \cdot (\alpha \rho_l \mathbf{v}_l) + \alpha \frac{\partial(\rho_l \phi s_l)}{\partial t} - \alpha \rho_l q_l = \mathbf{0}. \tag{2.1}$$

The first term in Eq. 2.1 is called the transmissibility term and describes the difference in the mass of phase l that flows with the velocity v_l in and out of a unit volume per unit time. The second term, called the accumulation term, describes the accumulation of the mass of phase l per unit time and volume. The third term, called source/sink term, describes the mass added or extracted from reservoir through the wells per unit time and volume. α stands for a geometric factor which accounts for the translation of the 1D or 2D model into the 3D. s_l $[-]$ stands for saturation of phase l, and t $[s]$ stands for time. ρ_l $\left[\frac{kg}{m^3}\right]$ denotes the density of the phase l, ϕ $[-]$ denotes the porosity of the rock. q_l $\left[\frac{1}{s}\right]$ is the rate of phase l in a well per unit volume. $\mathbf{v}_l$ $\left[\frac{m}{s}\right]$ is the velocity of the phase l in the porous media and is modeled by the Darcy's law:

$$\mathbf{v}_l = -\frac{k_{rl}}{\mu_l} \mathbf{K} (\nabla p_l - \rho_l g \nabla d) \tag{2.2}$$

where p_l $[Pa]$ denotes the pressure of phase l, μ_l $[Pa \cdot s]$ is the viscosity of the phase l, and k_{rl} $[-]$ is the relative permeability of phase l (which physical meaning will be discussed below). g $\left[\frac{m}{s^2}\right]$ is the gravity acceleration and d $[m]$ is the depth of the reservoir. $\mathbf{K}$ $[m^2]$ is the absolute permeability tensor, which is a full 3×3 matrix. Often the orientation of the coordinate system can be aligned with geological layering in the reservoir such that the permeability tensor becomes a diagonal matrix:

$$1D : \mathbf{K} = k,\ 2D : \mathbf{K} = \begin{bmatrix} k_x & 0 \\ 0 & k_y \end{bmatrix} \text{ and } 3D : \mathbf{K} = \begin{bmatrix} k_x & 0 & 0 \\ 0 & k_y & 0 \\ 0 & 0 & k_z \end{bmatrix}.$$

It is commonly assumed that horizontal permeability is isotropic (i.e. k_x=k_y). This never really occurs, but permeability along various directions of a formation is often close enough for calculation purposes. The reservoir model used in this thesis is simplified by assuming isotropic horizontal permeability field (thus $k_x = k_y = k$). By substituting the velocity of phase l in Eq. 2.1 by the Darcy's velocity the following system of partial differential equations is obtained:

$$\nabla \cdot (-\alpha \rho_l \frac{k_{rl}}{\mu_l} k(\nabla p_l - \rho_l g \nabla d)) + \alpha \frac{\partial(\rho_l \phi s_l)}{\partial t} - \alpha \rho_l q_l = 0,\ l \in \{o, w\}. \tag{2.3}$$

It is a system of two equations with four unknowns, however, additional information is given by the closure equations, i.e.

$$s_o + s_w = 1 \tag{2.4}$$

and

$$p_o - p_w = p_c(s_w), \tag{2.5}$$

where $p_c(s_w)$ $[Pa]$ is a capillary pressure (the physical meaning of capillary pressure will be discussed below). Using these equations two unknowns are eliminated. By introducing the dependencies of density on pressure or fluids composition Eq. 2.1 is changed adequately. In the simplistic model we assume that the density depends on pressure; therefore the accumulation term becomes

$$\frac{\partial(\rho_l \phi s_l)}{\partial t} = \frac{\partial \rho_l}{\partial p_l}\frac{\partial p_l}{\partial t}\phi s_l + \frac{\partial \phi}{\partial p_l}\frac{\partial p_l}{\partial t}\rho_l s_l + \rho_l \phi \frac{\partial s_l}{\partial t}. \tag{2.6}$$

In case of liquids (oil and water), the density depends mostly only weakly on pressure and its sensitivity can be expressed by using the compressibility of the phase, which for a constant temperature is defined as

$$c_l(p_l) := \frac{1}{\rho_l}\frac{\partial \rho_l}{\partial p_l}, \tag{2.7}$$

and is assumed to be constant over the range of pressures of interest and equal c_l, for $l \in \{o,\ w\}$. If the capillary pressure effects are very small, the oil pressure is approximately equal to the water pressure and the compressibility of the water can be expressed as function of the oil pressure, i.e.

$$c_w(p_o) := \frac{1}{\rho_w}\frac{\partial \rho_w}{\partial p_o}. \tag{2.8}$$

The compressibility of the rock, for a constant temperature, is defined as

$$c_r(p_l) := \frac{1}{\phi}\frac{\partial \phi}{\partial p_l}, \tag{2.9}$$

and is assumed to be constant over the range of pressures of interest and equal c_r. Combining Eq. 2.3, closure equations (Eq. 2.4 and Eq. 2.5) and the compressibility definitions one obtains:

$$\begin{aligned} & -\nabla\cdot(\alpha\rho_w\frac{k_{rw}}{\mu_w}k\left[\nabla p_o-\frac{\partial p_c}{\partial s_w}\nabla s_w-\rho_w g\nabla d\right]) \\ & +\alpha\rho_w\phi\left[s_w(c_w+c_r)\frac{\partial p_o}{\partial t}+\frac{\partial s_w}{\partial t}\right]-\alpha\rho_w q_w = \mathbf{0}, \qquad (2.10) \\ & -\nabla\cdot(\alpha\rho_o\frac{k_{ro}}{\mu_o}k(\nabla p_o-\rho_o g\nabla d)) \\ & +\alpha\rho_o\phi\left[(1-s_w)(c_o+c_r)\frac{\partial p_o}{\partial t}-\frac{\partial s_w}{\partial t}\right]-\alpha\rho_o q_o = \mathbf{0}. \qquad (2.11) \end{aligned}$$

This is a system of coupled nonlinear equations. In this thesis the gravity forces and capillary pressures are ignored.

Here short explanations of the physical meaning of relative permeability and capillary pressure (used in Eq. 2.2 and Eq. 2.5, respectively) are given.

Relative permeabilities

Relative permeabilities are required when several phases in the porous media are modeled. If several phases are flowing simultaneously through a porous medium, then the flow of each phase is slowed down by the resistance of the pore structure and additionally by the presence of the other phases. The resistance of the pore structure is expressed by the absolute permeability (denoted as **K**), while the relative permeability of a phase l, k_{rl}, accounts for the permeability reduction in the presence of other phases. The relative permeability is a dimensionless quantity in the range between zero to one, whereas the absolute permeability is a rock property and is measured in square meters or Darcy units ($\left[1 Darcy \approx 9.87\cdot 10^{-13} m^2\right]$). The effective permeability of a phase is the product of the absolute permeability and the relative permeability (see Eqs. 2.10 and 2.11). The relative permeability of the phase depends on the phase saturations. The main influence on the shape of the relative permeability functions is the wettability of the rock. The phase that preferentially sticks to the rock surface is called the wetting phase, whereas the phase that flows preferentially in the center of the pores is the non wetting phase. The remaining phase (if present) is called the intermediate phase. Commonly it is assumed that water is the wetting phase, gas is the non wetting phase and oil is the intermediate phase. The relative permeabilities are typically considered as functions of their own saturations and they are derived from experiments in a laboratory. By definition the relative permeability is zero when the saturation of the phase equals its residual saturation which is defined as the largest saturation for which the phase does not flow. The sum of the relative permeabilities is usually less than one. In this thesis the Corey model of the relative permeability was chosen, which defines the relatives permeabilities as:

$$\begin{aligned} k_{rw} &= k^0_{rw}s^{n_w}, \qquad (2.12) \\ k_{ro} &= k^0_{ro}(1-s)^{n_o}, \qquad (2.13) \end{aligned}$$

where

$$s = \frac{s_w - s_{wc}}{1 - s_{or} - s_{wc}}$$

and k_{ro}^0 and k_{rw}^0 are the end-point relative permeabilities for oil and water respectively, n_o and n_w are Corey exponents, s_{or} is the residual oil saturation and s_{wc} is the connate water saturation. The relative permeability curves used in this thesis are show in Fig. 2.1.

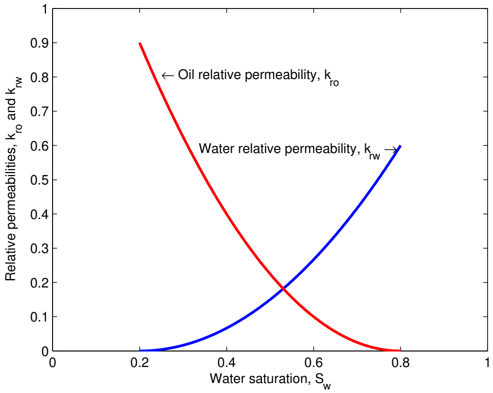


Figure 2.1: Corey type relative permeabilities.

Capillary pressures

The capillary pressure is defined as the difference in pressure across the interface between two immiscible phases, and thus defined as

$$p_c = p_{non-wetting\ phase} - p_{wetting\ phase}. \tag{2.14}$$

At the oil-water interface, water is typically the wetting phase, while at a gas-oil interface, oil is typically the wetting phase. In this thesis capillary pressures are neglected.

Well model

The average pressures and phase saturations in an unit volume grid block is described by Eqs. 2.10 and 2.11, however, the grid blocks in which wells are located require special modeling. A so-called *well model* gives a relationship between grid block pressures and pressures in the well (bottom hole pressure). The well flow equations used in reservoir simulation are a set of algebraic equations:

$$q_t = J_p(p_{bh} - p_{gb}), \tag{2.15}$$

where q_t is the total flow rate, p_{bh} is the bottom hole pressure, p_{gb} is the grid block pressure and J_p is the well index which is defined as a function of saturation, permeability, grid block geometry and reflects the effect of near-well flow. See Aziz and Settari (1979) for more details. A well operates at specified constraints, which control the maximum or minimum bottom hole pressure or maximum total flow rate. If the bottom hole pressure constraint is active, i.e. $p_{bh} = \bar{p}_{bh}$, then the total flow rate q_t can be calculated from Eq. 2.15 as

$$q_t = J_p(\bar{p}_{bh} - p_{gb}). \tag{2.16}$$

In case of operating a well at fixed total rate $q_t = \bar{q}$, the bottom hole pressure can be calculated as

$$p_{bh} = p_{gb} + J_p^{-1}\bar{q}. \tag{2.17}$$

Spatial discretization

Eqs. 2.10 and 2.11 define a system of coupled nonlinear partial differential equations, which cannot be solved analytically. A solution of this system is approximated at a given spatial and temporal grid. The numerical methods are required to approximate the terms of continuous partial differential equations in the discrete domains. In spacial discretization the reservoir is divided into block-centered grid, where the unknown variables are placed in the middle of each block. In case of a 2D model, each grid block is indexed by two indexes i, j, which describes their position in x and y direction, respectively. The transmissibility term at grid block (i, j) for water phase (when the gravity forces and capillary pressures are ignored) is approximated by

$$\nabla \cdot \left[\alpha \frac{k_{rw}}{\mu_w} k \nabla p_o\right]_{ij} \approx \frac{T(s_w)_{i+\frac{1}{2},j}(p_{i+1,j} - p_{i,j}) - T(s_w)_{i-\frac{1}{2},j}(p_{i,j} - p_{i-1,j})}{\Delta x \Delta y} + \frac{T(s_w)_{i,j+\frac{1}{2}}(p_{i,j+1} - p_{i,j}) - T(s_w)_{i,j-\frac{1}{2}}(p_{i,j} - p_{i,j-1})}{\Delta y \Delta x}$$

where the subscript o for pressure of oil is dropped for clarity, Δx and Δy are the grid block sizes in x and y directions, respectively, and

$$T(s_w)_{i\pm\frac{1}{2},j} := \frac{\Delta y}{\Delta x}\frac{\alpha}{\mu_w}(kk_{rw}(s_w))_{i\pm\frac{1}{2},j}, \tag{2.18}$$

$$T(s_w)_{i,j\pm\frac{1}{2}} := \frac{\Delta x}{\Delta y}\frac{\alpha}{\mu_w}(kk_{rw}(s_w))_{i,j\pm\frac{1}{2}} \tag{2.19}$$

are the transmissibilities between two grid blocks. The absolute permeability at the interface between grid blocks (i, j) and $(i \pm 1, j)$ is calculated as harmonic average, i.e.

$$k_{i\pm\frac{1}{2},j} = \frac{2}{\frac{1}{k_{i\pm 1,j}} + \frac{1}{k_{i,j}}} \tag{2.20}$$

and the relative permeability: $(k_{rw})_{i\pm\frac{1}{2},j}$ is approximated using the upstream weighting, i.e.

$$(k_{rw})_{i\pm\frac{1}{2},j} = \begin{cases} (k_{rw})_{i\pm 1,j}, \; p_{i\pm 1,j} \geq p_{i,j} \\ (k_{rw})_{i,j}, \; p_{i\pm 1,j} < p_{i,j} \end{cases}. \tag{2.21}$$

Similarly the transmissibility term for oil phase is approximated.

State space representation of the reservoir flow model

After the spatial discretization on a chosen grid the system of the partial differential equations becomes a system of ordinary differential equations which can be expressed as

$$\begin{aligned} \mathbf{E}(\mathbf{x}(t))\dot{\mathbf{x}}(t) &= \mathbf{A}(\mathbf{x}(t))\mathbf{x}(t) + \mathbf{B}(\mathbf{x}(t))\mathbf{u}(t), \\ \mathbf{x}(0) &= \mathbf{x}_{init}, \end{aligned} \tag{2.22}$$

where $\mathbf{x}(t)$ is a vector of all unknowns at time t, called a state vector, $\mathbf{x}^T = [\mathbf{p}^T\ \mathbf{s}^T]$, where $\mathbf{p}$ denotes an oil pressure vector and $\mathbf{s}$ denotes a water saturation vector. $\mathbf{E}$ is the accumulation matrix obtained after discretization of the accumulation term, matrix $\mathbf{A}$ is the transmissibility matrix which is obtained after discretization of the transmissibility term, and $\mathbf{u}$ is the vector of the well controls. Expression $\dot{\mathbf{x}}(t)$ denotes the derivative of the state vector with respect to time. $\mathbf{x}_{init}$ is the initial condition for the state vector. The initial condition is not known and is typically uncertain, but its uncertainty is much smaller compared to the uncertainty in parameters. In this thesis the initial condition is assumed to be perfect. The matrices $\mathbf{E}$, $\mathbf{A}$, $\mathbf{B}$ are sparse matrices which are functions of state $\mathbf{x}$, and therefore the differential equations are nonlinear. Matrix $\mathbf{A}$ depends on the state vector through the relative permeability term which is a function of saturation. In case of presence of capillary pressure (which is a function of saturation too) another source of state dependencies in this term would be introduced. The matrix $\mathbf{B}$, which is related to the sink/source term, is dependent on the state vector because in the well model J_p is a function of saturation (see Jansen (2011b)). In realistic reservoir models the nonlinearities are larger because of more complex modeling of density, porosity, viscosity or even permeability.

The discrete-time state space representation of the reservoir flow model

A commonly used time discretization in the reservoir simulation models is the fully implicit scheme, i.e. all dependent variables and inputs are evaluated at the next time step. Using Euler implicit time stepping Eq. 2.22 becomes

$$\begin{aligned} &\mathbf{E}(\mathbf{x}(t_n)) - \Delta t\mathbf{A}(\mathbf{x}(t_n))\mathbf{x}(t_n) = \mathbf{E}(\mathbf{x}(t_n))\mathbf{x}(t_{n-1}) + \Delta t\mathbf{B}(\mathbf{x}(t_n))\mathbf{u}(t_n), \\ &\mathbf{x}(0) = \mathbf{x}_{init}, \end{aligned} \tag{2.23}$$

where t_{n-1} denotes the current time step, t_n denotes the next time step, $Deltat = t_n - t_{n-1}$ and $n \in \{1, \ldots, N\}$, where t_N is the simulation length. This can be written as

$$\begin{aligned} &\mathbf{g}_n(\mathbf{x}(t_n), \mathbf{x}(t_{n-1}), \mathbf{u}(t_n)) = \mathbf{0}, \\ &\mathbf{x}(0) = \mathbf{x}_{init}. \end{aligned} \tag{2.24}$$

This system needs to be solved iteratively. Commonly, to solve the nonlinear system of flow equations the iterative Newton-Raphson method is used (Burden and Faires (2001)). At each simulation step a few Newton-Raphson iterations:

$$\begin{aligned} \mathbf{x}^k(t_n) &= \mathbf{x}^{k-1}(t_n) - [\frac{\partial \mathbf{g}_n}{\partial \mathbf{x}(t_n)}]^{-1}\mathbf{g}_n(\mathbf{x}^{k-1}(t_n), \mathbf{x}(t_{n-1}), \mathbf{u}(t_n)), \\ \mathbf{x}^0(t_n) &= \mathbf{x}(t_{n-1}) \end{aligned} \quad (2.25)$$

are required, where the Jacobian $\frac{\partial \mathbf{g}_n}{\partial \mathbf{x}(t_n)}$ is evaluated at $\mathbf{x}^{k-1}(t_n)$ and k stands for a Newton-Raphson iteration number. When the specified stopping criterion are satisfied an iteration terminates, i.e. the Newton-Raphson iteration converges, and $\mathbf{x}^k(t_n)$ is a solution of the system at time t_n. If Newton-Raphson does not succeed to converge then the time step size is reduced and the procedure is iterated again. In practice the inverse of Jacobian is never formed, instead at every iteration a linear system of equations is solved. The fully implicit method is unconditionally stable and therefore is commonly implemented. However, a too large time step may lead to many Newton-Raphson iterations or failure of convergence, and larger discretization errors.

In the remainder of this thesis a reservoir system is described by the implicit system of equations

$$\begin{aligned} &\mathbf{g}_n(\mathbf{x}(t_n), \mathbf{x}(t_{n-1}), \mathbf{u}(t_n), \boldsymbol{\theta}) = \mathbf{0}, \\ &\mathbf{j}_n(\mathbf{y}(t_n), \mathbf{x}(t_n), \mathbf{u}(t_n), \boldsymbol{\theta}) = \mathbf{0}, \\ &\mathbf{x}(t_0) = \mathbf{x}_{init} \end{aligned} \quad (2.26)$$

or the explicit one

$$\begin{aligned} &\mathbf{x}(t_n) = \mathbf{f}_n(\mathbf{x}(t_{n-1}), \mathbf{u}(t_n), \boldsymbol{\theta}), \\ &\mathbf{y}(t_n) = \mathbf{h}_n(\mathbf{x}(t_n), \mathbf{u}(t_n), \boldsymbol{\theta}), \\ &\mathbf{x}(t_0) = \mathbf{x}_{init} \end{aligned} \quad (2.27)$$

where $\mathbf{x}(t_n)$ is the state vector, a functional $\mathbf{f}_n$ and $\mathbf{g}_n$ describe the evolution of the state vector from time t_{n-1} to time t_n, $\boldsymbol{\theta}$ is a vector of uncertain parameters, $\mathbf{u}(t_n)$ is a vector of the well controls, $\mathbf{x}_{init}$ is a given initial state of the system and $\mathbf{y}(t_n)$ are the outputs of the system at time t_n given by the observation operator $\mathbf{h}_n$ or $\mathbf{j}_n$. In this thesis, the reservoir flow simulation always uses the implicit time discretization.

2.2 Adjoint model of the reservoir flow model

The reservoir flow models are used to solve parameter estimation or production optimization problems. Both of these problems can be formulated as optimization of a scalar objective function:

$$J = J(\mathbf{x}(t_1), ..., \mathbf{x}(t_N), \boldsymbol{\theta}), \ \ or \quad (2.28)$$

$$J = J(\mathbf{x}(t_1), ..., \mathbf{x}(t_N), \mathbf{u}), \quad (2.29)$$

where N corresponds to the total number of time steps and t_N represents the simulation time. J is typically a function of production data, and therefore depends on the state vectors (see Eqs. 2.27 or Eqs. 2.26). A computation of the gradient of the objective function with respect to model parameters or control variables requires the sensitivity of the system (given by Eqs. 2.26 or Eqs. 2.27), and is commonly performed by the adjoint method. In this section we present the derivation of the adjoint model of the reservoir system given by Eqs. 2.26 and the gradient of the objective function given by Eq. 2.28.

In the adjoint approach, first, a modified function $\bar{J}$ is obtained by adjoining the system equations (2.26) to J, and is given by

$$\bar{J} = J(\mathbf{x}_1, ..., \mathbf{x}_N, \boldsymbol{\theta}) + \sum_{n=1}^{N} \boldsymbol{\lambda}_n^T \mathbf{g}_n(\mathbf{x}_n, \mathbf{x}_{n-1}, \boldsymbol{\theta}) \tag{2.30}$$

where $\mathbf{x}_n$ represents $\mathbf{x}(t_n)$, $\boldsymbol{\lambda}_n$ represents the adjoint variables at time t_n, and (at this moment) is an arbitrary vector. The total variation of $\bar{J}$ which is sum of the variations with respect to $d\mathbf{x}_n$, for $n = 1, \dots, N$, and $d\boldsymbol{\theta}$, is

$$d\bar{J} = dJ + \sum_{n=1}^{N} \left\{ \boldsymbol{\lambda}_n^T \frac{\partial \mathbf{g}_n}{\partial \mathbf{x}_n} d\mathbf{x}_n \right\} + \sum_{n=1}^{N} \left\{ \boldsymbol{\lambda}_n^T \frac{\partial \mathbf{g}_n}{\partial \mathbf{x}_{n-1}} d\mathbf{x}_{n-1} \right\} + \sum_{n=1}^{N} \left\{ \boldsymbol{\lambda}_n^T \frac{\partial \mathbf{g}_n}{\partial \boldsymbol{\theta}} d\boldsymbol{\theta} \right\}. \tag{2.31}$$

Changing the index in terms which involve $\mathbf{x}(t_{n-1})$ yields

$$d\bar{J} = dJ + \sum_{n=1}^{N} \left\{ \boldsymbol{\lambda}_n^T \frac{\partial \mathbf{g}_n}{\partial \mathbf{x}_n} d\mathbf{x}_n \right\} + \sum_{n=0}^{N-1} \left\{ \boldsymbol{\lambda}_{n+1}^T \frac{\partial \mathbf{g}_{n+1}}{\partial \mathbf{x}_n} d\mathbf{x}_n \right\} + \sum_{n=1}^{N} \left\{ \boldsymbol{\lambda}_n^T \frac{\partial \mathbf{g}_n}{\partial \boldsymbol{\theta}} d\boldsymbol{\theta} \right\}. \tag{2.32}$$

Rearranging the limits of the sums involving $\mathbf{x}(t_n)$ yields

$$\begin{aligned} d\bar{J} \quad = \quad & dJ + \sum_{n=1}^{N} \left\{ \boldsymbol{\lambda}_n^T \frac{\partial \mathbf{g}_n}{\partial \mathbf{x}_n} d\mathbf{x}_n + \boldsymbol{\lambda}_{n+1}^T \frac{\partial \mathbf{g}_{n+1}}{\partial \mathbf{x}_n} d\mathbf{x}_n \right\} + \\ & \boldsymbol{\lambda}_1^T \frac{\partial \mathbf{g}_1}{\partial \mathbf{x}_0} d\mathbf{x}_0 - \boldsymbol{\lambda}_{N+1}^T \frac{\partial \mathbf{g}_{N+1}}{\partial \mathbf{x}_N} d\mathbf{x}_N + \sum_{n=1}^{N} \left\{ \boldsymbol{\lambda}_n^T \frac{\partial \mathbf{g}_n}{\partial \boldsymbol{\theta}} d\boldsymbol{\theta} \right\}. \end{aligned} \tag{2.33}$$

Since the initial conditions are fixed in our model, the term $\boldsymbol{\lambda}_1^T \frac{\partial \mathbf{g}_1}{\partial \mathbf{x}_0} d\mathbf{x}_0$ vanishes. The boundary term $\boldsymbol{\lambda}_{N+1}^T \frac{\partial \mathbf{g}_{N+1}}{\partial \mathbf{x}(t_N)} d\mathbf{x}_N$ is required to be zero for any differential $d\mathbf{x}_N$, which implies that $\boldsymbol{\lambda}_{N+1} = \mathbf{0}$. The total variation of J, which is a first term in Eq. 2.30, is the sum of variations with respect to all state variables $\mathbf{x}_n$ and parameters $\boldsymbol{\theta}$, and it is,

$$dJ = \sum_{n=1}^{N} \left\{ \frac{\partial J}{\partial \mathbf{x}_n} d\mathbf{x}_n \right\} + \frac{\partial J}{\partial \boldsymbol{\theta}} d\boldsymbol{\theta}. \tag{2.34}$$

It yields

$$\begin{aligned} d\bar{J} \quad = \quad & \sum_{n=1}^{N} \left\{ \frac{\partial J}{\partial \mathbf{x}_n} d\mathbf{x}_n \right\} + \frac{\partial J}{\partial \boldsymbol{\theta}} d\boldsymbol{\theta} + \\ & \sum_{n=1}^{N} \left\{ \boldsymbol{\lambda}_n^T \frac{\partial \mathbf{g}_n}{\partial \mathbf{x}_n} d\mathbf{x}_n + \boldsymbol{\lambda}_{n+1}^T \frac{\partial \mathbf{g}_{n+1}}{\partial \mathbf{x}_n} d\mathbf{x}_n \right\} + \sum_{n=1}^{N} \left\{ \boldsymbol{\lambda}_n^T \frac{\partial \mathbf{g}_n}{\partial \boldsymbol{\theta}} d\boldsymbol{\theta} \right\}. \end{aligned} \tag{2.35}$$

Rearranging again the terms results in

$$\begin{aligned} d\bar{J} &= \sum_{n=1}^{N} \left\{ \boldsymbol{\lambda}_n^T \frac{\partial \mathbf{g}_n}{\partial \mathbf{x}_n} + \boldsymbol{\lambda}_{n+1}^T \frac{\partial \mathbf{g}_{n+1}}{\partial \mathbf{x}_n} + \frac{\partial J}{\partial \mathbf{x}_n} \right\} d\mathbf{x}_n \\ &\quad + \sum_{n=1}^{N} \left\{ \boldsymbol{\lambda}_n^T \frac{\partial \mathbf{g}_n}{\partial \boldsymbol{\theta}} \right\} d\boldsymbol{\theta} + \frac{\partial J}{\partial \boldsymbol{\theta}} d\boldsymbol{\theta}. \end{aligned} \tag{2.36}$$

The adjoint model is obtained by setting all coefficients of $d\mathbf{x}(t_n)$ to zero, i.e.

$$\boldsymbol{\lambda}_n^T \frac{\partial \mathbf{g}_n}{\partial \mathbf{x}_n} + \boldsymbol{\lambda}_{n+1}^T \frac{\partial \mathbf{g}_{n+1}}{\partial \mathbf{x}_n} + \frac{\partial J}{\partial \mathbf{x}_n} = \mathbf{0}. \tag{2.37}$$

Taking the transpose of the above equations gives the system of adjoint equations

$$\left(\frac{\partial \mathbf{g}_n}{\partial \mathbf{x}_n} \right)^T \boldsymbol{\lambda}_n + \left(\frac{\partial \mathbf{g}_{n+1}}{\partial \mathbf{x}_n} \right)^T \boldsymbol{\lambda}_{n+1} + \left(\frac{\partial J}{\partial \mathbf{x}_n} \right)^T = \mathbf{0}, \tag{2.38}$$

for $n = \{1, \ldots, N\}$. This system is solved backward in time because of the given ending condition $\boldsymbol{\lambda}_{N+1} = \mathbf{0}$, which can be interpreted as the initial condition for the adjoint system.
If $\boldsymbol{\lambda}_n$ satisfy Eq. 2.38 then the total variations of $\bar{J}$ are given by

$$d\bar{J} = \frac{\partial J}{\partial \boldsymbol{\theta}} d\boldsymbol{\theta} + \sum_{n=1}^{N} \left\{ \boldsymbol{\lambda}_n^T \frac{\partial \mathbf{g}_n}{\partial \boldsymbol{\theta}} d\boldsymbol{\theta} \right\}, \tag{2.39}$$

and it follows that the total derivatives of J with respect to $\boldsymbol{\theta}$ are given by

$$\frac{d\bar{J}}{d\boldsymbol{\theta}} = \frac{\partial J}{\partial \boldsymbol{\theta}} + \sum_{n=1}^{N} \left\{ \boldsymbol{\lambda}_n^T \frac{\partial \mathbf{g}_n}{\partial \boldsymbol{\theta}} \right\}. \tag{2.40}$$

An important remark is that the reservoir equations are nonlinear, whereas, the adjoint equations are linear. The Newton-Raphson method requires few iterations to converge at each time step of the reservoir flow model, whereas the adjoint model uses the sensitivity of the system at the converged solution of the Newton-Raphson method and, therefore, requires only one system of equations at each time step to be solved.

2.3 Reservoir flow model under induced fracturing conditions

The process of waterflooding-induced hydraulic fracturing has been studied by Hagoort (1981), Perkins and Gonzalez (1985), Koning (1988), Dikken and Niko (1987), Clifford et al. (1991) and Tran et al. (2004), to mention a few. In the past, fracturing conditions were preferably avoided, even if it resulted in a very low injection rates and a very slow oil recovery. Recent studies showed that in low-mobility reservoirs it is practically impossible to avoid fracturing conditions and that water injection and part of enhanced oil recovery operations (polymer flooding and steam injection) nearly always take place under induced fracturing conditions. The induced fractures may grow to several hundreds of meters and therefore it introduces a risk of an unfavorable sweep. Van den Hoek et al. (2008) showed that induced fractures in a waterflooded reservoir are very sensitive to parameters such as fluid mobility, mobility ratio, position of wells, 3D saturation distribution, geological details and injection strategy. In this work the dynamic behavior of a fracture under different conditions was studied and optimization of the injection strategy using a simple finite difference approach was performed.

In this chapter a two way coupling of a fracture growth simulator and a reservoir fluid flow simulator described in Zwarts et al. (2006); Hustedt et al. (2008) is presented.

2.3.1 Stress calculation

To describe the induced fracturing conditions in waterflooded reservoir a geomechanical model is required. It provides a basis to model induced fracturing conditions. The geomechanical model derived in this thesis is described in more details in Zoback (2007). A key element of a comprehensive geomechanical model is knowledge of the current state of stress field. The in-situ stress field at depth, i.e. the local stress in a given rock mass at depth, in practice can, to some extent, be determined from borehole measurements and regional geological features.

In terms of continuum mechanics, the stresses acting on a homogeneous, isotropic body at depth are describable as a second-rank tensor, with components:

$$\sigma = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{bmatrix}. \tag{2.41}$$

The individual stress component represents a force acting in a specific direction on a unit area of given orientation. The stresses perpendicular to the plane (with two identical subscripts) may be compressive or tensile, while the stresses parallel to the plane are called shear.

Acting forces may result in a deformation. The deformation of a body can be described as a transformation from a reference configuration into another one (the displacement) and can be defined by considering the second-order strain tensor, defined as

$$\epsilon_{ij} := \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right), \tag{2.42}$$

where $x_i, x_j, i, j \in \{x, y, z\}$ and $\mathbf{u} = [u_x, u_y, u_z]$ is a displacement vector describing the deformation from a reference configuration. The deformation is caused by forces; if these are removed a purely elastic body comes back to its original form. A linearly elastic material is one in which stress and strain are linearly proportional and deformation is reversible. The rock formation is not really an elastic body but its behavior can be described as linearly elastic as long as the deformations are sufficiently small. Under these assumptions and for an isotropic medium, stress can be expresses in terms of strain by the following relation

$$\sigma_{ij} = \lambda \delta_{ij}(\epsilon_{xx} + \epsilon_{yy} + \epsilon_{zz}) + 2G\epsilon_{ij}, \tag{2.43}$$

where the Kronecker delta, δ_{ij}, is given by

$$\delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} . \tag{2.44}$$

The shear modulus G is defined as the ratio of an applied shear stress to a corresponding shear strain, i.e.

$$G = \frac{1}{2}\frac{\sigma_{ij}}{\epsilon_{ij}} \tag{2.45}$$

and it is a material property. The Lamé's parameter, λ, satisfies given relations

$$\lambda = \frac{2G\nu}{1-2\nu} = \frac{E\nu}{(1+\nu)(1-2\nu)}, \tag{2.46}$$

where ν denotes the Poisson's ratio which under a uni-directional stress in the x-direction is defined as

$$\nu = -\frac{\epsilon_{zz}}{\epsilon_{xx}} = -\frac{\epsilon_{yy}}{\epsilon_{xx}}, \tag{2.47}$$

and describes the proportion of the lateral expansion $\epsilon_{zz} = \frac{\partial u_z}{\partial z}$ and $\epsilon_{yy} = \frac{\partial u_y}{\partial y}$ to axial shortening $\epsilon_{xx} = \frac{\partial u_x}{\partial x}$, and it is a material property too. Young's modulus, E, is a stiffness of the rock in simple uni-axial compression, i.e.

$$E = \frac{\sigma_{xx}}{\epsilon_{xx}}, \tag{2.48}$$

and it is a material property as well. In an isotropic material the shear modulus is related to Young's modulus and the Poisson ration by

$$G = \frac{E}{2(1+\nu)}. \tag{2.49}$$

Eq. 2.43 describes the relation between stress and strain of a rock, however, a reservoir is saturated with fluids. In this thesis we assume that reservoir pores are uniformly saturated with fluid, that the volume of pores is small compared to the volume of the rock and that pressure in the pore is the total stress acting on the rock externally. The absolute stress in a porous medium is carried by the rock and the fluid in the pores of the rock. This additional stress related to the pore pressure is called backstress. When the formation behaves in a linear poro-elastic manner, the effective stress on the rock, σ'_{ij}, is related to the absolute stress σ_{ij} and the fluid pressure p by

$$\sigma'_{ij} = \sigma_{ij} - \delta_{ij} a_p p, \tag{2.50}$$

where a_p is the poro-elastic constant and $i, j \in \{x, y, z\}$ (see Terzaghi (1943) for more details). Eq. 2.50 says that only the normal components of the stress tensor are influenced by the pore pressure (and not the shear components). Assuming linear elasticity of the reservoir rock (Eq. 2.43) and linear poro-elasticity of a formation (Eq. 2.50) result in the following relation

$$\Delta\sigma_{ij} = \frac{E}{1+\nu}\left[\epsilon_{ij} + \frac{\nu}{1-2\nu}\delta_{ij}\left(\epsilon_{xx} + \epsilon_{yy} + \epsilon_{zz}\right)\right] + \frac{E\alpha_p}{1-2\nu}\Delta p\delta_{ij}, \tag{2.51}$$

where $\Delta\sigma_{ij}$ and Δp denote the change in the stress profile and pore pressure in reference to their initial conditions and

$$\alpha_p = \frac{1-2\nu}{E}a_p. \tag{2.52}$$

Eq. 2.51 says that the change in the stress field is described by a function that depends on the pore pressure and the strain. Moreover, the relation between stress and pressure is linear. The final formula for the stress calculation can be found in Koning (1988). When cold water is injected into a reservoir the formation temperature changes. Assuming that the expansion of the rock is proportional to the temperature change, thermo-elasticity can be described using the thermo-poro-elasticity theory. In this thesis, however, we consider isothermal conditions.

2.3.2 Fracture propagation or closure

If the formation contains a crack, then at each time step of the simulation the following situations can be observed:

1. stable or unstable fracture propagation,
2. fracture shrinkage,
3. no fracture propagation or shrinkage.

The propagation is called stable if it is caused by the fluid injection in the well, while it is called unstable if it grows without fluid injection. The calculation of new fracture size is based on the principle that the pressure inside the fracture and the in-situ minimum stresses around the fracture are balanced. In an elastic medium that is subjected to a non-uniform force the fracture propagation criterion is given by

$$K_I = K_{IC} \tag{2.53}$$

where K_{IC} is called a fracture toughness, which describes the ability of a material containing a crack to resist fracture, while K_I is called the stress intensity factor (mode I) at the particular tip of the fracture, which depends on the entire load profile on the fracture face (more about fracture mechanics can be found in Anderson (1995)). If we assume an uniform load on the fracture face then the amount of pressure sufficient to propagate the fracture is $\frac{K_{IC}}{\pi L}$, where L stands for the half-length of the fracture (see Barenblatt (1962)). If the fracture pressure is $\frac{K_{IC}}{\pi L}$ higher than the in-situ stress on the fracture tip, then the fracture grows. If we define the propagation pressure as

$$p_{prop} := \frac{K_{IC}}{\pi L} + \sigma \tag{2.54}$$

then the propagation criterion becomes

$$p_{frac} = p_{prop}, \tag{2.55}$$

p_{frac} is the pressure inside the fracture, and σ is the in-situ stress at the fracture tip. The fracture closes if the fracture pressure becomes smaller than the in-situ stress on the fracture tip. Defining the closure pressure as

$$p_{closure} := \sigma \tag{2.56}$$

the closure criterion becomes

$$p_{frac} = p_{closure}. \tag{2.57}$$

If the calculated fracture pressure is larger than the closure pressure but smaller that the propagation pressure then no shrinkage and no growth of the fracture takes place.

2.3.3 Coupling routine

In the fracture simulator considered for this thesis a fracture is modeled as a plane of an elliptical shape, perpendicular to the fixed minimum (mostly horizontal) stress direction, and with a small aperture. The center of the fracture is positioned in the well interval, where the fracture is initiated. A fracture is modeled by three parameters, $\mathbf{L} = [L_1, L_2, L_3]$ (left-right horizontal length (L_1) and upward-downward length ($L_2 - L_3$), see Fig. 2.2). Orientation, initial size and width of a fracture are assumed to be known.

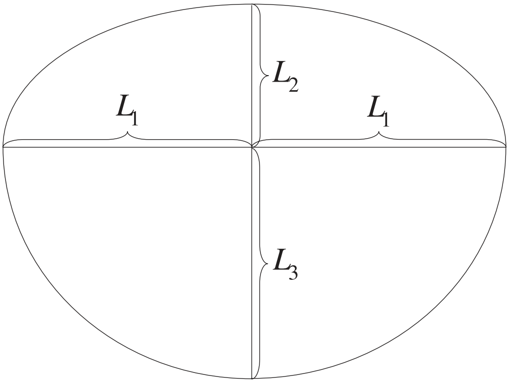


Figure 2.2: Section through the main axes of a fracture.

Changes in the reservoir pressures influence the in-situ stresses, changes in the in-situ stresses and fracture pressure influence the size of the fracture, while changes in the fracture size influence the reservoir pressure. It requires thus a two-way coupling. The reservoir pressure is calculated by the standard reservoir flow model, in which fracture is modeled by assigning new transmissibilities to the reservoir grid blocks that are intersected with the fracture or which are very close to it. Since the outflow area occurs on both side of the fracture, the grid block pressure around the fracture is modeled in similar way as the grid block pressure around a well. A flow around a small fracture is modeled using a radial flow model, and the transmissibility of the grid block affected by the fracture is defined by

$$T := \frac{2\pi kh}{ln\left(\frac{2R_P}{L}\right)}, \tag{2.58}$$

where R_P is the Peaceman radius (see Peaceman (1977)), k is the absolute permeability, h is the hight of the grid block, and L is one fracture size. A flow profile around a fracture with a size larger than $2R_P$ is assumed to be elliptical, which results in the following transmissibility of the affected grid block:

$$T := \frac{2\pi kh}{sinh^{-1}\left(\frac{2R_P}{L}\right)}. \tag{2.59}$$

From Eqs. 2.58 and 2.59 we see that the transmissibility becomes dependent on the fracture size L.

Using the updated transmissibility matrix the standard reservoir flow simulator is capable to model the flow under induced fracturing conditions. When the reservoir pressures is computed, the fracture growth simulator calculates the in-situ stress and the closure/propagation pressure. If the fracture propagates, then the size of the propagation is determined and a new fracture size is calculated. If the new fracture size satisfies the propagation criterion, then the new time step is iterated, otherwise the fracture size is corrected. One proceeds similarly when the fracture closes. When no closure and no propagation takes place, the fracture sizes are not updated, and a new time step is calculated.

2.4 Adjoint model of the reservoir flow model under induced fracturing conditions

In the described coupling the reservoir flow simulator is adapted to model the flow in porous media under induced fracturing conditions. This modeling does not require any internal code adjustment of the flow simulator but does require an internal adjustment of the adjoint code. Because our system is a two-way coupling, changes in the state vector cause changes in the fracture size, while changes in the fracture size cause changes in the transmissibility matrix and consequently changes in the state vector. It means that the transmissibility depends on the state vector, which has to be included in the adjoint model. In this section we present the adjoint equations for this coupled system and we discuss the possibility to combine it with the existing adjoint code.

Let us assume that the fluid flow in the reservoir with induced fracture is described by the following system of equations

$$\mathbf{c}_{i+1}(\mathbf{x}(t_{i+1}), \mathbf{x}(t_i), \mathbf{u}(t_{i+1})) = \mathbf{0}, i = 0, ..., N-1, \tag{2.60}$$

and the output data $\mathbf{y}(t_{i+1})$ is described by

$$\mathbf{h}_{i+1}(\mathbf{u}(t_{i+1}), \mathbf{x}(t_{i+1}), \mathbf{y}(t_{i+1})) = \mathbf{0}, i = 0, ..., N-1. \tag{2.61}$$

Let us assume that we are interested in the production optimization of this system. An objective function is defined then as an explicit function of the production/injection rates and can be written as

$$J(\mathbf{u}) = \sum_{i=0}^{N-1} J_{i+1}\left(\mathbf{u}_{i+1}, \mathbf{y}_{i+1}\right), \tag{2.62}$$

where $\mathbf{y}_{i+1}$ is given by Eq. 2.61 which requires an evaluation of the state vector $\mathbf{x}(t_{i+1})$ described by Eq. 2.60. It can be shown that the adjoint model for this coupled problem is

$$\boldsymbol{\lambda}_{i+1}^T \frac{\partial \mathbf{c}_{i+1}}{\partial \mathbf{x}_{i+1}} + \boldsymbol{\lambda}_{i+2}^T \frac{\partial \mathbf{c}_{i+2}}{\partial \mathbf{x}_{i+1}} + \left(\frac{\partial J_{i+1}}{\partial \mathbf{y}_{i+1}}\right)^T \frac{\partial \mathbf{h}_{i+1}}{\partial \mathbf{x}_{i+1}} = \mathbf{0}. \tag{2.63}$$

It looks similar to the adjoint model of the reservoir flow model without fracturing condition. However, since the coupled system imposes certain changes in the reservoir flow model, the Jacobians of the coupled system, $\frac{\partial \mathbf{c}_{i+1}}{\partial \mathbf{x}_{i+1}}$ and $\frac{\partial \mathbf{c}_{i+2}}{\partial \mathbf{x}_{i+1}}$ are changed. In the coupled system the time step evolution described by operator $\mathbf{c}_{i+1}$ consists of few trial runs of the flow simulator and the fracture simulator. During those runs an appropriate fracture size is calculated. To solve a coupled system first the reservoir flow equations are solved given the fracture dimensions $\mathbf{L}_{i+1} = \mathbf{L}(t_{i+1})$. The transmissibilities, $\mathbf{T}_{i+1}$, of the grid blocks which are affected by the fracture are recalculated using Eq. 2.58 or Eq. 2.59. This can be represented by the following operator:

$$\mathbf{T}_{i+1} = \boldsymbol{\psi}_{i+1}(\mathbf{L}_{i+1}), i = 0, ..., N-1. \tag{2.64}$$

Using updated $\mathbf{T}_{i+1}$ the standard flow equations are solved, i.e.

$$\mathbf{g}_{i+1}(\mathbf{x}_{i+1}, \mathbf{x}_i, \mathbf{u}_{i+1}, \mathbf{T}_{i+1}) = \mathbf{0}, i = 0, ..., N-1. \tag{2.65}$$

Next, the fracture growth simulator is used to calculate the backstresses and the propagation or closure criterion. Since stress depends linearly on pressure it can be described as follows

$$\boldsymbol{\sigma}_{i+1} = \mathbf{F}_{i+1}\mathbf{a}_{i+1},\ i = 0, ..., N-1 \tag{2.66}$$

where $\mathbf{F}_{i+1}$ denotes the linear operator, $\boldsymbol{\sigma}_{i+1}$ denotes the stress vector at time t_{i+1}, and $\mathbf{a}_{i+1}$ denotes the data required from reservoir flow simulation (i.e. the reservoir pressure), and can be interpreted as an operation on the state vector $\mathbf{x}$:

$$\mathbf{a}_{i+1} = \boldsymbol{\phi}(\mathbf{x}_{i+1}),\ i = 0, ..., N-1. \tag{2.67}$$

The stress σ_{i+1} is required to calculate the propagation (or closure) pressure $\mathbf{p}_{prop}$ (or $\mathbf{p}_{closure}$) and criterion $\mathbf{crit}_{i+1}$, which is defined as

$$\mathbf{crit}_{i+1}(\sigma_{i+1}, \mathbf{L}_{i+1}, \mathbf{a}_{i+1}) := \mathbf{p}_{prop/closure}(\sigma_{i+1}, \mathbf{L}_{i+1}) - \mathbf{p}_{frac}(\mathbf{a}_{i+1}), \tag{2.68}$$

and it is required to be zero (see Eqs. 2.55 and 2.57), i.e.

$$\mathbf{crit}_{i+1}(\sigma_{i+1}, \mathbf{L}_{i+1}, \mathbf{a}_{i+1}) = \mathbf{0}. \tag{2.69}$$

If the propagation (closure) criterion is satisfied for each fracture tip then the next time step can be evaluated; otherwise the fracture sizes are updated. The size of the fracture is updated using a gradient of the criterion $\mathbf{crit}_{i+1}$ with respect to $\mathbf{L}_{i+1}$. The gradient is approximated by finite differences, hence, it requires a few additional iterations of flow simulator. These intermediate steps, however, are not important for the adjoint calculations because the adjoint model is derived using the Jacobians of the system at the final solution of the coupled system, that is, when the propagation (or closure) criterion is satisfied. The adjoint code for the reservoir flow model under induced fracturing conditions requires the implementation of the Jacobians $\frac{\partial \mathbf{c}_{i+1}}{\partial \mathbf{x}_{i+1}}$ which are given by

$$\frac{\partial \mathbf{c}_{i+1}}{\partial \mathbf{x}_{i+1}} = \frac{\partial \mathbf{g}_{i+1}}{\partial \mathbf{x}_{i+1}} + \frac{\partial \mathbf{g}_{i+1}}{\partial \mathbf{T}_{i+1}} \frac{\partial \mathbf{T}_{i+1}}{\partial \mathbf{L}_{i+1}} \frac{\partial \mathbf{L}_{i+1}}{\partial \mathbf{x}_{i+1}}, \tag{2.70}$$

for $i = 0, ..., N-1$, and differ from the Jacobians of the reservoir flow model without induced fracturing conditions. The term $\frac{\partial \mathbf{g}_{i+1}}{\partial \mathbf{x}_{i+1}}$ is implemented already in reservoir flow simulator, while the terms $\frac{\partial \mathbf{T}_{i+1}}{\partial \mathbf{L}_{i+1}}$, $\frac{\partial \mathbf{L}_{i+1}}{\partial \mathbf{x}_{i+1}}$ and $\frac{\partial \mathbf{g}_{i+1}}{\partial \mathbf{T}_{i+1}}$ are new. The missing terms $\frac{\partial \mathbf{g}_{i+1}}{\partial \mathbf{T}_{i+1}}$ should be relatively easy to derive because reservoir flow simulator contains the adjoint-based gradient for the transmissibilities. $\frac{\partial \mathbf{g}_{i+1}}{\partial \mathbf{T}_{i+1}}$ is not directly available but is part of gradient calculation and can be retrieved from the code. The term $\frac{\partial \mathbf{T}_{i+1}}{\partial \mathbf{L}_{i+1}}$ is obtained by differentiation of Eq. 2.64, i.e.

$$\frac{\partial \mathbf{T}_{i+1}}{\partial \mathbf{L}_{i+1}} = \frac{\partial \boldsymbol{\psi}_{i+1}}{\partial \mathbf{L}_{i+1}} \tag{2.71}$$

and is easy to derive (see Eqs. 2.58 and 2.59). To calculate $\frac{\partial \mathbf{L}_{i+1}}{\partial \mathbf{x}_{i+1}}$ we need to differentiate Eq. 2.69. A small variation of the state vector, $\delta \mathbf{x}_{i+1}$, gives a rise to a small variation $\delta \mathbf{L}_{i+1}$ of the fracture size, a small variation $\delta \mathbf{a}_{i+1}$ to the data given to fracture growth simulator and a small variation $\delta \sigma_{i+1}$ of a stress profile, i.e.

$$\begin{aligned}\frac{\partial \mathbf{crit}_{i+1}}{\partial \mathbf{x}_{i+1}} &= \frac{\partial \mathbf{crit}_{i+1}}{\partial \mathbf{L}_{i+1}} \cdot \frac{\partial \mathbf{L}_{i+1}}{\partial \mathbf{x}_{i+1}} + \frac{\partial \mathbf{crit}_{i+1}}{\partial \mathbf{a}_{i+1}} \cdot \frac{\partial \mathbf{a}_{i+1}}{\partial \mathbf{x}_{i+1}} + \\ &\quad \frac{\partial \mathbf{crit}_{i+1}}{\partial \sigma_{i+1}} \cdot \frac{\partial \sigma_{i+1}}{\partial \mathbf{a}_{i+1}} \cdot \frac{\partial \mathbf{a}_{i+1}}{\partial \mathbf{x}_{i+1}}.\end{aligned} \tag{2.72}$$

Since Eq. 2.69 has a zero on the right hand side, its derivative has to be zero as well. Thus, $\frac{\partial \mathbf{L}_{i+1}}{\partial \mathbf{x}_{i+1}}$ is given by the following relation

$$\begin{aligned} \frac{\partial \mathbf{crit}_{i+1}}{\partial \mathbf{L}_{i+1}} \cdot \frac{\partial \mathbf{L}_{i+1}}{\partial \mathbf{x}_{i+1}} &= -\frac{\partial \mathbf{crit}_{i+1}}{\partial \mathbf{a}_{i+1}} \cdot \frac{\partial \mathbf{a}_{i+1}}{\partial \mathbf{x}_{i+1}} \\ &\quad -\frac{\partial \mathbf{crit}_{i+1}}{\partial \sigma_{i+1}} \cdot \frac{\partial \sigma_{i+1}}{\partial \mathbf{a}_{i+1}} \cdot \frac{\partial \mathbf{a}_{i+1}}{\partial \mathbf{x}_{i+1}}, \end{aligned} \tag{2.73}$$

where $\frac{\partial \mathbf{a}_{i+1}}{\partial \mathbf{x}_{i+1}}$ is given by

$$\frac{\partial \mathbf{a}_{i+1}}{\partial \mathbf{x}_{i+1}} = \frac{\partial \boldsymbol{\phi}_{i+1}}{\partial \mathbf{x}_{i+1}} \tag{2.74}$$

and $\frac{\partial \sigma_{i+1}}{\partial \mathbf{a}_{i+1}} = \mathbf{F}_{i+1}$ (see Eq. 2.66). The remaining terms ($\frac{\partial \mathbf{crit}_{i+1}}{\partial \sigma_{i+1}}$ and $\frac{\partial \mathbf{crit}_{i+1}}{\partial \mathbf{a}_{i+1}}$) are also simple to derive (see Eq. 2.69).

