Streamline based History Matching

Master thesis in Applied and Computational Mathematics

Tor Harald Sandve

Department of Mathematics
University of Bergen

May 8, 2009

Abstract

Streamline based reservoir simulation has been used with great success in the petroleum industry for decades. Fast computational speed together with the fact that sensitivities can be computed analytically along streamlines makes streamline based methods efficient for history matching problems. The key idea for streamline based methods is to approximate 3D fluid flow by a family of 1D transport equations along streamlines. This is done by introducing a time-of-flight coordinate variable. In this thesis a method called generalized travel-time inversion (GTTI) will be discussed for history matching. The main idea behind GTTI is to combine amplitude matching with travel-time matching. The inverse problem associated with the amplitude matching is fully nonlinear and can therefore give unstable and non-unique solutions, while the travel-time inversion has quasi-linear properties. GTTI is able to use all the data available and still preserve the quasi-linear properties of the travel-time inversion. Several authors has studied GTTI previously. In all these articles only the first order sensitivities are used in the history matching. Since the second order sensitivities can be computed along the streamlines at practically no cost, a method that includes information from the second order sensitivities will in theory show better convergence properties. Some synthetic examples shows that this is true for cases where almost no regularization is needed but not for more realistic and illposed problems. To further improve the convergence properties of the minimization a line search and a trust-region strategy is suggested. Both the line search strategy and the trust region method of Levenberg-Marquardt shows good performance. Both in the ability to reduce the objective function and in characterizing the permeability field. A method where the regularization parameters are chosen by the L-curve criteria is also considered and compared to the other methods. An advantage of this method is that the regularization term is reduced as the iterate enters into the neighborhood of the solution and less regularization is needed.

Preface

The work of this thesis is done as part of a master program in applied and computational mathematics, at Department of Mathematics, University of Bergen, Norway.

Contents

This master thesis consists of 7 chapters. In the first chapter the reader is introduced to the motivations and ideas behind history matching. The governing equations for fluid flow in porous medium are introduced in Chapter 2. In Chapter 3, the concept of streamlines and time-of-flight is introduced to formulate a simplified model for fluid flow. History matching is an inverse problem. Theory on inverse problem and regularization is found in Chapter 4. To solve non-linear inverse problems, the Jacobian and the Hessian is needed depending on the method