2.5 Difficulties implementing an adjoint model

In real life reservoir simulation, the level of modeled details can be very advanced and the system can be very complex. A simulator that offers an advanced modeling solves a nonlinear system of equations and consists of millions of lines, which were built by many different people in few decades time. In such a case, the implementation of the adjoint method is a collarwork, because the implementation of the partial derivatives of the system (see section 2.2) requires detailed knowledge of the simulation code. Even if the sensitivity matrices with respect to the current state vectors are available (because they are used in the Newton-Raphson iterations), the sensitivity matrices with respect to parameters can be very complicated to derive. Moreover, the sensitivity matrices with respect to the previous state vectors are not implemented; similarly, the sensitivities of the objective function are not available. Nowadays, if all partial derivatives of the system are available, then the adjoint code is relatively straightforward. Nevertheless, till a few years ago a storage capability was also a limitation for the adjoint code. To solve the adjoint equations it is necessary to save the state vectors of the forward simulation. This problem was solved by storing only a limited number of state vectors and by recomputing the remaining ones. Nowadays, all the simulated state vectors can be stored to disk.

For the purpose of this thesis we used three reservoir simulators: SIMSIM (SIMple SIMulator) developed at Delft University of Technology, MoReS (The Modular Reservoir Simulator) built at Shell International Exploration & Production and coupling of MoReS with FRAC-IT simulator (FRACture simulator) built at Shell International Exploration & Production too.
The SIMSIM simulator models two phase flow in a 2D reservoir, it assumes isothermal conditions, absence of capillary pressure, and constant fluid and rock parameters. It is an open source simulator, and therefore it was used for the initial research purposes. Additionally, it contains the implementation of the adjoint-based

gradient for the permeability field. We used those gradients for an evaluation of the proposed methods.
MoReS and FRAC-IT simulators are fully capable to model complex reservoir conditions, but the user has no access to the source code. The MoReS simulator contains the adjoint-based gradients for permeability field, porosity, controls and few more parameters. The coupled simulator, however, does not contain the adjoint-based gradients. In this research, we consider methods that do not require an access to the source code and are derivate-free.

3 CHAPTER

History matching

In typical reservoir simulation models many parameters are uncertain, e.g. grid block permeabilities and porosities, relative permeabilities, and fault multipliers. A prior reservoir characterization is generally based on wellbore and logging data that are interpolated to give regional descriptions of uncertain geological properties. The interpolation process introduces uncertainty in those parameters that directly translates into uncertainty in the reservoir behavior. The geological model can be improved by using surface and downhole production data (flow rates and bottom-hole pressures) or seismic data in a process called history matching. History matching identifies the parameter values that minimize an objective function representing the mismatch between modeled and observed production data. The objective function is usually defined as a sum of weighted squared differences between observed and modeled data. History matching is challenging and integral aspect of reservoir modeling. This chapter describes the data that is typically collected during the evaluation phase and the data that is available during the production process. Additionally, the history matching problem is formulated in the Bayesian framework and a brief review of different history matching methods is presented. The chapter concludes with a review of adjoint-free gradient-based history matching and parameterization methods.

3.1 Data

While modeling the reservoir two kind of data are typically available: the data collected at the exploration stage (prior data) and data collected during the production process (data measured at well locations and seismic data). They provide information about the reservoir and improve the quality of predictions.

3.1.1 The prior probability

In the exploration phase of the reservoir trial wells are drilled to inspect the presence of hydrocarbons. The drill cutting, i.e. the chips of rock removed from the reservoir, are gathered and inspected. In addition, cores are taken over important parts of the reservoirs. These core samples are long cylinders of rock approximately two or five inches in diameter. Various measurements are taken from smaller core plugs which are one or one and half inches in diameter. They are used to see fluid content, the depositional environment, the diagenetic effects and the presence of fractures. The rock and fluid properties such as a litologic description, porosity, permeability, grain density, fluid saturation, relative permeabilities, capillary pressure relations are estimated from these samples too. Those measurements are called wellbore data and they are the only direct measurements of rock formation and fluid properties. Another commonly used data are obtained by lowering down into a well a probe equipped with sensors. The signals are transmitted to the surface through a cable and then they are processed and interpreted. This process is called well-logging and taken measurements are called log data. They give information about water or salt content, the age of the rock and its porosity, presence of sand and shale minerals, fluid properties in near-well region. The well-logging measurements can be categorize roughly in three groups: electric, acoustic and radioactivity logging (Oliver et al. (2008)). The wellbore data and logs data are sometimes called the hard data. They are collected from a wellbore region that is much smaller than the size of a reservoir simulation gridblock. To cover a large area of a model seismic data should be collected. Seismic data are measurements of wave signals that have traveled from a source at the surface, through the formation, are reflected from a subsurface layer, and are recorded by receivers at the surface. The wave carry information about the properties of the formation through which they travel, and are used to map the formation structure and property fields. A seismic survey is the most widely used geophysical method in subsurface mapping. It covers a large area and is helpful in understanding sedimentary series, overall formation thickness and large-scale properties of sedimentation.

The hard data, seismic data and the geologic constraints are used to describe the prior information about geological model. The parameters of the geostatistical models describing the spatial relationship within a field are based on information present in those data. These data, however, contain noise. The core data is contaminated by random noise and interpretation errors. The log data may be contaminated by many errors: the randomness in the phenomenon measured, lack of calibration, incorrect choice of parameters, interpretation process. The errors are estimated based on a comparison between multiple estimates. In case of seismic data noise is introduced during an acquisition process and during complicated data processing procedures. The errors may be caused by poorly planted geophones, wind motion, wave motion in the water that vibrates the receivers and cables, static noise from recording instruments and production operations in an adjacent oil field. However, the procedures in the seismic data processing are often interpretive and the decisions on parameter settings may be based on geological intuition and personal preference. The quantification of the error level is therefore difficult.

3.1.2 Production data

One type of the measurements available during an operation process are the production data. They refer to measurements observed in production and injection wells. Since they are taken at well locations, they are usually very limited in space but they can be frequently repeated in time. Available production measurements are: bottom hole pressure, tubing head pressure, flow rates or water-oil ration, gas-oil ratio. Bottom hole pressure i.e. pressure measured at the bottom of a well by the downhole pressure sensors can be collected typically on a daily basis, unfortunately the pressure sensors are not available in most wells. Tubing head pressure is measured at the top of the well, typically on a daily basis. Flow rates, for a well or a group of wells, are measured on a monthly basis, after a separation of the produced fluid is performed. Similarly, the water-oil or gas-oil ratios are determined. Likewise the prior data, the production data contain errors. Nowadays used gaugers have often a resolution of 0.01 psi, but the accuracy of the pressure data is not as good. The reasons for that are transient temperature effects and drift in electronics. Rate measurements may be obtained from a separator whose accuracy could be estimated reasonably well. In many cases a well is tested at most once per month, and intermediate rates are estimated from a well-head pressure. It does not make much sense to take the accuracy of the instantaneous measurement as the accuracy of the monthly rate, because usually the modeling error is more important than the measurement error. In the history matching procedure it is very important to know the magnitude of the errors because it is useless to match the data to the greater accuracy than the accuracy of the measurements.

3.2 History matching problem in the Bayesian framework

In history matching uncertain observations are used to improve the uncertain model parameters, $\boldsymbol{\theta}$. A full solution of this inverse problem is the a-posteriori probability density function for the model parameters. The complete characterization of the probability distribution function is usually impractical. Only under the assumption that the model is linear, the prior model is Gaussian and the measurement noise is Gaussian, the a-posteriori distribution is Gaussian, and is completely characterized by mean and covariance. In the general case, by matching production data an estimate of $\boldsymbol{\theta}$ is obtained which is an approximate sample of the a-posteriori probability distribution function.

In the Bayesian framework the unknown model parameters are treated as random variables described by multivariate probability distributions. We let $f(\boldsymbol{\theta})$ denote the prior probability density function for the parameter $\boldsymbol{\theta}$ which is based on the hard data, geologic and seismic data. This represents the initial knowledge about the model parameters and their initial uncertainty. We look for an estimate of the conditional probability density function for model parameters given production data $\mathbf{d}(t_i)$, for $i = 1, \ldots, N_o$. The conditional probability density function should reduce the uncertainty about the unknown parameters below the level given by the prior model. Bayes's rule says that the conditional probability density function for model parameters given production data $\mathbf{d}(t_i)$ is

$$f(\boldsymbol{\theta}\,|\{\mathbf{d}(t_i) : i = 1, \ldots N_o\}\,) \quad = \quad \frac{f(\{\mathbf{d}(t_i) : i = 1, \ldots N_o\}\,|\boldsymbol{\theta}\,)f(\boldsymbol{\theta})}{f(\{\mathbf{d}(t_i) : i = 1, \ldots N_o\})}. \tag{3.1}$$

$f(\{\mathbf{d}(t_i) : i = 1, \ldots N_o\}\,|\boldsymbol{\theta}\,)$ is the conditional probability of the data given the model parameters $\boldsymbol{\theta}$. It is a measure of how likely the measurements are given a certain geological model. $f(\{\mathbf{d}(t_i) : i = 1, \ldots N_o\})$ is the probability of the measurements, and it does not depend on $\boldsymbol{\theta}$. It is calculated as a normalization factor for the a-posteriori probability density function. The model with the highest probability density $f(\boldsymbol{\theta}\,|\{\mathbf{d}(t_i) : i = 1, \ldots N_o\}\,)$ is called the maximum a-posteriori (MAP) estimate. Because of the imperfect measuring techniques, the observed data contains the noise, which can be modeled by the following relationship:

$$\mathbf{d}(t_i) = \mathbf{h}_i(\boldsymbol{\theta}) + \mathbf{v} \tag{3.2}$$

where $\mathbf{h}_i$ is an operator describing the relationship between measurements and parameters at time t_i, and $\mathbf{v}$ is the noise vector. Assuming that the measurements errors are Gaussian distributed with zero mean and covariance matrix $\mathbf{R}_i$, the likelihood function of observations, given a certain model $\boldsymbol{\theta}$, can be expressed as:

$$f(\{\mathbf{d}(t_i) : i = 1, \ldots N_o\}\,|\boldsymbol{\theta}\,) \propto e^{\left\{-\frac{1}{2}\sum_{i=1}^{N_o}(\mathbf{d}(t_i)-\mathbf{h}_i(\boldsymbol{\theta}))^T\,\mathbf{R}_i^{-1}(\mathbf{d}(t_i)-\mathbf{h}_i(\boldsymbol{\theta}))\right\}}. \tag{3.3}$$

Furthermore, assuming that model parameters are Gaussian distributed with mean $\boldsymbol{\theta}_{init}$ and covariance matrix $\mathbf{R}_b$, the prior probability density function is

$$f(\boldsymbol{\theta}) = e^{-\frac{1}{2}(\boldsymbol{\theta}-\boldsymbol{\theta}_{init})^T\mathbf{R}_b^{-1}(\boldsymbol{\theta}-\boldsymbol{\theta}_{init})} \tag{3.4}$$

and the a-posteriori probability density function is given by

$$f(\boldsymbol{\theta}\,|\{\mathbf{d}(t_i) : i = 1, \ldots N_o\}\,) \quad = \quad ce^{-J(\boldsymbol{\theta})} \tag{3.5}$$

where

$$J(\boldsymbol{\theta}) = \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}_{init})^T\mathbf{R}_b^{-1}(\boldsymbol{\theta} - \boldsymbol{\theta}_{init}) + \frac{1}{2}\sum_{i=1}^{N_o}(\mathbf{d}(t_i) - \mathbf{h}_i(\boldsymbol{\theta}))^T\,\mathbf{R}_i^{-1}\,(\mathbf{d}(t_i) - \mathbf{h}_i(\boldsymbol{\theta}))\,. \tag{3.6}$$

If the relation between the observations and model parameters is linear, then the a-posteriori probability density function is also Gaussian, with mean given as

$$\mathbf{E}(\boldsymbol{\theta}) \quad = \quad \boldsymbol{\theta}_{init} + \sum_{i=1}^{N_o}\left(\mathbf{R}_b^{-1} + \mathbf{H}_i^T\mathbf{R}_i^{-1}\mathbf{H}_i\right)^{-1}\mathbf{H}_i^T\mathbf{R}_i^{-1}\left(\mathbf{d}(t_i) - \mathbf{H}_i\boldsymbol{\theta}_{init}\right)$$

and the following covariance matrix

$$\mathbf{Cov}(\boldsymbol{\theta}) = \sum_{i=1}^{N_o} \left(\mathbf{R}_b^{-1} + \mathbf{H}_i^T \mathbf{R}_i^{-1} \mathbf{H}_i \right)^{-1},$$

where $\mathbf{H}_i$ is the Jacobian of the observation operator at time t_i (see Oliver et al. (2008)). The maximum a-posteriori estimate of the model parameters can be found by maximizing the logarithm of the posterior probability function given by Eq. 3.5 or by minimizing the objective function J. In case that measurement operator is nonlinear, the probability density function may have many modes, which correspond to different local minimum of J. In case of linear observation operator the MAP estimate is equal to the estimate of mean and is given as

$$\boldsymbol{\theta}_{MAP} = \boldsymbol{\theta}_{init} + \sum_{i=1}^{N_o} \left(\mathbf{R}_b^{-1} + \mathbf{H}_i^T \mathbf{R}_i^{-1} \mathbf{H}_i \right)^{-1} \mathbf{H}_i^T \mathbf{R}_i^{-1} \left(\mathbf{d}(t_i) - \mathbf{H}_i \boldsymbol{\theta}_{init} \right).$$

In this specific case the objective function is quadratic, which guarantees an unique solution.

3.3 Overview of history matching algorithms

History matching has been investigated for the last few decades and many algorithms have been developed; see Oliver et al. (2008) and Oliver and Chen (2011). They can be classified in four groups: gradient-based algorithms, Kalman filter algorithms, streamline-based algorithms and stochastic algorithms. They differ in the computational efficiency, implementation complexity, assumptions on the flow model, assimilation kind: recursive (sequential) or in time-windows, the capability to honor complex geological constraints and the sensitivity to local optima. Due to the high computing cost of a reservoir model simulation, a number of model simulations involved in history matching procedures is an important characteristic. Reviewing the methods we will emphasize the computational cost and implementation difficulty of the methods.

3.3.1 Gradient-based history matching algorithms

A particularly efficient class of history matching methods are gradient-based history matching algorithms, where the gradients of the objective function with respect to the model parameters are calculated by solving an adjoint, or co-state, problem as introduced by Courant and Hilbert (1953). Those methods have been studied by the meteorological community where they are called 4D-VAR (see e.g. Talagrand and Courtier (1987), Courtier et al. (1994), Courtier (1997)). In reservoir engineering the adjoint method was for the first time used by Chen et al. (1974) and later applied by, among others, Wasserman et al. (1975), Chavent et al. (1975), Watson et al. (1980), Lee and Seinfeld (1987), Yang et al. (1988), Zhang and

Reynolds (2002), and Oliver et al. (2008). In the adjoint approach the history-matching problem is treated as an optimal control problem where the control variables are the unknown model parameters, and where the objective function is minimized subject to the constraint that the state variables obey the prescribed reservoir model. Another use of adjoint-based methods in reservoir engineering is for recovery optimization or production optimization. In that case the objective function, which is ultimate recovery or net present value, is optimized with respect to control variables, which are the well rates, well pressures, or well valve settings. Initially adjoint-based method was applied to the optimization of tertiary recovery processes, see Ramirez (1987), and later followed by water flooding optimization, see e.g. Asheim (1988), Brouwer and Jansen (2004), Doublet et al. (2009), Sarma et al. (2008), Sudaryanto and Yortsos (2000) and Zandvliet et al. (2007). In both, the history matching and the recovery optimization problems, the necessary conditions for optimality lead to the gradients. These gradients can subsequently be used in gradient-based optimization routines. The adjoint approach is computationally very efficient because one gradient calculation requires just a single simulation of the reservoir model and a single simulation of the adjoint model, irrespective of the number of parameters. Usually, first order gradient-based minimization algorithms, which avoid explicit computation of the Hessian, are used to perform the minimization process. In particular with the Limited-memory Broyden-Fletcher-Goldfarb-Shanno (LBFGS) or the Levenberg-Marquardt algorithm, the history matching process can be performed efficiently (see for further details Oliver et al. (2008)). The disadvantage of this method is its difficult implementation, which was discussed in Chapter 2.

3.3.2 Streamline-based history matching algorithms

Streamline-based methods consist of a two-step inversion: first the breakthrough times at the production wells are lined up and secondly the production data are matched. Streamlines identify the instantaneous flow directions and the regions of the field that are flowing to each production well. As a consequence, it is possible to identify what parts of the reservoir impact the flow to each production well, and adjust the permeability and porosity of these regions to match production data. The computation of the sensitivities requires only a single simulation, which makes this routine very time efficient. This method has been introduced by Vasco and Datta-Gupta (1997), and then used by Wen et al. (1998), Vasco et al. (1999), Wang and Kovscek (2001), Caers (2003), Wu and Datta-Gupta (2002), Agarwal and Blunt (2003) and Cheng et al. (2004).

3.3.3 Stochastic history matching algorithms

Two popular stochastic algorithms are: the probability perturbation method and the gradual deformation method. The gradual deformation method was proposed by Roggero and Hu (1998) and then was extended by Hu et al. (2001); Hu and Jenni (2005). It is an iterative procedure where the direction of search for the best parameter is expressed as a realization sampled from the prior probability distribution function. At each iteration a new random realization is generated and estimate is expressed as a linear combination of the previous and new realization. The coefficients which provide the best match to the data are found by solving an inner 1D optimization problem. The method iterates until the data is matched. The restriction of this method is that it can only be used with a Gaussian distribution for the parameters, where the variogram model is used. Another popular stochastic method is probability perturbation method (PPM) proposed by Caers (2003). This approach uses a training image to approximate the multiple point statistics of the facies distribution. The training image should depict the desired geostatistical constraints. PPM is an iterative procedure. At each iteration a facies realization is perturbed in geologically consistent fashion, however, instead of perturbing the given realization, the probability model is perturbed. The optimal size of the perturbation is determined by solving an inner 1D optimization problem. The method forms two loop approach, which iterates until history data is matched or the maximum number of iterations is exceeded. Compared to the gradual deformation method this method can be applied to a much larger variety of geological continuity constraints. Both those methods are relatively easy to implement and they do not put any assumption on the flow model. They claim to be globally convergent due to the stochastic nature but they are computationally inefficient, because they require large number of iterations to converge, see Wu (2001).

3.3.4 Kalman filter-based history matching algorithms

The Kalman filter is a recursive method to estimate the state of a linear dynamical system from a series of noisy measurements, and it was proposed by Kalman (1960). The method is based on model equations, where the current state is associated with an uncertainty. The measurements are given by a linear operator of the state and are associated with uncertainty too. The predictions of the state are calculated by the model equations, in a so-called forecast steps, up to the moment that the new observations are available. In a so-called analysis step, an estimate of the state and its uncertainty are corrected by taking into account the available observations. The correction is weighted by the so-called Kalman gain matrix which combines the information from the uncertain data and from the uncertain state. The method assumes that the measurement noise is independent in time, there is no correlation between model noise and the measurement noise, and the model noise and the measurement noise have Gaussian distribution. Under these assumptions the conditional probability of the state given data is Gaussian and is fully characterized by the first two statistics, a mean and a covariance. The extension of the Kalman filter method to nonlinear systems was done by Jazwinski (1970), and is called the extended Kalman Filter (ExKF). In this approach the nonlinear model and measurement operators are replaced by tangent linear approximations of them and the Kalman filter analysis is applied. In both methods the covariance matrices of the state variables and parameters need to be updated at every analysis step. Since the number of states and parameters is typically very large in reservoir models these methods are computationally very expensive and consequently prohibitive for history matching. The ExKF method also requires the implementation of the tangent linear model, which is as difficult as the implementation of the adjoint model. The method which resolves the two major problems of which suffers the ExKF is the ensemble Kalman filter (EnKF). In this method the use of tangent linear models and the huge computational requirements associated with the storage and forward integration of the error covariance matrix are eliminated. Therefore, this method is computationally feasible for large systems and is relatively simple to implement. The method was originally proposed by Evensen (1994) as a stochastic or Monte Carlo alternative to the deterministic ExKF. The importance of including random observations to each ensemble member was pointed out by Burgers et al. (1998) and independently included in the formulation used by Houtekamer and Mitchell (1998). To petroleum engineering it has been introduced by Lorentzen et al. (2001), where the model parameters for a dynamic two-phase model of fluid flow in a well were calibrated. The improved predictions of pressure behavior of the well were obtained applying the full-scaled experimental data. This studies were continue in Lorentzen et al. (2003). For the first time the EnKF was used to estimate the permeability field for near well reservoir models by Nævdal et al. (2002, 2003). In those papers it was shown that the improved prediction and improvement in the quality of the estimated permeability field were obtained by assimilating additional data. In Nævdal et al. (2005) the permeability of a 2D field was estimated for a synthetic field. In Gu and Oliver (2004) it was used to estimate permeability and porosity field of the PUNQ-S3 model. The first published demonstration of the EnKF on the real field was given by Skjervheim et al. (2007), where real 4D seismic data from Northern Sea field were used. The EnKF has gained popularity because of its simple conceptual formulation and relative ease of implementation, e.g. it requires no derivation of a tangent linear operator or adjoint equations. It has been studied further by Evensen (2007), Aanonsen et al. (2009) and many more. In the EnKF an ensemble of model parameters or states is used (typically 40-100). This ensemble is supposed to reflect the uncertainty of the estimated variables and their probability distributions. Similarly to the Kalman filter, in the forecast step the elements of an ensemble are propagated in time until the first observations become available. The forecast step is based only on forward model equations. In the analysis step the state variable and model parameters are updated to honor the measurements. Each ensemble member is updated using the Kalman gain, resulting in a new ensemble. A given ensemble of finite size provides an approximation to the error covariance matrix, but instead of storing and computing the covariance matrix, one computes and stores only a square root of it. For a nonlinear problem, it may be difficult to update the state variables to be consistent with the updated model parameters without resolving the nonlinear forward problem to obtain the state variables, but doing so increases significantly the computational burden. A review of various issues with the EnKF, including covariance localization, filter divergence, sampling error because of small ensemble, and model error was given by Ehrendorfer (2007). For non-Gaussian problems, the EnKF does not produce the correct estimate of the posterior probability distribution because the analysis equations are based only on the first and second statistics (i.e. mean and covariance). The particle filter, Doucet et al. (2001), can give a better approximation of the a-posteriori distribution. In this method the forward step is computed for a large number of random samples and the posterior distribution after the analysis step is computed by combining the empirical distribution obtained from the samples with information provided by the measurements. This method is not feasible for large systems, because it is difficult to locate particles in regions of high likelihood.

3.3.5 Alternative history matching algorithms

The genetic algorithm (GA) for history matching was proposed by Romero et al. (2000). It is a randomized search techniques, which is based on an analogy to the mechanics of natural selection according to Darwinian evolutionary theory and the "survival of the fittest" principle. They draw ideas from genetics to describe solutions to the problem under consideration as individuals. GAs start with an initial population of feasible solutions (individuals), from which individual solutions are selected according to a stochastic process that favors the individuals with better performances, and their genetic information is recombined and modified following probabilistic transition rules such as the genetic operators, to form a new population. The process is repeated until a convergence is detected, or a specified maximum number of function evaluations is reached. The formulation of a GA for a specific problem requires the definition of three main issues: the design of the genome to contain the variables that define a possible solution and the generation of the phenotype (a realization of the reservoir simulation model), the selection and breeding structures used to generate solutions, and the genetic operators such as crossover and mutation used to generate new solutions. Those approaches are relatively easy to implement but computationally expensive.
The evolutionary algorithms have been demonstrated to history matching of three-phase black-oil reservoir (Schulze-Riegert et al. (2002)). The paper addresses the improvement of the convergence and parallel computations. The main step of the method is mutation of the ensemble of models, which are performed on the basis of individual step sizes. In the mutation process the model parameters are changed which increases the variability of a population. The goal of the selection process is to reduce the number of individuals on the basis of an objective function. Out of each population, two best sets of model parameters survive to define the starting point of the next generation. To improve the slow convergence, which is typical in the evolutionary algorithms, a heuristic reparameterization was applied. The parameters are divided into regions in which the parameters are assumed to be correlated. For each region a set of realization is generated from the prior geological information. The optimization algorithm chooses among the set of realizations instead of arbitrary combinations among model parameters. This approach requires relatively many simulations but is simple to implement.

3.4 Literature overview of adjoint-free gradient-based data assimilation algorithms

The adjoint-based gradient-based approach does have a few drawbacks: it can converge to a local minimum rather than the global one, it requires a significant storage of intermediate results, and it requires the implementation of an adjoint model. Nevertheless, if correctly implemented, it is one of the most efficient approaches existing today to solve the history matching problem.

In reservoir engineering, gradient-based adjoint-free methods have been proposed in recent years. We note that the naive solution of computing the gradients through finite differences, i.e. by perturbing the parameters one by one and computing the associated changes in the objective function, is not a realistic option for reservoir models which may have up to millions of parameters. A somewhat more feasible finite difference approach is one where reparameterization is used to reduce the number of parameters, e.g. with the aid of a limited number of spatial basis functions that are linear combinations of the original parameters. An example of this approach was presented by Dadashpour et al. (2009). However, also in this case the number of forward simulations required for the finite difference approach may be too large for realistic applications in reservoir engineering. In Gao et al. (2007) the simultaneous perturbation stochastic approximation (SPSA) method of Spall (1998) is used where an approximate gradient is calculated using a stochastic perturbation of all parameters together. The same method was applied by Wang et al. (2009b) for the recovery optimization problem and by Bangerth et al. (2006) for optimal placement of the oil wells. Another adjoint-free method uses an ensemble of control variables and computes the cross-covariance between the control variables and the objective function which can then be used as an approximate gradient. This ″ensemble optimization″ technique was proposed for recovery optimization by Lorentzen et al. (2006), and thereafter refined by Chen et al. (2009). For history matching a similar approximate gradient is obtained in the EnKF. Although somewhat disguised, because the EnKF assimilates data sequentially rather than all together, the basic idea of replacing an exact gradient by an ensemble-based correlation is also present in this method.

A way to overcome the problems coming from the complexity of the forward model is to replace it with a simplified model, for which the adjoint derivation is simpler. This approach has been proposed in weather prediction modeling by Courtier et al. (1994) as a solution to the computationally expensive large-scale inverse problem. They derived an incremental approach, in which the forward solution of a high-resolution nonlinear model is replaced by the solution of a lower resolution, approximate linear model. The disadvantage of this method is the need to develop a new, linear model, which is an approximation of the original model (with simplified physical parameterizations).

A different way to obtain an approximated model is by the use of reduced order modeling (ROM) techniques (in Lawless et al. (2008) a combination of reduced order modeling with incremental approach is presented). The application of ROM reduces the CPU time of the model simulation but does not change the complexity of the problem, and consequently, does not solve the implementation problem of the adjoint model. Vermeulen and Heemink (2006) proposed an approach based on model reduction, that avoids an implementation of the adjoint model of the tangent linear approximation of the original nonlinear model. First, they construct the reduced-order tangent linear approximation of the original forward model, and thereafter, thanks to the linear character of the reduced-order model, they easily derive the adjoint model. In this approach, the Proper Orthogonal Decomposition (POD) method (introduced by Karhunen (1946) and Loéve (1946) as a statistical tool to analyze random process data) is used to obtain an approximate low-dimensional version of the tangent-linear model. In the POD method the projection subspace is determined by processing data obtained from numerical simulations of the high-dimensional model, which are expected to provide relevant information about the dynamic behavior of the system. The high-dimensional equations are projected on the low-dimensional subspace resulting in a low-dimension model. Since the high-order tangent linear model is not available, Vermeulen and Heemink (2006) uses the POD method in a somewhat different manner than the conventional application of the POD method for state-space reduction. Afterwards, due to the reduced model size the minimization problem is solved efficiently. Vermeulen and Heemink (2006) applied this method to groundwater flow problems where it proved to be computationally very efficient. Later it was applied by Altaf et al. (2009) for water depth estimation in a coastal engineering problem and also there it proved to be very efficient. In this thesis we apply this method to the history matching problem. This new algorithm is not better in terms of efficiency or robustness than the classical adjoint-based approach, but it does not require the implementation of the adjoint code, it operates in a low-dimensional space, which makes it much more efficient than the naive finite difference method, and it can easily be implemented with any simulator without having access to simulation source code.

3.5 Parameterization

Since the vector $\boldsymbol{\theta}$ may consists of properties in each grid block of the model, e.g. permeabilities or porosities, the number of uncertain parameters can be very large. In real reservoir models it ranges from 10^4 to 10^6. In case of a gradient-based history matching algorithm, one needs to calculate the gradient of the objective function with respect to the parameters. Using the adjoint-based method it can be calculated by one solution of the reservoir model and one solution of the corresponding adjoint model independently of the number of model parameters. Moreover, to efficiently update the Hessian matrix during the quasi-Newton optimization the Limited Memory BFGS method can be used. Hence, gradient-based methods can efficiently history match data even for large scale reservoir models. However, in case of matching only the production data, the available data are very sparse. This implies that the identifiability of the parameter space is very low (see Zandvliet et al. (2008)) and the history matching is ill-posed problem. It means that this problem does not have an unique solution (see Gavalas et al. (1976)). The resulting history matched field might honor observed data but it might provide incorrect predictions or it might represent an unrealistic estimate. This gives the motivation for using a smaller number of parameters, i.e. to make the problem better-posed. The objective of parameterization is to express the model parameters by a lower number of new variables and to preserve the important geological features in the new parametrization. During the recent decade, many methods to reduce the number of parameters have been developed, and some of them are briefly discussed here.

3.5.1 Zonation

Zonation was introduced by Jacquard and Jain (1965) and it is one of the earliest and the most straightforward method for a low dimensional parameterization. In this approach the reservoir is divided into a small number of subregions in which the parameters are assumed to be homogeneous. Then the model parameters are adjusted for each subregion by history matching dynamic data. The gradzone procedure (Bissell (1994)) or adaptive multiscale methods (Grimstad and Mannseth (2000)) represent modern extensions of this method.

3.5.2 Pilot point method

The pilot point method was proposed by de Marsily et al. (1984). In this approach one chooses only a limited number of parameters, at which the permeability values are adjusted during the history matching procedure and then those updated values are interpolated to the neighboring points by kriging. RamaRao et al. (1995) proposed to place the pilot points at locations where the data is the most sensitive, hence their potential for decreasing the objective function is the highest. This approach is not fully automatic because it is not very clear how many points one should use.

3.5.3 Discrete cosine transform

The discrete cosine transform (DCT) for parameterization of the permeability field has been proposed by Jafarpour and McLaughlin (2009). This method originates from image processing and is widely used as an image-compression standard. In this approach the permeability field is expressed as a linear combination of predefined basis functions weighted by uncorrelated random coefficients. The basis functions are defined as cosine functions. The total number of basis functions is equal to the number of permeability values. For a known permeability field, only the basis functions with the highest coefficients in the expansion are retained, which reduces the number of parameters. In history matching the parameter field is unknown; therefore it is not possible to identify the most relevant basis functions by ordering the coefficients, but prior information in the form of a training set of permeability fields can be used to identify the most dominant basis function. The absolute value of the coefficients of the permeability fields from a training set is taken and the basis functions with the highest average values of coefficients are selected to create the DCT parameterization. If no prior fields are available but there is an information about the orientation of the channels, this can be used to make a decision which cosine functions should be retained in the DCT parameterization.

3.5.4 Wavelets

A discrete wavelet transform proposed by Sahni and Horne (2005) uses Haar wavelets to parametrize the permeability field. The permeability field is expressed as linear combination of Haar wavelets weighted by uncorrelated random coefficients. First the wavelet coefficients that are the most sensitive to the history match data are fixed and the history matching procedure is performed. In the second step, the remaining free coefficients are modified in order to integrate the variogram information into the reservoir description. This process is done by minimizing the objective function which is defined as a norm between the reference variogram and the variogram of the history matched field. Since the wavelet coefficients constraining the history match are not changed, the history match is retained after the second step.

3.5.5 Karhunen-Loéve expansion

The Karhunen-Loéve expansion (Karhunen (1946), Loéve (1946)) is a powerful parameterization approach very well suited for history matching. It has been applied for permeability and porosity parameterization by Gavalas et al. (1976), Reynolds et al. (1996). In this approach the permeability at each gridblock is expressed as a linear combination of deterministic basis functions weighted by uncorrelated random coefficients. The basis functions are defined by the eigenvectors of a specified permeability covariance matrix and are obtained by solving an eigenvalue problem. The covariance matrix is often approximated as the sample covariance matrix, which significantly reduces the size of an eigenvalue problem. In such a case the parameterization is data-dependent and the reliability of it depends on the representativeness of a set of samples. Usually only the eigenvectors corresponding to the largest eigenvalues are used in the Karhunen-Loéve expansion. The neglected part corresponds to the shortest correlation lengths present in the sample.

3.5.6 Kernel Principal Component Analysis

Sarma et al. (2007) has proposed the kernel PCA method as an extension to the Karhunen-Loéve expansion (also know as the PCA). Unlike the standard PCA which preserves only second moment statistic, this approach preserves higher order statistics, and hence it can be used to parameterize more complex geological models than Gaussian distributed fields. In the posterior probability distribution we want to preserve the geological features of the field that are known from the prior information. The realizations of the geological model are transformed to the feature space which is defined as a nonlinear mapping of the original space. In the feature space the dot product between elements is defined by a kernel function. Polynomial kernels of higher order are applied. The PCA is performed in the feature space, which results in a parameter expansion in the feature space too. To come back to the original space the pre-image problem is solved. The obtained parameterization is a continuously differentiable implicit nonlinear parameterization. Due to its implicit character it requires some additional methods to compute the coefficients of the expansion.

3.5.7 Other parameterizations

Shah et al. (1978) has proposed the parameterization method that is based on information coming from the model outputs. They perform the eigenvalue decomposition of $\left(\frac{\partial \mathbf{h}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right)^T \frac{\partial \mathbf{h}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$, where $\frac{\partial \mathbf{h}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$ is the sensitivity matrix of the model outputs with respect to model parameters. In this approach the permeability at each grid block is expressed as a linear combination of the largest eigenvectors resulting from this decomposition.

Similar approach was presented in Van Doren et al. (2008), where the identifiable parameterizations based on a singular value decomposition of the information matrix was proposed. The information matrix is expressed in terms of controllability and observability properties of the system and the sensitivities of the system matrices with respect to the parameter vector. The columns of the left-side singular vectors corresponding to non-zero Hankel singular values are regarded as the directions in the reduced parameter space. This reparameterization was motivated by the fact that only parameters changes that result in state perturbations of the observable part of the system can be identified (see Van Doren (2010) for more details).

4 CHAPTER

Reduced Order Modeling

Re duced Order Modeling (ROM) aims at capturing dynamics of high-order dynamical system by means of a low-order approximation. This chapter presents two model reduction techniques: Proper Orthogonal Decomposition (POD) and Balanced Proper Orthogonal Decomposition (BPOD). Both methods are based on the projection of high-order system equations onto a subspace derived from the data. The chapter starts with a motivation for the choice of ROM methods followed by their description, and concludes with remarks about the connection between the POD and the BPOD methods.

4.1 Motivation

Oil & gas reservoirs have been produced under increasingly more difficult conditions. As consequence, there is a compelling need for the accurate simulation of the dynamical behavior of reservoir systems. This lead to large and complex models. *Reduced order modeling* (ROM) techniques, also called *model reduction techniques* or *model order reduction techniques*, play an indispensable role in providing efficient computational prototyping tools to replace large-scale models by approximate smaller models, which are capable of capturing critical dynamical behavior and faithfully preserving essential properties of the larger models. A good model reduction technique tries to keep the necessary information and throws away the irrelevant. The speed-up of simulation is the main reason for using model reduction techniques. Consequently, if scientists and engineers can significantly reduce the simulation time, that is, the experiments run in hours instead of days, then they can re-evaluate the methods which were considered computationally too expensive or improve the results by allowing for more simulations. An accurate and effective reduced-order model can be applied then for production optimization, history matching or prediction uncertainty quantification. In this thesis we use the reservoir model for the production optimization and parameter estimation purposes. In those cases we are not interested in a very accurate approximation of all state variables, but rather in a good approximation of the production data. Therefore, we should be able to compress the high-dimensional model significantly. The conventional application of the ROM for state-space reduction reduces the CPU time of the model simulation but does not change the complexity of the model, and consequently, does not solve the implementation problem of the adjoint model. In this application the model reduction is not used in the standard way, that is, it is mainly a technique to obtain an approximate reduced-order version of the tangent-linear reservoir model and its adjoint. The Jacobians of these models are approximated in the reduced-order space by finite differences and therefore the reduced-order model is constructed without projection of the original model equations. The detailed explanation of this method is described in Chapter 5 while this chapter introduces the ROM techniques in their standard form.

4.2 Projection-based Reduced Order Modeling methods

The balanced POD method and the POD method fall in the category of projection-based methods. The basic idea of projection-based reduced-order modeling is to project the high dimensional state space X onto a low dimensional subspace that contains the essential dynamics of the system. A *projection* is a linear transformation and can be represented by projection matrix $\mathbf{\Pi} \in \mathbb{R}^n \times \mathbb{R}^n$, which satisfies the condition $\mathbf{\Pi} = \mathbf{\Pi}\mathbf{\Pi}^T$. A projection $\mathbf{\Pi}$ is called an *orthogonal projection* if it is self-adjoint, i.e. $\mathbf{\Pi} = \mathbf{\Pi}^T$. A *Galerkin projection* $\mathbf{\Pi}$ is an orthogonal projection defined as $\mathbf{\Pi} = \mathbf{\Phi}\mathbf{\Psi}^T$, where $\mathbf{\Phi}$, $\mathbf{\Psi} \in \mathbb{R}^{n \times l}$, $\mathbf{\Psi}^T\mathbf{\Phi} = \mathbf{I}_l$, $\mathbf{\Psi} = \mathbf{\Phi}$, and the columns of $\mathbf{\Phi}$ form an orthogonal set. In case that $\mathbf{\Phi} \neq \mathbf{\Psi}$, the projection $\mathbf{\Pi}$ is called a *Petrov-Galerkin projection*. The various projection-based techniques differ in the choice of the matrices $\mathbf{\Phi}$ and $\mathbf{\Psi}$. Roughly speaking, three main strategies for choosing $\mathbf{\Phi}$ and $\mathbf{\Psi}$ can be identified: heuristic schemes (include choosing $\mathbf{\Phi}$ and $\mathbf{\Psi}$ based on eigenvector analysis), methods based on rigorous system-theoretic analysis (balanced truncation), and parameter-matching techniques (Krylov methods), see Antoulas (2005) for more details.

The projection-based methods consist of the following steps:

- to determine the l-dimensional subspaces S_Φ and S_Ψ, and their basis $\mathbf{\Phi}$ and $\mathbf{\Psi}$,
- to compress the state information by projecting the state variables onto a subspace spanned by $\mathbf{\Phi}$,
- to compress the system equations by multiplying them by $\mathbf{\Psi}^T$.

To describe those steps let us consider the Hilbert space $X \subset \mathbb{R}^{n \times n}$, where n is the number of state variables needed to describe the system. Let the dynamic of the system be modeled by the following equations

$$\mathbf{x}(t_i) = \mathbf{f}_i[\mathbf{x}(t_{i-1}), \boldsymbol{\theta}], \tag{4.1}$$

$$\mathbf{y}(t_i) = \mathbf{h}_i[\mathbf{x}(t_i), \boldsymbol{\theta}], \tag{4.2}$$

$$\mathbf{x}(t_0) = \mathbf{x}_{init},$$

where $i = \{1, ..., N\}$ indicates the time steps, $\mathbf{x}(t_i)$ is a solution of a functional $\mathbf{f}_i$ at time t_i, $\mathbf{x}(t_0) = \mathbf{x}_{init}$ is a given initial state of the system and $\mathbf{y}(t_i)$ are the outputs of the system at time t_i given by the observation operator $\mathbf{h}_i$. We assume that for all time instants $\mathbf{x}(t_i)$ belongs to the Hilbert space X. For given matrices $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$ the system described by equations (4.1) and (4.2) is approximated by the reduced-order system in the following form

$$\begin{aligned} \mathbf{z}(t_i) &= \boldsymbol{\Psi}^T \mathbf{f}_i(\boldsymbol{\Phi}\mathbf{z}(t_{i-1}), \boldsymbol{\theta}), \\ \mathbf{y}(t_i) &= \mathbf{h}_i(\boldsymbol{\Phi}\mathbf{z}(t_i), \boldsymbol{\theta}), \\ \mathbf{z}(t_0) &= \boldsymbol{\Psi}^T \mathbf{x}(t_0), \end{aligned} \tag{4.3}$$

where $\mathbf{z}(t_i)$ is the reduced-order state variable which operates in the subspace $S_\Phi \subset X$ of lower dimension l spanned by $\boldsymbol{\Phi}$, and $\mathbf{z}(t_0)$ is the initial state of the reduced-order system. The reduced-order system (4.3) describes the dynamic evolution of the l dimensional vector $\mathbf{z}$. Of course the arbitrary choice for matrices $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$ would bring us to a model that is unable to reproduce the dynamics of the original model. The choice of those matrices is the main obstacle of the ROM methods.

4.3 Proper Orthogonal Decomposition

The *Proper Orthogonal Decomposition* (POD) method is also known as the *Karhunen-Loève* method, *principal component analysis* (PCA) or the *method of empirical orthogonal functions*. The origin of this method goes back to 1946, when it was introduced independently by Karhunen (1946) and Loéve (1946) as a statistical tool to analyze random process data. According to Lumley it was introduced independently by even more people including Kosambi (1943), Pougachev (1953) and Obukbov (1954). The method was called for the first time POD by Lumley (1967), when it was used for the study of turbulent flow. It has been used in various disciplines including random variables (Papoulis (1965)), image processing (Rosenfeld and Kak (1982)), signal analysis (Algazi and Sakrison (1969)), data compression (Andrews et al. (1967)), and it has been successfully applied to increase the computational efficiency of reservoir simulation by Heijn et al. (2004), Markovinović et al. (2002), Markovinović and Jansen (2006), and Cardoso et al. (2009). The use of the POD method in combination with adjoint-based production optimization in reservoir simulation has been investigated by Van Doren et al. (2004) and by Markovinović (2009). The POD method has desirable features for our application: it allows nonlinearities in the system, it is suitable for model reduction of spatio-temporal systems and it is feasible for large-scale systems.

4.3.1 Snapshots

In the POD method the projection subspace S_Φ is determined by processing data obtained from numerical simulations of the high-dimensional model (4.1). This data consists of selected state vectors of a forward simulation of the large-scale numerical model, called *snapshots*. The selected data should provide relevant information about the dynamical behavior of the system, and should be able to represent the behavior of the system for different spatial and time instants. Typically, the snapshots are created in the following way:

$$\mathbf{x}_j(t_i) = \mathbf{f}_i[\mathbf{x}(t_{i-1}), \boldsymbol{\theta}], \tag{4.4}$$

where $t_i = t_1, \ldots, t_N$ and $\boldsymbol{\theta}$ is a parameter vector and $\mathbf{x}(t_{i-1})$ is a state vector determined by $\boldsymbol{\theta}$. When the reduced-order model is used for parameter estimation, the snapshots should be able to represent the behavior of the system also for modified parameter values, which results in the following snapshots

$$\mathbf{x}_j(t_i) = \mathbf{f}_i[\mathbf{x}^{pert}(t_{i-1}), \boldsymbol{\theta} + \boldsymbol{\theta}_{pert}], \tag{4.5}$$

where $\boldsymbol{\theta}_{pert}$ is a perturbation to the initial parameter values and $\mathbf{x}^{pert}(t_{i-1})$ is a state vector determined by the perturbed parameters. These snapshots are the sampled solutions of numerical model at particular time steps and at particular parameter values. In case of using the reduced-order model for production optimization the control variables have to be perturbed while generating snapshots. All snapshots are put as columns in matrix

$$\mathbf{X} = \{\mathbf{x}_1(t_1), ..., \mathbf{x}_1(t_N), ..., \mathbf{x}_p(t_1), ..., \mathbf{x}_p(t_N)\} \tag{4.6}$$

of size $n \times s$, where p is the number of parameters perturbations and $s = N \times p$ is the total number of snapshots. In general it is assumed that $s \ll n$.

4.3.2 Projection matrices

One of the main issues of the POD method is to express the essential information present in the collected data by means of a few basis vectors $\mathbf{\Phi}$. POD gives an optimal set of basis vectors, i.e. it guarantees the best reconstruction of the data set $\mathbf{X}$ in the mean square error sense among all linear bases of the same size. Here we give a short explanation how the POD basis are constructed and how to calculate them efficiently. As it was said, the aim of the POD is to find an orthogonal $n \times l$ basis $\mathbf{\Phi}$ such that the total mean squared error between the snapshots and their projections on the subspace spanned by basis $\mathbf{\Phi}$ is minimized for any l, i.e. the given value

$$\frac{1}{s}\sum_{i=1}^{s}\left\|\mathbf{x}_i - \mathbf{\Phi}\mathbf{\Phi}^T\mathbf{x}_i\right\|_2^2, \tag{4.7}$$

is minimal, where $\|\cdot\|_2$ is an L_2 norm and $\mathbf{x}_i \in \mathbf{X},\ i = 1, \ldots s$, where $1 \leq s \ll n$ is assumed. This problem can be expressed as

$$\min_{\Phi:\boldsymbol{\phi}_i^T\boldsymbol{\phi}_j=0,\forall i\neq j\|\boldsymbol{\phi}_i\|^2=1} \frac{1}{s}\sum_{i=1}^{s}\left\|\mathbf{x}_i - \mathbf{\Phi}\mathbf{\Phi}^T\mathbf{x}_i\right\|_2^2. \tag{4.8}$$

Due to the identity

$$\left\|\mathbf{x}_i - \mathbf{\Phi}\mathbf{\Phi}^T\mathbf{x}_i\right\|_2^2 = \|\mathbf{x}_i\|_2^2 - \left\|\mathbf{\Phi}\mathbf{\Phi}^T\mathbf{x}_i\right\|_2^2 \tag{4.9}$$

which holds for any orthogonal projection, the minimization problem (4.8) is equivalent to the following maximization problem

$$\max_{\Phi:\boldsymbol{\phi}_i^T\boldsymbol{\phi}_j=0,\forall i\neq j\|\boldsymbol{\phi}_i\|_2^2=1} \sum_{i=1}^{s} \left\|\mathbf{\Phi}\mathbf{\Phi}^T\mathbf{x}_i\right\|_2^2 \tag{4.10}$$

where each element $\left\|\mathbf{\Phi}\mathbf{\Phi}^T\mathbf{x}_i\right\|_2$ of this sum can be rewritten as

$$\begin{aligned}
\left\|\mathbf{\Phi}\mathbf{\Phi}^T\mathbf{x}_i\right\|_2 &= \sqrt{\left(\mathbf{\Phi}\mathbf{\Phi}^T\mathbf{x}_i\right)^T\left(\mathbf{\Phi}\mathbf{\Phi}^T\mathbf{x}_i\right)} \\
&= \sqrt{\mathbf{x}_i^T\mathbf{\Phi}\mathbf{\Phi}^T\mathbf{\Phi}\mathbf{\Phi}^T\mathbf{x}_i} \\
&= \sqrt{\mathbf{x}_i^T\mathbf{\Phi}\mathbf{\Phi}^T\mathbf{x}_i} \\
&= \left\|\mathbf{\Phi}^T\mathbf{x}_i\right\|_2.
\end{aligned}$$

Therefore, problem (4.8) is equivalent to the following maximization problem

$$\max_{\Phi:\boldsymbol{\phi}_i^T\boldsymbol{\phi}_j=0,\forall i\neq j\|\boldsymbol{\phi}_i\|_2^2=1} \sum_{i=1}^{s} \left\|\mathbf{\Phi}^T\mathbf{x}_i\right\|_2^2 \tag{4.11}$$

which is a constrained optimization problem that can be solved by considering the first-order necessary optimality conditions. First, we solve this problem in case that $l = 1$, i.e. $\mathbf{\Phi}$ consists of a single vector $\boldsymbol{\phi}$. Let L be the Lagrangian functional associated with (4.11)

$$L(\boldsymbol{\phi},\lambda) = \sum_{i=1}^{s} \left\|\boldsymbol{\phi}^T\mathbf{x}_i\right\|_2^2 - \lambda(1 - \|\boldsymbol{\phi}\|_2^2). \tag{4.12}$$

A first-order necessary optimality condition is given by

$$\nabla L(\boldsymbol{\phi},\lambda) = 0, \tag{4.13}$$

and the gradient of L with respect to $\boldsymbol{\phi}$ is

$$\begin{aligned}
\frac{\partial L}{\partial\boldsymbol{\phi}}(\boldsymbol{\phi},\lambda) = \frac{\partial}{\partial\boldsymbol{\phi}}\left(\sum_{i=1}^{s}(\boldsymbol{\phi}^T\mathbf{x}_i)^T(\boldsymbol{\phi}^T\mathbf{x}_i) + \lambda(1-\boldsymbol{\phi}^T\boldsymbol{\phi})\right) &= \\
\frac{\partial}{\partial\boldsymbol{\phi}}\left(\sum_{i=1}^{s}(\mathbf{x}_i^T\boldsymbol{\phi}\boldsymbol{\phi}^T\mathbf{x}_i) + \lambda(1-\boldsymbol{\phi}^T\boldsymbol{\phi})\right) &= \\
2\sum_{i=1}^{s}\boldsymbol{\phi}^T\mathbf{x}_i\mathbf{x}_i^T - 2\lambda\boldsymbol{\phi}^T &= \\
2\left(\boldsymbol{\phi}^T\mathbf{X}\mathbf{X}^T - \lambda\boldsymbol{\phi}^T\right)&.
\end{aligned} \tag{4.14}$$

Thus,

$$\mathbf{X}\mathbf{X}^T\boldsymbol{\phi} - \lambda\boldsymbol{\phi} = \mathbf{0}. \tag{4.15}$$

The above equation yields the eigenvalue problem

$$\mathbf{X}\mathbf{X}^T\boldsymbol{\phi} = \lambda\boldsymbol{\phi}. \tag{4.16}$$

It means that $\boldsymbol{\phi}$ is the eigenvector corresponding to the largest eigenvalue of problem (4.16), and the maximum reached value is equal to λ. Using (finite) mathematical induction it can be proven (see Volkwein (2008)) that for any $l \in \{1, ..., s\}$ the solution to problem (4.8) is given by the first l largest eigenvectors of matrix $\mathbf{X}\mathbf{X}^T$. Summarizing, the optimization problem (4.8) is solved by $\boldsymbol{\Phi}$ consisting of l eigenvectors corresponding to l first largest eigenvalues of the correlation matrix

$$\mathbf{R} := \frac{1}{s}\mathbf{X}\mathbf{X}^T. \tag{4.17}$$

Alternatively, instead of correlation matrix $\mathbf{R}$, the covariance matrix

$$\tilde{\mathbf{R}} := \frac{1}{s-1}\sum_{i=1}^{s}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T \tag{4.18}$$

can be used, where $\bar{\mathbf{x}}$ represents a mean of the snapshots, i.e. $\bar{\mathbf{x}} = \frac{1}{s}\sum_{i=1}^{s}\mathbf{x}_i$. The decision about l is usually based on the *cumulative relative importance* defined as

$$\sum_{i=1}^{l}\frac{\lambda_i}{\sum_{j=1}^{s}\lambda_j} > \alpha^e \tag{4.19}$$

where l is chosen as the smallest number, such that

$$\sum_{j=1}^{l}\alpha_j \geq \alpha^e. \tag{4.20}$$

By choosing l patterns with the largest relative importance, defined as $\alpha_i = \frac{\lambda_i}{\sum_{j=1}^{s}\lambda_j}$, an orthonormal projection matrix $\boldsymbol{\Phi}$ is created by taking vectors $\boldsymbol{\phi}_i$ as its columns, for $i \in \{1, ..., l\}$. The accuracy of a reduced-order model increases with an increase in the number of patterns in the matrix $\boldsymbol{\Phi}$. For a given number of snapshots, the highest accuracy is achieved by taking the expected variance equal to 100%. On the other hand, more patterns means lower computational efficiency. Therefore, there is a trade-off between efficiency and accuracy.

Numerical issues
$\mathbf{X}\mathbf{X}^T \in \mathbb{R}^{n\times n}$ is a symmetric matrix satisfying

$$\phi^T(\mathbf{X}\mathbf{X}^T)\boldsymbol{\phi} = (\mathbf{X}^T\boldsymbol{\phi})^T(\mathbf{X}^T\boldsymbol{\phi}) \geq 0\;\forall\boldsymbol{\phi} \in \mathbb{R}^n$$

which means that $\mathbf{X}\mathbf{X}^T$ is positive semi-definite, and therefore has non-negative eigenvalues. Since $\mathbf{R}$ is approximated by s snapshots we know that it has at most s eigenvalues $\lambda_1 \geq \ldots \geq \lambda_s \geq 0$ and corresponding s eigenvectors. For large system solving eigenvalue problem given by Eq. 4.16 is computationally demanding. This eigenvalue problem is formulated as a spatial eigenvalue problem, however, an equivalent formulation can be derived in which the eigenvalue problem is formulated in the snapshots coordinate. It is a so-called method of snapshots (Sirovich (1987)) and has a numerical advantage, because typically $s \ll n$. If we multiply both sides of Eq. 4.16 with $c_i \mathbf{X}^T$ ($c_i \neq 0$) it gives

$$\mathbf{X}^T \mathbf{X} \mathbf{v}_i = \lambda_i \mathbf{v}_i, \tag{4.21}$$

where $\mathbf{v}_i = c_i \mathbf{X}^T \boldsymbol{\phi}_i$. Thus, $\mathbf{X}^T\mathbf{X}$ and $\mathbf{X}\mathbf{X}^T$ have the same s (or $s-1$) eigenvalues and their eigenvectors satisfy $\mathbf{v}_i = c_i \mathbf{X}^T \boldsymbol{\phi}_i$. Because the problem (4.21) works on much smaller matrix $\mathbf{X}^T\mathbf{X} \in \mathbb{R}^{s\times s}$ compared to problem (4.16), it is much more efficient to solve. The condition that the eigenvectors are normalized, i.e.

$$1 \equiv \|\mathbf{v}_i\|_2 = \mathbf{v}_i^T \mathbf{v}_i = \left(c_i \mathbf{X}^T \boldsymbol{\phi}_i\right)^T c_i \mathbf{X}^T \boldsymbol{\phi}_i = c_i^2 \boldsymbol{\phi}_i^T \mathbf{X}\mathbf{X}^T \boldsymbol{\phi}_i = c_i^2 \boldsymbol{\phi}_i^T \lambda_i \boldsymbol{\phi}_i = c_i^2 \lambda_i \tag{4.22}$$

results in the following relation

$$c_i = \lambda_i^{-\frac{1}{2}} \tag{4.23}$$

and thus

$$\boldsymbol{\phi}_i = \lambda_i^{-\frac{1}{2}} \mathbf{X} \mathbf{v}_i. \tag{4.24}$$

From relation (4.24) we see also that the POD patterns are linear combination of the collected snapshots, which was observed by Sirovich (1987).

Another alternative way to calculate the POD patters is to perform a singular value decomposition (SVD) on the snapshot matrix $\mathbf{X}$. It avoids the calculation of the product $\mathbf{X}^T\mathbf{X}$ (or $\mathbf{X}\mathbf{X}^T$) which are dense matrices in our application and the matrix $\mathbf{X}$ is decomposed as

$$\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^T \tag{4.25}$$

where $\mathbf{U} \in \mathbb{R}^{n\times n}$, $\mathbf{V} \in \mathbb{R}^{s\times s}$ are orthogonal ($\mathbf{U}^T\mathbf{U} = \mathbf{I}_n$, $\mathbf{V}^T\mathbf{V} = \mathbf{I}_s$) and $\Sigma \in \mathbb{R}^{n\times s}$ is an pseudo diagonal matrix with elements ordered in the decreasing manner, i.e. $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_s \geq 0$. If we consider again the correlation matrix $\mathbf{X}\mathbf{X}^T$ then we see that

$$\mathbf{X}\mathbf{X}^T = \mathbf{U}\Sigma\mathbf{V}^T(\mathbf{U}\Sigma\mathbf{V}^T)^T = \mathbf{U}\Sigma\mathbf{V}^T\mathbf{V}\Sigma^T\mathbf{U}^T = \mathbf{U}\Sigma\Sigma^T\mathbf{U}^T. \tag{4.26}$$

It follows that the POD patterns $\boldsymbol{\Phi}$ can be computed as left singular vectors $\mathbf{U}$, whereas $\lambda_i = \sigma_i^2$. Alternatively we can also express them using right singular vectors, $\mathbf{V}$, because for $\mathbf{X}^T\mathbf{X}$ holds

$$\mathbf{X}^T\mathbf{X} = (\mathbf{U}\Sigma\mathbf{V}^T)^T\mathbf{U}\Sigma\mathbf{V}^T = \mathbf{V}\Sigma^T\mathbf{U}^T\mathbf{U}\Sigma\mathbf{V}^T = \mathbf{V}\Sigma^T\Sigma\mathbf{V}^T. \tag{4.27}$$

4.3.3 Reduced order model

The projection $\boldsymbol{\Pi} = \boldsymbol{\Phi}\boldsymbol{\Psi}^T$ is defined as a Galerkin projection with $\boldsymbol{\Psi} = \boldsymbol{\Phi}$, where $\boldsymbol{\Phi}$ consists of l eigenvectors $\boldsymbol{\phi}_i$ corresponding to l leading eigenvalues. The state of the system is then expressed in the new space as linear combination of the POD patterns, i.e. $\mathbf{x}(t_i) = \mathbf{\Phi}\mathbf{z}(t_i)$ or $\mathbf{x}(t_i) = \bar{\mathbf{x}} + \mathbf{\Phi}\mathbf{z}(t_i)$ if the mean value was subtracted from snapshots, and the reduced-order system is described by

$$\begin{aligned} \mathbf{z}(t_i) &= \mathbf{\Phi}^T \mathbf{f}_i(\mathbf{\Phi}\mathbf{z}(t_{i-1}), \boldsymbol{\theta}) \\ \mathbf{y}(t_i) &= \mathbf{h}_i(\mathbf{\Phi}\mathbf{z}(t_i), \boldsymbol{\theta}), \\ \mathbf{z}(t_0) &= \mathbf{\Phi}^T \mathbf{x}(t_0), \end{aligned} \tag{4.28}$$

This reduced-order system preserves the states which characteristics were present in the snapshots.

4.3.4 Comments on the POD method

POD method has a few drawbacks.

- Optimality of a particular basis does not imply optimality of the reduced-order dynamic model. Since it is a data-driven method, the reduced order model is highly dependent on the given data. Therefore, it is essential to choose snapshots which represent the behavior of model in the range of the parameters or control variables that we are interested in. However, there is no guarantee that the reduced-order model will give the dynamic behavior corresponding to the original model.
- Using the POD method we can not calculate error bounds because it is system representation dependent. Hence, we can not estimate the uncertainty of the solution coming from the reduced-order model.
- Furthermore, the original sparse structure of the partial differential equations is lost during the projection of the equations on the reduced-order space. Hence, the resulting reduced-order model has a dense structure. It means that in some applications the advantage of having a reduced-order model does not lead to remarkable gain in simulation time.
- Finally, the POD method does not respect the underlying model equations, it means, that the underlying physics is not preserved exactly in the reduced-order model.
- In case of nonlinear systems the system matrices, $\mathbf{f}_i(\mathbf{\Phi}\mathbf{z}(t_{i-1}), \boldsymbol{\theta})$, need to be recomputed at every time step of the simulation and projected to lower space. This takes a lot of time, and therefore POD is much less efficient than for the linear cases, where system matrices have to be projected only once.

4.4 Balanced Proper Orthogonal Decomposition

Model reduction aims to eliminate the part of the state variables which modeling is not relevant for the chosen application. It is important to understand which properties of the system we want to preserve and which we can neglect. In this section we present another projection-based technique which is an approximation to a well known *balanced truncation* method. First, the description of the balanced truncation method is given, and then the Balanced Proper Orthogonal Decomposition is presented. Balanced truncation is based on a system-theoretic analysis, i.e. the input-output response of the system. We present a description of the balanced truncation method for linear time invariant systems, for which the theory of balancing was derived. The use of balancing for model reduction was proposed in 1981 by Moore (1981). Nowadays, it is a well-developed method that appears in standard textbooks (see Antoulas (2005)) and it has been very successfully used in control design. In reservoir applications this method has been investigated by Vakili-Ghahani (2010), Markovinović (2009), and Van Doren (2010).

4.4.1 Controllability and observability of Linear Time Invariant systems

The controllability and observability of a dynamic system reflect to what extent the inputs influence the states, and to what extent the states influence the outputs.

Controllability

Let us consider an n-dimensional, stable and minimal, time continuous linear time invariant system:

$$\begin{aligned} \frac{d\mathbf{x}}{dt} &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t),\ \mathbf{x}(0) = \mathbf{x}_0, \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \end{aligned} \tag{4.29}$$

with p inputs and m outputs.

Definition 4.1 *The dynamic system (4.29) is called controllable if for any $x_0 \in \mathbb{R}^n$ and the final state x_T there exists a finite time T and a (piecewise continuous minimum amount of finite control) input u such that the solution to (4.29) satisfies $x(t_N) = x_T$. Otherwise the system is called uncontrollable.*

To answer a question if the system is controllable one can check a rank of the *controllability matrix C_n* which is defined as

$$C_n := \left[\mathbf{B}\ \mathbf{AB}\ \mathbf{A}^2\mathbf{B}\ldots\mathbf{A}^{n-1}\mathbf{B}\right] \in \mathbb{R}^{n\times np} \tag{4.30}$$

If C_n has a full rank, then system (4.29) is controllable and state $\mathbf{x}$ can be reached by a proper choice of $\mathbf{u}$. Alternatively, we can verify the controllability of a linear dynamic system with the aid of the controllability Gramian, see e.g. Antoulas (2005). The controllability Gramian for finite time is defined as

$$P_n := \sum_{i=0}^{n-1} (\mathbf{A}^i\mathbf{B})(\mathbf{A}^i\mathbf{B})^T = C_n C_n^T, \tag{4.31}$$

while for infinite time it can be computed by solving a Lyapunov equation given by

$$\mathbf{A}P\mathbf{A}^T + \mathbf{B}\mathbf{B}^T = P. \tag{4.32}$$

The system (4.29) is controllable if a controllability Gramian is positive definite. Hence, the following conditions are equivalent: the given system is controllable or the controllability Gramian is positive definite or the controllability matrix C_n has a full row rank. The significance of the Gramian comes from the fact presented in Glover (1984), which says that in a controllable system the minimal energy required to steer a state $\mathbf{x}_0$ to a state $\mathbf{x}_t$ which are n time steps apart is

$$(\mathbf{x}_t - \mathbf{x}_0)^T P_n^{-1} (\mathbf{x}_t - \mathbf{x}_0) > 0. \tag{4.33}$$

Thus there exist a sequence of inputs $\mathbf{u}_i$ such that

$$\sum_{i=1}^{n} \mathbf{u}_i^T \mathbf{u}_i = (\mathbf{x}_t - \mathbf{x}_0)^T P_n^{-1} (\mathbf{x}_t - \mathbf{x}_0) > 0 \tag{4.34}$$

and it is the minimal energy[1]that is required. From $(\mathbf{x}_t - \mathbf{x}_0)^T P_n^{-1} (\mathbf{x}_t - \mathbf{x}_0)$ it follows that P_n^{-1} gives us a measure to say which state requires the least energy to be reached. For a system which is controllable the inverse of the Gramian P_n exists and is also positive definite. The eigenvectors of the controllability Gramian and its inverse are the same, only their corresponding eigenvalues are inverted. Hence, the eigenvector with the smallest eigenvalue in P_n^{-1} corresponds to the eigenvector with the largest eigenvalue in P_n. The largest eigenvalues of P_n indicates the dominant directions of the most controllable states, while small eigenvalues indicate the directions that play a minor role in the controllable space.

Observability

Now we define the observability of a dynamic system which reflects to what extent the states influence the outputs.

Definition 4.2 *The dynamic system (4.29) is called observable if for any time $t \in (0, T]$, the initial condition $\mathbf{x}_0 \in \mathbb{R}^n$ can be determined from the time history of the input $\mathbf{u}(t)$ and the output $\mathbf{y}(t)$ in interval $[0, t] \subset [0, T]$. Otherwise, the system is called unobservable.*

The observability of the system can be determined by analyzing the rank of the observability matrix defined as

$$O_n := \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \dots \\ \mathbf{CA}^{n-1} \end{bmatrix} \in \mathbb{R}^{nm \times n}. \tag{4.35}$$

[1] The term 'energy' is used loosely here, motivated by the fact that energy can often be written as a quadratic form (e.g. potential energy as a function of squared pressure).

If O_n has full rank, then system (4.29) is observable and the state $\mathbf{x}_0$ can be detected from $\mathbf{y}(t)$ and $\mathbf{u}(t)$. Alternatively, we can check the observability of the system (4.29) by using an *observability Gramian*, which for a finite time is defined as

$$Q_n := \sum_{i=0}^{n-1} (\mathbf{CA}^i)^T (\mathbf{CA}^i) = O_n^T O_n, \tag{4.36}$$

and for an infinite time can be computed by solving a Lyapunov equation given by

$$\mathbf{A}^T Q \mathbf{A} + \mathbf{C}^T \mathbf{C} = Q. \tag{4.37}$$

The system (4.29) is observable if the observability Gramian is positive definite. Hence, the following conditions are equivalent: the given system is observable, or the observability Gramian is positive definite, or the observability matrix Q_n has a full row rank. Similarly, a very important fact is presented in Glover (1984), namely, in the observable system the maximal energy of the output of the model with initial state $\mathbf{x}_0$ and a current state $\mathbf{x}_t$ that are n time steps apart is given by

$$(\mathbf{x}_t - \mathbf{x}_0)^T Q_n (\mathbf{x}_t - \mathbf{x}_0) > 0. \tag{4.38}$$

Thus

$$\sum_{i=1}^{n} \mathbf{y}_i^T \mathbf{y}_i = (\mathbf{x}_t - \mathbf{x}_0)^T Q_n (\mathbf{x}_t - \mathbf{x}_0) > 0, \tag{4.39}$$

and it is the maximal energy of the given outputs. Therefore, Q_n gives us a measure how much of energy a state gives. The largest eigenvalues of Q_n indicates the dominant directions of the most observable states, while small eigenvalues indicate the directions which play a minor role in the observable space. Moreover, the observability Gramian of the system (4.29) is the same as the controllability Gramian of the adjoint system (see Antoulas (2005)) defined as

$$\begin{aligned} \frac{d\boldsymbol{\lambda}}{dt} &= -\mathbf{A}^T \boldsymbol{\lambda}(t) - \mathbf{C}^T \mathbf{v}(t), \ \boldsymbol{\lambda}(t_{n+1}) = \mathbf{0}, \\ \mathbf{u}(t) &= \mathbf{B}^T \boldsymbol{\lambda}(t) + \mathbf{D}^T \mathbf{y}(t). \end{aligned} \tag{4.40}$$

4.4.2 Balanced representation of the Linear Time Invariant systems

Balancing is a technique that combines the controllability and observability properties of a system and finds a linear transformation that produces an equivalent model with equal and diagonal controllability and observability Gramians. The equality of the Gramians implies that the states which are difficult to reach are simultaneously difficult to observe.

Definition 4.3 *The system (4.29) is called balanced if*

$$P_n = Q_n = \boldsymbol{\Sigma} \tag{4.41}$$

where $\boldsymbol{\Sigma}$ is the diagonal matrix and its elements on the diagonal are called Hankel singular values.

Each state of the balanced system has the same controllability and observability degree. Since the Gramians are basis dependent we can look for a basis for which the Gramians are equal and diagonal. Such a basis is called a *balanced basis*. These balanced Gramians are obtained by applying the transformation $\mathbf{T}$, such that,

$$\mathbf{T}P_n\mathbf{T}^T = \mathbf{T}^{-T}Q_n\mathbf{T}^{-1} = \boldsymbol{\Sigma}, \tag{4.42}$$

which is found by solving the following minimization problem

$$\min_{\mathbf{T}} trace[\mathbf{T}P_n\mathbf{T}^T + \mathbf{T}^{-T}Q_n\mathbf{T}^{-1}]. \tag{4.43}$$

The balancing consists of the simultaneous diagonalization of the controllability and observability Gramians. The minimum value of the above expression is twice the sum of the Hankel singular values, and the minimizing $\mathbf{T}$ is called a *balancing transformation*. In Antoulas (2005) four numerical approaches for computing a balancing transformation are presented. Here we follow one of them. First, we express the controllability Gramian as $P_n = \mathbf{X}\mathbf{X}^T$, and similarly, the observability Gramian as $Q_n = \mathbf{Y}\mathbf{Y}^T$. Then the SVD of the product $\mathbf{X}^T\mathbf{Y}$ can be written as

$$\mathbf{X}^T\mathbf{Y} = \mathbf{W}\boldsymbol{\Sigma}\mathbf{V}^T \tag{4.44}$$

where $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{n\times s}$, $\mathbf{W} \in \mathbb{R}^{s\times s}$ and $\mathbf{V} \in \mathbb{R}^{s\times s}$ are orthogonal matrices and $\boldsymbol{\Sigma} \in \mathbb{R}^{s\times s}$ is the diagonal matrix, $s \leq n$.

Lemma 4.1 *Given the controllable, observable and stable system and the corresponding Gramians P_n and Q_n, a balancing transformation is given as*

$$\begin{aligned} \mathbf{T} &= \Sigma^{-1/2}\mathbf{V}^T\mathbf{Y}^T, \\ \mathbf{T}^{-1} &= \mathbf{X}\mathbf{W}\Sigma^{-1/2}. \end{aligned}$$

To verify that $\mathbf{T}^{-1}$ is an inverse of $\mathbf{T}$ we need to check that

$$\begin{aligned} \mathbf{T}^{-1}\mathbf{T} &= \mathbf{X}\mathbf{W}\Sigma^{-1/2}\Sigma^{-1/2}\mathbf{V}^T\mathbf{Y}^T \\ &= \mathbf{X}\mathbf{W}\Sigma^{-1}\mathbf{V}^T\mathbf{Y}^T = \mathbf{I}, \end{aligned}$$

$$\begin{aligned} \mathbf{T}\mathbf{T}^{-1} &= \Sigma^{-1/2}\mathbf{V}^T\mathbf{Y}^T\mathbf{X}\mathbf{W}\Sigma^{-1/2} \\ &= \Sigma^{-1/2}\mathbf{V}^T\mathbf{V}\boldsymbol{\Sigma}\mathbf{W}^T\mathbf{W}\Sigma^{-1/2} = \mathbf{I}. \end{aligned}$$

The fact that it is a balancing transformation is proved by the following

$$\begin{aligned} \mathbf{T}P_n\mathbf{T}^T = \Sigma^{-1/2}\mathbf{V}^T\mathbf{Y}^T\mathbf{X}\mathbf{X}^T\left(\Sigma^{-1/2}\mathbf{V}^T\mathbf{Y}^T\right)^T &= \\ \Sigma^{-1/2}\mathbf{V}^T\mathbf{Y}^T\mathbf{X}\mathbf{X}^T\mathbf{Y}\mathbf{V}\Sigma^{-T/2} &= \\ \Sigma^{-1/2}\mathbf{V}^T(\mathbf{X}^T\mathbf{Y})^T\mathbf{X}^T\mathbf{Y}\mathbf{V}\Sigma^{-T/2} &= \\ \Sigma^{-1/2}\mathbf{V}^T(\mathbf{W}\boldsymbol{\Sigma}\mathbf{V}^T)^T(\mathbf{W}\boldsymbol{\Sigma}\mathbf{V}^T)\mathbf{V}\Sigma^{-T/2} &= \\ \Sigma^{-1/2}\mathbf{V}^T(\mathbf{V}\boldsymbol{\Sigma}^T\mathbf{W}^T)(\mathbf{W}\boldsymbol{\Sigma}\mathbf{V}^T)\mathbf{V}\Sigma^{-T/2} &= \Sigma, \end{aligned}$$

and in similar way it can be shown that

$$\mathbf{T}^{-T} Q_n \mathbf{T}^{-1} = \Sigma.$$

Then the system is transform to a balanced representation such that

$$\begin{aligned} \frac{d\hat{\mathbf{x}}}{dt} &= \mathbf{TAT}^{-1}\hat{\mathbf{x}}(t) + \mathbf{TBu}, \\ \hat{\mathbf{x}}(0) &= \mathbf{Tx}_0, \\ \mathbf{y} &= \mathbf{CT}^{-1}\hat{\mathbf{x}} + \mathbf{Du}. \end{aligned} \tag{4.45}$$

Furthermore, if we partition

$$\mathbf{W} = [\mathbf{W}_1 \mathbf{W}_2], \boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_1 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_2 \end{pmatrix}, \mathbf{V} = [\mathbf{V}_1, \mathbf{V}_2] \tag{4.46}$$

where $\mathbf{W}_1, \mathbf{V}_1$ consist of l columns and $\boldsymbol{\Sigma}_1 \in \mathbb{R}^{l \times l}$, the transformation

$$\begin{aligned} \mathbf{T}_1 &= \Sigma_1^{-1/2} \mathbf{V}_1^T \mathbf{Y}^T \\ \mathbf{T}_1^{-1} &= \mathbf{X}\mathbf{W}_1 \Sigma_1^{-1/2} \end{aligned}$$

satisfies $\mathbf{T}_1 \mathbf{T}_1^{-1} = \mathbf{I}_l$. A reduced-order system representation, known as *balanced truncation*, is obtained by expressing the system dynamics in terms of a limited number of linear combinations of the Hankel singular vectors corresponding to the leading Hankel singular values $\boldsymbol{\Sigma}_1$, i.e. by taking $\boldsymbol{\Phi} = \mathbf{T}_1^{-1}$ and $\boldsymbol{\Psi}^T = \mathbf{T}_1$ the system is transformed to the truncated representation:

$$\begin{aligned} \frac{d\mathbf{z}}{dt} &= \boldsymbol{\Psi}^T \mathbf{A} \boldsymbol{\Phi} \mathbf{z}(t) + \boldsymbol{\Psi}^T \mathbf{Bu}, \\ \mathbf{z}(0) &= \boldsymbol{\Psi}^T \mathbf{x}_0, \\ \mathbf{y} &= \mathbf{C}\boldsymbol{\Phi}\mathbf{z} + \mathbf{Du}, \end{aligned} \tag{4.47}$$

where a new trajectory $\mathbf{z}$ operates in lower dimension l.

4.4.3 Balanced truncation for nonlinear systems using empirical Gramians

Extensions of the method to nonlinear systems have been proposed by Scherpen (1993) (based on energy functions), Lall et al. (2002) (based on empirical Gramians), Hermann and Krener (1977) (based on linearization). In the case of large-scale nonlinear system only two approaches can be considered: Lall et al. (2002) and Hermann and Krener (1977). In the approach based on linearization, the nonlinear model is first linearized along a state trajectory resulting in an Linear Time Varying model (LTV) and then the controllability and observability properties are analyzed with linearized system matrices or by using the Gramians of a linearized model. This approach has been used in Van Doren (2010) to analyze the controllability and observability of two phase flow in a porous medium but it can not be applied under assumptions of this thesis, since the tangent linear approximation of the original reservoir system is not available. The second way to handle the nonlinearities of the system is by using the Balanced Proper Orthogonal Decomposition (BPOD) method, which is an approximation to balanced truncation. In linear systems the Gramians are computed by solving Lyapunov equations but for large systems this is computationally very demanding, if possible at all. When balanced truncation is generalized to nonlinear systems, or when a linear system is too large then empirical Gramians are used. Empirical Gramians were proposed by Moore (1981) although this was not directly connected to balanced truncation. For balanced truncation empirical Gramians were used by Lall et al. (1999), who combined the features of the POD method and balanced truncation. This method requires only standard matrix computations, and makes use of data from either simulation or experiment to identify the dynamics relevant to the input-output map of the system. The resulting reduced-order model is nonlinear, and has inputs and outputs suitable for control. When it is applied to linear systems it results in the standard balanced truncation, Lall et al. (2002). Here, we follow the approach of Rowley (2005), and we compute the balancing transformation directly from snapshots of empirical Gramians, without computing the Gramians themselves. Similar approaches were used by Willcox and Peraire (2002) and Antoulas (2005).

Controllability Gramian using snapshots

The most dominant direction in the controllability Gramian indicates the states that can be reached most easily, i.e. we can reach them with a minimal energy. If we collect the snapshots of the nonlinear system with a specific initial condition $\mathbf{x}_0$ for different input strategies $\mathbf{u}_1, \ldots, \mathbf{u}_p$ then these snapshots span the subspace to which we can steer a given system in a given time t_N. The correlation matrix of the states approximated by the snapshots is an approximation of the controllability Gramian and we refer to it as the empirical controllability Gramian. The following definition is taken from Lall et al. (2002).

Definition 4.4 *Let T be a set of r orthogonal $p \times p$ matrices, $\{T_1, \ldots, T_r\}$, and let M be a set of o positive constants, $\{c_1, \ldots, c_o\}$. Further define $E = \{\mathbf{e}_1, \ldots, \mathbf{e}_p\}$ standard unit vectors in $\mathbb{R}^p$. The input signal is constructed as*

$$\mathbf{u}_k(t) = c_i \mathbf{T}_j \mathbf{e}_k \delta(t) + \bar{\mathbf{u}} \tag{4.48}$$

where c_i is the input size, $\mathbf{T}_j \mathbf{e}_k$ describes the input pattern, $\delta(t)$ is a delta function and $\bar{\mathbf{u}}$ is the mean input. Then using the system trajectories corresponding to different inputs $\mathbf{u}_k$ the empirical controllability Gramian is defined by

$$P_n = \sum_{i=1}^{o} \sum_{j=1}^{r} \sum_{k=1}^{p} \frac{1}{rpc_i^2} \sum_{t=1}^{N} \mathbf{\Phi}_t^{ijk} \tag{4.49}$$

where $\mathbf{\Phi}_t^{ijk}$ is given by

$$\mathbf{\Phi}_t^{ijk} = (\mathbf{x}_t^{ijk} - \bar{\mathbf{x}}^{ijk})(\mathbf{x}_t^{ijk} - \bar{\mathbf{x}}_t^{ijk})^T \tag{4.50}$$

and where $\bar{\mathbf{x}}_t^{ijk}$ is the state of the system at time step t when a constant in time input strategy $\bar{\mathbf{u}}$ was applied, $\mathbf{x}_t^{ijk}$ is the state of the system at time step t with modified input $\mathbf{u}_k$.

The inputs should be chosen such that they represent the typical inputs of the system. The matrix $\mathbf{T}_i$ is usually chosen as $\mathbf{I}$ or $-\mathbf{I}$, and refers to positive and negative perturbations in the input variables, respectively. It has been shown in Lall et al. (2002) that for a stable, linear system using impulse inputs, the empirical Gramian is identical to the controllability Gramian. In order to compute the empirical controllability Gramian for a system with p inputs, the state responses to the perturbation of the inputs are collected and are put as columns in the matrix

$$\mathbf{X} = \{\mathbf{x}_1(t_1), ..., \mathbf{x}_1(t_N), ..., \mathbf{x}_p(t_1), ..., \mathbf{x}_p(t_N)\} \tag{4.51}$$

of size $n \times s$, where s is the total number of snapshots and is equal to $p \times N$. Then the empirical controllability Gramian is given by

$$P_n = \mathbf{X}\mathbf{X}^T. \tag{4.52}$$

Observability Gramian using snapshots

The observability Gramian indicates how much energy comes from a specific state vector. An approximation of the observability Gramian can be calculated from samples of the system with perturbed initial condition and fixed inputs $\bar{\mathbf{u}}$. This approach may require a very large data set, since the initial state is of large size. Using the fact that the observability Gramian is equal to the controllability Gramian of an adjoint system (see Antoulas (2005)) we can define the empirical observability Gramian as follows:

Definition 4.5 *Let T be a set of r orthogonal $n \times n$ matrices, $\{T_1, \dots, T_r\}$, and let M be a set of o positive constants, $\{c_1, \dots, c_o\}$. Further define $E = \{\mathbf{e}_1, \dots, \mathbf{e}_m\}$ standard unit vectors in $\mathbb{R}^m$. The input signal of the adjoint system (see Eqs. 4.40) is constructed as*

$$\mathbf{v}_k(t) = c_i \mathbf{T}_j \mathbf{e}_k \delta(t) + \bar{\mathbf{v}} \tag{4.53}$$

where c_i is the input size, $\mathbf{T}_j\mathbf{e}_k$ describes the input pattern, $\delta(t)$ is a delta function and $\bar{\mathbf{v}}$ is the mean input of the adjoint system. Then using the adjoint system trajectories corresponding to different adjoint inputs $\mathbf{v}_k$ the empirical observability Gramian is defined by

$$O_n = \sum_{i=1}^{o} \sum_{j=1}^{r} \sum_{k=1}^{m} \frac{1}{rmc_i^2} \sum_{t=1}^{N} \mathbf{\Psi}_t^{ijk} \tag{4.54}$$

where $\mathbf{\Psi}_t^{ijk}$ is given by

$$\mathbf{\Psi}_t^{ijk} = (\boldsymbol{\lambda}_t^{ijk} - \bar{\boldsymbol{\lambda}}^{ijk})^T (\boldsymbol{\lambda}_t^{ijk} - \bar{\boldsymbol{\lambda}}_t^{ijk}) \tag{4.55}$$

and where $\bar{\boldsymbol{\lambda}}_t^{ijk}$ is the state of the adjoint system at time step t when a constant in time input strategy $\bar{\mathbf{v}}$ was applied, $\boldsymbol{\lambda}_t^{ijk}$ is the state of the adjoint system at time step t with modified input $\mathbf{v}_k$.

To compute the empirical observability Gramian we compute the impulse responses of the adjoint model and we put them as columns in matrix

$$\mathbf{Y} = \{\boldsymbol{\lambda}(t_1), ..., \boldsymbol{\lambda}_1(t_N), ..., \boldsymbol{\lambda}_m(t_1), ..., \boldsymbol{\lambda}_m(t_N)\}. \tag{4.56}$$

Then the empirical observability Gramian is given by

$$Q_n = \mathbf{Y}\mathbf{Y}^T. \tag{4.57}$$

The adjoint variable can be interpreted as a multiplier which indicates the relevance of each state in the calculation of the output of the system.

Balanced POD using the method of snapshots
Following Rowley (2005) it is not necessary to compute the empirical Gramians explicitly, instead we can use the square roots of them ($P_n = \mathbf{X}\mathbf{X}^T$ and $Q_n = \mathbf{Y}\mathbf{Y}^T$). In the method of snapshots used here, the balancing patterns are computed by solving the SVD of the matrix $\mathbf{Y}^T\mathbf{X}$, i.e.

$$\mathbf{Y}^T\mathbf{X} = \mathbf{W}\mathbf{\Sigma}\mathbf{V}^T = \begin{bmatrix} \mathbf{W}_1 & \mathbf{W}_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & \mathbf{0} \\ \mathbf{0} & \Sigma_2 \end{bmatrix} \begin{bmatrix} \mathbf{V}_1^T \\ \mathbf{V}_2^T \end{bmatrix}, \tag{4.58}$$

where $\mathbf{Y}^T\mathbf{X} \in \mathbb{R}^{s_P \times s_Q}$ and s_P and s_Q are the number of columns in $\mathbf{X}$ and $\mathbf{Y}$, respectively. Since usually the number of snapshots is much smaller than the number of states, solving the SVD problem is computationally feasible. The computation time required to build the projection subspace is about twice more expensive than in the case of the POD method, because in this approach we need to generate snapshots of the adjoint model which is computationally almost as expensive as generating snapshots of the forward model.

Given the SVD of the matrix $\mathbf{Y}^T\mathbf{X}$, the balancing transformation is defined as

$$\mathbf{T} = \mathbf{\Sigma}_1^{-1/2}\mathbf{W}_1^T\mathbf{Y}^T \tag{4.59}$$

and its inverse is given by:

$$\mathbf{T}^{-1} = \mathbf{X}\mathbf{V}_1\mathbf{\Sigma}_1^{-1/2}. \tag{4.60}$$

The projection $\mathbf{\Pi} = \mathbf{\Phi}\mathbf{\Psi}^T$ is defined as a Petrov-Galerkin projection, where $\mathbf{\Psi}^T$ consists of first l columns of matrix $\mathbf{T}$ corresponding to the largest singular values, and $\mathbf{\Phi}$ consists of the first l rows of $\mathbf{T}^{-1}$. The reduced-order system is then described by

$$\begin{aligned} \mathbf{z}(t_i) &= \mathbf{\Psi}^T\mathbf{f}_i(\mathbf{\Phi}\mathbf{z}(t_{i-1}), \boldsymbol{\theta}) \\ \mathbf{y}(t_i) &= \mathbf{h}_i(\mathbf{\Phi}\mathbf{z}(t_i), \boldsymbol{\theta}), \\ \mathbf{z}(t_0) &= \mathbf{\Psi}^T\mathbf{x}(t_0). \end{aligned} \tag{4.61}$$

This reduced-order system preserves the states which are relatively easy to control and at the same time relatively easy to observe.
The drawbacks of the POD method presented in the previous section are also present in the BPOD method.

4.5 Connection between the POD and the BPOD methods

There is a connection between the POD and the BPOD methods. In the POD method the inner product used for computing the POD patterns and for projecting the dynamic system is arbitrary. Usually the standard dot product is used, as in our case, because it is not obvious which inner product is the most suitable. For stable linear systems the BPOD method can be seen as the POD method using an impulse response dataset and using the observability Gramian to define an inner product as

$$\forall\, \mathbf{a},\, \mathbf{b} \in \mathbb{R}^n \ < \mathbf{a}, \mathbf{b} >_{Q_n} := \mathbf{a}^T Q_n \mathbf{b} \tag{4.62}$$

where Q_n is the observability Gramian.

In the POD method to find the projection matrix $\mathbf{\Phi}$ the following eigenvalue problem is solved:

$$\mathbf{X}\mathbf{X}^T Q_n \boldsymbol{\phi}_i = \lambda_i \boldsymbol{\phi}_i. \tag{4.63}$$

If the POD method uses an impulse respond dataset then $P_n \approx \mathbf{X}\mathbf{X}^T$. Using the assumption that $Q_n \approx \mathbf{Y}\mathbf{Y}^T$ and multiplying both sides of Eq. 4.63 by $c_i \mathbf{Y}^T$ with $c_i \neq 0$ we obtain the following equivalent problem

$$\left(\mathbf{Y}^T\mathbf{X}\right)\left(\mathbf{Y}^T\mathbf{X}\right)^T \left(c_i \mathbf{Y}^T \boldsymbol{\phi}_i\right) = \lambda_i \left(c_i \mathbf{Y}^T \boldsymbol{\phi}_i\right). \tag{4.64}$$

If $\mathbf{Y}^T\mathbf{X} = \mathbf{W}\mathbf{\Sigma}\mathbf{V}^T$ then

$$\left(\mathbf{Y}^T\mathbf{X}\right)\left(\mathbf{Y}^T\mathbf{X}\right)^T = \mathbf{W}\mathbf{\Sigma}^2\mathbf{W}^T, \tag{4.65}$$

and thus

$$c_i \mathbf{Y}^T \boldsymbol{\phi}_i = \mathbf{w}_i. \tag{4.66}$$

Multiplying Eq. 4.66 by $\mathbf{w}_i^T$ and using the fact that $\mathbf{w}_i^T \mathbf{w}_i = 1$ we obtain

$$c_i \mathbf{w}_i^T \mathbf{Y}^T \boldsymbol{\phi}_i = 1, \tag{4.67}$$

which defines $\boldsymbol{\psi}_i^T$:

$$\boldsymbol{\psi}_i^T = c_i \mathbf{w}_i^T \mathbf{Y}^T, \tag{4.68}$$

where $c_i = \sigma_i^{-1/2}$. The obtained POD-based projection matrix $\mathbf{\Psi}$ is the same as in the BPOD method. Moreover, using the decomposition $\mathbf{Y}^T\mathbf{X} = \mathbf{W}\mathbf{\Sigma}\mathbf{V}^T$ we know that

$$\left(\mathbf{Y}^T\mathbf{X}\right)^T \left(\mathbf{Y}^T\mathbf{X}\right) = \mathbf{V}\mathbf{\Sigma}^2\mathbf{V}^T. \tag{4.69}$$

Thus,

$$\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X}\mathbf{v}_i = \sigma_i{}^2\mathbf{v}_i. \tag{4.70}$$

Multiplying it by $c_i\mathbf{X}$ we obtain

$$\mathbf{X}\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\left(c_i\mathbf{X}\mathbf{v}_i\right) = \sigma_i^2\left(c_i\mathbf{X}\mathbf{v}_i\right). \tag{4.71}$$

Hence,

$$\boldsymbol{\phi}_i = c_i\mathbf{X}\mathbf{v}_i, \tag{4.72}$$

where $c_i = \sigma_i{}^{-1/2}$. The POD-based projection matrices are the same as the BPOD-based projection matrices. Therefore, the BPOD method can be seen as the POD method with norm defined by (4.62) and snapshots $\mathbf{X}$ such that $P \approx \mathbf{X}\mathbf{X}^T$.

5 CHAPTER

Model-reduced gradient-based history matching

Gr adient-based history matching algorithms can be used to adapt the uncertain parameters in a reservoir model using production data. However, they require the implementation of an adjoint model and sensitivity matrices of the reservoir model with respect to uncertain parameters, which usually involves a considerable programming effort. We propose a new approach to gradient-based history matching based on model reduction, where the original (nonlinear and high-order) forward model is replaced by a linear reduced-order forward model and, consequently, the adjoint of the tangent linear approximation of the original forward model is replaced by the adjoint of a linear reduced-order forward model. The reduced-order model is constructed by means of the Proper Orthogonal Decomposition (POD) method or Balanced Proper Orthogonal Decomposition (BPOD) method. The reduced-order model is not, however, obtained by projection of the nonlinear system of equations as in the conventional projection-based ROM techniques, but is built in the reduced subspace. The conventional POD method requires the availability of the high-order tangent model, i.e. of the Jacobians with respect to the states that we do not have. Model-reduced method obtains a reduced-order approximation of the tangent linear model directly by computing approximate derivatives of the reduced-order model. Then due to the linear character of the reduced model, the corresponding adjoint model is easily obtained. The gradient of the objective function is approximated and the minimization problem is solved in the reduced space; the procedure is iterated with the updated estimate of the parameters if necessary. The POD-based approach is adjoint-free and can be used with any reservoir simulator, while the BPOD-based approach requires an adjoint model but does not require the sensitivity matrices of the model with respect to uncertain parameters. The methods are compared with a few other gradient-based history matching algorithms, including: a classical adjoint-based history matching algorithm, a simultaneous perturbation stochastic approximation algorithm, and a gradient-based finite-difference algorithm.

5.1 Introduction

History matching problem
The history matching problem can be interpreted as a constrained minimization problem of the given objective function

$$\begin{aligned} J(\boldsymbol{\theta}) &= \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}_{init})^T \mathbf{R}_b^{-1}(\boldsymbol{\theta} - \boldsymbol{\theta}_{init}) + \\ &\quad + \frac{1}{2}\sum_{i=1}^{N_o} [\mathbf{d}(t_i) - \mathbf{h}_i[\mathbf{x}(t_i), \boldsymbol{\theta}]]^T \mathbf{R}_i^{-1} [\mathbf{d}(t_i) - \mathbf{h}_i[\mathbf{x}(t_i), \boldsymbol{\theta}]] . \end{aligned} \tag{5.1}$$

where the constraints are formed by the system equations:

$$\mathbf{x}(t_i) = \mathbf{f}_i(\mathbf{x}(t_{i-1}), \boldsymbol{\theta}, \mathbf{u}(t_i)) \tag{5.2}$$

with $\mathbf{x}(t_0) = \mathbf{x}_{init}$. N_o is the number of time instants at which observations were collected and $\boldsymbol{\theta}_{init}$ is the prior parameter vector. The system equations and observation operator $\mathbf{h}_i$ have been defined in Chapter 4 (see Eq. 4.1 and Eq. 4.2), while the objective function J has been defined in Chapter 3 (see Eq. 3.6).

Reparameterization
Since the vector $\boldsymbol{\theta}$ may consists of properties in each grid block of the model, e.g. permeabilities or porosities, the number of uncertain parameters p can be very large. In case of the model-reduced method and the finite-difference method it is necessary to reparameterize the vector $\boldsymbol{\theta}$. In this thesis we use the POD method (the Karhunen-Loéve expansion) to reparameterize the parameter space and, consequently, to reduce the number of parameters to be estimated. In particular we consider reparameterization of the permeability field as preparation to the numerical example to be discussed in the next chapter. The permeability vector $\boldsymbol{\theta}$ is approximated by

$$\boldsymbol{\theta} \approx \boldsymbol{\theta}_{init} + \boldsymbol{\Phi}_\eta \boldsymbol{\eta} \Longleftrightarrow \Delta\boldsymbol{\theta} = \boldsymbol{\Phi}_\eta \boldsymbol{\eta}, \tag{5.3}$$

where the permeability patterns creating $\boldsymbol{\Phi}_\eta$ are the first p_{red} dominant eigenvectors of a low-rank approximation of the covariance matrix of the permeability field, represented by an ensemble of prior permeability fields and $\boldsymbol{\theta}_{init}$ is the average of the ensemble. Alternatively, other reparameterization methods such as the discrete cosine transform (Jafarpour and McLaughlin (2009)), or a discrete wavelet transform (Sahni and Horne (2005)) may be chosen, as long as they lead to a significant reduction in the number of parameters. See Chapter 3 for their short descriptions.

History matching problem with reparameterization
After replacing $\boldsymbol{\theta}$ with $\boldsymbol{\theta}_{init} + \boldsymbol{\Phi}_\eta \boldsymbol{\eta}$ the objective function can be rewritten as

$$J(\boldsymbol{\eta}) = \frac{1}{2}\boldsymbol{\eta}^T \boldsymbol{\Phi}_\eta^T \mathbf{R}_b^{-1} \boldsymbol{\Phi}_\eta \boldsymbol{\eta} + \tag{5.4}$$

$$+\frac{1}{2}\sum_{i=1}^{N_o} \left[\mathbf{d}(t_i) - \mathbf{h}_i[\mathbf{x}(t_i), \boldsymbol{\theta}_{init} + \boldsymbol{\Phi}_\eta \boldsymbol{\eta}]\right]^T \mathbf{R}_i^{-1} \left[\mathbf{d}(t_i) - \mathbf{h}_i[\mathbf{x}(t_i), \boldsymbol{\theta}_{init} + \boldsymbol{\Phi}_\eta \boldsymbol{\eta}]\right] .$$

5.2 Gradient-based model-reduced history matching algorithm

Classical gradient-based history matching formulated for a general model represents a nonlinear constrained optimization problem that is very difficult to solve (see the next section). It can be greatly simplified with the hypothesis that the objective function can be made quadratic by assuming that the $\mathbf{f}_i$ and $\mathbf{h}_i$ operators can be linearized. The quadratic problem has theoretical advantages since it involves solving only linear equations and it has a unique solution. Therefore, the tangent linear model of the original model is written using the first-order Taylor formula in the vicinity of the background parameter $\boldsymbol{\theta}^b$; this results in the following equation:

$$\Delta\bar{\mathbf{x}}(t_i) = \frac{\partial \mathbf{f}_i[\mathbf{x}(t_{i-1}),\boldsymbol{\theta}]}{\partial \mathbf{x}(t_{i-1})}\Delta\bar{\mathbf{x}}(t_{i-1}) + \sum_{j=1}^{p} \frac{\partial \mathbf{f}_i[\mathbf{x}(t_{i-1}),\boldsymbol{\theta}]}{\partial \theta_j}\Delta\theta_j, \tag{5.5}$$

where p is the size of the parameter space and $\Delta\bar{\mathbf{x}}$ is a deviation of the model from the background trajectory. The partial derivatives in Eq. (5.5) are evaluated at $(\mathbf{x}^b(t_{i-1}),\boldsymbol{\theta}^b)$, where $\mathbf{x}^b$ is the nonlinear state vector with a parameter value $\boldsymbol{\theta}^b$, that is, $\mathbf{x}^b(t_i) = \mathbf{f}_i[\mathbf{x}^b(t_{i-1}),\boldsymbol{\theta}^b]$. Eq. (5.5) can be rewritten as

$$\begin{bmatrix} \Delta\bar{\mathbf{x}}(t_i) \\ \Delta\boldsymbol{\theta} \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathbf{f}_i[\mathbf{x}(t_{i-1}),\boldsymbol{\theta}]}{\partial \mathbf{x}(t_{i-1})} & \frac{\partial \mathbf{f}_i[\mathbf{x}(t_{i-1}),\boldsymbol{\theta}]}{\partial \boldsymbol{\theta}} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \Delta\bar{\mathbf{x}}(t_{i-1}) \\ \Delta\boldsymbol{\theta} \end{bmatrix}, \tag{5.6}$$

which is a linear model in terms of variations. The minimization process is then performed by the following iterative steps:

- linearize the model around the background geological parameters $\boldsymbol{\theta}^b_k$, where k denotes the iteration number and $\boldsymbol{\theta}^b_1 = \boldsymbol{\theta}_{init}$,
- find $\Delta\boldsymbol{\theta}_k$, such that it minimizes the quadratic objective function (details are given in the next sections, see Eq. 5.27),
- update $\boldsymbol{\theta}^b_{k+1} = \boldsymbol{\theta}^b_k + \Delta\boldsymbol{\theta}_k$.

The history matching procedure turns into a scheme that consists of two loops: an inner loop, which finds the minimum of the quadratic objective function and an outer loop, where the original model is used to redefine the model trajectory and to calculate the original objective function. It iterates until some predefined convergence criteria are met. In practice, in order to make the computation more efficient, the tangent linear model is replaced by a simpler low-resolution linear model. This is called the incremental approach. It has been described in Courtier et al. (1994) and is used in operational systems for weather forecasting. The difference with the model-reduced approach is that we do not need to develop and implement a simpler linear model. We construct it by using a reduced-order modeling technique and information from the perturbations of the original model. Because the reduced-order model is designed to capture the dominant dynamics of the original system we expect that we can successfully use it for a fast minimization of the objective function in the inner loop. We note, however, that there is no guarantee that the reduced-order model will always lead to the same response with respect to parameter variations as the full-order model.

5.2.1 Initialization

The initial parameter vector, $\boldsymbol{\theta}_{init}$, is chosen as a first guess of the uncertain parameters, $\boldsymbol{\theta}_1^b = \boldsymbol{\theta}_{init}$. To keep the notation simple we skip the index indicating the outer iteration and we denote the reference parameters as $\boldsymbol{\theta}^b$. Next the high-order reservoir model is simulated with parameters values $\boldsymbol{\theta}^b$, and the initial objective function is calculated.

5.2.2 Collection of snapshots and pattern selection

POD-based model reduction

The POD method can be used to obtain an approximate low-order version of the tangent-linear model. This is a somewhat different procedure than the conventional application of the POD method for state-space reduction. In the conventional method, a data matrix is built from selected state vectors of a forward simulation of the large-scale numerical model, called *snapshots*. Spatial basis functions are obtained by computing a low-rank approximation of the covariance matrix of the data, and by selecting the leading eigenvectors. Next, the (tangent-) linear model equations are projected on the low-order basis formed by the basis functions. It results in a reduced-order model, in terms of coefficients multiplying the basis functions, which is still able to reproduce the dominant dynamic behavior of the original model, see e.g. Cardoso et al. (2009) or Van Doren et al. (2004) for further details. If the original model is nonlinear, the matrix coefficients of the tangent-linear high-order model need to be recomputed every time step before applying the reduction, which significantly reduces the computational advantage of the POD method as compared to the linear case. Moreover, the conventional POD method requires the availability of the high-order tangent model, i.e. of the Jacobians with respect to the states. In our approach we aim at obtaining a reduced-order approximation of the tangent linear model directly by computing approximate derivatives of the reduced-order model.

In this application the reduced-order model is used for parameter estimation and therefore the snapshots should be able to represent the behavior of the system for modified parameter values. They are created in the following way:

$$\mathbf{x}_j(t_i) = \mathbf{f}_i[\mathbf{x}^b(t_{i-1}), \boldsymbol{\theta}^b + \boldsymbol{\phi}_{\boldsymbol{\eta}}^j \eta_j] - \mathbf{f}_i[\mathbf{x}^b(t_{i-1}), \boldsymbol{\theta}^b], \tag{5.7}$$

where $\boldsymbol{\phi}_{\boldsymbol{\eta}}^j$ is the jth pattern in matrix $\boldsymbol{\Phi}_{\boldsymbol{\eta}}$ and η_j is the size of the perturbation, and they are put as columns in matrix

$$\mathbf{X} = \{\mathbf{x}_1(t_1), ..., \mathbf{x}_1(t_N), ..., \mathbf{x}_{p_{red}}(t_1), ..., \mathbf{x}_{p_{red}}(t_N)\} \in \mathbb{R}^{n \times s}, \tag{5.8}$$

where s is the total number of snapshots, n is the dimension of the state vector and p_{red} is the dimension of the parameter space after reparameterization. The given set of snapshots spans a subspace on which the reduced-order model is built. Since typically $s \ll n$, instead of solving the eigenvalue problem of matrix $\mathbf{X}\mathbf{X}^T \in \mathbb{R}^{n \times n}$, the reduced eigenvalue problem

$$(\mathbf{X}^T\mathbf{X})\mathbf{w}_i = \lambda_i \mathbf{w}_i; \;\; i \in \{1, ..., s\} \tag{5.9}$$

is solved. The eigenvalues of those problems are the same and the following relationship holds between the eigenvectors: $\boldsymbol{\phi}_i = \lambda_i^{-\frac{1}{2}} \mathbf{X}\mathbf{w}_i$, for $i \in \{1, ..., s\}$. The eigenvectors corresponding to the largest eigenvalues of matrix $\mathbf{X}\mathbf{X}^T$ represent the dominant directions present in the snapshots, called *patterns*. For each pattern its relative importance α_i, defined as the percentage of the eigenvalues in the total sum of eigenvalues, is calculated, that is, $\alpha_i = \frac{\lambda_i}{\sum_{j=1}^{s} \lambda_j}$. By choosing the n_{red} patterns with the largest relative importance an orthonormal projection matrix $\mathbf{\Phi}$ is created by taking vectors $\boldsymbol{\phi}_i$ as its columns, for $i \in \{1, ..., n_{red}\}$, where $n_{red} \leq s$.
In order to select the number n_{red} we use the relative importance α_i. n_{red} is chosen as the smallest number, such that $\sum_{j=1}^{n_{red}} \alpha_j \geq \alpha^e$, where α^e is some percentage of the total relative importance. One can notice that if the last eigenvector which is included in the set of patterns $\mathbf{\Phi}$ has the same eigenvalue as the next eigenvector which is not selected because the required level of relative importance is satisfied, then this procedure is not uniquely defined and depends on the sorting routine used to order the eigenvalues. Hence, one may check the next eigenvalue and include a corresponding eigenvector if the eigenvalues are equal.

Balanced POD-based model reduction

When the BPOD method is used to obtain an approximate reduced-order tangent-linear model, then forward and adjoint snapshots are required, and they are collected in the matrices $\mathbf{X}$ and $\mathbf{Y}$, respectively. Typically, the balanced truncation and the BPOD methods are applied to the systems which can be controlled. In this application we can control the system but it is not of our interest. Because the history matching routine analyzes the past data, the inputs which were used in the past are known (to certain extent) and they can not be changed. We aim, however, to steer the uncertain parameters to the values that match the past data. We adopted in this work the terminology and the ideas of balanced truncation to history matching problems, i.e. we aim to control the numerical model by changing the uncertain parameters. To compute the empirical controllability Gramian for a system with p_{red} parameters, the state responses to the perturbations of the parameters are collected exactly in the same way as in the POD method. To generate the data for the observability Gramian calculation we should excite the adjoint model with impulses to obtain an approximation to the true Gramian. Since we analyze past data, the outputs of the 'true' system are known. Therefore, we excite the adjoint model with the observed data weighted with the measurement covariance matrix. If N_o is the number of time instants where the outputs are observed and m is the number of measurement at each time instant then the Gramian calculation requires $N_o \times m$ integrations of the adjoint system. Since observations are sparse we take all measurements available at a certain point in time together and we integrate the adjoint model once. The adjoint state takes non-zero values from the time instant at which the observation is included, and it is integrated backward in time to quantify which part of the state is relevant in the gradient calculation. We include all time instants at once as well. Because the system is nonlinear with respect to parameters we collect snapshots using adjoint models corresponding to the forward models that are used to collect snapshots for the controllability Gramian.

In the method of snapshots used here, the balancing patterns are computed by solving the SVD of the matrix $\mathbf{Y}^T\mathbf{X}$, which can be decomposed as

$$\mathbf{Y}^T\mathbf{X} = \mathbf{W}\boldsymbol{\Sigma}\mathbf{V}^T = \begin{bmatrix} \mathbf{W}_1 & \mathbf{W}_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & \mathbf{0} \\ \mathbf{0} & \Sigma_2 \end{bmatrix} \begin{bmatrix} \mathbf{V}_1^T \\ \mathbf{V}_2^T \end{bmatrix}, \tag{5.10}$$

where $\mathbf{Y}^T\mathbf{X}$ has dimensions equal to the number of snapshots collected for the controllability Gramian times the number of snapshots collected for the observability Gramian. The Gramians themselves never need to be computed and only a single SVD is solved. Since usually the number of snapshots is much smaller than the number of states, this is computationally efficient.

The balancing transformation is given by the matrix:

$$\mathbf{T} = \boldsymbol{\Sigma}_1^{-1/2}\mathbf{W}_1^T\mathbf{Y}^T \tag{5.11}$$

and its inverse is given by:

$$\mathbf{T}^{-1} = \mathbf{X}\mathbf{V}_1\boldsymbol{\Sigma}_1^{-1/2}. \tag{5.12}$$

The projection $\boldsymbol{\Pi} = \boldsymbol{\Phi}\boldsymbol{\Psi}^T$ is defined as a Petrov-Galerkin projection, where $\boldsymbol{\Psi}$ consists of n_{red} columns of matrix $\mathbf{T}$ corresponding to the largest Hankel singular values that we want to retain, and $\boldsymbol{\Phi}$ consists of the first n_{red} rows of $\mathbf{T}^{-1}$. In order to select the number n_{red} we use, as before, the relative importance of patterns.

Remarks

The tangent linear approximation of the original model is

$$\Delta\bar{\mathbf{x}}(t_i) = \frac{\partial \mathbf{f}_i[\mathbf{x}(t_i),\boldsymbol{\theta}]}{\partial \mathbf{x}(t_{i-1})}\Delta\bar{\mathbf{x}}(t_{i-1}) + \frac{\partial \mathbf{f}_i[\mathbf{x}(t_i),\boldsymbol{\theta}]}{\partial \boldsymbol{\theta}}\boldsymbol{\Delta\theta}, \tag{5.13}$$

and its adjoint model is

$$\boldsymbol{\lambda}(t_i) = \left(\frac{\partial \mathbf{f}_{i+1}[\mathbf{x}(t_i),\boldsymbol{\theta}]}{\partial \mathbf{x}(t_i)}\right)^T \boldsymbol{\lambda}(t_{i+1}) + \left(\frac{\partial \mathbf{h}(t_i)[\mathbf{x}(t_i),\boldsymbol{\theta}]}{\partial \mathbf{x}(t_i)}\right)^T \mathbf{v}(t_i), \tag{5.14}$$

where $\mathbf{v}(t_i) = \mathbf{R}_i^{-1}\left[\mathbf{d}(t_i) - \mathbf{h}_i[\mathbf{x}(t_i),\boldsymbol{\theta}]\right]$. Projecting the equations of the tangent linear model using $\boldsymbol{\Pi}$ we obtain the following reduced-order linear model

$$\mathbf{z}(t_i) = \boldsymbol{\Psi}^T\frac{\partial \mathbf{f}_i[\mathbf{x}(t_i),\boldsymbol{\theta}]}{\partial \mathbf{x}(t_{i-1})}\boldsymbol{\Phi}\mathbf{z}(t_{i-1}) + \boldsymbol{\Psi}^T\frac{\partial \mathbf{f}_i[\mathbf{x}(t_i),\boldsymbol{\theta}]}{\partial \theta}\boldsymbol{\Delta\theta}. \tag{5.15}$$

The adjoint model of the reduced-order linearized model has the form

$$\gamma(t_i) = \left(\mathbf{\Psi}^T \frac{\partial \mathbf{f}_{i+1}[\mathbf{x}(t_i),\boldsymbol{\theta}]}{\partial \mathbf{x}(t_i)}\mathbf{\Phi}\right)^T \gamma(t_{i+1}) + \left(\frac{\partial \mathbf{h}(t_i)[\mathbf{x}(t_i),\boldsymbol{\theta}]}{\partial \mathbf{x}(t_i)}\mathbf{\Phi}\right)^T \mathbf{v}(t_i). \quad (5.16)$$

One can notice that if we require that $\boldsymbol{\lambda}(t_i) \approx \mathbf{\Psi}\gamma(t_i)$, then the projection of the original adjoint model corresponds to the adjoint of the reduced-order model. This does not have to be true if the external projection $\mathbf{\Psi}$ is chosen only based on the reconstruction of the forward model. Patterns that accurately represent the original state do not necessary accurately represent the adjoint state. Therefore, we should aim to find a projection that represents both, the original forward model and the adjoint model with acceptable accuracy. This is achieved by the BPOD method, where we reduce the model by constructing the projection matrix $\mathbf{\Pi}$ using both the forward and the adjoint states.

5.2.3 Building the reduced-order linear model

A model can be reduced by reconstructing the incremental state $\Delta\bar{\mathbf{x}}(t_i)$ according to

$$\Delta\bar{\mathbf{x}}(t_i) = \mathbf{\Phi}\mathbf{z}(t_i), \quad (5.17)$$

where $\mathbf{\Phi}$ is the projection matrix derived in the previous subsection and $\mathbf{z} \in \mathbb{R}^{n_{red}}$ is the reduced state vector. Combining equations (5.17), (5.5) and (5.3) results in

$$\mathbf{z}(t_i) = \mathbf{N}_i\mathbf{z}(t_{i-1}) + \mathbf{N}_i^{\theta}\boldsymbol{\eta}, \quad (5.18)$$

where

$$\mathbf{N}_i = \mathbf{\Psi}^T \frac{\partial \mathbf{f}_i[\mathbf{x}(t_{i-1}),\boldsymbol{\theta}]}{\partial \mathbf{x}(t_{i-1})}\mathbf{\Phi}, \quad (5.19)$$

$$\mathbf{N}_i^{\theta} = \mathbf{\Psi}^T \frac{\partial \mathbf{f}_i[\mathbf{x}(t_{i-1}),\boldsymbol{\theta}]}{\partial \boldsymbol{\theta}}\mathbf{\Phi}_{\eta}, \quad (5.20)$$

and where $\mathbf{z}$ represents the response of the original model to a perturbation of the parameters, such that we can write:

$$\begin{bmatrix} \mathbf{z}(t_i) \\ \boldsymbol{\eta} \end{bmatrix} = \begin{bmatrix} \mathbf{N}_i & \mathbf{N}_i^{\theta} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{z}(t_{i-1}) \\ \boldsymbol{\eta} \end{bmatrix}. \quad (5.21)$$

The partial derivatives in formulas (5.19) and (5.20) can not be easily computed explicitly and should be approximated by

$$\left[\frac{\partial \mathbf{f}_i[\mathbf{x}(t_{i-1}),\boldsymbol{\theta}]}{\partial \boldsymbol{\theta}}\right]_j \approx \frac{\mathbf{f}_i[\mathbf{x}(t_{i-1}),\boldsymbol{\theta}+\epsilon_\theta \mathbf{i}_j^{\theta}] - \mathbf{f}_i[\mathbf{x}(t_{i-1}),\boldsymbol{\theta}]}{\epsilon_\theta}, \quad (5.22)$$

and

$$\left[\frac{\partial \mathbf{f}_i[\mathbf{x}(t_{i-1}),\boldsymbol{\theta}]}{\partial \mathbf{x}(t_{i-1})}\right]_j \approx \frac{\mathbf{f}_i[\mathbf{x}(t_{i-1})+\epsilon \mathbf{i}_j,\boldsymbol{\theta}] - \mathbf{f}_i[\mathbf{x}(t_{i-1}),\boldsymbol{\theta}]}{\epsilon} \quad (5.23)$$

where ϵ and ϵ_θ are the intervals in which the partial differentials are linearized, and $\mathbf{i}_j$ and $\mathbf{i}_j^\theta$ are the jth columns of identity matrices with dimension, $n \times n$ and $p \times p$ respectively. Of course, those perturbations are too expensive for large-scale models. From (5.19) and (5.20), however, we see that there is no need to calculate those partial derivatives explicitly, and instead of approximating terms $\frac{\partial \mathbf{f}_i[\mathbf{x}(t_{i-1}),\boldsymbol{\theta}]}{\partial \mathbf{x}(t_{i-1})}$ we approximate the terms $\frac{\partial \mathbf{f}_i[\mathbf{x}(t_{i-1}),\boldsymbol{\theta}]}{\partial \mathbf{x}(t_{i-1})}\boldsymbol{\Phi}$ as

$$\frac{\partial \mathbf{f}_i[\mathbf{x}(t_{i-1}),\boldsymbol{\theta}]}{\partial \mathbf{x}(t_{i-1})}\boldsymbol{\phi}_j \approx \frac{\mathbf{f}_i[\mathbf{x}(t_{i-1})+\epsilon\boldsymbol{\phi}_j,\boldsymbol{\theta}] - \mathbf{f}_i[\mathbf{x}(t_{i-1}),\boldsymbol{\theta}]}{\epsilon}, \tag{5.24}$$

where $\boldsymbol{\phi}_j$ is the jth column of $\boldsymbol{\Phi}$. In the same manner we approximate the terms $\frac{\partial \mathbf{f}_i[\mathbf{x}(t_{i-1}),\boldsymbol{\theta}]}{\partial \boldsymbol{\theta}}\boldsymbol{\Phi}_\eta$ as

$$\frac{\partial \mathbf{f}_i[\mathbf{x}(t_{i-1}),\boldsymbol{\theta}]}{\partial \boldsymbol{\theta}}\boldsymbol{\phi}_\eta^j \approx \frac{\mathbf{f}_i[\mathbf{x}(t_{i-1}),\boldsymbol{\theta}+\epsilon_\theta\boldsymbol{\phi}_\eta^j] - \mathbf{f}_i[\mathbf{x}(t_{i-1}),\boldsymbol{\theta}]}{\epsilon_\theta}, \tag{5.25}$$

where $\boldsymbol{\phi}_\eta^j$ is the jth column of $\boldsymbol{\Phi}_\eta$. It results in a reduced-order model which is linearized along the patterns. The dimension of the reduced-order model depends on the number of patterns n_{red}, and on the number of variables p_{red} to be estimated, which are both expected to be small (see Chapter 3). Moreover, the matrices in the reduced-order model are known explicitly and the reduced-order model is linear. Therefore, the CPU time of a single reduced-order model simulation is negligible. However, the computational overhead to construct the reduced-order model may be significant.

We described how to build the reduced-order linear model, which is a low-order approximation to the high-order tangent linear approximation of the original reservoir model. We presented the formulation of the method using an explicit form of the reservoir equations which was motivated by the resulting simplification in the description of the method. An explicit method calculates the state of a system at a current time step explicitly from the parameters and the state of the system at the previous time step, while an implicit method finds a solution by solving an equation involving both: the current state of the system and the previous one. Since, the tangent linear model in both cases can be rewritten such that it calculates the approximate state at a current time step from the approximate state at the previous time step, and from the parameters, the choice of an explicit or an implicit method is irrelevant.

5.2.4 Improving parameters in the reduced space

When the reduced-order approximate linear model is available, the adjoint model is easy to implement (due to its linear character) and is governed by the following equation

$$\underline{\boldsymbol{\lambda}}(t_i) = \mathbf{N}_{i+1}^T \underline{\boldsymbol{\lambda}}(t_{i+1}) + \left(\frac{\partial \mathbf{h}_i[\mathbf{x}(t_i), \boldsymbol{\theta}]}{\partial \mathbf{x}(t_i)} \boldsymbol{\Phi} \right)^T \mathbf{R}_i^{-1}$$
$$\left[\mathbf{d}(t_i) - \mathbf{h}_i[\mathbf{x}^b(t_i), \boldsymbol{\theta}^b] - \frac{\partial \mathbf{h}_i[\mathbf{x}(t_i), \boldsymbol{\theta}]}{\partial \mathbf{x}(t_i)} \boldsymbol{\Phi} \mathbf{z}(t_i) - \frac{\partial \mathbf{h}_i[\mathbf{x}(t_i), \boldsymbol{\theta}]}{\partial \boldsymbol{\theta}} \boldsymbol{\Phi}_{\eta} \boldsymbol{\eta} \right], \tag{5.26}$$

where $\underline{\boldsymbol{\lambda}}$ represents a vector of the reduced adjoint state and $\underline{\boldsymbol{\lambda}}(t_{N+1}) = \mathbf{0}$. The objective function for the reduced-order model is

$$\hat{J}(\boldsymbol{\eta}, \mathbf{z}(t_i)) = \frac{1}{2} \sum_{i=1}^{N_o} \left[\mathbf{d}(t_i) - \mathbf{h}_i[\mathbf{x}^b(t_i), \boldsymbol{\theta}^b] - \frac{\partial \mathbf{h}_i[\mathbf{x}(t_i), \boldsymbol{\theta}]}{\partial \mathbf{x}(t_i)} \boldsymbol{\Phi} \mathbf{z}(t_i) - \frac{\partial \mathbf{h}_i[\mathbf{x}(t_i), \boldsymbol{\theta}]}{\partial \boldsymbol{\theta}} \boldsymbol{\Phi}_{\eta} \boldsymbol{\eta} \right]^T$$
$$\mathbf{R}_i^{-1} \left[\mathbf{d}(t_i) - \mathbf{h}_i[\mathbf{x}^b(t_i), \boldsymbol{\theta}^b] - \frac{\partial \mathbf{h}_i[\mathbf{x}(t_i), \boldsymbol{\theta}]}{\partial \mathbf{x}(t_i)} \boldsymbol{\Phi} \mathbf{z}(t_i) - \frac{\partial \mathbf{h}_i[\mathbf{x}(t_i), \boldsymbol{\theta}]}{\partial \boldsymbol{\theta}} \boldsymbol{\Phi}_{\eta} \boldsymbol{\eta} \right]$$
$$+ \frac{1}{2} \left[\boldsymbol{\theta}_{init} - \boldsymbol{\theta}^b - \boldsymbol{\Phi}_{\eta} \boldsymbol{\eta} \right]^T \mathbf{R}_b^{-1} \left[\boldsymbol{\theta}_{init} - \boldsymbol{\theta}^b - \boldsymbol{\Phi}_{\eta} \boldsymbol{\eta} \right], \tag{5.27}$$

and the gradient is

$$\left(\frac{d\hat{J}}{d\boldsymbol{\eta}} \right)^T = - \sum_{i=1}^{N} \left(\mathbf{N}_i^{\theta} \right)^T \underline{\boldsymbol{\lambda}}(t_i) - \boldsymbol{\Phi}_{\eta}^T \mathbf{R}_b^{-1} \left[\boldsymbol{\theta}_{init} - \boldsymbol{\theta}^b - \boldsymbol{\Phi}_{\eta} \boldsymbol{\eta} \right] -$$
$$\sum_{i=1}^{N_o} \left(\frac{\partial \mathbf{h}_i[\mathbf{x}(t_i), \boldsymbol{\theta}]}{\partial \boldsymbol{\theta}} \boldsymbol{\Phi}_{\eta} \right)^T \mathbf{R}_i^{-1} \left[\mathbf{d}(t_i) - \mathbf{h}_i[\mathbf{x}^b(t_i), \boldsymbol{\theta}^b] - \right. \tag{5.28}$$
$$\left. \frac{\partial \mathbf{h}_i[\mathbf{x}(t_i), \boldsymbol{\theta}]}{\partial \mathbf{x}(t_i)} \boldsymbol{\Phi} \mathbf{z}(t_i) - \frac{\partial \mathbf{h}_i[\mathbf{x}(t_i), \boldsymbol{\theta}]}{\partial \boldsymbol{\theta}} \boldsymbol{\Phi}_{\eta} \boldsymbol{\eta} \right].$$

If not available, we can approximate the partial derivatives of the observation model in the same manner as we did for the forward model, namely

$$\frac{\partial \mathbf{h}_i[\mathbf{x}(t_i), \boldsymbol{\theta}]}{\partial \mathbf{x}(t_i)} \boldsymbol{\phi}_j \approx \frac{\mathbf{h}_i[\mathbf{x}(t_i) + \epsilon \boldsymbol{\phi}_j, \boldsymbol{\theta}] - \mathbf{h}_i[\mathbf{x}(t_i), \boldsymbol{\theta}]}{\epsilon} \tag{5.29}$$

and

$$\frac{\partial \mathbf{h}_i[\mathbf{x}(t_i), \boldsymbol{\theta}]}{\partial \boldsymbol{\theta}} \boldsymbol{\phi}_{\eta}^j \approx \frac{\mathbf{h}_i[\mathbf{x}(t_i), \boldsymbol{\theta} + \epsilon_{\theta} \boldsymbol{\phi}_{\eta}^j] - \mathbf{h}_i[\mathbf{x}(t_i), \boldsymbol{\theta}]}{\epsilon_{\theta}}. \tag{5.30}$$

The minimization is performed using a quasi-Newton optimization where the Hessian of the objective function is updated using the BFGS method. No scaling of the parameters in BFGS is used because after reparameterization they are of the same order of magnitude. The minimization algorithm terminates when

- the objective function hardly changes, i.e.

$$\frac{|\hat{J}(\boldsymbol{\eta}_{k+1}) - \hat{J}(\boldsymbol{\eta}_k)|}{max\{|\hat{J}(\boldsymbol{\eta}_{k+1})|, 1\}} < \epsilon_J, \tag{5.31}$$

and

- the estimates hardly changes, i.e.

$$\frac{|\boldsymbol{\eta}_{k+1} - \boldsymbol{\eta}_k|}{max\{|\boldsymbol{\eta}_{k+1}|, 1\}} < \epsilon_\eta, \tag{5.32}$$

where $\epsilon_J = 10^{-4}$ and $\epsilon_\eta = 10^{-3}$ and $| \cdot |$ is L_2 norm. The alternative methods can also be used to solve the minimization of the objective function $\hat{J}$ because this part of the algorithm is not computationally expensive. One can even write a formula for the Hessian calculation and use a Newton's like update. We use the BFGS update because the same method is used in the history matching routine with the original reservoir model and its adjoint. Additionally, it converges in less iterations than the number of parameters plus one (Fletcher (1991)).

5.2.5 Improving parameters in high-order space

The solution of the inner loop is an optimum for the reduced-order linearized system but not necessarily for the original one. Therefore, we check the value of the original objective function for the new parameters $\boldsymbol{\theta}^b_{k+1} = \boldsymbol{\theta}^b + \boldsymbol{\Phi}_\eta \boldsymbol{\eta}_{opt}$, where $\boldsymbol{\eta}_{opt}$ is the solution of minimization process. The process is repeated as long as the objective function satisfies

$$mN_o - 2\sqrt{2mN_o} \leq 2J(\boldsymbol{\theta}^b_{k+1}) \leq mN_o + 2\sqrt{2mN_o}, \tag{5.33}$$

where m denotes number of observations collected at each time t_i and N_o denotes the number of those time instants. This convergence criterion is a consequence of the assumption that the model is linear and $\boldsymbol{\theta}$ has a normal distribution. Moreover, model errors are not considered. In this case the minimum of objective function J has a χ^2 distribution with mN_o degrees of freedom (see Tarantola (1987)). As stated in Oliver et al. (2008) it is reasonable to assume that this fact also approximately applies in the nonlinear case. The constraint (5.33) tells us that the objective function should fall in the interval of two standard deviations ($2\sqrt{2mN_o}$) distance from the mean expected value (mN_o). If J has a χ^2 distribution with mN_o degrees of freedom, then as mN_o tends to infinity, the distribution of J tends to a standard normal distribution. This property justifies the interval of two standard deviations, which implies that if we would repeat the numerical experiment with different random measurement errors drawn from the same distribution, about 95% of the results would be within two standard deviations away from the mean. If the objective function does not obey the stopping criterion, then the process is iterated with a new background parameter $\boldsymbol{\theta}^b_{k+1}$. Since a relationship exists between patterns and model parameterization, whenever the background parameter is changed, a new background state and a new set of patterns need to be identified. Thereafter, a new reduced-order model is built and the inner loop is performed again. The advantage of this criterion compared to alternative criteria is that it works even if only a single outer loop is performed. The history matching algorithm described in this section has been summarized by the flow chart in Fig. 5.1.

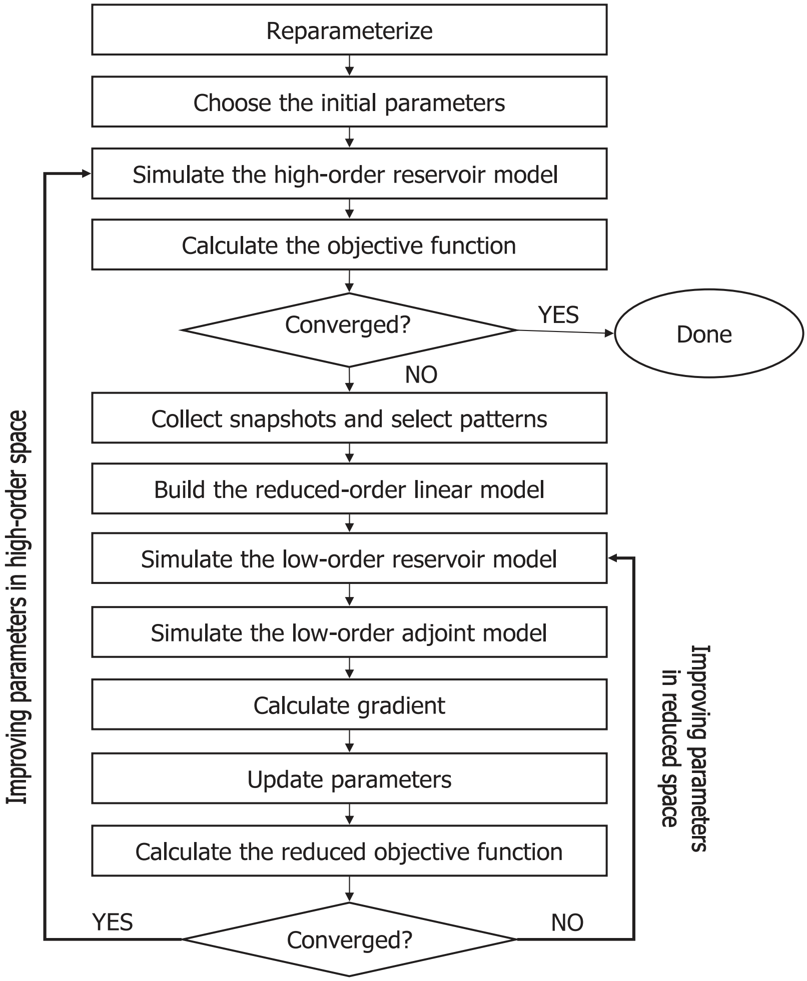

Reparameterize
Choose the initial parameters
Simulate the high-order reservoir model
Calculate the objective function
Converged?
YES
Done
NO
Collect snapshots and select patterns
Build the reduced-order linear model
Simulate the low-order reservoir model
Simulate the low-order adjoint model
Calculate gradient
Update parameters
Calculate the reduced objective function
YES
Converged?
NO
Improving parameters in high-order space
Improving parameters in reduced space

Figure 5.1: Model-reduced gradient-based history matching algorithm.

5.2.6 Computational complexity

The overall computational costs of the model-reduced approach consist of two parts:
(i) **Preprocessing costs of constructing the reduced-order model:**

- Cost of generating representative snapshots which span a large portion of the variability of the permeability field:
  The generation of 'good' snapshots is an important part of the POD and BPOD methods. Our strategy is as follows: First, we check the range of the permeability values that are present in the ensemble of prior realizations.

Then, we choose a number of permeability patterns, which we change during one simulation. We can reduce the CPU time by changing a few patterns per simulation during initial outer loops. The size of the perturbation depends on the number of outer iterations as well, namely, we perturb with the maximum range in first loop and we reduce it in subsequent loops. We have not yet established a fully systematic approach which guarantees the best compromise between efficiency and representativeness of the snapshots but from our experience the described approach works well.

- Cost of solving the reduced eigenvalue problem:
  This cost is equivalent to the cost of a Singular Value Decomposition of matrix $\mathbf{X}$ in case of POD method and of matrix $\mathbf{Y}^T\mathbf{X}$ in case of BPOD method. Since the number of snapshots can be kept low, this cost is low. Moreover, we may use a Lanczos procedure to only compute the n_{red} largest singular values.

- Cost of approximating partial derivatives along patterns:
  This is computationally the most expensive part of the proposed procedure and its CPU time is comparable to the time of $p_{red} + n_{red}$ simulations of the high-order model.

(ii) **Cost of solving the reduced system and simplified history matching problem:**

- Cost of solving the reduced system:
  The cost of solving a system of n_{red} linear equations is equal to $O\left((n_{red})^2\right)$, which can be neglected in our case.

- Cost of the optimization procedure:
  In case of a quadratic objective function the quasi-Newton routine iterates approximately $p_{red} + 1$ times; see Fletcher (1991).

5.2.7 Additional issues

Pressure and saturation variables have a totally different physical behavior, a different order of numerical values and generally a different variability in their values. The POD procedure favors variables which values show greater variability; this can have negative consequences on the pattern identification. If all variables are of the same kind, then the data can be preprocessed before the POD procedure by normalizing them to have unit variance. Therefore, we decided to collect snapshots for pressures and saturations separately, following Van Doren et al. (2004). Instead of solving one eigenvalue problem to obtain the POD matrix $\mathbf{\Phi}$, we solve two separate eigenvalue problems (one for the matrix of pressure snapshots $\mathbf{X}_p \in \mathbb{R}^{\frac{n}{2}\times s}$ and one for the matrix of saturation snapshots $\mathbf{X}_s \in \mathbb{R}^{\frac{n}{2}\times s}$). It yields two different POD matrices, $\mathbf{\Phi}_p \in \mathbb{R}^{\frac{n}{2}\times n_p}$ and $\mathbf{\Phi}_s \in \mathbb{R}^{\frac{n}{2}\times n_s}$ for pressure, and saturation, respectively, and the final POD matrix $\mathbf{\Phi}$ is formed as

$$\mathbf{\Phi} = \begin{bmatrix} \mathbf{\Phi}_p & \mathbf{0}_{\frac{n}{2}\times n_s} \\ \mathbf{0}_{\frac{n}{2}\times n_p} & \mathbf{\Phi}_s \end{bmatrix} \in \mathbb{R}^{n\times n_{red}},$$

where n_p and n_s are number of patterns for pressure, and saturation, respectively, and $n_{red} = n_p + n_s$. The same separation is applied when the snapshots of the adjoint model are collected in the BPOD method. By this separation we assume that pressure and saturation are not correlated, which is not the case, but it is a way to solve the problem associated with different scales of state variables.

5.3 Gradient-based adjoint-based history matching algorithms

The history matching problem can be solved using iterative gradient-based minimization methods, where each iteration consists of the following steps:

- determine a direction g_k in parameter space, which leads to a lower objective function,i.e. $g_k = -\frac{dJ}{d\boldsymbol{\theta}}$,
- find the size of the step length $\boldsymbol{\alpha}_k$ along that direction, such that it minimizes $J(\boldsymbol{\theta}_k + \boldsymbol{\alpha}_k g_k)$,
- set the new estimate $\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \boldsymbol{\alpha}_k g_k$.

The down-hill direction is given by the opposite gradient of the objective function with respect to the parameters and it can be obtained by reformulating the minimization problem as an unconstrained one, through inclusion of the constraints with the aid of Lagrange multipliers; see Chapter 2 and e.g. Oliver et al. (2008). This leads to:

$$\begin{aligned}\left(\frac{dJ}{d\boldsymbol{\theta}}\right)^T &= -\sum_{i=1}^{N}\left(\frac{\partial \mathbf{f}_i[\mathbf{x}(t_{i-1}),\boldsymbol{\theta}]}{\partial\boldsymbol{\theta}}\right)^T \boldsymbol{\lambda}(t_i) - \mathbf{R}_b^{-1}(\boldsymbol{\theta}_{init} - \boldsymbol{\theta}) - \\ &\sum_{i=1}^{N_o}\left(\frac{\partial \mathbf{h}_i[\mathbf{x}(t_i),\boldsymbol{\theta}]}{\partial\boldsymbol{\theta}}\right)^T \mathbf{R}_i^{-1}\left[\mathbf{d}(t_i) - \mathbf{h}_i[\mathbf{x}(t_i),\boldsymbol{\theta}]\right],\end{aligned}$$

where $\boldsymbol{\lambda}$ represents a vector of adjoint states (or Lagrangian multipliers) which satisfies the following equation:

$$\begin{aligned}\boldsymbol{\lambda}(t_i) &= \left(\frac{\partial \mathbf{f}_{i+1}[\mathbf{x}(t_i),\boldsymbol{\theta}]}{\partial \mathbf{x}(t_i)}\right)^T \boldsymbol{\lambda}(t_{i+1}) + \\ &\left(\frac{\partial \mathbf{h}_i[\mathbf{x}(t_i),\boldsymbol{\theta}]}{\partial \mathbf{x}(t_i)}\right)^T \mathbf{R}_i^{-1}\left[\mathbf{d}(t_i) - \mathbf{h}_i[\mathbf{x}(t_i),\boldsymbol{\theta}]\right]\end{aligned} \tag{5.34}$$

for $i = N, ..., 1$ with an end condition $\boldsymbol{\lambda}(t_{N+1}) = \mathbf{0}$. The size of the step length $\boldsymbol{\alpha}_k$ is usually determined by line search methods. In an exact line search method the optimal α_k is found by solving an optimization problem

$$\min_{\alpha_k} J(\boldsymbol{\theta}_k + \boldsymbol{\alpha}_k g_k), \tag{5.35}$$

which requires multiple runs of the simulator. Since this can be computationally costly, typically an exact line search is not used and instead conditions which guarantee a sufficient decrease in the objective function at each iteration are applied,

e.g. the strong Wolfe conditions (see e.g. Oliver et al. (2008)).
The algorithm proceeds until the minimum of the objective function is found or certain stopping criteria are satisfied. When the problem is not well scaled, the steepest descent method may converge very slowly. This can be improved by using Newton-like methods, where the exact Hessian matrix, i.e. the second derivative of the objective function with respect to parameters, or its approximation is used. Then the iterative steps become:

- determine a direction g_k in parameter space, which leads to a lower objective function, i.e. $g_k = -\frac{dJ}{d\boldsymbol{\theta}}$
- approximate the Hessian matrix $\mathbf{H}_k(\boldsymbol{\theta}_k)$,
- find the size of the step length $\boldsymbol{\alpha_k}$ along that direction, such that it minimizes $J(\boldsymbol{\theta}_k + \boldsymbol{\alpha}_k \mathbf{H}_k^{-1} g_k)$,
- set the new estimate $\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \boldsymbol{\alpha}_k \mathbf{H}_k^{-1} g_k$.

The Hessian matrix can be approximated by using a Gauss-Newton Hessian matrix or alternatively, using an iterative manner, such as the BFGS formula (see Fletcher (1991) and Oliver et al. (2008) for more details). In this thesis the BFGS formula is used. In case of a reparametrized parameter vector, the gradient is given by

$$\left(\frac{dJ}{d\boldsymbol{\eta}}\right)^T = \left(\frac{dJ}{d\boldsymbol{\theta}}\right)^T \boldsymbol{\Phi}_\eta \tag{5.36}$$

and the same iterative steps are applicable.

5.4 Simultaneous Perturbation Stochastic Approximation algorithm

In the Simultaneous Perturbation Stochastic Approximation (SPSA) algorithm proposed by Spall (see Spall (1992), Spall (2000)), one minimizes the objective function J in the following iterative steps:

- determine a stochastic direction $\mathbf{g}_k$ in parameters space, which leads to a lower objective function,
- find the size of the step length $\boldsymbol{\alpha_k}$ along that direction,
- set the new estimate $\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \boldsymbol{\alpha}_k \mathbf{g}_k$.

Let us define $\mathbf{g}_k = -\hat{\mathbf{g}}_k(\boldsymbol{\theta}_k)$, where $\hat{\mathbf{g}}_k(\boldsymbol{\theta}_k)$ is a stochastic approximation of the gradient of the objective function with respect to $\boldsymbol{\theta}$ evaluated at the old iterate, $\boldsymbol{\theta}_k$, and a realization of this stochastic gradient is calculated by

$$\hat{g}_k(\boldsymbol{\theta}_k) = \frac{J(\boldsymbol{\theta}_k + c_k \boldsymbol{\Phi}_\eta \boldsymbol{\Delta}_k) - J(\boldsymbol{\theta}_k - c_k \boldsymbol{\Phi}_\eta \boldsymbol{\Delta}_k)}{2c_k} \times \boldsymbol{\Delta}_k^{-1}, \tag{5.37}$$

where c_k is a positive coefficient, $\mathbf{\Delta}_k := \left[\Delta_{k,1}, \Delta_{k,2}, \ldots, \Delta_{k,p_{red}}\right]^T \in \mathbb{R}^{p_{red}}$ is a simultaneous perturbation vector which elements, $\Delta_{k,i}$, are independent samples from the symmetric ±1 Bernoulli distribution (i.e. $\Delta_{k,i}$ can take only the values $+1$ or -1) and $\mathbf{\Delta}_k^{-1} := \left[\frac{1}{\Delta_{k,1}}, \frac{1}{\Delta_{k,2}}, \ldots, \frac{1}{\Delta_{k,p_{red}}}\right]^T$. In the SPSA method it is required that the random vector $\mathbf{\Delta}$ is a vector of p_{red} mutually independent mean-zero random variables symmetrically distributed around zero satisfying the condition that the $E\left|\Delta_{k,i}^{-1}\right|$ and $\left|\Delta_{k,i}\right|$ are bounded, where E stands for an expected value. The Bernoulli distribution obeys those conditions. If additionally for almost all $\boldsymbol{\theta}_k$ the third derivative of the objective function exists in an open neighborhood of $\boldsymbol{\theta}_k$ and is bounded, then the bias in the stochastic gradient goes to zero as the number of iterations goes to infinity.

Parameters α_k and c_k have to be chosen according to the guideline presented by Spall in (Spall (1992), Spall (2000)). According to his experience and theoretical derivations, they should be chosen as

$$\alpha_k = \frac{a}{(A + k + 1)^\alpha} \tag{5.38}$$

and

$$c_k = \frac{c}{(k+1)^\gamma}, \tag{5.39}$$

where A, a, c, α and γ are positive real numbers, $0 < \alpha \leq 1$ and $\alpha - \gamma > 0.5$. Those parameters are case dependent, but it is recommended to start with $\alpha = 0.602$, $\gamma = 0.101$ and A equal to $5 - 10\%$ of the maximum number of allowed iterations. The SPSA method was designed for noisy objective functions and for those problems c is recommended to be equal to the standard deviation of the noise in J. In this application c is chosen as a small ratio of the prior parameter value, while a is chosen based on at least one gradient sample.

The SPSA method requires only two evaluations of the objective function at each iteration (i.e. $J(\boldsymbol{\theta}_k + c_k \mathbf{\Phi}_\eta \mathbf{\Delta}_k)$ and $J(\boldsymbol{\theta}_k - c_k \mathbf{\Phi}_\eta \mathbf{\Delta}_k)$) and in case that $\mathbf{\Delta}$ is only a single value, Eq. 5.37 represents a central difference approximation of the gradient. In case of reparametrized parameter space the SPSA gradient, $\hat{g}_k$, is created by simultaneous perturbation of all model patterns.

Since in case of nonlinear problems the steepest descent method is known to converge very slowly, Spall (2000) proposed a modification to his approach, called adaptive SPSA or second order SPSA,

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \boldsymbol{\alpha}_k \mathbf{H}_k^{-1} g_k, \tag{5.40}$$

where the Hessian matrix is approximated by the following equation:

$$\mathbf{H}_k = \frac{k}{k+1}\mathbf{H}_{k-1} + \frac{1}{k+1}\hat{\mathbf{H}}_k, \tag{5.41}$$

and

$$\mathbf{H}_k^{-1} = \mathbf{f}_k(\mathbf{H}_k), \tag{5.42}$$

where $\mathbf{f}_k$ is a mapping that represents the modification of the matrix $\mathbf{H}_k$ to ensure that the resulting $\mathbf{H}_k$ is positive definite, and $\hat{\mathbf{H}}_k$ is given by

$$\hat{\mathbf{H}}_k = \frac{1}{2}\left[\boldsymbol{\Delta}_k^{-1}\delta\mathbf{g}_k^T\left(\boldsymbol{\Delta}_k^{-1}\delta\mathbf{g}_k^T\right)^T\right], \tag{5.43}$$

where

$$\delta\mathbf{g}_k = \frac{1}{2c_k}\left[g_k^+(\boldsymbol{\theta}_k + c_k\boldsymbol{\Phi}_\eta\boldsymbol{\Delta}_k) - g_k^-(\boldsymbol{\theta}_k - c_k\boldsymbol{\Phi}_\eta\boldsymbol{\Delta}_k)\right], \tag{5.44}$$

with

$$\mathbf{g}_k^+(\boldsymbol{\theta}_k + c_k\boldsymbol{\Phi}_\eta\boldsymbol{\Delta}_k) = \frac{J(\boldsymbol{\theta}_k + c_k\boldsymbol{\Phi}_\eta\boldsymbol{\Delta}_k + \tilde{c}_k\tilde{\boldsymbol{\Delta}}_k) - J(\boldsymbol{\theta}_k + c_k\boldsymbol{\Phi}_\eta\boldsymbol{\Delta}_k)}{\tilde{c}_k} \times \tilde{\boldsymbol{\Delta}}_k^{-1}$$

and

$$\mathbf{g}_k^-(\boldsymbol{\theta}_k + c_k\boldsymbol{\Phi}_\eta\boldsymbol{\Delta}_k) = \frac{J(\boldsymbol{\theta}_k + c_k\boldsymbol{\Phi}_\eta\boldsymbol{\Delta}_k - \tilde{c}_k\tilde{\boldsymbol{\Delta}}_k) - J(\boldsymbol{\theta}_k + c_k\boldsymbol{\Phi}_\eta\boldsymbol{\Delta}_k)}{\tilde{c}_k} \times \tilde{\boldsymbol{\Delta}}_k^{-1},$$

where $\tilde{c}_k$ is a perturbation size, $\tilde{\boldsymbol{\Delta}}_k = \left[\tilde{\Delta}_{k,1}, \tilde{\Delta}_{k,2}, \ldots, \tilde{\Delta}_{k,p_{red}}\right]^T$ is a simultaneous perturbation vector which elements, $\tilde{\Delta}_{k,i}$, are independent samples from the ± 1 Bernoulli distribution, and $\tilde{\boldsymbol{\Delta}}_k^{-1} = \left[\frac{1}{\tilde{\Delta}_{k,1}}, \frac{1}{\tilde{\Delta}_{k,2}}, \ldots, \frac{1}{\tilde{\Delta}_{k,p_{red}}}\right]^T$. g_k^+ and g_k^- are one-sided gradient approximations which reduces the number of model simulations compared to two-sided approximation.

Another way to improve the convergence of the SPSA method is by using an average of few stochastic gradients, i.e.

$$\mathbf{g}_k = \frac{1}{M}\sum_{i=1}^{M}\hat{\mathbf{g}}_k(\boldsymbol{\theta}_k), \tag{5.45}$$

where M is the number of gradient samples. This averaged gradient should be a better approximation of the true gradient, and therefore should result in faster convergence of the method.

5.5 Gradient-based finite-difference algorithm

This procedure has exactly the same iterative steps as the classical adjoint-based algorithm, but the calculation of the gradient is done according to the formula:

$$\frac{dJ}{d\eta_j} \approx \frac{J(\boldsymbol{\theta}_k + c_k\boldsymbol{\Phi}_\eta\eta_j) - J(\boldsymbol{\theta}_k)}{c_k}, \tag{5.46}$$

where c_k is the size of the perturbation. It is a one-sided gradient approximation and it requires $p_{red} + 1$ model simulations for a single gradient calculation. Since the history matching problems can take between 50-100 iterations to converge, this approach even after reparameterization can be computationally very demanding. In this method the iterative steps are:

- approximate a direction g_k, which leads to a lower objective function, by using Eq. 5.46, i.e. $g_k = -\frac{dJ}{d\eta_j}$,
- approximate the Hessian matrix $\mathbf{H}_k(\boldsymbol{\theta}_k)$,
- find the size of the step length $\boldsymbol{\alpha}_k$, such that it minimizes $J(\boldsymbol{\theta}_k + \boldsymbol{\alpha}_k \mathbf{H}_k^{-1} g_k)$,
- set the new estimate $\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \boldsymbol{\alpha}_k \mathbf{H}_k^{-1} g_k$.

The Hessian matrix can be approximated by the BFGS formula.

5.6 Parallelization

When frequent model evaluations are required parallel computing is a very powerful form of computation. However, parallel computing is beneficial only if many calculations can be carried out simultaneously. Parallelism has been employed for many years, and nowadays clusters of the multiple processors are available to engineers. The communication and synchronization between the different subtasks, which are typically one of the greatest obstacles to getting good parallel program performance, are not an issue in the methods presented here. A parallel computer program for those methods is no more difficult to write than a sequential one. Obviously, for the presented methods, each node of the cluster has to contain the necessary reservoir simulation software.

In the model-reduced method, the simulation of the model trajectory around which the system is linearized can be computed in parallel with the snapshot simulations. It means that the snapshot preparation can be performed in the time of one model simulation. Then, just as in the sequential computer program, the POD method is performed and the patterns are identified. After this step the parallel computations can start again. Each sensitivity along each pattern is computed using information about the background trajectory and the selected patterns. It means that the all sensitivity calculation can be computed in a single time step evaluation, assuming that the sensitivity at each time step along each pattern is calculated on a separate processor. This requires too many nodes in the cluster; therefore we assume that only the sensitivity with respect to each pattern is computed simultaneously, which requires the time of about one full simulation. Additional single run is required to verify the optimal result of the inner loop. Summarizing, one outer iteration of the model-reduced approach can be performed in the time of three model simulations, if the required number of nodes is available. Furthermore, if the reduced order approach requires the repetition of the outer loop, then a total computational time is equal to three times the number of outer loops simulations. The number of nodes, necessary to accomplish the calculations of the model-reduced method in about three simulations, depends strongly on the number of parameters and on the number of selected patterns. For example, estimating 50 parameters for which 25 snapshot simulations are performed and 60 patterns are selected, the maximum required number of nodes is 110 (in the first part of the calculations we require 1+25 nodes in the cluster to generate snapshots, but in the second part of the calculations we need 50+60 nodes to calculate the sensitivities, thus the maximum of these two values gives the required number of nodes). If the model-reduced method requires, for example, 3 outer iterations, then all calculations can be accomplished in 9 $(= 3 \cdot 3)$ model simulations. If the maximum number of nodes in the cluster is e.g. 55, then the model-reduced method requires 12 $(= 3 \cdot 4)$ model simulations.

Now we compare the potential of the parallelization capability of the model reduced approach to the finite-difference and SPSA methods. In the finite difference method the simulation of the model with the reference parameter and all its perturbations can be computed simultaneously. This means that one gradient computation requires the time of one simulation. The optimization technique is an iterative routine which uses the gradient at each iteration. The total cost of the parallelized finite difference approach depends therefore on the number of iterations required to converge, i.e. it is about the time required to simulate the model as many times as the number of iterations required in the optimization routine. The number of nodes necessary in the finite-difference approach is equal to the number of parameters plus one. If the finite-difference method requires, for example, 50 iterations to converge, then this routine costs 51 $(= 50 \cdot 1 + 1)$ model simulations.

The simultaneous perturbation method becomes less attractive than the finite difference method if parallel computing is involved. This method approximates the gradient by making finite difference approximation simultaneously for all parameters at once. The calculated gradient may be less accurate than the finite difference gradient, hence it may require more iterations to converge than the finite difference approach. The gradient can be calculated in the time of one simulation and the total computational time is equal to the number of iterations of the optimization routine.

6 CHAPTER

Application to history matching problems

The model-reduced approaches described in Chapter 5 were compared with two other gradient-based history matching algorithms: a classical adjoint-based history matching algorithm and a gradient-based finite-difference algorithm. The first comparisons were done on a very simple two-phase 2D reservoir. For this case study, we found that model-reduced approaches were much more efficient than the finite-difference based approach. Furthermore, they showed to be at most twice less efficient than the adjoint-based method. The second experiment was performed for a water-flood 3D reservoir with a channelized permeability field. A comparison with an adjoint-based history matching procedure showed that the model-reduced approach produces a comparable quality of history matches and predictions. For this problem, the computational efficiency of the model-reduced approach was at most six times less efficient than of an adjoint-based approach, but at least twice more efficient than of the finite-difference based approach.

6.1 Application to a simple 2D model

6.1.1 Model settings

In the numerical experiments presented in this section the model-reduced approaches, the classical adjoint-based approach and the finite-difference approach were used to estimate the permeability field of a simple synthetic 2D reservoir model, which describes iso-thermal slightly compressible two-phase (oil-water) flow in a five-spot well configuration (see Fig. 6.1). The SPSA method described in the previous chapter showed very slow convergence compared to the model-reduced methods, and is therefore not presented.

All experiments were performed using a simple in-house 2D two-phase reservoir simulator SIMSIM (described in Jansen (2011b)). All the model parameters, except the permeability field, were assumed to be known. The uncertainty of the permeability field was expressed by an ensemble of model realizations. The "true" field was chosen as one realization out of a given ensemble of 1000 members. We performed a so-called "twin experiment" in which synthetic noisy production data was used. The synthetic data was created using the true model and by adding random errors, which were assumed to be Gaussian distributed random variables with zero mean and known variance equal to a fixed percentage of the true data. Additionally, all observations were assumed to be independent.

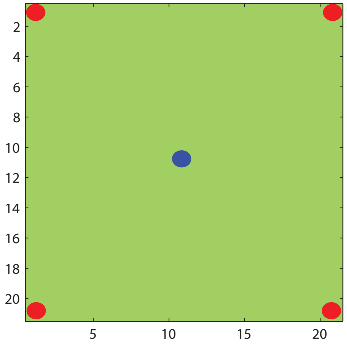


Figure 6.1: The well locations in the reservoir used in this section. The red dots correspond to the production wells and the blue dot to the injection well.

- *Model assumptions:*
  We assumed that the reservoir operates under water flooding condition. During that process dispersive and diffusive effects take place. The diffusive effect of capillary pressure plays a role during the displacement process on a relatively small length scale and is usually much smaller than the dispersion. The dispersive effect of geological heterogeneities takes place and should be modeled using a velocity-dependent dispersion tensor, see Russell and Wheeler (1983). Additionally to diffusion and dispersion caused by physical phenomena, diffusion caused by numerical errors of the discretization occurs. Since in many cases this numerical diffusion is of the same order of magnitude as the physical diffusion and dispersion, we neglected the capillary forces and dispersion all together. We assumed that the effects of gravity can be neglected, that the permeability is isotropic, and that all parameters are pressure-independent.

- *Fluids properties:*
  The relative permeabilities which represent the additional resistance to flow of a phase caused by the presence of the other phase has a Corey model. Corey exponents for oil and water are equal 2, and the end point relative permeabilities for oil and water equals 0.9 and 0.6, respectively. The residual oil saturation and the connate water saturation equal 0.2. The oil and water viscosity equals $0.5 \cdot 10^{-3}\ Pa \cdot s$ and $1.0 \cdot 10^{-3}\ Pa \cdot s$, respectively. The water, oil and rock compressibility equal $1.0 \cdot 10^{-9}\ 1/Pa$. The residual oil saturation and connate water saturation equal 0.2.

- *Reservoir geometry:*
  The total size of the reservoir is $700m \times 700m \times 2m$ and it is divided into $21 \times 21 \times 1$ uniform Cartesian grid blocks ($33.33m \times 33.33m \times 2m$).

- *Reservoir properties:*
  The porosity is assumed to be homogeneous and equals 0.3. The "true" permeability fields on a log scale are shown in Fig. 6.6 and 6.30, and are assumed to be unknown.

- *Initial conditions:*
  The initial reservoir pressures is $30 \cdot 10^6 Pa$ and the initial water saturation is taken as connate water saturation.

- *Well locations and constraints:*
  Water is injected at a rate of 0.003 m^3/s in a vertical injector placed in the middle of the reservoir, while four vertical producers at the corners are operated at constant pressures of $25 \cdot 10^6 Pa$. The distribution of the wells is depicted in Fig. 6.1.

6.1.2 History matching settings

All the assimilation experiments were performed with the same settings. Namely, the assimilation period was chosen to be 250 days and observations were taken from 4 production wells and one injection well after each 30 days of production, resulting in 40 observations. During this period no water was observed in the production wells. In the injection well the bottom hole pressure was measured and was assumed to have 10% error. In the production wells the total production rates were measured and were assumed to have 5% error. The errors were generated from Gaussian distributions with zero mean and a standard deviation equal to 10% and 5% of actual data in the injection and production wells, respectively. We assumed that distinct observations were affected by physically independent noise, resulting in a diagonal covariance matrix for the observation errors. The uncertainty of the permeability field was expressed by an ensemble of 1000 realizations. Each realization was a linear combination of the eigenvectors of the permeability covariance matrix conditioned to the hard data. The coefficients of this linear expansion were sampled from Gaussian distributions with zero means and standard deviations equal to the corresponding eigenvalues. The hard data was assumed to be perfect and therefore, the values of permeability were assumed to be known at well locations, and were not estimated. The true permeability field was removed from the given ensemble and the remaining ensemble members were used to create the background permeability field and to estimate the background covariance matrix. The covariance matrix used to create the realizations was not available. Fig. 6.2 presents a random sample of 9 out of 999 ensemble members.

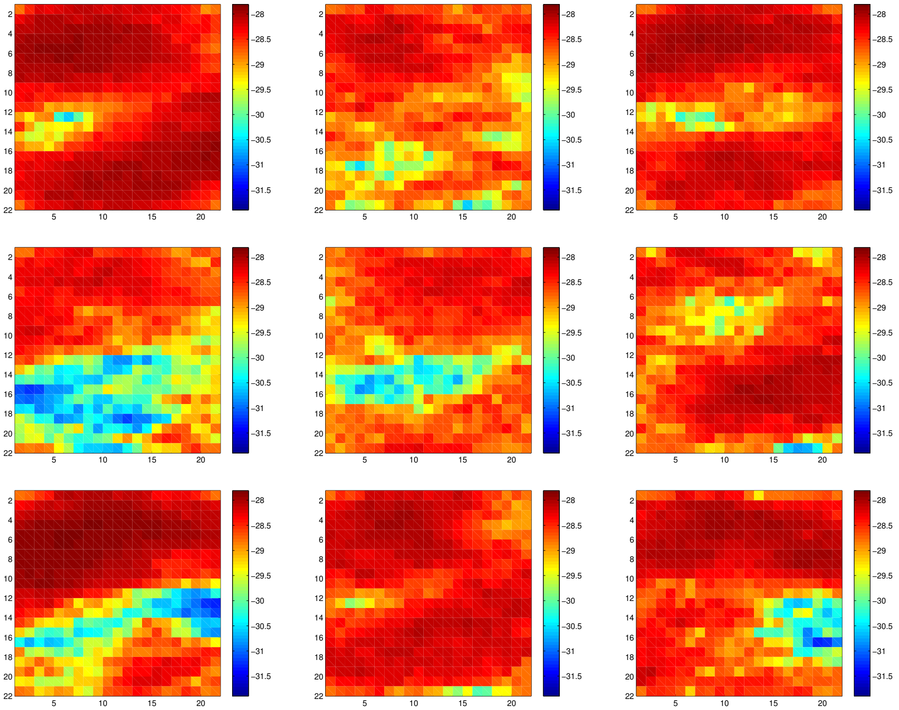


Figure 6.2: 9 realizations of the log of permeability field chosen randomly from an ensemble of 1000 realizations (units: $[m^2]$).

The natural logarithm of the permeability was reparametrized using the Karhunen-Loéve expansion. We have decided to parametrize the permeability field using 20 parameters, i.e. $\boldsymbol{\theta} = \boldsymbol{\theta}_{prior} + \boldsymbol{\Phi}_\eta \boldsymbol{\eta}$, where $\boldsymbol{\eta} = [\eta_1, \ldots, \eta_{20}]$, which gave a good reconstruction of the field. This parameterization represents 91% of the total relative importance present in the given ensemble. The chosen patterns are presented in Fig. 6.3. In Fig. 6.3 the relative importance of the patterns decreases moving from the top to the bottom of the figure starting from the left side of each row. The most important patterns represent the most common shapes which can be recognized in many realizations. For example, the first pattern is evidently present in the realizations presented in Fig. 6.4, while the second pattern can be clearly recognized in the realizations shown in Fig. 6.5. The first rows of Figs. 6.4 and 6.5 correspond to the permeability fields which have very high negative coefficients for the given patterns while the second rows correspond to the permeability fields with high positive coefficients. The patterns with lower relative importance represent more detailed aspects of the fields, and they are not as frequently present in the realizations as the patterns with the high relative importance, and they are difficult to point out only by an visual inspection.

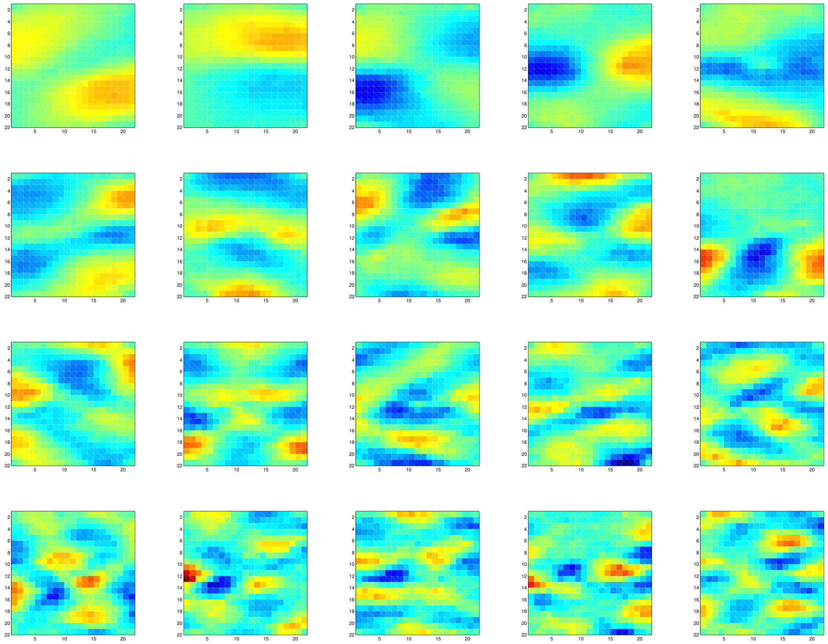


Figure 6.3: 20 patterns of the log permeability field identified in the given ensemble. The most important pattern is located in the left top corner, while the least important pattern is located in the right bottom corner (units: $[-]$).

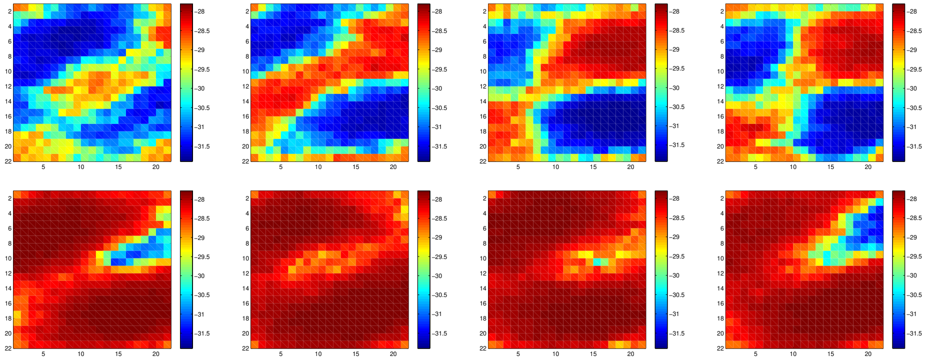


Figure 6.4: 8 realizations of the log permeability field that after reparameterization have the largest negative (the first four) or the largest positive (the last four) coefficients for the first pattern shown in Fig. 6.3 (units: $[m^2]$).

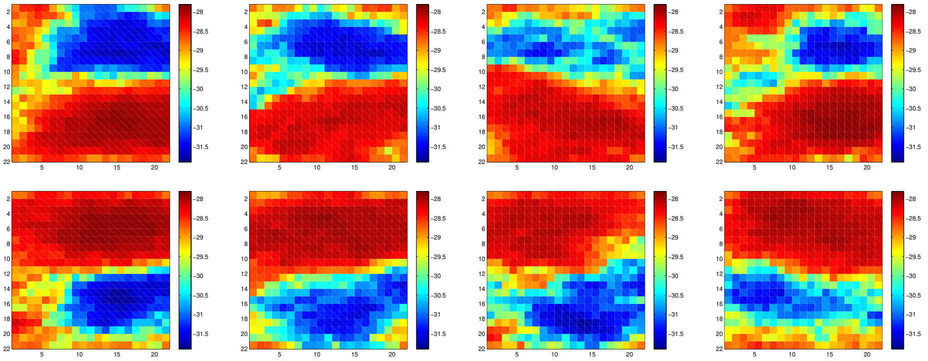


Figure 6.5: 8 realizations of the log permeability field that after reparameterization have the largest negative (the first four) or the largest positive (the last four) coefficients for the second pattern shown in Fig. 6.3 (units: $[m^2]$).

The true permeability field and its reparameterized representations are presented in Fig. 6.6. The prior permeability field (presented in Fig. 6.7) after reparameterization stays unchanged (see Eq. 5.3).

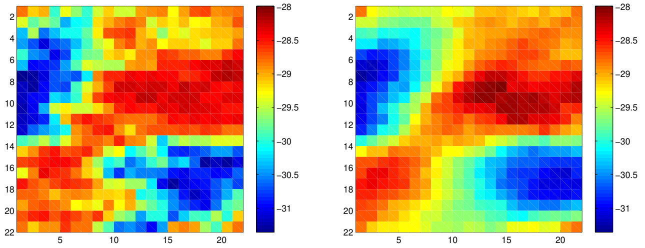


Figure 6.6: True (left) and reconstructed true (right) permeability fields $[m^2]$.

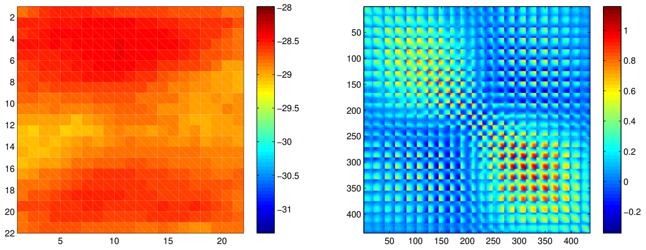


Figure 6.7: The average and covariance matrix approximations derived based on the given ensemble of 999 permeability fields.

The spatial dependencies between permeability values are described by the covariance matrix presented in Fig. 6.7. After reparameterization this covariance matrix becomes a diagonal matrix with the eigenvalues of the most important patterns on the main diagonal. The differences between original and reparametrized permeability field result in differences in the simulated saturation and pressure fields, however, these differences are very small (see Figs. 6.8 and 6.9). In case of saturation the differences are mostly visible along the fluid front.

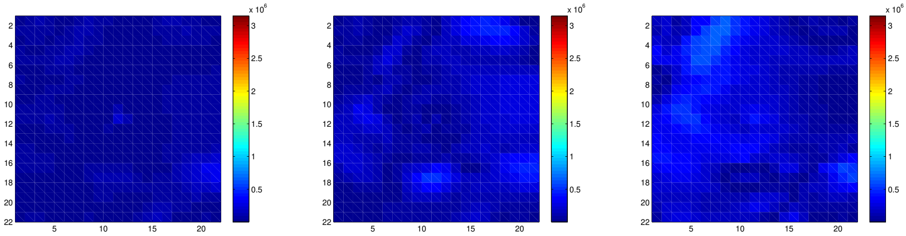


Figure 6.8: The differences (at different time instants) between the pressure field resulting from the simulation using the true permeability field and the pressure field resulting from the simulation using the reparametrized permeability field (units: $[Pa]$).

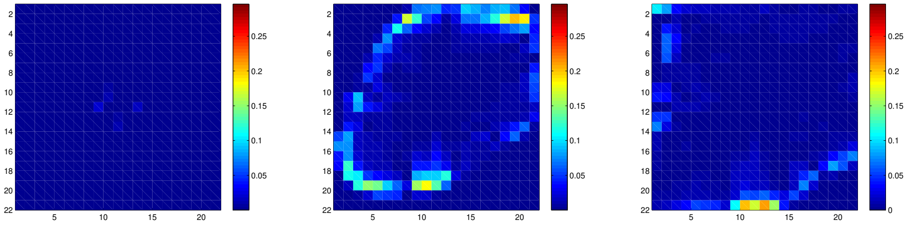


Figure 6.9: The differences (at different time instants) between the saturation field resulting from the simulation using the true permeability field and the saturation field resulting from the simulation using the reparametrized permeability fields (units: $[-]$).

Figs. 6.10, 6.11 and 6.12 present the rates of the produced water in the production wells, and bottom hole pressure in the injection well simulated with original and reparametrized permeability fields. Since the differences in the production data are small, we decided to use the chosen reparameterization.

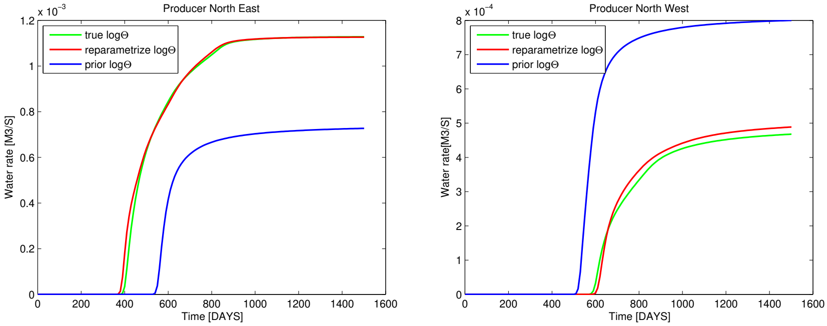


Figure 6.10: Comparison between the predictions of water breakthrough time and water flow rates for North-East and North-West production wells, located at the top of the reservoir, using the true, reconstructed and prior permeability fields.

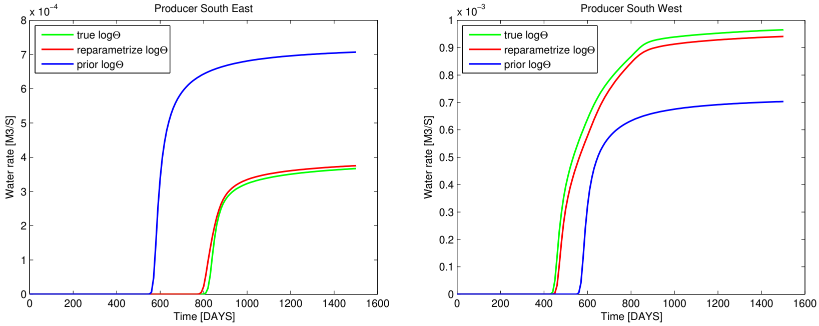


Figure 6.11: Comparison between the predictions of water breakthrough time and water flow rates for South-East and South-West production wells, located at the bottom of the reservoir, using the true, reconstructed and prior permeability fields.

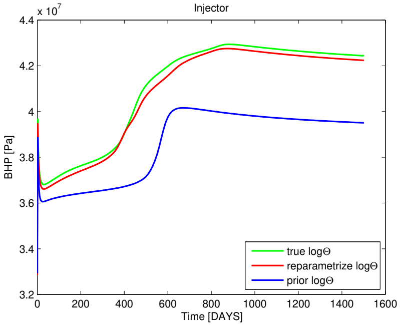


Figure 6.12: Comparison between the predictions of bottom hole pressure in the injection well, located in the middle of the reservoir, using the true, reconstructed and prior permeability field.

6.1.3 Reduced-order model settings

Pressure and saturation variables have a totally different physical behavior and generally a different variability in their values, and therefore we considered them separately. Figs. 6.15 and 6.16 present the pressure fields at different time instants, for two different permeability fields: the true field and the prior field, while Figs. 6.13 and 6.14 show the corresponding saturation fields. As noticed in Heijn et al. (2004), Van Doren et al. (2004) and Cardoso et al. (2009) the pressure behavior shows less variability than the saturation behavior. Therefore, we expect that more patterns are required to represent the saturation field than the pressure field.

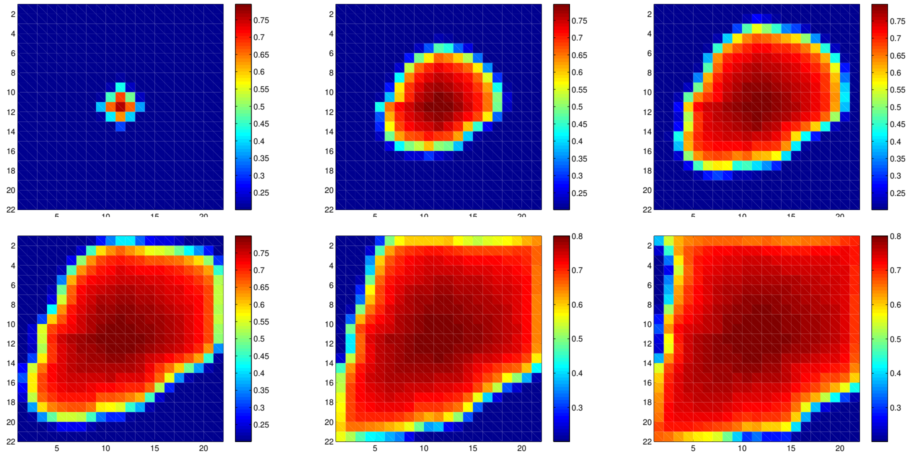


Figure 6.13: Water saturation distributions at different time instants simulated for the true permeability field (units: $[-]$).

Additional settings are required for the model-reduced approaches, in particular, the number of snapshots and the number of patterns used to built the reduced-order model. In order to cover the dynamic behavior of the system, snapshots should be taken during all the assimilation times for different permeability fields. Snapshots were created by perturbing the permeability field and simulating the reservoir behavior for the period of time when data were available. We perturbed each parameter separately, and therefore the number of snapshot simulations was equal to the number of parameters. The size of the perturbation was based on the ensemble of realizations; i.e. each realization was projected to the space spanned by the permeability patterns and the average of the absolute value of the coefficients was taken as the size of perturbation. When BPOD method is used, additional adjoint snapshots are required. The adjoint snapshots were taken separately for pressure and saturation. Moreover, since the adjoint model changes when different permeability field is used, the adjoint model was simulated for each forward simulation used to generate forward snapshots. The settings related

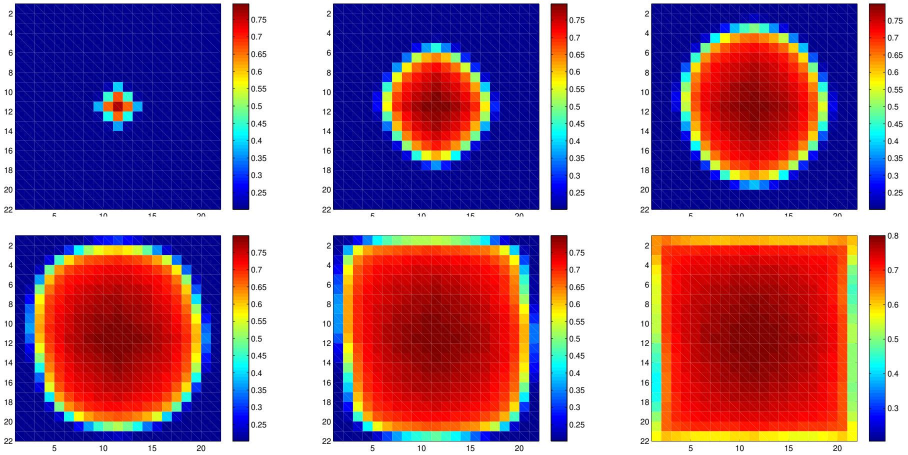


Figure 6.14: Water saturation distributions at different time instants simulated for the prior permeability field (units: $[-]$).

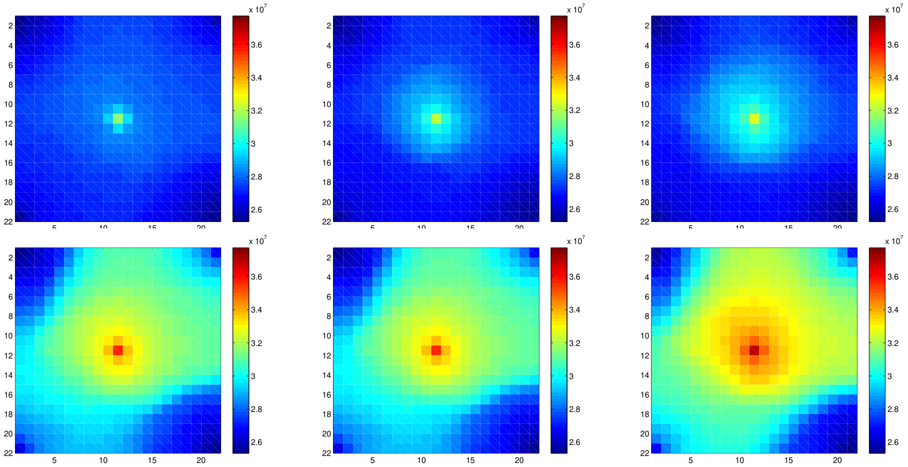


Figure 6.15: Oil pressure distributions at different time instants simulated for the true permeability field (units: $[Pa]$).

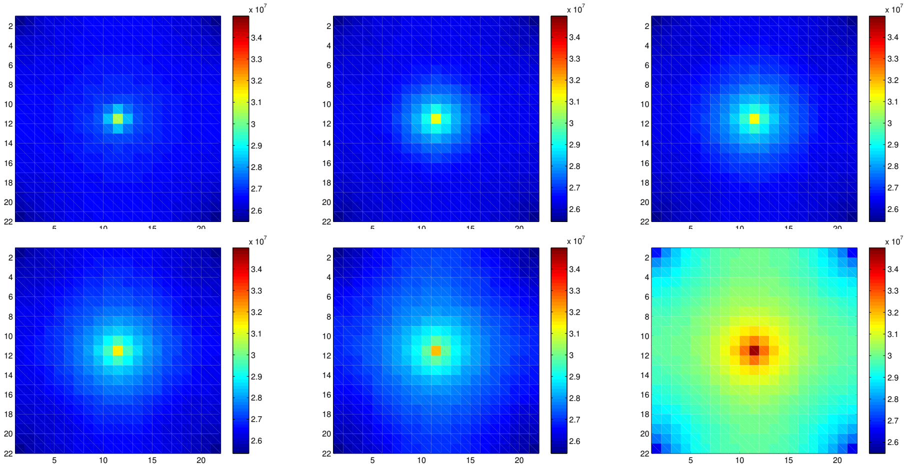


Figure 6.16: Oil pressure distributions at different time instants simulated for the prior permeability field (units: $[Pa]$).

to the number of patterns were determined by choosing the relative importance. Because we considered pressure and saturation separately it gave us the flexibility to choose different levels of relative importance for each of them. In case of the POD method the relative importance was set to 99% for saturation and 99.9% for pressure, while for the BPOD method it was fixed to 99.9% for both of them. In the BPOD method the number of patterns was lower, and therefore a higher range of relative importance could be afforted.

6.1.4 History matching results: experiment 1

In history matching our aim is to find a model that is geologically consistent and gives an acceptable history-match. In this chapter we aim to show that the model-reduced approach provides such a model in less computation time than the finite-difference based approach. We performed parameter estimation using the adjoint approach, the finite difference approach, and the model-reduced approaches and in all cases we iterated until the objective function reached comparably low values. The number of measurements used in the assimilation routine was 40. According to criterion given by Eq. 5.33 the value of the objective function for the estimated parameters $\boldsymbol{\theta}$ should fall in the interval:

$$11.06 \leq J(\boldsymbol{\theta}) \leq 28.94. \tag{6.1}$$

The inequality given in Eq. 6.1 is the first quantitative measure to evaluate the quality of obtained estimates. Another measure is the *Root Mean Squared Error* (*RMSE*) defined as the square root of the mean squared difference between the true parameter vector and its estimate. Since the objective function derived for this problem contains a background term, the true permeability field is not an argument that minimizes the given objective function, and therefore, RMSE can be only an indication of the quality of the estimate obtained.

The results of the adjoint-based method are summarized in Table 6.1. The initial objective function was 1657 and resulted from a simulation with the prior permeability field. The adjoint-based history matching converged after 67 iterations and resulted in the objective function equal 18.65. The minimization was performed using a quasi-Newton optimization where the Hessian of the objective function was updated using the BFGS formula (see Fletcher (1991)). The total computation time of this method was about 135 simulations (67 evaluations of the adjoint model and 68 evaluations of the forward model). The estimated permeability field is presented in Fig. 6.17. The final estimate closely resembles the true field, and its root mean squared error is 0.55. The same figure shows that the estimate obtained after 30 iterations is already very similar to the true field and gives the objective function that falls in the middle of the interval given by Eq. 6.1.

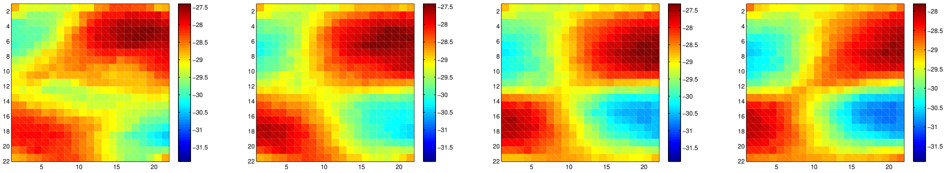


Figure 6.17: Experiment 1. The estimated permeability fields using the adjoint-based method after 10, 20, 30, and 67 iterations $[m^2]$.

Iter no	J	n_{red}	p_{red}	n_{sim}
0	1657	–	20	1
10	31.22	–	20	$21(= 1 + 2 \cdot 10)$
20	22.72	–	20	$41(= 21 + 2 \cdot 10)$
30	20.05	–	20	$61(= 41 + 2 \cdot 10)$
40	19.57	–	20	$81(= 61 + 2 \cdot 10)$
50	19.01	–	20	$101(= 81 + 2 \cdot 10)$
60	18.74	–	20	$121(= 101 + 2 \cdot 10)$
67	18.65	–	20	$135(= 121 + 2 \cdot 7)$

Table 6.1: Adjoint-based gradient-based history matching. p_{red}, n_{red} and n_{sim} denote the number of parameters, the number of patterns and the number of simulations, respectively.

The estimation results of the POD-based model-reduced method are given in Table 6.2 and Fig.6.18. We selected 31 pressure and 41 saturation patterns, which means that the reduced-order model operates in dimension $72 + 20$. Some of those patterns are presented in Figs. 6.19 and 6.20. Those patterns describe the deviations of the pressure and saturation fields from their background trajectories. From Table 6.2 one can see that a significant decrease in the objective function was obtained after the first outer iteration. Additional outer iterations were performed using the same system matrices of the reduced-order model and by updating only the trajectory. The third outer loop terminated with estimate for which the value of the objective function decreased to 20.27 and the root mean squared error decreased to 0.57. A new reduced-order model could be created in a very closed neighborhood of the current estimate and additional outer iteration could be performed. Since it is computationally expensive to rebuild the reduced-order model and the value of the objective function already indicated a good quality estimate, we did not iterate further.

Outer iter	J	n_{red}	p_{red}	n_{sim}
0	1657	–	20	1
1	43.52	$72(=31+41)$	20	$113(=1+20+72+20)$
2 (1)	20.90	72	20	$114(=113+1)$
3 (1)	20.27	72	20	$115(=114+1)$

Table 6.2: The POD-based model-reduced gradient-based history matching. p_{red}, n_{red} and n_{sim} denote the number of parameters, the number of patterns and the number of simulations, respectively.

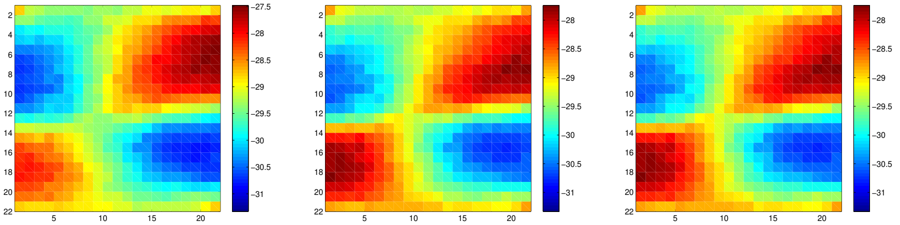


Figure 6.18: Experiment 1. The estimated permeability fields from the POD-based method after first, second and third outer iterations $[m^2]$.

The estimation results using the BPOD-based method are summarized in Table 6.3 and Fig. 6.21. The BPOD-based reduced-order model required 6 pressure and 42 saturation patterns (all pressure and first eight dominant saturation patterns are presented in Figs. 6.23 and 6.22, respectively). Performing six additional outer loops resulted in the objective function value that falls in the middle of interval given by Eq. 6.1. The root mean squared error $e_{RMSE} = 0.55$ is exactly the same as the one obtained when using the adjoint method.

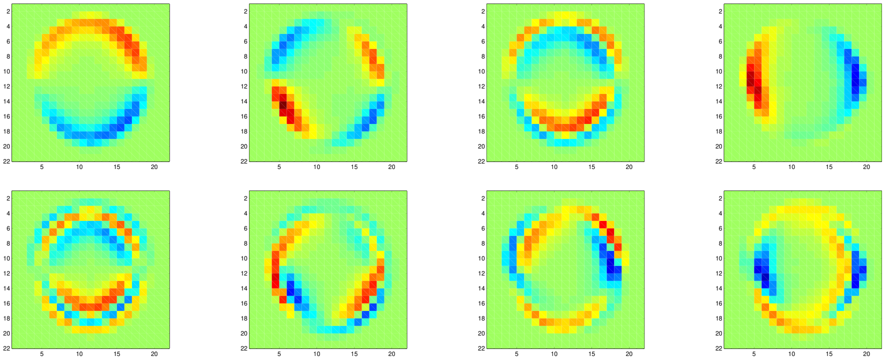


Figure 6.19: Experiment 1. 8 patterns of the saturation field with the largest importance identified by the POD method.

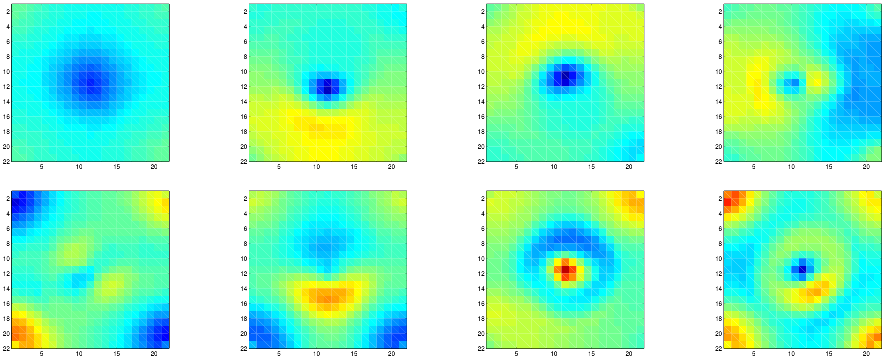


Figure 6.20: Experiment 1. 8 patterns of the pressure field with the largest importance identified by the POD method.

Outer iter	J	n_{red}	p_{red}	n_{sim}
0	1657	–	20	1
1	66.93	$48(= 6 + 42)$	20	$109(= 1 + 2 \cdot 20 + 48 + 20)$
2 (1)	27.12	48	20	$110(= 109 + 1)$
3 (1)	20.23	48	20	$111(= 110 + 1)$
4 (1)	19.02	48	20	$112(= 111 + 1)$
5 (1)	18.76	48	20	$113(= 112 + 1)$
6 (1)	18.73	48	20	$114(= 113 + 1)$

Table 6.3: The BPOD-based model-reduced gradient-based history matching. p_{red}, n_{red} and n_{sim} denote the number of parameters, the number of patterns and the number of simulations, respectively.

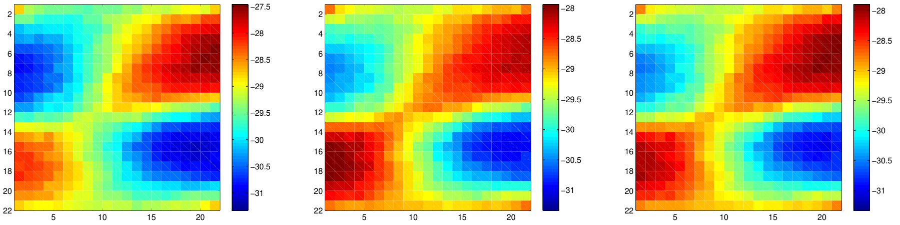


Figure 6.21: Experiment 1. The estimated permeability fields from the BPOD-based method after first, second, and sixth outer iterations $[m^2]$.

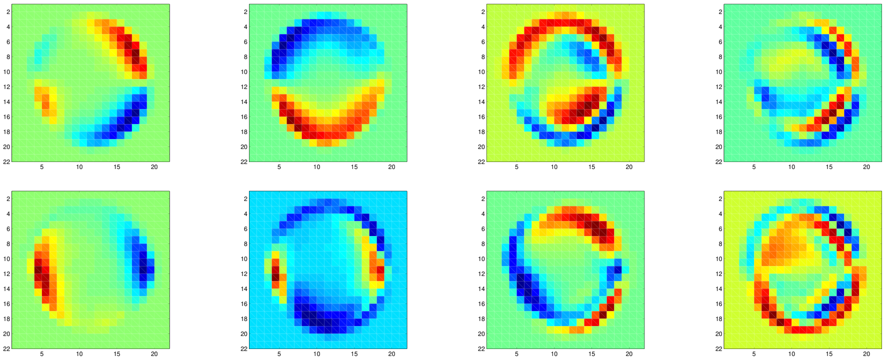


Figure 6.22: Experiment 1. 8 patterns of the saturation field with the largest importance identified using the BPOD method.

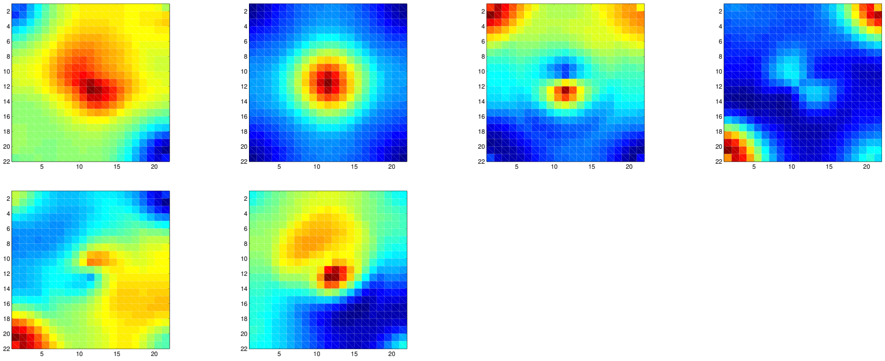


Figure 6.23: Experiment 1. 6 patterns of the pressure field with the largest importance identified using the BPOD method.

In order to compare the computational efficiency of the model-reduced approaches we performed an additional computation using a finite-difference approach. The results of this approach are summarized in Table 6.4 and Fig. 6.24. We terminated the iterations when the objective function reached value 19.85. This required 43 iterations and about 1054 simulations of the forward model. The estimated permeability field closely resembles the true field, and has a root mean squared error $e_{RMSE} = 0.68$. Moreover, the estimate obtained after 30 iterations is already very similar to the true field.

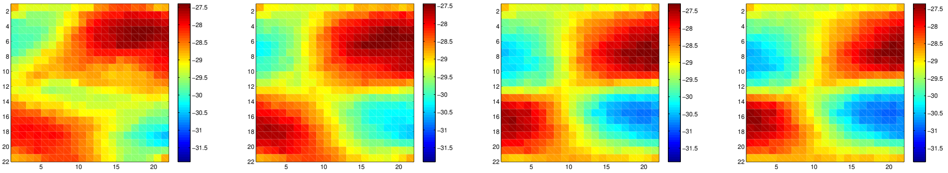


Figure 6.24: Experiment 1. The estimated permeability fields from the finite-difference based method after 10, 20, 30, and 43 iterations $[m^2]$.

Iter no	J	n_{red}	p_{red}	n_{sim}
0	1657	–	20	1
10	31.01	–	20	$210 (= 21 \cdot 10)$
20	22.46	–	20	$420 (= 210 + 21 \cdot 10)$
30	20.17	–	20	$630 (= 420 + 21 \cdot 10)$
40	19.91	–	20	$840 (= 630 + 21 \cdot 10)$
43	19.85	–	20	$1054 (= 840 + 21 \cdot 10 + 4)$

Table 6.4: Experiment 1. Finite-difference based gradient-based history matching. p_{red}, n_{red} and n_{sim} denote the number of parameters, the number of patterns and the number of simulations, respectively.

We showed that the presented history matching methods were able to find the geologically consistent models that gave the acceptable history-matches. It is expected that those models give also better predictions of production curves, such as oil rates and water rates. An important aspect during water flooding is the right prediction of the moment when water breaks through in the production wells. From that moment on, the injected water follows a preferable path to the production well and the well starts to produce mostly water. Such a well in the end becomes unprofitable and has to be closed. Therefore, the next measure to compare the results of the assimilation routines is the *water breakthrough time prediction capability*. The water breakthrough time was evaluated over a period of 1500 days and is depicted in Figs. 6.25 and 6.26. The curves represent produced water rates for the true permeability field (red), the prior permeability field (blue), the permeability field obtained from the adjoint-based method (green), the estimate obtained from the finite-difference based approach (light blue), the estimate obtained from the POD-based model-reduced approach (pink) and the BPOD-based model-reduced approach (yellow). We see that the history matching exercises resulted in the improved predictions for all the estimates compared to the prior permeability field. It is difficult to point out the estimate which gave the best predictions, because it differs from well to well. In the North-East well the best prediction was given by the POD-based estimate, while in the North-West by the BPOD-based and the adjoint-based estimates. In the South-West and South-East the POD-based estimate performed the best.

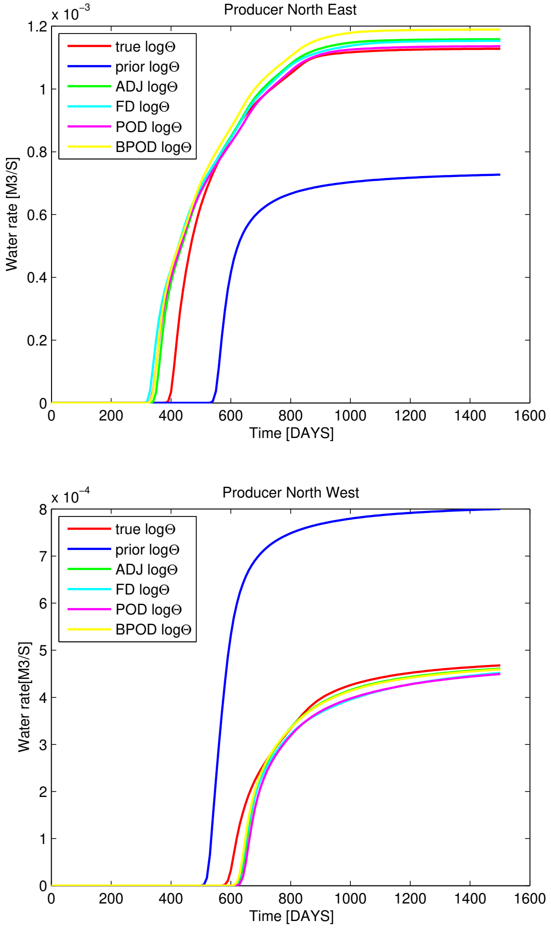


Figure 6.25: Experiment 1. Comparison between the predictions of water breakthrough time and water flow rates for two production wells, located at the top of the reservoir, using the estimates obtained with different methods.

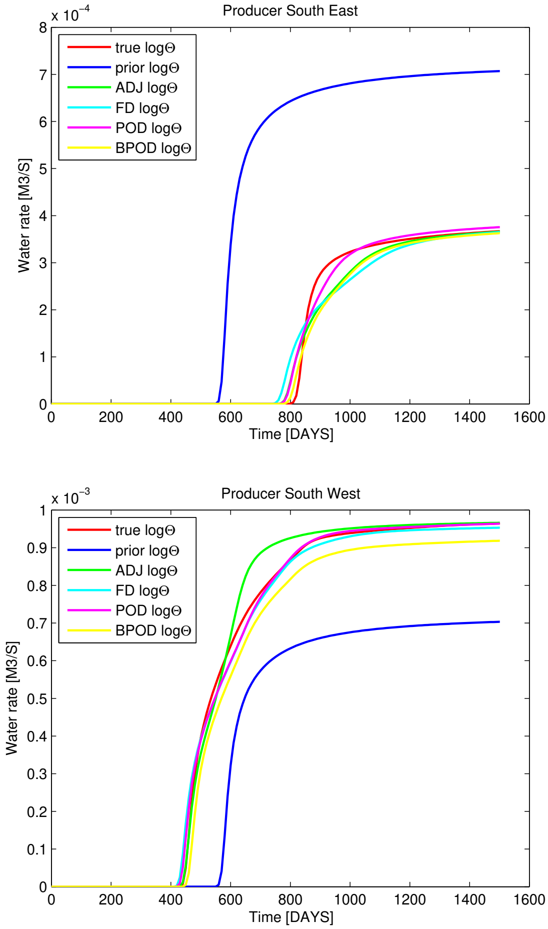


Figure 6.26: Experiment 1. Comparison between the predictions of water breakthrough time and water flow rates for two production wells, located at the bottom of the reservoir, using the estimates obtained with different methods.

In model-reduced data assimilation the calculated gradients are only the approximations of the true gradients. In Fig. 6.27 the exact gradient used in the first step of the optimization is compared with the gradients approximated using the BPOD-based and the POD-based reduced-order forward and adjoint models. We see that the approximated gradients look very much like the true gradient. Examining carefully Fig. 6.27 one can notice that the BPOD-based gradient resembles the true gradient better than the POD-based gradient. The state reconstruction, however, looks slightly better for the POD-based model than the BPOD-based model.

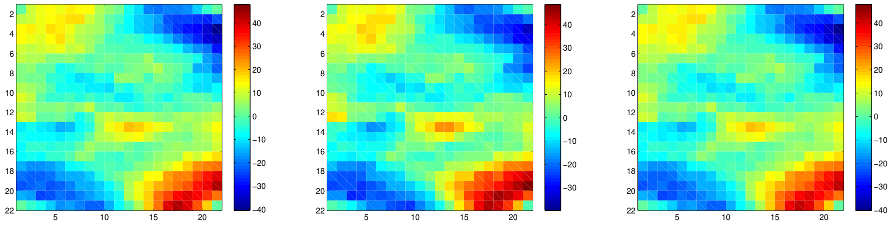


Figure 6.27: Experiment 1. The adjoint-based gradient (left), the POD-based gradient (middle) and the BPOD-based gradient (right).

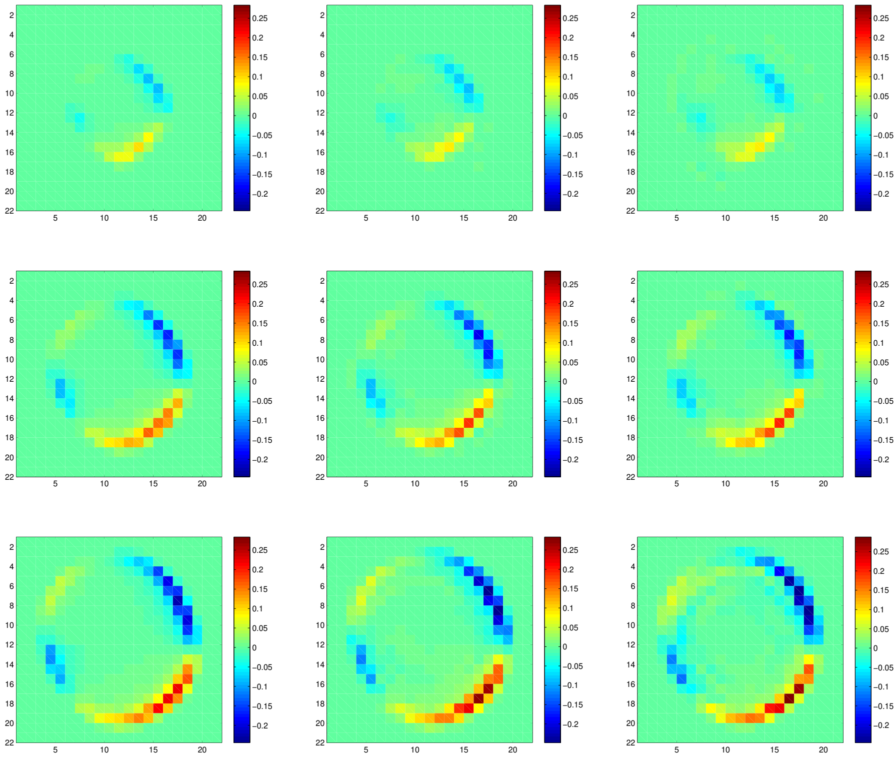


Figure 6.28: Experiment 1. Comparison of the saturation fields at different time steps (20th, 30th and the last one) from the original model (left), the POD-based model (middle) and the BPOD-based model (right).

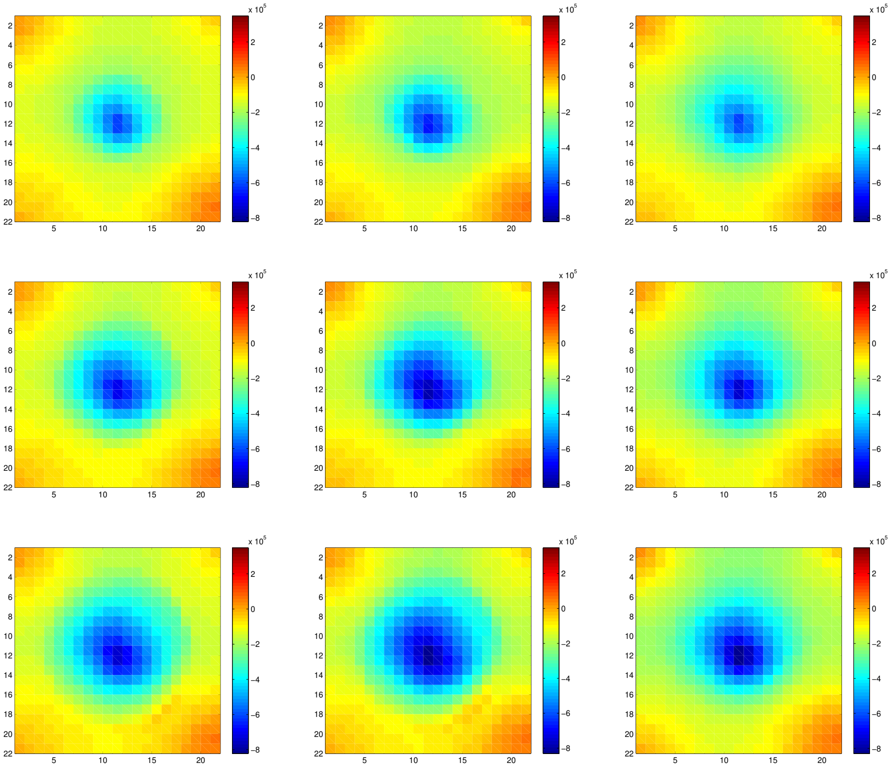


Figure 6.29: Experiment 1. Comparison of the pressure fields at different time steps (20th, 30th and the last one) from the original model (left), the POD-based model (middle) and the BPOD-based model (right).

In general, the adjoint-based approach is computationally very effective. In this experiment the largest decrease in the objective function was observed after the first iterations, and already after 30 iterations it reached a value in the middle of the statistical interval given by Eq. 6.1. It is known that in case of a quadratic function the quasi-Newton optimization should converge in a number of iterations smaller or equal to the number of parameters plus one (which in this experiment is 21). In our case the objective function was not quadratic and therefore the number of iterations was larger, and equal 67. The objective function might be more irregular closer to the optimum, which might cause slower convergence. Since we used reparametrized parameters, scaling was not an issue; new parameters were of the same order of magnitude. The total cost of the adjoint-based routine was approximately 135 simulations. The BPOD-based routine required about 114 simulations. About 40 simulations were performed to collect the snapshots of the forward model and the adjoint model; 48 simulations were used to approximate the partial derivatives of the system along selected patterns; 20 additional simulations were required to approximate the partial derivatives of the model along the permeability patterns; 6 simulations were used to update the trajectory of the system. In this experiment the computational cost of the BPOD-based approach was slightly lower than of the adjoint-based approach. The POD-based method required about 115 simulations, which is a summation of 20 simulations during which snapshots were collected, 72 simulations required to approximate the partial derivatives of the model along the state patterns, and 20 simulations required to approximate the partial derivatives along the permeability patterns and 3 simulations to calculate the original trajectory of the model. The final value of the objective function resulting from the POD-based method was comparable to the value of the objective function resulting from the adjoint routine after 30 iterations, which computational cost was about 61 simulations. Hence, for this experiment the POD-based method was about twice as expensive as the adjoint-based approach. The finite-difference routine was very expensive, after 43 iterations it reached the value of the objective function equal 19.85, which required about 1054 simulations. This method was therefore more than 9 times as expensive as the BPOD-based method. After 30 iterations the finite-difference approach reached the objective function equal 20.17 which was more than 5 times as expensive as the POD-based approach.

6.1.5 History matching results: experiment 2

The shape of the true permeability field and the well distribution used in previous experiment is favorable to the estimation process. The high permeable streak connecting the injector with two producers results in very informative production data. Therefore, the true permeability field of the above example turned out to be easy to estimate, and the conclusions drawn from this case might not be representative for more complicated shapes. The new true field and its reconstruction are presented in Fig. 6.30.

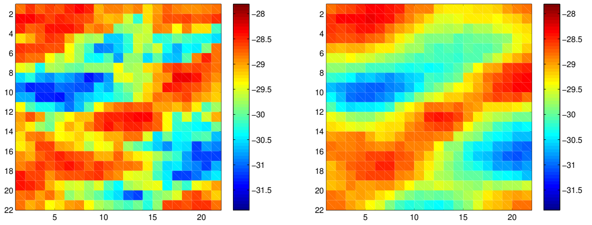


Figure 6.30: Experiment 2. True (left) and reconstructed true (right) permeability fields $[m^2]$.

The results of the adjoint-based method are summarized in Table 6.5 and in Fig. 6.31. The starting value of the objective function was equal 226.49. The routine terminated after 56 iterations, which required about 113 simulation. The objective function reached value of 20.33, while the estimated permeability field decreased the root mean squared error to 0.75. The estimate obtained after 30 iterations was already very similar to the true field. The computational cost was in this case comparable to about 61 simulations.

Iter no	J	n_{red}	p_{red}	n_{sim}
0	226.49	–	20	1
10	23.97	–	20	$21(= 1 + 2 \cdot 10)$
20	22.01	–	20	$41(= 21 + 2 \cdot 10)$
30	21.21	–	20	$61(= 41 + 2 \cdot 10)$
40	20.60	–	20	$81(= 61 + 2 \cdot 10)$
50	20.42	–	20	$101(= 81 + 2 \cdot 10)$
56	20.33	–	20	$113(= 101 + 2 \cdot 6)$

Table 6.5: Adjoint-based gradient-based history matching. p_{red}, n_{red} and n_{sim} denote the number of parameters, the number of patterns and the number of simulations, respectively.

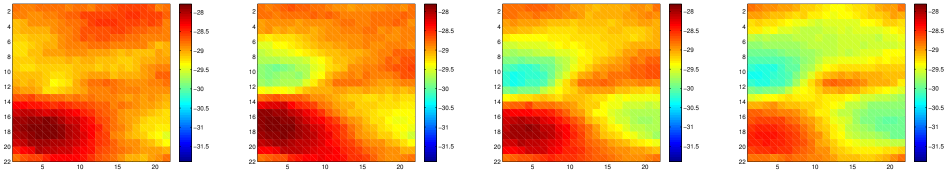


Figure 6.31: Experiment 2. The estimated permeability fields from adjoint-based method after 10, 20, 30, and 56 iterations $[m^2]$.

The estimation results of the POD-based method are summarized in Table 6.6 and Fig. 6.32. We selected 30 pressure and 42 saturation patterns. After first outer loop the objective function decreased significantly to the value 22.38. The next outer loop was performed using the same matrices of the reduced-order model and only updating the trajectory. The additional outer loop resulted in a small decrease of the objective function. The root mean squared error of the final estimate is 0.69.

Outer iter	J	n_{red}	p_{red}	n_{sim}
0	226.49	–	20	1
1	22.38	$72(= 30 + 42)$	20	$113(= 1 + 20 + 72 + 20)$
2 (1)	22.04	72	20	$114(= 113 + 1)$

Table 6.6: The POD-based model-reduced gradient-based history matching. p_{red}, n_{red} and n_{sim} denote the number of parameters, the number of patterns and the number of simulations, respectively.

The results of the BPOD-based reduced-order model are summarized in Table 6.7 and Fig. 6.32. The dimension of the reduced-order model was $9 + 49$ (resulting from 9 pressure and 49 saturation patterns). After three outer loops the estimate improved significantly and resulted in the value of the objective function 20.44 that falls in the center of the interval given by Eq. 6.1. The root mean squared error of the final estimate equals 0.66. The additional iterations were performed using the same matrices of the reduced-order model.

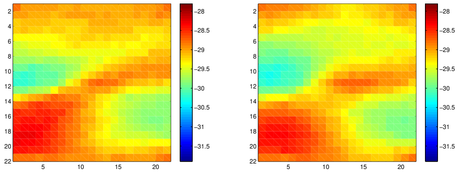


Figure 6.32: Experiment 2. The estimated permeability fields from the POD-based method (left) and the BPOD-based method (right) $[m^2]$.

Outer iter	J	n_{red}	p_{red}	n_{sim}
0	226.49	–	20	1
1	26.28	$58(= 9 + 49)$	20	$119(= 1 + 2 \cdot 20 + 58 + 20)$
2 (1)	20.60	58	20	$120(= 119 + 1)$
3 (1)	20.44	58	20	$121(= 120 + 1)$

Table 6.7: The BPOD-based model-reduced gradient-based history matching. p_{red}, n_{red} and n_{sim} denote the number of parameters, the number of patterns and the number of simulations, respectively.

The results of the finite-difference based approach are summarized in Table 6.8 and Fig. 6.33. The routine converged after 36 iterations, which required the time comparable to about 851 simulations of the forward model. The estimated permeability field resembles the true field, and has a root mean squared error $e_{RMSE} = 0.70$. Already after 30 iterations very similar results were obtained.

Iter no	J	n_{red}	p_{red}	n_{sim}
0	226.49	–	20	1
10	23.94	–	20	$210(= 21 \cdot 10)$
20	22.13	–	20	$420(= 210 + 21 \cdot 10)$
30	21.65	–	20	$635(= 420 + 21 \cdot 10 + 5)$
36	21.56	–	20	$851(= 635 + 21 \cdot 10 + 6)$

Table 6.8: Finite-difference based gradient-based history matching. p_{red}, n_{red} and n_{sim} denote the number of parameters, the number of patterns and the number of simulations, respectively.

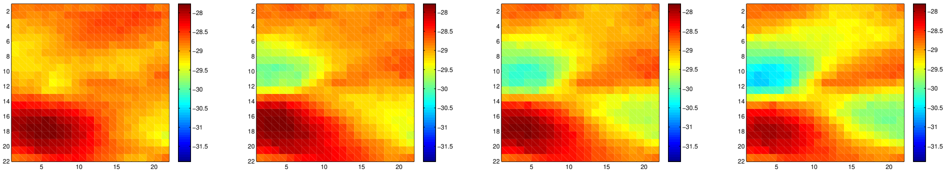


Figure 6.33: Experiment 2. The estimated permeability fields from finite-difference based method after 10, 20, 30, and 36 iterations $[m^2]$.

The water breakthrough time was evaluated over a period of 1500 days and is depicted in Figs. 6.34 and 6.35. The colors of the curves remained the same as in Experiment 1. Figs. 6.34 and 6.35 show that for North-West and South-West wells the predictions of water breakthrough time improved for all the estimates (compared to those obtained from the prior permeability field) but are not perfect. For the South-East well all the estimates showed the improvement of the magnitude of the produced water but the water breakthrough moment was predicted worse compared to the prediction obtained for the prior estimate. In the North-East well no considerable changes were observed. Since all obtained estimates resulted in almost identical predictions, the assimilation routines might require more data to improve the predictions.

In this experiment the adjoint-based approach converged after 56 iterations, which required about 113 simulations. After 20 iterations of this method the objective function decreased to 22.01, which is similar to the result obtained from the POD-based method. The computational effort of those 20 iterations is comparable to about 41 simulations, while the computational effort of the POD-based method is comparable to about 114 simulations. In this case the POD-based approach was about 3 times more expensive than the adjoint-based approach. The BPOD-based routine converged to a very low value using about 121 simulations, which was slightly more than for the adjoint-based approach. The finite-difference based routine was again very expensive. After 36 iterations it reached the value of the objective function equal to 21.56, which required about 851 simulations. It is therefore more than 7 times as expensive as the BPOD-based routine. After 20 iterations the finite-difference approach reached the objective function equal to 22.13 which is about 4 times as expensive as the POD-based approach.

6.1.6 Conclusions

In the experiments presented in this chapter, two different realizations from an ensemble of realizations were chosen to define the true permeability field and the model-reduced history matching methods were tested. The following conclusions can be drawn from those studies:

- In both cases presented the model-reduced approaches performed very well.

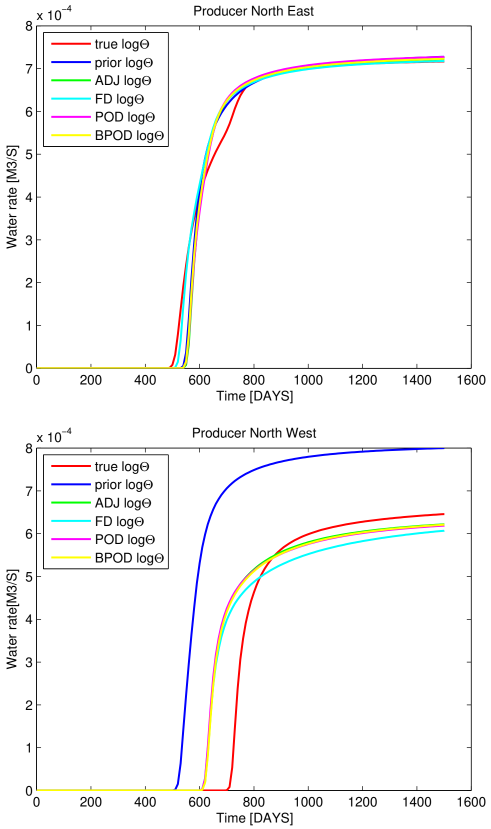


Figure 6.34: Experiment 2. Comparison between the predictions of water breakthrough time and water flow rates for North-East and North-West production wells, located at the top of the reservoir, using the estimates obtained with different methods.

The obtained estimates of the permeability field significantly improved compared to the prior fields and gave an acceptable history-match.

- The quality of the prediction capabilities of the models estimated by the model-reduced methods were very high and comparable to those obtained by the classical adjoint-based approach.

- The computational time of the presented methods differed. The adjoint-based method was computationally very effective. In the presented cases the POD-based method was approximately twice as expensive as the classical approach, but the BPOD-based method was as efficient as the adjoint-based method. Both methods were considerably cheaper than the finite-difference based approach. For the small examples considered (882 states and 441 parameters) the model-reduced approach showed to perform very well.

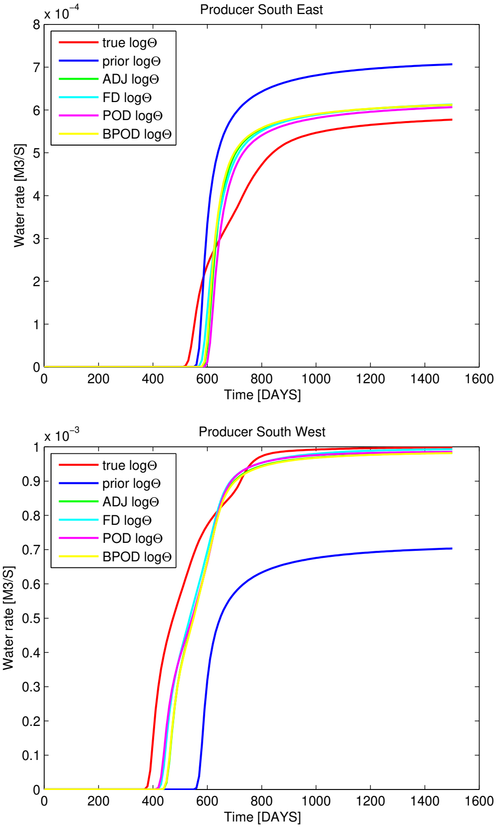


Figure 6.35: Experiment 2. Comparison between the predictions of water breakthrough time and water flow rates for South-East and South-West production wells, located at the bottom of the reservoir, using the estimates obtained with different methods.

- The classical adjoint-based approach gives the exact gradients of the objective function with respect to the parameters. The model-reduced history matching procedure uses a simplified reduced-order forward model and a corresponding reduced-order adjoint model, and therefore the calculated gradients are not exact. In the presented cases the approximated gradients were very close to the true gradients.

- For complex models the classical adjoint-based history matching procedure is very difficult to implement. The presented POD-based approach is adjoint-free, is relatively easy to implement and can be used with any reservoir simulator. The balanced POD-based method requires the adjoint model but does not require the Jacobians with respect to the parameters.

- The presented cases are probably too small to draw general conclusions. In large scale models, it might be computationally very expensive to use 99.9%

of the relative importance to construct the reduced-order model. However, the reduced-order model offers many tuning parameters that might be used to reduce the computational time, e.g. the number of simulations to obtain snapshots and the number of patterns used to build the reduced-order model can be changed. Another relevant parameter in this approach is the size of the perturbations used to generate snapshots.

- Several tests were performed to assess the impact of these parameters on the performance of the model-reduced assimilation scheme. Those tests showed that the perturbation size of the permeability patterns used to generate the snapshots is a relevant parameter. Obviously, the perturbations which bring the estimate to the true values would be the most desirable, but they are not essential to construct a good reduced-order model. Because the reduced-order model is a linearized form of the original equations, the snapshots should be taken in a relatively close neighborhood of the linearized trajectory. Another relevant parameter is the number of simulations used to generate snapshots. Choosing less snapshots and less patterns results in less variability, and therefore the optimization may not converge to a solution comparable to that of the adjoint-based method. In the model-reduced approaches if the assimilation routine does not result in an improvement, the user may increase the number of patterns used to create the reduced-order model. This step does not require to recompute all partial derivatives, but only the partial derivatives along the new patterns. This is not the case when the number of snapshots have to be increased. The new snapshots can be added to the existing set of snapshots but since the patterns are recomputed all partial derivative have to be recomputed too. This is computationally an expensive part of the model-reduced approach, and therefore, the set of snapshots should be chosen carefully to avoid the unsuccessful outer loops.

- From our experience we can conclude that using half of the collected snapshots or less, an accurate reduced-order model and very good data assimilation results can be obtained. However, to get the results as good as with the adjoint-based method we need to keep the accuracy relatively high. Based on several tests we built a guideline for assisted history matching with model-reduced approaches, i.e.

  1. Create snapshots while perturbing two patterns simultaneously, e.g. perturb in one simulation the most dominant pattern together with the least dominant ones, and so on.

  2. Build the reduced-order model with 80%-90% of the relative importance and perform the assimilation.

  3. Make additional iterations using exactly the same system matrices and update only the trajectory.

  4. Add patterns to the model if necessary, perform assimilation and go to Step 3 again.

The coming experiment is focused on quantifying the computational benefits of the model-reduced methods for larger and more complex models.

6.2 Application to a complex 3D model

In the simulator used for the experiments considered in this section, the adjoint variable is accessible only for the saturation state and results from the adjoint model of the implicit form of the reservoir equations. This is different than the adjoint of the explicit form that is assumed in our methodology. Therefore, we present here only results obtained with the POD-based method.

6.2.1 Model settings

We performed another 'twin experiment' in which we used the described approaches to estimate the uncertain permeability field of a 3D reservoir model from synthetic noisy production data. The model, originally introduced by Van Essen et al. (2009), describes iso-thermal slightly compressible two-phase (oil-water) flow, in a channelized reservoir with eight injection and four production wells; see Fig. 6.36. The model is based on the assumptions that the effects of capillary pressure can be neglected and that the permeability is isotropic.

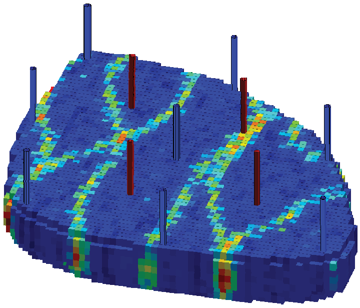


Figure 6.36: The 'true' reservoir permeability field with well locations. Producers are indicated in red, injectors in blue.

- *Reservoir geometry:*
  The reservoir model is a part of a quadrilateral of the size $480m \times 480m \times 28m$, divided into $60 \times 60 \times 7$ uniform Cartesian grid blocks, of which there are 18553 active, forming an egg-shaped domain with no-flow boundaries.

- *Reservoir properties :*
  The porosity is assumed to be uniform and equals 0.2. The 'true' permeability field on a log scale is shown in Fig. 6.37. The permeability contrast between the channels and the background fill is relatively mild (about a factor of ten). The rock compressibility $c_r = 0 \cdot 1/Pa$.

- *Initial conditions :*
  The initial reservoir pressures is $40MPa$ and the initial water saturation is

taken as connate water saturation $S_{wc} = 0.2$.

- *Well locations and constraints :*
  Water is injected at a rate of $32m^3/d$ in the eight vertical injectors, while the four vertical producers are operated at constant bottom-hole pressures of $39.5MPa$. All wells are perforated at each layer of the reservoir.

- *Fluid properties :*
  The oil and water viscosities are identical: $\mu_o = \mu_w = 10^{-3} Pa \cdot s$, and so are the compressibilities: $c_w = c_o = 10^{-10} 1/Pa$. The densities of oil and water are given by $\rho_o = 900kg/m^3$, and $\rho_w = 1000kg/m^3$.

- *Relative permeabilities:*
  We assumed Corey-type relative permeabilities with Corey exponents for oil and water: $n_o = 4$ and $n_w = 3$. The residual oil saturation was taken as $S_{or} = 0.15$, and the connate water saturation as $S_{wc} = 0.2$.

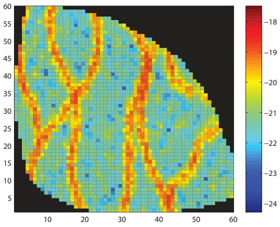


Figure 6.37: Natural log of true permeability field in the 4th layer $[m^2]$.

6.2.2 History matching settings

The assimilation period was chosen to be three years during which observations were taken from 4 production and 8 injection wells every 60 days, resulting in 18 time instances. During this assimilation period water breakthrough occurred in one well, after about 500 days. Noisy observations were generated from the model with the 'true' permeability field, and consisted of bottom-hole pressures in the injectors, and water and oil rates in the producers, with a random normally distributed noise of 5% of their values. It resulted in 72 oil rates and 5 water rates measured in the producers and 144 bottom-hole pressures measured in the injectors, which gives in total 221 data points. In the assimilation procedure, the errors statistics were assumed to be known, i.e. errors were represented by a normal distribution with zero mean and a standard deviation equal to 5% of the observed data. Moreover it was assumed, correctly, that distinct observations were affected by physically independent noise, resulting in a diagonal covariance matrix for the observation errors. The ensemble of reservoir models was created using in-house geostatistical software with an anisotropic variogram. We used an ensemble of 100 realizations which are Gaussian distributed random fields in each layer, conditioned to the well data corrupted with an error in the order of 10^{-4} of the true value. All realizations were conditioned to the same hard data. Each layer had a randomly-oriented ellipsoidal covariance with randomly-chosen correlation length between 4 to 8 grid blocks for each layer at each realization. No vertical correlation was assumed. An ensemble of 100 permeability fields was used to create the background permeability $\boldsymbol{\theta}_{init}$, taken as the ensemble average, and to estimate the background error covariance matrix. The permeability field was reparameterized using a KL-expansion, resulting in 22 permeability patterns with 22 corresponding parameters. Such a small number of parameters to represent the 18553 gridblock permeability values is motivated by the fact that the available data are very sparse, which implies that the identifiability of the parameter space is very low (Zandvliet et al. (2008)).

6.2.3 Reduced-order model settings

Additional numerical parameter settings are required for the model-reduced approach, in particular, the number of snapshots and the number of patterns used to build the reduced-order model. In order to cover the dynamic behavior of the system, snapshots should be collected during all the assimilation times for different ranges of the permeability fields. However, to get a rough idea of the direction of the parameter update, we started with a lower number of snapshots in the first outer iteration (reducing also the dimension of the eigenvalue problem) and we increased it in the successive ones. We selected a snapshot every 60 days from 11 simulations, where each simulation was based on a different permeability field created by perturbing two different parameters per simulation. We collected snapshots for pressures and saturations separately, which resulted in 200 snapshots for pressures and 200 snapshots for saturations, and two eigenvalue problems to be solved. This separation allowed us to choose different relative importance levels for pressure and for saturation, which varied over the iterations. We started with a lower relative importance in the first outer iteration and increased it in the successive ones. The relative importance for the pressures was initiated at 85% for the first iteration and was set to increase 5% in each successive iteration up to 95%. Because the saturation field displays a much larger spatial variability than the pressure field, it results in a higher number of patterns for the same level of relative importance. We fixed the relative importance for saturation to 80% in the first iteration, increasing it in the successive iterations up to 90%, in steps of 5%.

6.2.4 History matching results: experiment 1

It was noticed in Heijn et al. (2004), Van Doren et al. (2004), Cardoso et al. (2009) and previous experiments that the pressure behavior can be well represented by a few patterns, but that the situation is worse for the saturation behavior, which is caused by the moving fluid interface. A significantly larger number of patterns is therefore required to satisfy the chosen accuracy level. We selected 54 + 11 patterns for the first iteration (54 for saturation and 11 for pressure), which means that the reduced-order model operates in dimension $65 + 22$. The results of the first data assimilation experiment are summarized in Table 6.9.

Outer iter	J	n_{red}	p_{red}	n_{sim}
0	338	–	22	1
1	113	65	22	$98(= 11 + 54 + 11 + 22)$

Table 6.9: Model-reduced gradient-based history matching. p_{red}, n_{red} and n_{sim} denote the number of parameters, the number of patterns and the number of simulations, respectively.

The initial value of the objective function was 338 and resulted from a simulation with the prior parameters, as depicted in Fig. 6.38.

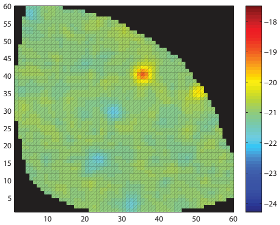


Figure 6.38: Experiment 1. Log of prior permeability field in layer 4 $[m^2]$.

The first outer iteration terminated with parameters for which the original objective function decreased to 113. The stopping criterion for the outer loop, (5.33), was satisfied ($179 \leq 2J(\boldsymbol{\theta}) \leq 263$), and the procedure terminated. The computational cost of the model-reduced approach was about equal to the time required to simulate 98 high-order model evaluations. More precisely, 98 is a summation of 11 simulations during which snapshots are collected, 54 + 11 simulations required to approximate the partial derivatives along saturation and pressure patterns and 22 simulations required to approximate the partial derivatives with respect to the parameters. The final estimate is presented in Fig. 6.39.

From a geological point of view the obtained estimate does not represent the true reservoir (see Fig. 6.37 and Fig. 6.39). Clearly, the information coming from sparse well data is not sufficient to capture the complicated geological structure, a well known issue in petroleum reservoir history matching. However, if we consider the match between observed data and predictions in Fig. 6.40, we see that the history matching exercise resulted in an improved match for past data in all four producers and improved predictions in three out of the four producers.

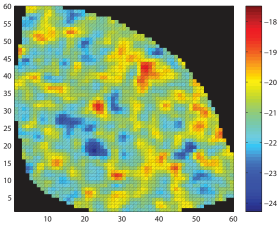


Figure 6.39: Experiment 1. Log of permeability field from the model-reduced approach in layer $4[m^2]$.

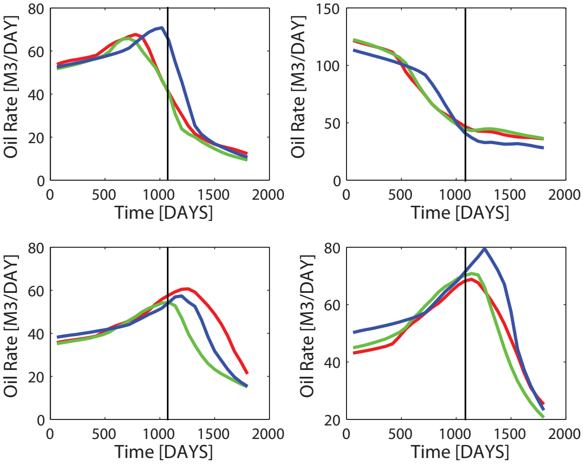


Figure 6.40: Experiment 1. Oil rate in the production wells, obtained with the prior permeability field (blue), the true permeability field (red), and the estimated permeability field (green) using the model-reduced approach. The black vertical lines indicate the end of the assimilation period.

In an attempt to further reduce the computing time, we repeated the model-reduced history matching exercise with an even smaller number of patterns to build the reduced-order model. We obtained similar results in approximately 68 iterations; see Table 6.10 for further details.

Outer iter	J	n_{red}	p_{red}	n_{sim}
0	338	–	22	1
1	114	35	22	$68(= 11 + 29 + 6 + 22)$

Table 6.10: Model-reduced gradient-based history matching with fewer patterns. p_{red}, n_{red} and n_{sim} denote the number of parameters, the number of patterns and the number of simulations, respectively.

Next, we compared the results with those of the classical approach using the adjoint of the tangent linear model. Because the convergence of any minimization problem strongly depends on the number of estimated parameters, we used the same reparameterization for the adjoint-based case as for the model-reduced case. It was stopped according to the criterion defined by Eq. 5.31 and Eq. 5.32.

Iterations	J	n_{red}	p_{red}	n_{sim}
0	338	–	22	1
15	97	–	22	$79(= 15 \times 2 + 49)$

Table 6.11: Classical adjoint-based history matching. p_{red} and n_{sim} denote the number of parameters and the number of simulations, respectively.

The adjoint-based history matching exercise converged after 15 iterations (see Fig. 6.41 and Table 6.11).

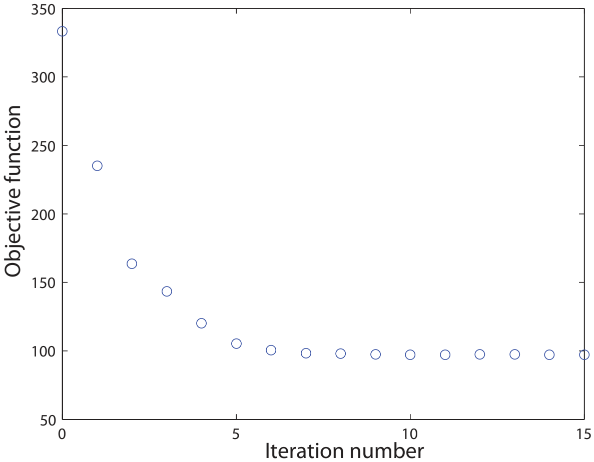


Figure 6.41: Experiment 1. Successive iterations of the objective function value.

The minimization was performed using a quasi-Newton optimization where the Hessian of the objective function was updated using the BFGS method; see Fletcher (1991). It required 15 evaluations of the adjoint model and 49 evaluations of the forward model (additional evaluations of forward model were caused by the line-search procedure) and obtained a minimal value of the objective function of 97. The estimated parameters and the liquid rates are presented in Fig. 6.42 and Fig. 6.43, respectively.

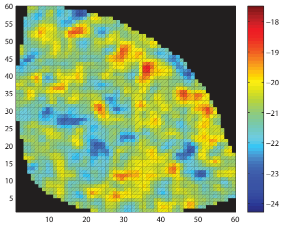


Figure 6.42: Experiment 1. Log of permeability field from the adjoint-based approach in layer 4 $[m^2]$.

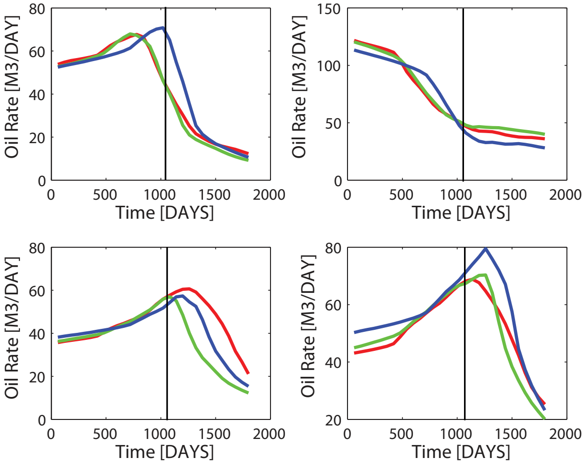


Figure 6.43: Experiment 1. Oil rate in the production wells, obtained with the prior permeability field (blue), the true permeability field (red), and the estimated permeability field (green) using the adjoint-based approach. The black vertical lines indicate the end of the assimilation period.

The results closely resemble those of the model-reduced approach. Both methods converged to solutions that give good matches for all producers and good predictions for three out of four producers. In both cases, the difficulty in the prediction is related to the same well, located in the region that is swept least, and for which the observed production rates are the lowest. Summarizing, for the example considered, the approximate derivatives as obtained from the model-reduced approach resulted in a similar behavior of the optimizer as the 'exact' derivatives obtained from the adjoint-based approach.

In order to compare the computational effort of the model-reduced approach we performed an additional computation using a finite-difference approach. We iterated the optimization until we obtained a value of the objective function comparable to the result of the model-reduced approach. Table 6.12, Fig. 6.44 and Fig.

6.45 represent those results.

Iterations	J	n_{red}	p_{red}	n_{sim}
0	338	–	22	1
6	111	–	22	$141(= 6 \times 23 + 3)$

Table 6.12: Finite-difference based history matching. p_{red} and n_{sim} denote the number of parameters and the number of simulations, respectively.

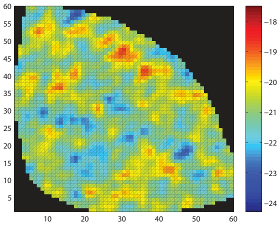


Figure 6.44: Experiment 1. Log of permeability field from the finite-difference based approach in layer 4 $[m^2]$.

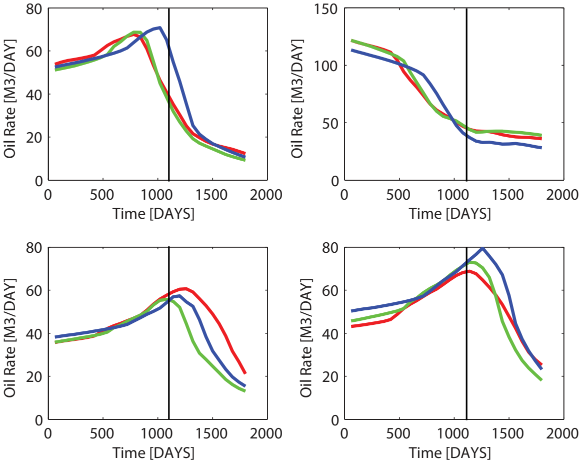


Figure 6.45: Experiment 1. Oil rate in the production wells, obtained with the prior permeability field (blue), the true permeability field (red), and the estimated permeability field (green) using the finite-difference based approach. The black vertical lines indicate the end of the assimilation period.

As expected, the numerical performance of the adjoint-based approach is better than that of the model-reduced approach. In this example we required 79 and

98 simulations for each of the methods respectively. In fact we could have used less iterations in the adjoint-based case to achieve a similar reduction in the objective function as for the model-reduced case: see Fig. 6.41 which indicates that we could have used just seven iterations. In that case the adjoint-based approach would have required only about seven forward, seven backward and an estimated five line search iterations bringing the total to 19. I.e. the adjoint-based approach would then be about five times as efficient as the model-reduced approach. At the other hand there is also scope to improve the efficiency of the model-reduced approach: as indicated by Table 6.10, an almost identical objective function value could have been obtained with 35 (=29+6) patterns, which reduces the number of iterations from 98 to 68. More importantly, the model-reduced approach is more effective than a finite difference approach, even when using the reparameterization as applied in our example. A finite difference approach required six iterations and one additional function evaluation for the line search routine, which gives $6 \times (22+1)+3 = 141$ simulations which makes the model-reduced approach with 35 patterns more than twice as efficient as a finite difference approach.

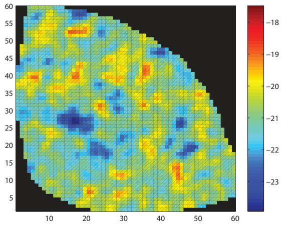


Figure 6.46: Experiment 1. Log of permeability field from the model-reduced approach with fewer patterns in layer $4[m^2]$.

6.2.5 History matching results: experiment 2

In the second experiment we repeated the history matching procedure with a different initial guess for the permeability field. All other settings for model reduction and history matching stayed the same. For this case we assumed, following Van Essen et al. (2009), that the main direction of the channels was known, e.g. from seismic measurements, but that no specific knowledge about the channel configuration was available. The set of 100 realizations of the reservoir were sketched by hand based on the geological insight, with a strong vertical correlation, and each realization displayed an alternative channel configuration. The generated ensemble of permeability fields was used to create the background permeability $\boldsymbol{\theta}_{init}$, taken as the ensemble average, and to estimate the background error covariance matrix. The results of this experiment are summarized in Table 6.13.

Outer iter	J	n_{red}	p_{red}	n_{sim}
0	838	–	22	1
1	100	51	22	$84(=11+47+4+22)$

Table 6.13: Model-reduced gradient-based history matching. p_{red}, n_{red} and n_{sim} denote the number of parameters, the number of patterns and the number of simulations, respectively.

We selected 51 (=47+4) patterns for the first outer iteration (47 for saturation and 4 for pressure), which means that the reduced model operates in dimension $51+22$. We observed again that the pressure behavior is well represented by a few patterns, only 4, but that the saturation requires significantly more patterns, namely 47. The initial value of the objective function was 838, and results from the prior parameter field presented in Fig. 6.47.

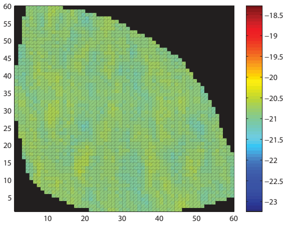


Figure 6.47: Experiment 2. Log of prior permeability field in layer 4 $[m^2]$.

The first outer iteration terminated with parameters for which the objective function decreased to 100. The stopping criterion for the outer loop, (5.33), was satisfied ($179 \leq 2J(\boldsymbol{\theta}) \leq 263$), and the procedure terminated. In this example, the computational cost of the model-reduced approach was about the time required to simulate 84 high-order model evaluations (11 simulations during which snapshots are collected, about 47 + 4 simulations required to approximate the partial derivatives along saturation and pressure patterns and 22 simulations required to approximate the partial derivatives with respect to parameters). In Fig. 6.48, the comparison of oil production curves resulting from different permeability fields is depicted. The estimate is able to reproduce the past data from all four producers, while the prediction is nearly perfect for two wells (top left and top right plots), and improved for one well (bottom right plot). Similar to the first experiment, the worst predictions were obtained for the well depicted on the bottom left plot. The final estimated parameters are presented in Fig. 6.49.

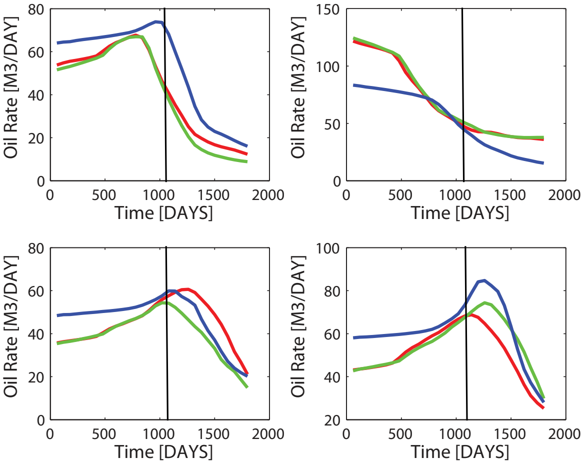


Figure 6.48: Experiment 2. Oil rate in the production wells, obtained with the prior permeability field (blue), the true permeability field (red), and the estimated permeability field (green) using the model-reduced approach. The black vertical lines indicate the end of the assimilation period.

Also for this second example we checked the performance of the model-reduced approach in comparison to the classical adjoint-based approach and a finite-difference approach (see Tables 6.13, 6.14 and 6.15), respectively.

Iterations	J	n_{red}	p_{red}	n_{sim}
0	838	–	22	1
13	92	–	22	$59(= 13 \times 2 + 33)$

Table 6.14: Classical adjoint-based history matching. p_{red} and n_{sim} denote the number of parameters and the number of simulations, respectively.

Iterations	J	n_{red}	p_{red}	n_{sim}
0	838	–	22	1
11	101	–	22	$254(= 11 \times 23 + 1)$

Table 6.15: Finite-difference based history matching. p_{red} and n_{sim} denote the number of parameters and the number of simulations, respectively.

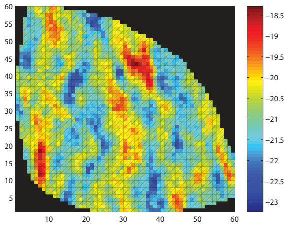


Figure 6.49: Experiment 2. Log of permeability field from the model-reduced approach in layer 4 $[m^2]$.

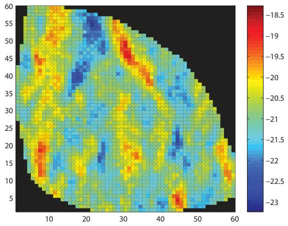


Figure 6.50: Experiment 2. Log of permeability field from the adjoint-based approach in layer 4 $[m^2]$.

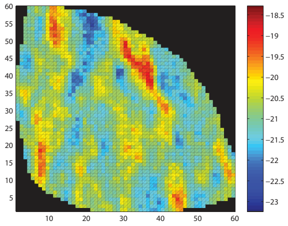


Figure 6.51: Experiment 2. Log of permeability field from the finite-difference based approach in layer 4 $[m^2]$.

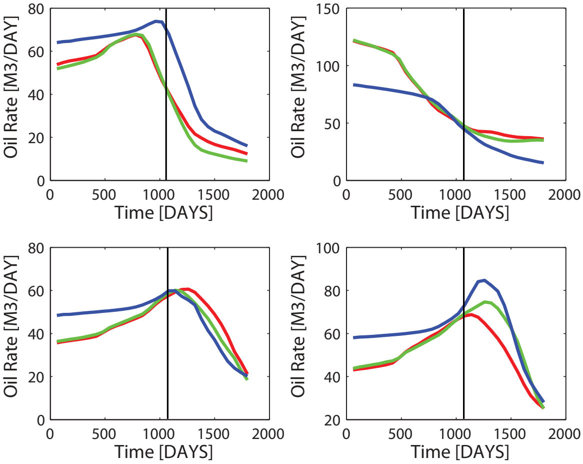


Figure 6.52: Experiment 2. Oil rate in the production wells, obtained with the prior permeability field (blue), the true permeability field (red), the estimated permeability field (green) using the adjoint-based approach. The black vertical lines indicate the end of the assimilation period.

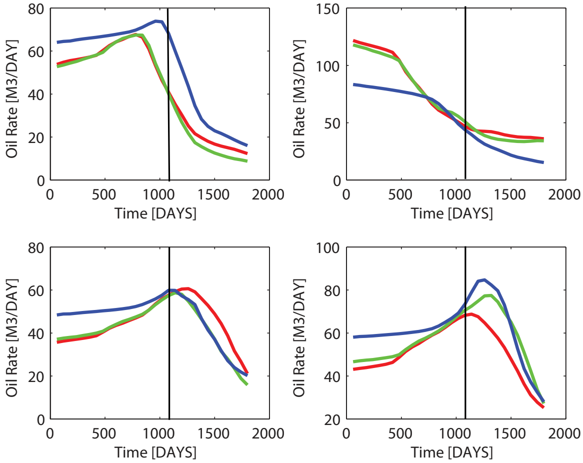


Figure 6.53: Experiment 2. Oil rate in the production wells, obtained with the prior permeability field (blue), the true permeability field (red), the estimated permeability field (green) using the finite-difference based approach. The black vertical lines indicate the end of the assimilation period.

Comparison of Fig. 6.49, 6.50 and 6.51, and comparison of Fig. 6.48, 6.52 and 6.53, respectively, shows that in the second example the results of the model-reduced

approach and of a finite-difference based approach come very close to those of the adjoint-based approach.
The adjoint case converged after 13 iterations (see Fig. 6.54).

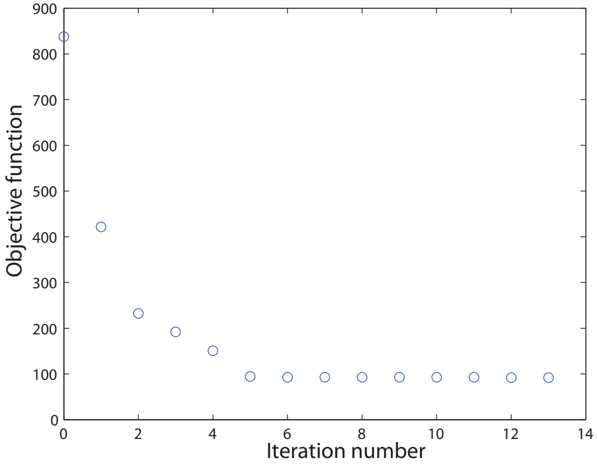


Figure 6.54: Experiment 2. Successive iterations of the objective function value.

It required 13 evaluations of the forward model, another 13 of the adjoint model and 33 line searches, bringing the total to 59 iterations to reach the minimal objective function value of 92. Just like in example one, we could have used fewer iterations, five in this case, to obtain a similar objective value function, which would have brought the total number of iterations for the adjoint-based case to about 15. In other words, for this second example, the adjoint-based approach is up to about 6 times as efficient as the model-reduced approach. However, also in this case there is scope to improve the efficiency of the model-reduction by further reducing the number of patterns. Moreover, the model-reduced approach is now about three times more efficient efficient than the finite difference approach.

6.2.6 Conclusions

The model-reduced history matching method was used to adapt the permeability field in a two-phase (oil-water) reservoir simulation model containing 18553 grid blocks. Two different ensembles of realizations were considered as the prior information. The following conclusions can be drawn from these synthetic cases:

- The results from two numerical experiments with different prior permeability fields showed that the model-reduced approach gives results comparable to those of a classical adjoint-based history matching procedure.
- In the examples considered the adjoint-based approach was about five to six times as efficient as the model-reduced approach, but the latter was about

two to three times as efficient as a finite difference approach. Moreover, in case of a larger number of wells, allowing for the estimation of a larger number of (reparameterized) model parameters, it is expected that the efficiency compared to a finite difference approach improves. If the parallelization would be applied (see Chapter 5) then all computations would take the time of about 3 model simulations because only single outer loops were performed.

- The classical gradient-based history matching procedure is a very efficient one, but for complex models it is very difficult to implement. The model-reduced approach is computationally more expensive, but it does not require access to the simulation code and it is relatively easy to implement.
- The classical adjoint-based approach gives the exact downhill directions, while the model-reduced history matching procedure generates only approximations of them. In all cases analyzed by us they proved to be accurate enough to decrease the objective function in all iteration steps. There is, however, no guarantee that the model-reduced approach will always result in a decline of the objective function.

6.3 Summary

We presented numerical examples of model-reduced gradient-based history matching based on the POD and the BPOD method. The POD technique identifies a low-dimensional subspace of the state space, in which the dynamics of the reservoir state are well represented, while the BPOD technique identifies a low-dimensional subspace which is relevant for the state representation but also relevant for the observations. Using those methods simplified linear forms of reservoir flow equations are derived and used to minimize the mismatch between observed and simulated outputs. In all the presented cases the model-reduced methods performed better than the finite difference based approach. The results indicate that comparable quality estimates to those from the classical adjoint-based approach can be achieved.

7 CHAPTER

Production optimization of a coupled flow/geomechanics system

In this chapter, the Net Present Value function is optimized with respect to model inputs during water flooding operation, where water injection takes place under induced fracturing conditions. In this case, the gradient of the Net Present Value function with respect to model inputs is not available. Based on experience, we can try to find a good operation strategy by changing the scenarios manually and choosing the best among them. However, this trial-and-error approach can be computationally very expensive. First, we present the model-reduced approach adapted to compute the optimal operation strategy and we explain the numerical difficulties associated with this coupling. Then, we present the results of the simultaneous perturbation stochastic approximation algorithm and the finite-difference-based method.

7.1 Introduction

Another use of adjoint-based methods in reservoir engineering is for production optimization or recovery optimization. In this case the objective function is ultimate recovery or net present value (NPV), and the control variables are the well rates, well pressures, or well valve settings. Initially, this was done for the optimization of tertiary recovery processes, see Ramirez (1987), later followed by water flooding optimization, see e.g. Asheim (1988), Brouwer and Jansen (2004), Doublet et al. (2009), Sarma et al. (2008), Sudaryanto and Yortsos (2000), Zandvliet et al. (2007) and Jansen (2011a). In this chapter we aim to optimize the NPV function with respect to model inputs during the water flooding operation, where the water injection takes place under induced fracturing conditions. In this case, a gradient of the NPV function with respect to model inputs is not available. An alternative gradient-based algorithm is the simultaneous perturbation stochastic approximation (SPSA) method of Spall (1998), where an approximated gradient is calculated using a stochastic perturbation of all parameters together, which was proposed for use in the recovery optimization problem by Wang et al. (2009b).

Another method uses an ensemble of control variables and computes the cross-covariance between the control variables and the objective function which can then be used as an approximate gradient. This "ensemble optimization" technique was proposed for recovery optimization by Lorentzen et al. (2006), and thereafter refined by Chen et al. (2009). The model-reduced production optimization method, similar to the model-reduced history matching method, can be considered as alternative to these methods. In this chapter we present the adaptation of the model-reduced approach to production optimization problem and we explain the numerical difficulties which hampered the application of this method to the coupled system under consideration. The SPSA did not suffer from these numerical problems, and therefore was applied. The "ensemble optimization" technique of Lorentzen et al. (2006) has been compared with the SPSA by Wang et al. (2009b). Since in their case it performed significantly worse, we did not consider it here, and we compared the SPSA results with the finite-difference based approach.

7.2 Production optimization

The injection rates in water flooding strongly influence the growth/shrinkage of induced fractures around the injection wells (see Van den Hoek et al. (2008)). We aim to optimize the waterflooding under induced fracturing conditions by changing the injection rates over a time. We search for a configuration of the injection rates that optimizes the production profit, which is evaluated by the Net Present Value (NPV) defined as:

$$J = \int_0^{t_N} \phi(t)dt, \tag{7.1}$$

where t_N stands for the total simulation time and the integrand $\phi(t)$ is a linear combination of the well rates, and in our case it is defined as

$$\phi(t) = \sum_p \left(r_o^p(t)q_o^p(t) - r_w^p(t)q_w^p(t) \right) - \sum_i r_w^i(t)q_w^i(t), \tag{7.2}$$

where the first summation is done over all production wells, while the second summation is done over all injection wells, the rates $q_o^p(t)$, $q_w^p(t)$ stand for the volume rates of oil and water at reservoir conditions in the production well p at time t, $q_w^i(t)$ stands for the volume rate of injected water in the injection well i at time t, the weight factors $r_o^p(t)$ and $r_w^p(t)$ stand for the (monetary or other) value assigned to a unit volume of the oil and water components at time t, the weight factor $r_w^i(t)$ stands for the value assigned to a unit volume of injected water in the injection well i at time t. For each produced fluid the annual interest rate is taken into account by defining $r_l^p(t)$ as

$$r_l^p(t) = r_l^p(t_0)f(t)(1+b_l)^{-(t-t_0)/year} \tag{7.3}$$

where $l \in \{o, w\}$ (o stands for oil and w stands for water) and $r_l^p(t_0)$ is the unit price of component l at reference time t_0, $f(t)$ is the price factor for component l

at time t, b_l is the annual discount factor for component l and t_0 is the discount reference time.

The oil and water production rates in the NPV are functions of the control vector $\mathbf{u}$ and the dynamic state vector $\mathbf{x}$, and therefore the NPV can be written as a function of $\mathbf{u}$ and $\mathbf{x}$, $J = J(\mathbf{u}, \mathbf{x})$. We aim to adjust the control vector $\mathbf{u}$ such that the resulting objective function reaches its maximum. Typically, to prevent unphysical values the control vector is restricted by specifying lower and upper bounds: $\mathbf{u}^{low}$ and $\mathbf{u}^{up}$, respectively. Then the production optimization can be written as

$$\begin{aligned} &\max_{\mathbf{u}=[\mathbf{u}_1,\ldots,\mathbf{u}_N]} && J(\mathbf{u}, \mathbf{x}_{1:N}) = \sum_{i=0}^{N-1} J_{i+1}\left(\mathbf{u}_{i+1}, \mathbf{x}_{i+1}\right)), \\ &\textit{subject to} && \mathbf{u}^{low} \leq \mathbf{u} \leq \mathbf{u}^{up}. \end{aligned} \tag{7.4}$$

One way to transform the constrained optimization problem (7.4) into an unconstrained one is by applying the log-transformation on the input vector $\mathbf{u}$ (as in Gao and Reynolds (2006)) given by

$$s_i = ln\left(\frac{u_i - u_i^{low}}{u_i^{up} - u_i}\right), \tag{7.5}$$

where u_i, u_i^{up} and u_i^{low} stand for i-th component of the control vector $\mathbf{u}$, $\mathbf{u}^{up}$ and $\mathbf{u}^{low}$, respectively. As u_i approaches u_i^{low} the transformed variable s_i approaches $-\infty$, and as u_i approaches u_i^{up} the transformed variable s_i approaches $+\infty$. The update of the control vector in the optimization routine is performed on the log-transformed vector $\mathbf{s}$. To calculate the objective function at the updated control vector the inverse log-transformation needs to be applied, which is given by

$$u_i = \frac{e^{s_i} u_i^{up} + u_i^{low}}{1 + e^{s_i}}. \tag{7.6}$$

7.3 Model-reduced gradient-based production optimization

In the previous chapter the model-reduced approach was tested on history-matching problems. Now, we apply the same approach to solve production optimization problem. In production optimization we aim to maximize an objective function with respect to control vector, and therefore the linearization of the reservoir system has to be made around a control vector, $\mathbf{u}^b$ (instead of the parameter vector $\boldsymbol{\theta}^b$). The resulting tangent linear model describes the variation in the state caused by changes in the inputs, i.e.

$$\Delta\bar{\mathbf{x}}(t_i) = \frac{\partial \mathbf{c}_i[\mathbf{x}(t_{i-1}), \mathbf{u}]}{\partial \mathbf{x}(t_{i-1})} \Delta\bar{\mathbf{x}}(t_{i-1}) + \frac{\partial \mathbf{c}_i[\mathbf{x}(t_{i-1}), \mathbf{u}]}{\partial \mathbf{u}(t_i)} \Delta\bar{\mathbf{u}}(t_i), \tag{7.7}$$

where $\Delta\bar{\mathbf{x}}(t_i)$ is a state vector of the tangent linear model and $\Delta\bar{\mathbf{u}}(t_i)$ is the variation from the reference input vector $\mathbf{u}^b(t_i)$. The reduction of the linearized reservoir model is then performed using the POD method. The snapshots are generated by simulating the model with perturbed settings of the control vector. The perturbations of inputs are relatively small, because only in such a space the linearized model is valid. Using the POD method the dominant patterns for pressure and saturation are identified, and used to build the projection matrix $\mathbf{\Phi}$. Then a high-order model (7.7) is projected onto the subspace defined by $\mathbf{\Phi}$ resulting in the reduced-order linearized model:

$$\mathbf{z}(t_i) = \mathbf{N}_i\mathbf{z}(t_{i-1}) + \mathbf{N}_i^u\Delta\mathbf{u}(t_i) \tag{7.8}$$

where

$$\mathbf{N}_i = \mathbf{\Phi}^T\frac{\partial\mathbf{c}_i[\mathbf{x}(t_{i-1}),\mathbf{u}]}{\partial\mathbf{x}(t_{i-1})}\mathbf{\Phi} \quad , \quad \mathbf{N}_i^u = \mathbf{\Phi}^T\frac{\partial\mathbf{c}_i[\mathbf{x}(t_{i-1}),\mathbf{u}]}{\partial\mathbf{u}(t_i)}, \tag{7.9}$$

where $\mathbf{z}$ represents the response of the reduced-order model to a perturbation of $\mathbf{u}$. The partial derivatives in the formulas (7.9) can not be easily computed explicitly and should be approximated in the same manner as described in Chapter 5, i.e. instead of approximating terms $\frac{\partial\mathbf{c}_i[\mathbf{x}(t_{i-1}),\mathbf{u}]}{\partial\mathbf{x}(t_{i-1})}$ we approximate the terms $\frac{\partial\mathbf{c}_i[\mathbf{x}(t_{i-1}),\mathbf{u}]}{\partial\mathbf{x}(t_{i-1})}\mathbf{\Phi}$ as

$$\frac{\partial\mathbf{c}_i[\mathbf{x}(t_{i-1}),\mathbf{u}]}{\partial\mathbf{x}(t_{i-1})}\boldsymbol{\phi}_j \approx \frac{\mathbf{c}_i[\mathbf{x}(t_{i-1})+\epsilon\boldsymbol{\phi}_j,\mathbf{u}] - \mathbf{c}_i[\mathbf{x}(t_{i-1}),\mathbf{u}]}{\epsilon}. \tag{7.10}$$

This results in a reduced-order model that is linearized along the patterns. The output model is linearized as well, and its partial derivatives are approximated in the same manner. Then, the adjoint model for a reduced-order model is easily implemented. Using the reduced-order reservoir model and its adjoint model the approximated NPV function and its gradient can be calculated, and the maximization problem can be solved. A new solution is an optimum for the reduced-order linearized system but not necessarily for the original one. Therefore, the value of the original objective function is checked and the process is repeated if necessary. The described production optimization algorithm has been summarized by the flow chart in Fig. 7.1.

The model-reduced gradient-based production optimization method requires many system perturbations, and therefore it requires high accuracy of the simulation tool. In the coupled simulator to find the fracture size $\mathbf{L}_{i+1}$ satisfying condition given by Eq. 2.69 the following steps are iterated:

Step k=0

Take $\mathbf{L}_{i+1}^k = \mathbf{L}_i$ and calculate $\mathbf{crit}(\mathbf{a}_{i+1},\sigma_{i+1},\mathbf{L}_{i+1}^k)$ defined by Eq. 2.68. If condition given by Eq. 2.69 is satisfied, then the fracture size is correct (there is no closure and no propagation) and the iteration stops. If condition given by Eq. 2.69 is not satisfied, then the fracture should propagate or close. To know how much to change the fracture size we go to the next step $k+1$.

Step k+1

Using a Taylor expansion we can rewrite the criterion at each fracture tip as

$$\mathbf{crit}_{i+1}(\sigma_{i+1},\mathbf{L}_{i+1}^{k+1},\mathbf{a}_{i+1}) \approx \mathbf{crit}_{i+1}(\sigma_{i+1},\mathbf{L}_{i+1}^{k},\mathbf{a}_{i+1}) + \frac{\partial\mathbf{crit}_{i+1}}{\partial\mathbf{L}_{i+1}}\Delta\mathbf{L}_{i+1}^{k+1}, \tag{7.11}$$

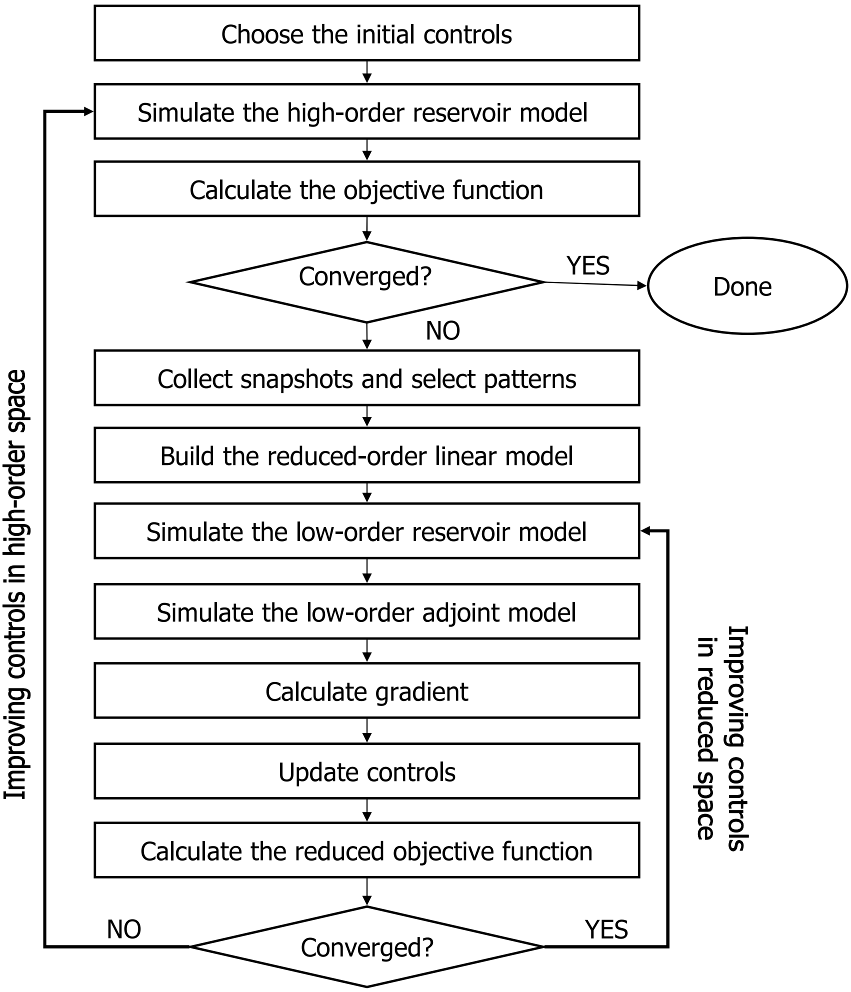

Choose the initial controls
Simulate the high-order reservoir model
Calculate the objective function
Converged?
YES
Done
NO
Collect snapshots and select patterns
Build the reduced-order linear model
Simulate the low-order reservoir model
Simulate the low-order adjoint model
Calculate gradient
Update controls
Calculate the reduced objective function
NO
Converged?
YES
Improving controls in high-order space
Improving controls in reduced space

Figure 7.1: Model-reduced gradient-based production optimization algorithm.

where the total derivatives $\frac{\partial \mathbf{crit}_{i+1}}{\partial \mathbf{L}_{i+1}}$ are evaluated for $\mathbf{L}_{i+1}^{k}$ and $\Delta\mathbf{L}_{i+1}^{k+1}$ is defined as

$$\Delta\mathbf{L}_{i+1}^{k+1} = \mathbf{L}_{i+1}^{k+1} - \mathbf{L}_{i+1}^{k}. \tag{7.12}$$

The change of one fracture size influences the pressure vector, hence it might influence the criterion for the remaining fracture sizes. $\frac{\partial \mathbf{crit}_{i+1}}{\partial \mathbf{L}_{i+1}}$ is a square matrix, which elements describe the sensitivity of the criterion to all fracture sizes. From Eq. 7.11 we see that it is reasonable to update the solution using the following equality

$$\Delta \mathbf{L}_{i+1}^{k+1} = -\left(\frac{\partial \mathbf{crit}_{i+1}}{\partial \mathbf{L}_{i+1}}\right)^{-1} \mathbf{crit}_{i+1}(\sigma_{i+1}, \mathbf{L}_{i+1}^{k}, \mathbf{a}_{i+1}), \tag{7.13}$$

assuming inversion of $\frac{\partial \mathbf{crit}_{i+1}}{\partial \mathbf{L}_{i+1}}$. The fracture sizes are updated then according to

$$\mathbf{L}_{i+1}^{k+1} = \mathbf{L}_{i+1}^{k} + \Delta \mathbf{L}_{i+1}^{k+1} \tag{7.14}$$

and the criterion is recalculated. If criterion (2.69) is satisfied, then the iteration stops; otherwise step $k+1$ is repeated.

Step $k+1$ should be iterated until the given condition is satisfied (to a certain numerical precision). In the fracture simulator described in Chapter 2 the explicit solver is used, which performs only one update of the fracture sizes per simulation time step. If the simulation time steps are very small then probably one update will be sufficient, while more updates are expected when time steps are larger. The explicit solver uses therefore very small time steps and stops the fracture sizes update even if the stopping criterion is not satisfied precisely. Because of this implementation it turned out to be impossible to accurately calculate the sensitivity matrices of the system with respect to state patterns and system controls. Since fracture sizes are not computed accurately, the state of the system is not computed accurately as well. Therefore, the resulting finite-difference sensitivities are not a good approximation of the sensitivity matrices, and they result in an unstable reduced-order model. Another problem may be caused by the large number of time steps which is required in the explicit form of the fracture simulator. Since each sensitivity matrix introduces an error in the reduced-order system, a large number of time steps might result in a large modeling error when the time propagates, because those errors accumulate. A solution to this problem would be an implicit form of the fracture size update.

7.4 SPSA versus finite-difference production optimization

7.4.1 Model settings

We performed optimization experiments on a 3D two-phase reservoir model with one induced fracture around the injection well. A unit cell of a 5-spot pattern with one injection and one production well is modeled. Periodic boundary conditions at the model boundary are used. The model is based on the assumptions that the effects of capillary pressure can be neglected and that the permeability is uniform.

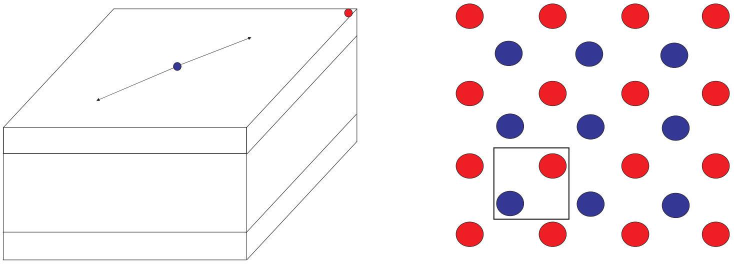


Figure 7.2: Reservoir structure and a five spot pattern. The producer is indicated in red, the injector in blue.

- *Reservoir geometry:*
  The reservoir model is a quadrilateral with flow boundaries of the size $800m \times 800m \times 50m$, divided into $50 \times 50 \times 5$ uniform Cartesian grid blocks, of which 7500 are active, while the top and bottom layers represent the cap and base rock and are not part of the reservoir.

- *Reservoir properties :*
  The porosity and permeability are assumed to be uniform and equal 0.3 and $100mD$, respectively. The rock compressibility $c_r = 15 \cdot 10^{-10} \cdot 1/Pa$.

- *Initial conditions :*
  The initial reservoir pressure is $7.5MPa$ and the initial water saturation is taken as connate water saturation $S_{wc} = 0.2$.

- *Well locations and constraints :*
  Water is injected at a rate of $1650m^3/d$ in the vertical injector, while the vertical producer is operated at constant bottom-hole pressure of $74.9MPa$. All wells are perforated at each layer of the reservoir.

- *Fluid properties :*
  The oil and water viscosities are identical: $\mu_o = \mu_w = 5.9 \cdot 10^{-4} Pa \cdot s$. The compressibility of water is: $c_w = 4.4 \cdot 10^{-10}$ while the compressibility of oil is:$c_o = 3.5 \cdot 10^{-10} 1/Pa$. The densities of oil and water are given by $\rho_o = 920kg/m^3$, and $\rho_w = 1002kg/m^3$.

- *Relative permeabilities:*
  We assumed Corey-type relative permeabilities with Corey exponents for oil and water: $n_o = 1.3$ and $n_w = 1.3$. The residual oil saturation was taken as $S_{or} = 0.1$, and the connate water saturation as $S_{wc} = 0.2$. The end point relative permeability to water and oil is 1.

- *Fracture properties:*
  We assumed that the fracture is located at the injection well and has an orientation to the production well, 45^o with respect to unit cell boundary. The initial high of the fracture is $5m$ and is constrained to stay between $0.1m$ to $25m$. The initial length of the fracture is $5m$ and is constrained to be between $0.1m$ to $560m$. The width of the fracture is fixed to $0.1m$. The stress gradient

is $15.8 \cdot 10^3 Pa/m$, the Young's modulus is $1.8 \cdot 10^9 Ps$, the Poisson ration is 0.2 and the fracture toughness is $1.0 \cdot 10^6 Pa/\sqrt{m}$.

7.4.2 Production settings

For the production optimization considered here, we assumed that the price of oil is 100$ per barrel, the cost of produced water is 0$ per barrel and the cost of injected water is 3$, which represents onshore conditions. The price factor $f(t)$ is set to one and the discount factor b_o is set to 0.15. The lower and upper bounds are $\mathbf{u}^{low} = 100m^3/d$ and $\mathbf{u}^{up} = 3000m^3/d$, respectively. Injecting water at a rate higher than $\mathbf{u}^{up}$ results in an induced fracture that grows significantly and, may break through in an adjacent producer. By reducing the injection rate the fracture can close but the reservoir can not come back to its original condition. Preferably, the optimized injection strategy should avoid the fracture breakthrough condition and therefore the maximum injection rate is specified.
To specify the initial injection strategy, three trial simulations with three different injection rates (constant over time) were performed. Fig. 7.3 depicts the cumulative NPVs for injection rates equal $1000m^3/d$, $1650m^3/d$ and $3000m^3/d$.

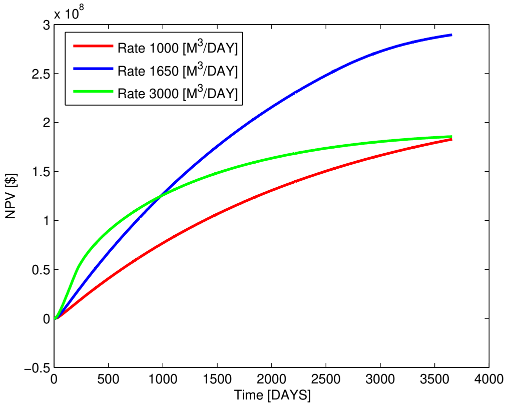


Figure 7.3: NPV for three different constant-over-time injection rates: $1000m^3/d$, $1650m^3/d$ and $3000m^3/d$.

While injecting water with rate of $1000m^3/d$ the fracture grew relatively fast in first few days. After an initial transient phase the fracture stopped to grow (see Fig. 7.4), i.e. the constant rate of $1000m^3/d$ was not sufficient to open it significantly and the fracture remained relatively small. The horizontal length of this induced fracture was $5.57m$, the upper vertical length was $6.63m$, while the lower vertical length was $4.39m$. The NPV resulted from this injection strategy was equal to $1.8293 \cdot 10^8$$.

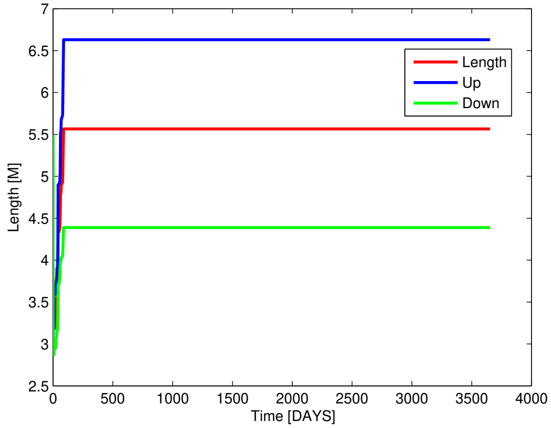


Figure 7.4: Fracture sizes in the period of 10 years for injection rate of $1000m^3/d$.

The injection rate of $3000m^3/d$ opened the fracture to its maximum allowed sizes, i.e. the horizontal size reached $520m$, and both (up and down) vertical sizes reached $25m$. The total recovery in this case was equal to $1.8563 \cdot 10^8\$$.

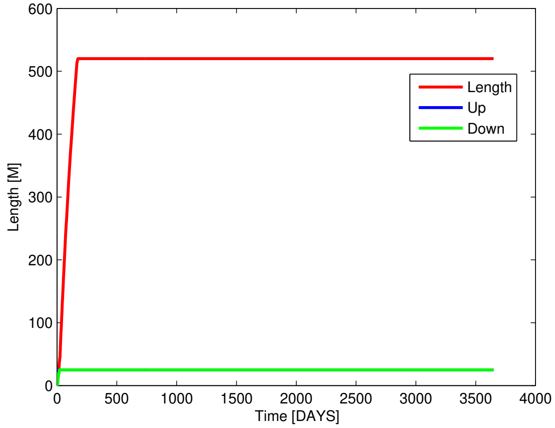


Figure 7.5: Fracture sizes over a period of 10 years for injection rate of $3000m^3/d$.

An injection rate of $1650m^3/d$ was chosen as an initial injection strategy. It resulted in the NPV equal $2.8949 \cdot 10^8\$$ (see Fig. 7.6), that was considerably higher compared to $1000m^3/d$ and $3000m^3/d$ rates. During an initial transient phase the induced fracture grew to a total length of $43.4m(= 2 \cdot 21.7m)$ and height of $32.66m(= 25m + 7.66m)$.

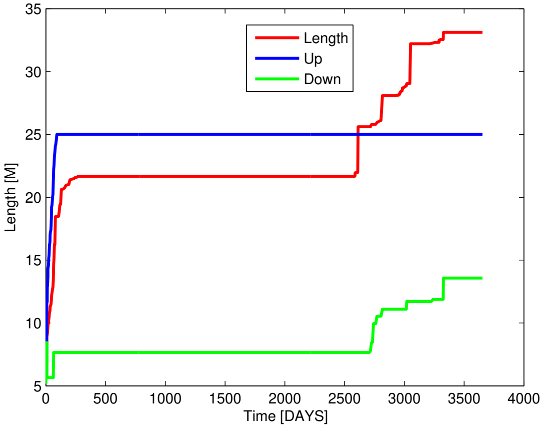


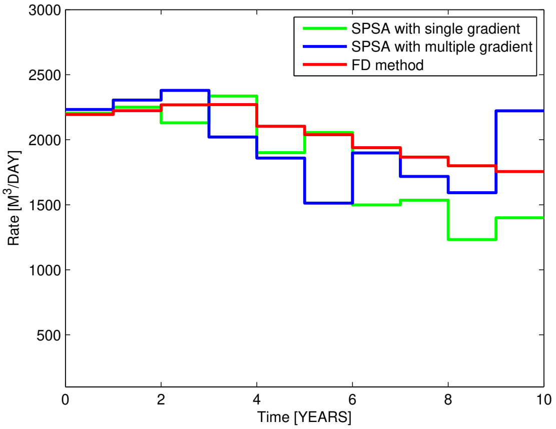
Figure 7.6: Fracture sizes over a period of 10 years for injection rate of $1650m^3/d$.

The fracture continued to grow when the water broke through in the producer. The changes in the fracture sizes over a period of 10 years are depicted in Fig.7.6. According to Van den Hoek et al. (2008) the moment in which water breaks through the producer is the optimal timing for reduction of the injection rate, when a relatively high injection rate is applied.

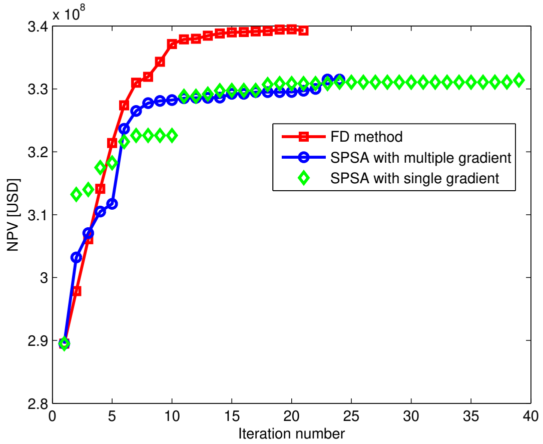
7.4.3 Production optimization results

Two different optimization algorithms were applied to improve the initial recovery strategy: the finite-difference based method (FD method) and the SPSA method, see Chapters 5.5 and 5.4. The SPSA method using a single gradient and an average of three gradients was applied. The injection rate was allowed to be changed at a yearly basis, which resulted in 10 control variables. The production bottom hole pressure was fixed at $74.9MPa$. The stopping criterion similar to (5.31) and (5.32) were selected to terminate the optimization routine, and a complementary stopping criterion that stops the routine if the number of simulations exceeds 200 was chosen.

The injection strategies obtained by the FD method and the SPSA method (with single and multiple gradients) are presented in Fig. 7.7(a).



(a) Injection rates



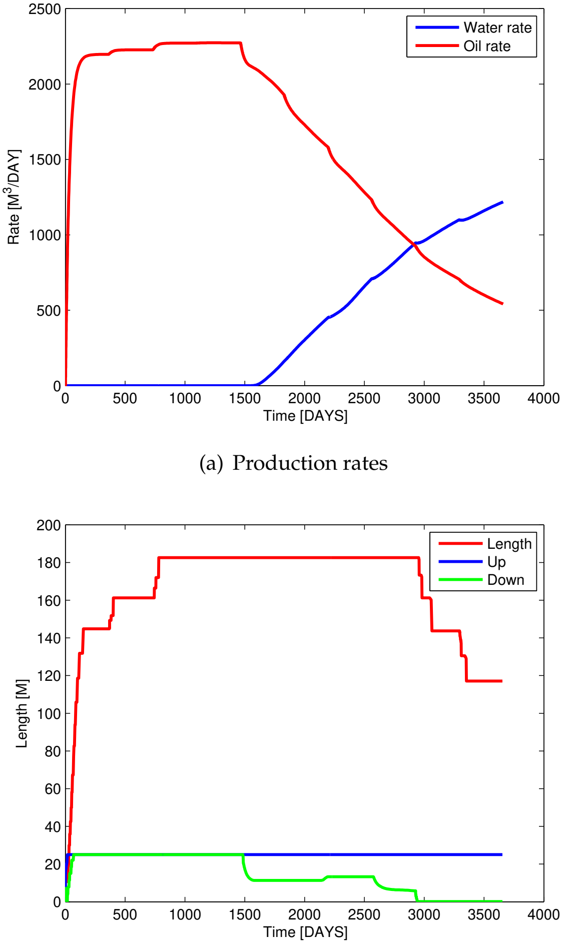
(b) The NPV function iterations

Figure 7.7: Injection strategies obtained from the FD and SPSA method (a). Convergence of the FD and SPSA method (b).

The successive iterations of the NPV function of the SPSA and the FD methods are presented in Fig. 7.7(b). The curve with squares represents the performance of the FD method while the curves with diamonds and circles represent the performance of SPSA with single and multiple gradients, respectively. The FD method converged after 20 iterations, and its total computational time was about 170 simulations. Using the finite difference formula one gradient calculation requires about 6.5 simulations of the coupled system. Namely, to calculate the perturbation with respect to $\mathbf{u}_1$ the full simulation is required, however, to calculate the perturbation with respect to $\mathbf{u}_2$ a model can be restarted from the moment when this input control becomes active. In case that inputs are splitted in 10 equal intervals, it means that this perturbation requires 0.9 simulation. The perturbation with respect to $\mathbf{u}_3$ requires 0.8 simulation and so on. The perturbation with respect to the last control $\mathbf{u}_{10}$ requires only 0.1 simulation. The total time of those perturbations is equal to the sum of an arithmetic sequence and is equal to $5.5(=10\frac{1+0.1}{2})$ simulations. One additional simulation is required to calculate the NPV for the reference case at which the gradient is approximated. Using the FD method, NPV increased from 2.9×10^8\$ to 3.4×10^8\$, which gave an increase of 17%. The performance of the SPSA method with multiple gradients (an average of three gradients) is represented in Fig. 7.7 by the curve with circles. The convergence of this method was slower compared to the FD method and the final objective function was $3.315 \cdot 10^8$\$, which gave an improvent of 14%. The total computational cost of the SPSA method using the multiple gradient was 200 simulations (one iteration of the SPSA method using the multiple gradient requires 7 simulations). The SPSA with a single gradient converged after 39 iterations and the NPV increased to $3.3141 \cdot 10^8$\$. The total computational cost of the SPSA method with single gradient was 200 simulations too. The convergence curves in Fig. 7.7(b) indicate that it might be beneficial to use the SPSA method in the first iterations to get the first rough update and then continue the improvement with the FD method.

The final control vector resulting from the SPSA methods and the FD method are presented in Fig. 7.7(a). The strategy obtained by the FD method shows a significant increase in the injection rate at the initial time. After a period of four years the injection is gradually reduced. The operation strategy resulting from the SPSA method with a single gradient shows a similar trend. The SPSA method with multiple gradients resulted in similar strategy for the first 9 years, but in the last year the injection rate is significantly higher. All the operation strategies, however, agreed about larger injection rate in the initial operation stage, which is followed by the decrease in later stage. This is in line with the results of Van den Hoek et al. (2008).

The production rates and fracture sizes for the best operation scenario are presented in Fig. 7.8. A drop in the injection rate after 1500 days results in a reduction of the fracture hight. The water breaks through the adjacent production well after this change.



(a) Production rates

(b) Fracture sizes

Figure 7.8: Water (blue) and oil(red) rates in the production well and fracture sizes over a period of 10 years.

7.5 Conclusion

This chapter concerns life cycle optimization of a reservoir operating under induced fracturing conditions.

- The dynamic changes in the reservoir flow are modeled using a fully coupled simulator that includes the induced fracture within existing reservoir fluid-flow simulator, as described in Zwarts et al. (2006); Hustedt et al. (2008). An advantage of this implementation is its integration in the framework of standard tools, while an disadvantage is an explicit form of the fracture growth calculation, which makes it computationally expensive, and does not assure a high accuracy of the fracture sizes at each time step of the simulation.

- Since the induced fracture is not included in the implementation of the gradient of the NPV function a standard optimization package can not be used. Therefore, we first applied the model-reduced approach to production optimization problem but it resulted in the numerical difficulties. Secondly, we performed the production optimization using two adjoint-free routines: the SPSA method and the finite-difference based method. Both approaches were computationally expensive, however, both methods resulted in the optimization strategies that improved the total recovery. Moreover, the shapes of those injection strategies are in line with work of Van den Hoek et al. (2008). Using an automatic optimization routine for complex models might help to find effective optimization strategies that might be difficult to find by manual adjustments. Combining the experience of reservoir engineers about fracture behavior with an automatic optimization routine might bring significant improvement in the recovery of reservoirs operating under induced fracturing conditions.

- Summarizing, computer-assisted optimization of the waterflooding process in a reservoir operating under induced fracturing conditions is possible, but computationally it is costly. Therefore, there is a need for deriving the missing part of the adjoint model, or for implementing an implicit fracture growth simulation and applying the model-reduced production optimization method.

8 CHAPTER

Conclusions

The models used for reservoir flow simulation are of a large scale. The number of state variables and model parameters are typically of the order of $10^5 - 10^6$. The most important model parameter to describe flow is the permeability, which is a property of the reservoir rock and therefore it is not accurately known. Control of the subsurface flow is, to a limited extent, possible by changing pressures or flow rates in the wells. The permeability parameters can be calibrated by using past production data while the control variables can be optimized by using predictions of the production data. The research objective of this thesis was to evaluate the performance of a gradient-based history matching routine which involves a model-order reduction. Additionally, the use of adjoint-free methods for production optimization of a reservoir operating under induced fracturing conditions was considered. This chapter presents the conclusions of this thesis and recommendations for future research.

8.1 Conclusions

- The model-reduced history matching routine consists of two nested loops. In the outer loop, a reduced-order model is constructed by using snapshots and perturbations of the original high-order system. In the inner loop, an adjoint-based history matching problem is solved where the reservoir model is replaced with its reduced-order proxy model. We have presented model-reduced gradient-based history matching procedures based on the Proper Orthogonal Decomposition (POD) and the Balanced Proper Orthogonal Decomposition (BPOD) methods. The POD technique identifies a low-dimensional subspace of the state space, in which the dynamics of the reservoir state corresponding to different model parameters patterns are well represented. The BPOD technique aims to preserve only those parts of the state that are influenced by the parameter patterns and which are relevant for observations. Because the reduced-order model is designed to capture the dominant dynamics of the original system (relevant for the considered problem) we could successfully use it for a fast minimization of the objective function in the inner loop.

- The POD-based method was used to adapt the permeability field in two-phase (oil-water) reservoir simulation models. Two synthetic reservoir models that varied in complexity and size were used. In case of simplistic reservoir models the model-reduced approaches performed very well. The computed estimates of the permeability field significantly improved with respect to the prior fields and gave an acceptable history match. The quality of the predictions of the models estimated by the model-reduced methods was comparable to the quality of the predictions obtained by the classical adjoint-based approach. Computationally, the POD-based method was approximately twice as expensive as the classical approach. The results from the numerical experiments using more complex and larger simulation models showed that the model-reduced approach gives results comparable to those of a classical adjoint-based history matching procedure. In the examples considered, the adjoint-based approach was about five to six times more efficient than the model-reduced approach, but the latter was about two to three times more efficient than a finite difference approach.

- The BPOD-based method was used to adapt the permeability field in simplistic two-phase (oil-water) reservoir simulation models. It showed to be as efficient as the adjoint-based method or even more efficient than the adjoint-based method, and of course much faster than the finite difference approach. Using the BPOD-method we achieved very good matches and very good predictions quality.

- The classical adjoint-based approach gives the exact gradients of the objective function with respect to the parameter patterns. The model-reduced history matching procedure uses a simplified reduced-order forward model and a corresponding reduced-order adjoint model, and therefore the calculated gradients are not exact. In the considered cases they proved to be accurate enough to decrease the original objective function in all iteration steps and to bring it to an acceptable value.

- The classical gradient-based history matching procedure is very efficient but it is very difficult to implement for complex models. The POD-based model-reduced approach is computationally more expensive, but it does not require access to the simulation code and is relatively easy to implement. The BPOD-based method requires the adjoint states but does not require the Jacobians with respect to the parameters, and may therefore be used as a fast method to add functionality for assimilating additional parameters to an existing adjoint code.

- In the model-reduced approach, the approximated objective function is quadratic, and therefore the optimization problem solved using reduced-order models is simpler than the optimization problem in terms of the original objective function. The model-reduced gradient can be interpreted as an ensemble-based gradient, which sometimes may help to overcome convergence to local optima in case of a 'rough' objective function.

- The model-reduced approach is very attractive for parallel processing and the implementation of the code for parallel computation of the model-reduced method is as simple as the implementation of the sequential code. The total computational time in case of parallel implementation is comparable to the time of three simulations times the number of outer loops. (Per outer iteration: a parallel run to establish a base-line and to collect snapshots, a parallel run to compute the perturbations, and another single run to verify the optimal result using the full-order model.)

- Using only a few (parameter and state) patterns the model-reduced methods provide approximate solutions for history matching problems and relatively good approximations of the gradients. However, when very accurate gradients are required those methods may require high numbers of patterns. Alternatively the model-reduced methods could be used in combination with conventional techniques. For instance, after one outer loop of the model-reduced method a global optimization method could be considered as a method to improve the matches.

- The approach proposed in this work has a few limitations. In particular, the perturbations needed in the model-reduced method restrict its use to a limited number of parameter patterns or controls. Since the sensitivities of the system with respect to these patterns or controls have to be calculated by finite-differences, a number of controls or parameter patterns of order larger than 100 can be computationally demanding. Additionally, it can result in a very large number of snapshot simulations, and consequently in very large eigenvalue problems and many state patterns.

- Modeling the reservoir fluid flow under induced fracturing conditions is difficult. It requires the coupling of a reservoir flow model with a geomechanical model that describes the changes in the stress field during reservoir operation. The change in the pressure field causes changes in the stress field, while the changes in the stress field cause a fracture propagation or closure, i.e. changes in the fracture sizes. The changes in the fracture sizes influence the changes of the pressure field. Therefore, modeling this situation requires a two-way coupling. It has been shown that a fracture significantly influences the flow in the reservoir and that by changing the injection rates the injection induced fracture can be controlled.

- We attempted to optimize production under induced fracturing conditions using an existing combination of a proprietary reservoir simulator explicitly coupled to a semi-analytical fracture simulator. Because of numerical problems, application of the model-reduced approach turned out to be infeasible. As an alternative, production was therefore optimized using the simultaneous perturbation stochastic approximation algorithm (SPSA) and the steepest ascent method with finite difference gradients. Both methods resulted in optimization strategies that improved the total recovery, but they were computationally relatively expensive. Using a classical gradient-based routine, the production could be efficiently optimized under induced fracturing conditions if an automatic gradient calculation were available for this coupled system. Therefore, there is a need for deriving the missing part of the adjoint model, or for implementing an implicitly coupled fracture growth simulation and applying the model-reduced production optimization method.

8.2 Recommendations

The proposed model-reduced methods need to be further tested.

- In this thesis, we mainly focused on estimation of a permeability field. In reservoir simulation models there are other parameters that are uncertain and for which the adjoint-based gradient is often not available, e.g. the parameters of the relative permeability model or capillary pressure model. The same tests could be performed and assessed for those parameters.

- During the analysis of the BPOD method we used the fact that the adjoint model remains the same for all parameters (or controls) as long as the observed data remain the same. Therefore, once we have an adjoint model available, it can be used in gradient calculation of all remaining parameters (or controls). However, besides the adjoint states, the sensitivities of the system to the parameter patterns (or controls) under consideration are also required. Those sensitivities are typically not available. One could try to derive the sensitivity of the model to the uncertain parameter patterns or controls by using perturbations, and then used them in an iterative optimization routine. Since the calculation of the sensitivities is computationally expensive they should be computed only when the next iteration step does not result in a further reduction of the objective function. This approach might be compared to other adjoint-free methods.

- Many parameters can be tuned in the model-reduced method. The limited number of models used in this thesis does not allow us to general conclusions about the performance of this method. The performance of the method or the quality of an obtained gradient should be studied in detail for a larger number of models, in terms of the number of patterns, the frequency of snapshots, the perturbation size, the number of snapshot simulations and the percentage of the relative importance.

Bibliography

Aanonsen, S. I., G. Nædval, D. S. Oliver, A. C. Reynolds and B. Valles (2009). The Ensemble Kalman Filter in reservoir engineering – a review. *SPE Journal* **14**(3), 393–412.

Agarwal, A. and M. J. Blunt (2003). Streamline-based method with full-physics forward simulation for history-matching performance data of a north see field. *SPE Journal* **8**(2), 171–180.

Algazi, V. R . and D. Sakrison (1969). On the of the karhunen-loéve expansion. *IEEE Trans. biform.* **15**, 319– 321.

Altaf, M.U., A.W. Heemink and Verlaan M. (2009). Inverse shallow-water flow modeling using model reduction. *International Journal for Multiscale Computational Engineering* **7**(6), 577–594.

Anderson, T. L. (1995). *Fracture Mechanics: Fundamentals and Applications*. CRC Press.

Andrews, C. A., J. M. Davies and G. R. Schwartz (1967). Adaptive data compression. *IEEE* **55**, 267–277.

Antoulas, A.C. (2005). *Approximation of Large-Scale Dynamical Systems*. Philadelphia, US: SIAM.

Asheim, H. (1988). Maximization of water sweep efficiency by controlling production and injection rates. In *Paper SPE 18365 presented at the SPE European Petroleum Conference*, London, UK.

Aziz, K. and A. Settari (1979). *Petroleum reservoir simulation*. London, UK: Applied Science Publishers.

Bangerth, B. L., W. H. Klie, M. F. Wheeler, P. L. Stoffa and M. K. Sen (2006). On optimization algorithms for the reservoir oil well placement problem. *Computational Geosciences* **10**, 303–319.

Barenblatt, G.I. (1962). The mathematical theory of equilibrium cracks in brittle fracture. *Advances in Applied Mechanics* **7**, 55–129.

Bennett, A. F. (2002). *Inverse modelling of the ocean and atmosphere*. Cambridge University Press, Cambridge.

Bissell, R. C. (1994). Calculating optimal parameters for history matching. In *4th European Conference on the Mathematics of Oil Recovery*, Roros.

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

Brouwer, D.R., G. Nævdal, J.D. Jansen and C.P.J.W. Van Kruijsdijk (2004). Improved reservoir management through optimal control and continuous model updating. In *Paper SPE 90149 presented at the SPE Annual Technical Conference and Exhibition*, Houston, US.

Burden, R. L. and J. D. Faires (2001). *Numerical analysis*. Brooks/Cole.

Burgers, G., P. J. Van Leeuwen and G. Evensen (1998). Analysis scheme in the ensemble kalman filter. *Mon. Wea. Rev.* **126**, 1719–1724.

Caers, J. (2003). History matching under training-image-based geological model constraints. *SPE Journal* **8**(3), 218–226.

Cardoso, M. A., L. Durlofsky and P. Sarma (2009). Development and application of reduced-order modeling procedures for subsurface flow simulation. *International Journal for Numerical Methods in Engineering* **77**(9), 1322–1350.

Chavent, G., M. Dupuy and P. Lemonnier (1975). History matching by use of optimal theory. *SPE Journal* **15**(1), 74–86.

Chen, W. H., G. R. Gavalas, J. H. Seinfeld and M. L. Wasserman (1974). A new algorithm for automatic history matching. *SPE Journal* **14**(4), 593–608.

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

Cheng, H., X. Wen, W. J. Milliken and A. Datta-Gupta (2004). Field experiences with assisted and automated history matching. In *SPE paper 89857 presented at the SPE Annual Technical Conference and Exhibition*, Houston, USA.

Clifford, P. J., P. J. Berry and H. Gu (1991). Modeling the vertical confinement of injection-well thermal fractures. In *SPEPE*, pp. 377–383.

Courant, R. and D. Hilbert (1953). *Methods of Mathematical Physics*. US: Wiley Interscience.

Courtier, P. (1997). Dual formulation of four-dimensional variational assimilation. *Quarterly Journal of the Royal Meteorological Society* **123**, 2449–2461.

Courtier, P., J-N. Thépaut and A. Hollingsworth (1994). A strategy for operational implementation of 4d−var, using an incremental approach. *Quarterly Journal of the Royal Meteorological Society* **120**, 1367–1387.

Dadashpour, M., D. Echeverria Ciaurri, T. Mukerji, J. Kleppe and M. Landrø(2009). Simple zonation and principle component analysis for speeding up porosity and permeability estimation from 4d seismic and production data. In *Proceeding of 71st EAGE Conference and Exhibition*, Amsterdam, The Netherlands.

de Marsily, G., G. Lavedan, M. Boucher and G. Fasanino (1984). Interpretation of interference tests in a well field using geostatistical techniques to fit the permeability distribution in a reservoir model. *Geostatistics for Natural Resources Characterization*, 831–849.

Dikken, B. J. and H. Niko (1987). Waterflood-induced fractures: a simulation study of their propagation and effects on waterflood sweep efficiency. Aberdeen, United Kingdom.

Doublet, D.C., S.I. Aanonsen and X-C Tai (2009). An efficient method for smart well production optimisation. *Journal of Petroleum Science and Engineering* **69**(1-2), 25–39.

Doucet, A., N. de Freitas and N. Gordon (2001). *Sequential Monte Carlo method in practice*. New York: Statistics for engineering and information scince. Springer.

Ehrendorfer, M. (2007). A review of issues in ensemble-based kalman filtering. *Meteroligische Zeitschrift* **16**(6), 795–818.

Evensen, G. (1994). Sequential data assimilation with nonlinear quasi-geostrophic model using monte carlo methods to forecast error statistics. *Journal of Geophysical Research* **99**(C6), 10143–10162.

Evensen, G. (2007). *Data assimilation. The Ensemble Kalman Filter*. Springer.

Ewing, R. E. (1983). *The mathematics of reservoir simulation*. Frontier in Applied Mathematics.

Fletcher, R. (1991). *Practical methods of optimization*. Padstow: Wiley-Interscience.

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

Gao, G. and A. C. Reynolds (2006). An improved implementation of the LBFGS algorithm for automatic history matching. *SPE Journal (SPE 90058-PA)* **11**(1), 5–17.

Gavalas, G. R., P. C. Shah and J. H. Seinfeld (1976). Reservoir history matching by bayesian estimation. *SPE Journal* **16**(6), 337–350.

Glover, K. (1984). All optimal hankel-norm approximations of linear multivariable systems and their $\mathcal{L}^{\infty}-$ error bounds. *International Journal of Control* **39**(6), 1115–1193.

Grimstad, A. A. and T. Mannseth (2000). Nonlinearity, scale and sensitivity for parameter estimation problems. *SIAM Journal for Scientific Computing* **21**(6), 2096–2113.

Gu, H. and D.S. Oliver (2004). History matching of the PUNQ-3 reservoir model using the ensemble kalman filter. In *SPE Annual Technical Conference and Exhibition*, Houston, Texas. SPE. History Matching.

Hagoort, J. (1981). *Waterflood-induced hydraulic fracturing*. Delft, The Netherlands: Ph.D. thesis, Delft University of Technology.

Heijn, T., R. Markovinović and J.D. Jansen (2004). Generation of low-order reservoir models using system-theoretical concepts. *SPE Journal* **9**(2), 202–218.

Hermann, R. and A. J. Krener (1977). Nonlinear controllability and observability. *IEEE Trans. Automat. Control* **22**(5), 728.

Houtekamer, P.L. and H.L. Mitchell (1998). Data assimilation using an ensemble kalman filter technique. *Mon. Weather Rev.* **126**, 796–811.

Hu, L. Y., G. Blanc and B. Noetinger (2001). Gradual deformation and iterative multiwell pressure data and prior information. *Mathematical Geology* **33**, 475–489.

Hu, L. Y. and S. Jenni (2005). History matching of object-based stochastic reservoir models. *SPE Journal* **10**, 312–323.

Hustedt, B., D. Zwarts, H. P. Bjoerndal, R. Masfry and P. J. Van den Hoek (2008). Induced fracturing in reservoir simulations: application of a new coupled simulator to waterflooding field examples. **11**(3), 569–576.

IEO (2009). International energy outlook 2009, technical report doe/eia-0484(2009). Technical report, Energy Information Administration, U.S. Department of Energy.

Jacquard, P. and C. Jain (1965). Permeability distribution from field pressure data. *SPE Journal* **5**(4), 281–294.

Jafarpour, B. and D.B. McLaughlin (2009). Reservoir characterization with the discrete cosine parameterization. part i, parameterization. *SPE Journal* **14**(1), 182–188.

Jansen, J. D. (2011a). Adjoint-based optimization of multiphase flow through porous media – a review. *Computers and Fluids*, Published online. DOI: 10.1016/j.compfluid.2010.09.039.

Jansen, J. D. (2011b). Systems theory for reservoir management. lecture notes, tu delft.

Jansen, J. D., O. H. Bosgra and P. M. J. Van den Hof (2008). Model-based control of multiple flow in subsurface oil reservoirs. *Journal of Process Control* **18**, 846–855.

Jansen, J. D., S. D. Douma, D. R. Brouwer, P. M. J. V. den Hof, O. H. Bosgra and A. W. Heemink (2009). Closed-loop reservoir management. In *SPE Reservoir Simulation Symposium*, The Woodlands, Texas, US.

Jazwinski, A. H. (1970). *Stochastic processes and filtering theory*. Academic Press, INC.

Kaleta, M. P., R. G. Hanea, A. W. Heemink and J. D. Jansen (2010). Model-reduced gradient-based history matching. In *Proc. 5th European Conference on Computational Fluid Dynamics ECCOMAS CFD*, Lisbon, Portugal.

Kaleta, M. P., R. G. Hanea, A. W. Heemink and J. D. Jansen (2011). Model-reduced gradient-based history matching. *Computational Geosciences* **15**(1), 135–153.

Kaleta, M. P., A. W. Heemink, J. D. Jansen and R. G. Hanea (2008). Model-reduced variational data assimilation for reservoir model updating. In *9th European Conference on Mathematics in Oil Recovery (ECMOR XI)*, Bergen, Norway.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 35–45.

Karhunen, K. (1946). Zur spektral theorie stochasticher prozesse. *Annales Academiae Scientiarum Fennicae*.

Koning, E. J. L. (1988). *Waterflooding under fracturing conditions*. Delft, The Netherlands: Ph.D. thesis, Delft University of Technology.

Kosambi, D. D. (1943). Statistics in function space. *J. Indian Math. Soc.* **7**, 76–88.

Lall, S., J.E. Marsden and S. Glavaski (1999). Empirical model reduction of controlled nonlinear systems.

Lall, S., J. E. Marsden and S. A. Glavaski (2002). A subspace approach to balanced truncation for model reduction of nonlinear control systems. *International Journal of Robust Nonlinear Control* **12**, 519.

Lawless, A. S., N. K. Nichols, C. Boess and A. Bunse-Gerstner (2008). Using model reduction methods within incremental four-dimensional variational data assimilation. *Monthly Weather Review* **136**(4), 1511–1522.

Lee, T. Y. and J. H. Seinfeld (1987). Estimation of two-phase petroleum reservoir properties by regularization. *Journal of Computational Physics* **69**, 397–419.

Loéve, M. (1946). Functions aleatoire de second ordre. *Revue Science* **84**, 195–206.

Lorentzen, R.J., A.M. Berg, G. Nævdal and E.H. Vefring (2006). A new approach for dynamic optimization of waterflooding problems. In *Paper SPE 99690 presented at the SPE Intelligent Energy Conference and Exhibition*, Amsterdam, The Netherlands.

Lorentzen, R. J., J. Frøyen, A. C. V. M. Lage, G. Nævdal and E. H. Vefring (2001). Underbalanced drilling: real time data interpretation and decision support.

Lorentzen, R. J., G. Nævdal and A. C. V. M. Lage (2003). Tuning of parameters in a two-phase flow model using an ensemble kalman filter. *International Journal of Multiphase Flow* **29**(8), 1283–1309.

Lumley, J. L. (1967). The structure of inhomogeneous turbulence. *Atmospheric Turbulence and Radio Wave Propagation, A. M. Yaglom and V. I. Tatarsky (editors) (Nauka, Moscow)*, 166–178.

Markovinović, R. (2009). *System-Theoretical Model Reduction for Reservoir Simulation and Optimization.* Ph.D. Thesis, Delft University of Technology.

Markovinović, R., E. L. Geurtsen, T. Heijn and J. D. Jansen (2002). Generation of low-order reservoir models using POD, empirical gramians and subspace identification. In *8th European Conference on the Mathematics of Oil Recovery*, Freiberg.

Markovinović, R. and J.D. Jansen (2006). Accelerating iterative solution methods using reduced-order models as solution predictors. *International Journal for Numerical Methods in Engineering* **68**(5), 525–541.

Moore, B. C. (1981). Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE Trans. Automat. Control* **26**, 17–32.

Nædval, G., D. R. Brouwer and J. D. Jansen (2006). Waterflooding using closed-loop control. *Computational Geosciences* **10**(1), 37–60.

Nævdal, G., L. M. Johnsen, S. I. Aanonsen and E. H. Vefring (2003). Reservoir monitoring and continuous model updating using ensemble kalman filter.

Nævdal, G., L. M. Johnsen, S. I. Aanonsen and E. H. Vefring (2005). Reservoir monitoring and continuous model updating using ensemble kalman filter. *SPE Journal (SPE 84372)* **10**(1), 66–74.

Nævdal, G., T. Mannseth and E.H. Vefring (2002). Near-well reservoir monitoring through ensemble kalman filter. In *Paper SPE 75235 presented at the SPE/DOE Improved Oil Recovery Symposium.*

Obukbov, A. M. (1954). Statistical description of continuous fields. *Tr. Geophys. Int. Akad. Nauk. SSSR*, 3–42.

Oliver, D. S. and Y. Chen (2011). Recent progress on reservoir history matching: a review. *Computational Geosciences* **15**(1), 185–221.

Oliver, D. S., A. C. Reynolds and N. Liu (2008). *Inverse theory for petroleum reservoir characterization and history matching*. Cambridge: Cambridge University Press.

Papoulis, A. (1965). *Probability, Random Variables, and Stochastic Processes*. New York, US.

Peaceman, D. W. (1977). *Fundamentals of numerical reservoir simulation*. Elsevier.

Perkins, T. K. and J. A. Gonzalez (1985). The effect of thermo-elastic stresses on injection well fracturing. *SPE Journal* **25**(1), 78–88.

Pougachev, V. S. (1953). General theory of random functions. *Izv. Akud. Nauk. Set. Mat.* **17**, 1401–2.

RamaRao, B. S., A. M. La Venue, G. de Marsily and M. G. Marietta (1995). Pilot point metho- dology for automated calibration of an ensemble of conditionally simulated transmissivity fields. 1. theory and computational experiments. *Water Resources Research* **31**(3), 475–493.

Ramirez, W. F. (1987). *Applications of Optimal Control Theory to Enhanced Oil Recovery*. Elsevier Science Publishers.

Reynolds, A. C., Nanqun He, Lifu Chu and D. S. Oliver (1996). Reparameterization techniques for generating reservoir descriptions conditioned to variograms and well-test pressure data. *SPE Journal* **1**(4), 413–426.

Roggero, F. and L. Y. Hu (1998). Gradual deformation of continuous geostatistical models for history matching.

Romero, L.A., D.C. Ghiglia, C.C. Ober and S.A. Morton (2000). Phase encoding of shot records in prestack migration. *Geophysics* **65**, 426–436.

Rosenfeld, A. and A. C. Kak (1982). Digital picture. New York, US.

Rowley, C. W. (2005). Model reduction for fluids, using balanced proper orthogonal decomposition. *Int. J. on Bifurcation and Chaos* **15**(3), 997–1013.

Russell, T. F. and M. F. Wheeler (1983). *Finite element and finite difference methods for continuous flows in porous media*. Philadelphia, USA: Richard E. Ewing, Frontier in applied mathematics, SIAM.

Sahni, I. and R.N. Horne (2005). Multiresolution wavelet analysis for improved reservoir description. *SPE Reservoir Evaluation and Engineering* **8**(1), 53–69.

Sarma, P., W.H. Chen, L.J. Durlofsky and K. Aziz (2008). Production optimization with adjoint models under nonlinear control-state path inequality constraints. *SPE Reservoir Evaluation and Engineering* **2**, 326–339.

Sarma, P., L. J. Durlofsky, K. Aziz and W. H. Chen (2007). A new approach to automatic history matching using Kernel PCA. In *SPE Reservoir Simulation Symposium (SPE 106176)*, Houston.

Scherpen, J. M. A. (1993). Balancing of nonlinear systems. *Syst. Control Lett.* **21**, 143–153.

Schulze-Riegert, R. W., J. K. Axmann and O. Haase (2002). Evolutionary algorithms applied to history matching of complex reservoirs. *SPE Reservoir Evaluation & Engineering* **5**(2), 163–173.

Shah, P. C., G. R. Gavalas and J. H. Seinfeld (1978). Error analysis in history matching: the optimum level of parameterization. *SPE Journal (SPE 6508-PA)* **18**(3), 219–228.

Sirovich, L. (1987). Turbulence and the dynamics of coherent structures, parts i-iii. *Quarterly applied mathematics* **45**, 561–590.

Skjervheim, J.-A., G. Evensen, Aanonsen S. I. and T. A. Johansen (2007). Incorporating 4d seismic data in reservoir simulation model using ensemble kalman filter. *SPE Journal* **12**(3), 282–292.

Spall, J. C. (1992). Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transactions on automatic control* **37**, 332–341.

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

Spall, J. C. (2000). Adaptive stochastic approximation by the simultaneous perturbation method. *IEEE Transactions on automatic control*, 1839–1853.

Sudaryanto, B. and Y.C. Yortsos (2000). Optimization of fluid front dynamics in porous media using rate control. *Physics of Fluids* **12**, 1656–1670.

Talagrand, O. and P. Courtier (1987). Variational assimilation of meteorological observations with the adjoint vorticity equation. I:Theory. *Quarterly Journal of the Royal Meteorological Society* **113**, 1311–1328.

Tarantola, A. (1987). *Inverse problem theory. Methods for data fitting and model parameter estimation*. Elsevier.

Terzaghi, K. (1943). *Theoretical Soil Mechanics*. New York, USA: John Wiley and Sons.

Tran, D., A. Settari and L. Nghiem (2004). New iterative coupling between a reservoir simulator and a geomechanics module. *SPE Journal* **9**(3), 362–369.

Vakili-Ghahani, S. A (2010). *Control relevant upscaling*. Ph.D. thesis, Delft University of Technology.

Valstar, J. R., D. McLaughlin, C. B. M. te Stroet and F. C. Van Geer (2004). A representer based inverse method for groundwater flow and transport applications. *Water Resources Research* **50**, W05116.

Van den Hoek, P. J., R. A. Masfry, D. Zwarts, J. D. Jansen, B. Hustedt and L. Van Schijndel (2008). Optimizing recovery for waterflooding under dynamic induced fracturing conditions. Tulsa, Oklahoma, US.

Van Doren, J. F. M. (2010). *Model structure analysis for model-based operation of petroleum reservoirs*. Ph.D. thesis, Delft University of Technology.

Van Doren, J. F. M., R. Markovinović and J.D. Jansen (2004). Reduced-order optimal control of waterflooding using pod. In *Proc. 9th European Conference on Mathematics in Oil Recovery (ECMOR IX)*, Cannes, France.

Van Doren, J. F. M., P. M. J. Van den Hof, J. D. Jansen and O.H. Bosgra (2008). Determining identifiable parameterizations for large-scale physical models in reservoir engineering. In P. Misra M.J. Chung and H. Shim (Eds.), *7th IFAC World Congress*, Seoul, Korea, pp. 11421–11426.

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

Vasco, D. W. and A. Datta-Gupta (1997). Integrating field production history in stochastic reservoir characterization. *SPE Formation Evaluation* **12**(3), 149–156 (36567–PA).

Vasco, D. W., S. Soon and A. Datta-Gupta (1999). Integrating dynamic data into high-resolution reservoir models using streamline-based analytic sensitivity coefficients. *SPE Journal* **4**(4), 389–399.

Vermeulen, P. T. M. and A. W. Heemink (2006). Model-reduced variational data assimilation. *Monthly Weather Review* **134**(10), 2888–2899.

Volkwein, S. (2008). Proper orthogonal decomposition: Application in optimization and control. lecture notes, cea-edf-inria summer school numerical analysis summer school model reduction and reduced basis methods: Application in optimization. saint-lambert-des-bois, frankreich, juni 23 to juli 4, 2008, 74 pages.

Wang, C., G. Li and A.C. Reynolds (2009a). Production optimization in closed-loop reservoir management. *SPE Journal* **14**, 506–523.

Wang, C., G. Li and A. C. Reynolds (2009b). Production optimization in closed-loop reservoir management. *SPE Journal* **14**(3), 506–523.

Wang, Y. and A. R. Kovscek (2001). Streamline approach for history matching production data. *SPE Journal* **5**, 353–363.

Wasserman, M. L., A. S. Emanuel and J. H. Seinfeld (1975). Practical applications of optimal-control theory to history-matching multiphase simulator models. *SPE Journal* **15**(4), 347–355.

Watson, A. T., J. H. Seinfeld, G. R. Gavalas and P. T. Woo (1980). History matching in two-phase petroleum reservoirs. *SPE Journal* **20**(6), 521–532.

Wen, X.-H., C. V. Deutsch and A. S. Cullick (1998). Integrating pressure and fractional flow data in reservoir modeling with fast streamline-based inverse method. In *SPE paper 48971 presented at the SPE Annual Technical Conference and Exhibition*, New Orleans, USA.

Willcox, K. and J. Peraire (2002). Balanced model reduction via the proper Orthogonal Decomposition. *Journal of American Institute of Aeronautics and Astronautics* **40**(11), 2323–2330.

Wu, Z. (2001). A newton-raphson iterative scheme for integrating multiphase production data into reservoir models. *SPE Journal* **6**, 343–351.

Wu, Z. and A. Datta-Gupta (2002). Rapid history matching using a generalized travel time inversion method. *SPE Journal* **7**, 113–122.

Yang, P., A. T. Watson and R. V. Armasu (1988). Automatic history matching with variable-metric methods. *SPE Reservoir Evaluation and Engineering* **3**(3), 995–1001.

Zandvliet, M.J., O.H. Bosgra, J.D. Jansen, P.M.J. Van den Hof and J.F.B.M. Kraaijevanger (2007). Bang-bang control and singular arcs in reservoir flooding. *Journal of Petroleum Science and Engineering* **58**, 186–200.

Zandvliet, M.J., J.F.M. Van Doren, O.H. Bosgra, J.D. Jansen and P.M.J. Van den Hof (2008). Controllability, observability and identifiability in single-phase porous media flow. *Computational Geosciences* **12**(4), 605–622.

Zhang, F. and A. C. Reynolds (2002). Optimization algorithms for automatic history matching of production data. In *Proceeding of the 8th European Conference on the Mathematics of Oil Recovery*, Freiberg, Germany.

Zoback, M. (2007). *Reservoir Geomechanics*. Cambridge University Press, UK.

Zwarts, D., B. Hustedt and P.J. Van den Hoek (2006). Modelling induced fracture mechanics in reservoir simulations- a new coupled simulator. In *Paper presented at the ECMOR X Technical Conference*, Ansterdam, The Netherlands.

Summary

Since the world's energy demand increases every year, the oil & gas industry makes a continuous effort to improve fossil fuel recovery. Physics-based petroleum reservoir modeling and closed-loop model-based reservoir management concept can play an important role here. In this concept measured data are used to improve the geological model, while the improved model is used to increase the recovery from a field. Both problems can be formulated as optimization problem, i.e. history matching identifies the parameter values that minimize an objective function that represents the mismatch between modeled and observed data while production optimization identifies wells controls that maximize the total oil recovery or monetary profit. One of the most efficient class of methods to solve history matching and production optimization problems are gradient-based methods where the gradients are calculated with the use of an adjoint method. The implementation of the adjoint method for parameter estimation and control optimization is, however, very difficult if no Jacobians of the model are available. This implies that there is a need for gradient-based, but adjoint-free optimization methods. A requirement becomes even more pressing if reservoir simulation is combined with another simulation, e.g. simulation of geomechanics or rock physics, with a code for which no Jacobians are available. The research objective of this thesis was to evaluate the performance of a model-reduced gradient-based history matching routine that does not require a difficult implementation and involves the reduction of the reservoir system. Additionally, the use of model-reduced method for production optimization of a reservoir operating under induced fracturing conditions was considered.

In history matching problems one deals with a large number of uncertain parameters (usually of order of 10^4-10^6) and very sparse observations, while in the production optimization one controls a large dimensional system by adjusting a limited number of controls. Consequently, the values of many model parameters cannot be verified with measurements due to a relatively few information content present in them, while in the production optimization only a limited part of the system can be indeed controlled. In this thesis we proposed a new method inspired by the results in reduced order modeling (ROM) and system-theoretical concepts of controllability and observability of the reservoir system. The new approach assumes that the reservoir dynamics relevant for history matching or production optimization can be represented accurately by a much smaller number of variables than the number of grid cells used in the simulation model. Consequently, the original (nonlinear and high-order) forward model is replaced by

a linear reduced-order forward model and the adjoint of the tangent linear approximation of the original forward model is replaced by the adjoint of a linear reduced-order forward model. The reduced-order model is constructed by means of the Proper Orthogonal Decomposition (POD) method or Balanced Proper Orthogonal Decomposition (BPOD) method. The reduced-order model is not, however, obtained by the projection of the nonlinear system of equations as in the conventional projection-based ROM techniques, but instead it is approximated in the reduced subspace. The conventional POD method requires the availability of the high-order tangent model, i.e. of the Jacobians with respect to the states which are not available. The model-reduced method obtains a reduced-order approximation of the tangent linear model directly by computing approximate derivatives of the reduced-order model. Then due to the linear character of the reduced model, the corresponding adjoint model is easily obtained. The gradient of the objective function is approximated and the minimization problem is solved in the reduced space; the procedure is iterated with the updated estimate of the parameters if necessary. The POD-based approach is adjoint-free and can be used with any reservoir simulator, while the BPOD-based approach requires an adjoint model but does not require the Jacobians of the model with respect to uncertain parameters or controls.

At first the model-reduced method was applied to history matching problems and was evaluated based on its computational efficiency and robustness. In order to make a valuable judgment this approach was compared to the classical adjoint-based method, which was available for the estimation of the permeability field. Permeabilities are described at each cell of the model, and therefore they need to be re-parameterized. The KL-expansion was used to reduce the parameters space. The significant reduction of the dimension of the dynamic reservoir model and parameter space made the approximation of the reduced-order system feasible in acceptable computation time. In the first studies simplistic reservoir models were used, for which the model-reduced approach showed to perform very well. The obtained estimates of the permeability field significantly improved compared to the prior fields and gave the acceptable history-matches; the quality of the prediction capabilities of the estimated models were very high and comparable to those obtained by the classical adjoint-based approach. The POD-based method was approximately twice as expensive as the classical approach, but the BPOD-based method was comparable to the adjoint-based method. Moreover, both methods were considerably cheaper than the finite difference approach. These preliminary results were the first applications of the model-order reduction to history matching problems. After this proof of concept, further studies were carried on more complex and larger models. The proposed method was capable to obtain satisfactory match with a computational efficiency about five times lower than the adjoint-based method. Similarly, an improvement in the prediction was obtained. The second problem considered in this research was to apply the adjoint-free methods to production optimization of the reservoir operating under special conditions that required coupling of two simulators and for which the adjoint code is not available. The model-reduced method could not be applied because of a low accuracy of the simulation solution which in case of long time simulations resulted in large approximation errors. Therefore, simultaneous perturbation stochastic algorithm (SPSA) was applied together with the finite difference gradient-based method to solve the production optimization problem. SPSA is a gradient-based method where the gradients are approximated by random perturbations of all controls in once, while the finite difference method approximates the gradients by perturbation of each control separately. Both approaches were very simple to implement, they resulted in the improvement of the production, but they were computationally relatively expensive.

Samenvatting

Door de elk jaar toenemende vraag naar energie in de wereld, is de olie- en gasindustrie voortdurend bezig met het verbeteren van de winning van fossiele brandstoffen. Hierbij kunnen op natuurkundige wetten gebaseerde reservoirmodellen en het closed-loop model gebaseerde reservoir beheer concept een belangrijke rol spelen. In dit concept worden de gemeten data gebruikt om het geologisch model te verbeteren, zodat vervolgens dit model gebruikt kan worden voor het bepalen van een betere strategie bij het winnen van de olie uit een veld. Beide problemen kunnen beschreven worden als een optimalisatieprobleem. Bij history matching worden de parameterwaardes bepaald door het minimaliseren van een doelfunctie als maat voor het verschil tussen de gemodelleerde data en de waarnemingen, terwijl bij productie-optimalisatie de stuurparameters van de putten die de totale opbrengst van het veld maximaliseren, worden geidentificeerd. Eén van de meest efficiënte methodes om history matching en productie-optimalisatieproblemen op te lossen zijn gradiënt gebaseerde methoden, waarbij de gradiënten worden berekend met behulp van het geadjugeerde model. De implementatie van deze methode voor parameterschatting en optimalisatieproblemen is echter heel moeilijk als er geen Jacobiaan van het systeem beschikbaar is. Dit impliceert dat er een gradiënt gebaseerde, maar "adjoint-free" optimalisatiemethode nodig is. Een dergelijke aanpak wordt nog belangrijker als de reservoirsimulatie gecombineerd wordt met andere simulaties, zoals van geo-mechanica of rots-fysica, met een code waarvoor geen Jacobiaan beschikbaar is. Het doel van dit proefschrift was het ontwikkelen van een nieuwe "model-reduced gradient-based history matching" routine die relatief eenvoudig te implementeren is. Daarnaast is de methode tevens toegepast ten behoeve van de optimalisatie van de productie van een reservoir.
In history matching problemen werkt men met een groot aantal onzekere parameters (normaliter aantallen van $10^4 - 10^6$) en relatief weinig meetgegevens, terwijl men in de productie-optimalisatie hoog dimensionale systemen bestuurt door het aanpassen van een beperkt aantal stuurparameters. Door de beperkte hoeveelheid informatie kunnen de waarden van de modelparameters niet geverifieerd worden met behulp van metingen, terwijl in de productie-optimalisatie alleen een beperkt gedeelte van het systeem daadwerkelijk bestuurd kan worden. In dit proefschrift stellen wij een nieuwe methode voor die geïnspireerd is door de reduced order modeling (ROM) aanpak en de systeemtheoretische concepten bestuurbaarheid en waarneembaarheid van een reservoirsysteem. De nieuwe benadering veronderstelt dat het dynamisch gedrag van het reservoir nauwkeurig

gerepresenteerd kan worden door een veel kleiner aantal variabelen dan het aantal roosterpunten gebruikt in het simulatiemodel. Derhalve wordt de lineaire benadering van het originele (niet-lineaire en hoog dimensionale) voorwaartse model in de klassieke aanpak vervangen door een lineair gereduceerd voorwaarts model. Het gereduceerde model wordt samengesteld met behulp van de Proper Orthogonal Decomposition (POD) methode of de Balanced Proper Orthogonal Decomposition (BPOD) methode. Het gereduceerde model is hierbij niet verkregen door de projectie van de niet-lineaire modelvergelijkingen zoals in de conventionele ROM-technieken, maar in plaats daarvan wordt het benaderd in de gereduceerde deelruimte. De conventionele POD-methode vereist de beschikbaarheid van het gelineariseerde model, bijvoorbeeld van de Jacobiaan met betrekking tot de toestanden. In de praktijk is deze informatie echter meestal niet beschikbaar. Bij de in dit proefschrift voorgestelde aanpak wordt een benadering van het gelineariseerde model rechtstreeks verkregen door het berekenen van numerieke benaderingen van de afgeleiden van het gereduceerde model. Vervolgens is door het lineair karakter van het gereduceerde model ook het bijbehorende adjugeerde model makkelijk te bepalen. De gradiënt van de doelfunctie kan vervolgens worden benaderd en het minimalisatieprobleem kan dan worden opgelost in de gereduceerde ruimte. De procedure kan daarna worden herhaald met de aangepaste parameters indien nodig. De POD-gebaseerde benadering is "adjoint-free" en kan gebruikt worden in combinatie met iedere reservoirsimulator, terwijl de BPOD-gebaseerde benadering wel een geadjugeerd model vereist, maar geen Jacobiaan van het model met betrekking tot de onzekere modelparameters of stuurparameters. De ROM-methode is eerst toegepast op history matching problemen en is daarbij uitvoerig geëvalueerd op efficiëntie en robuustheid. De ROM-aanpak is hierbij tevens vergeleken met de klassieke geadjugeerde methode, die beschikbaar was voor de cshatting van het permeabiliteitsveld. De permeabiliteit is gedefinieerd in iedere cel van het model en moet daarom eerst geparameteriseerd worden. De KL-expansie werd hierbij gebruikt om de parametersruimte te reduceren. De significante reductie van de dimensie van het dynamische reservoirmodel en de parametersruimte maakte de benadering van het gereduceerde systeem in een acceptabele rekentijd mogelijk.

In de eerste studies werden relatief eenvoudige reservoirmodellen gebruikt, waarbij de POD-benadering goede resultaten liet zien. De verkregen schattingen van het permeabiliteitsveld waren significant beter en de kwaliteit van de voorspellingen van het gereduceerde model was erg hoog en vergelijkbaar met die verkregen met de klassieke geadjugeerde benadering. De BPOD-gebaseerde methode was ook vergelijkbaar met de klassieke geadjugeerde methode. Beide methoden waren aanzienlijk goedkoper dan de bekende eindige differentiebenadering voor het bepalen van de gradiënt. Na dit "proof of concept" werden andere studies gedaan met complexere en grotere modellen. De voorgestelde methode bleek in staat om bevredigende resultaten te krijgen met een efficiëntie van ongeveer vijfmaal minder dan de klassieke geadjugeerde methode.

Het tweede probleem bestudeerd in dit onderzoek was het toepassen van de nieuwe "adjoint-free" methode op de optimalisatie van de productie van een reservoir onder speciale condities die een koppeling vereisten van twee simulatoren en waarbij de geadjugeerde code niet beschikbaar was. De ROM-methode kon helaas niet worden toegepast door een lage nauwkeurigheid van de simulatie van het originele model, die vooral in geval van lange simulatieperioden resulteerde in significante numerieke fouten. Derhalve werden de "simultaneous perturbation stochastic algorithm" (SPSA) en de eindige differentie gradiënt-gebaseerde methode beide toegepast om het optimalisatieprobleem op te lossen. SPSA is een gradiënt-gebaseerde methode waarbij de gradiënten worden benaderd door random perturbaties van alle stuurparameters tegelijkertijd, terwijl de eindige differentie methode de gradiënten benadert door perturbaties van ieder stuurparameter afzonderlijk. Beide benaderingen waren erg gemakkelijk om te implementeren en resulteerden in een duidelijke verbetering van de productie, maar ze waren relatief duur om te berekenen.

About The Author

Małgorzata Paulina Kaleta was born on January 6, 1980 in Kalisz, Poland. She received her secondary education between 1995 and 1999 at Liceum Ogólnokształcące im. Władysława Broniewskiego in Krosno Odrzańskie, Poland. From 1999 to 2004 she studied Mathematics, Informatics and Econometrics at University of Zielona Góra in Poland, from which she received her master degree in Mathematics in 2004. In the same year she started a Master of Science Programme in Risk and Environmental Modelling at Delft University of Technology and obtained her M.Sc. degree in Applied Mathematics in 2006. From November 2006 until October 2010 she continued her academic education in the group of Mathematical Physics, Faculty of Electrical Engineering, Mathematics and Computer Science, at Delft University of Technology, as a Ph.D. student sponsored by the ISAPP Consortium. During the Ph.D. project she took graduate courses at the Centre for Technical Geoscience (CTG) and she received the CTG Research School certificate. In December 2010 she joint Shell Global Solutions International as a reservoir engineer in quantitative reservoir management team.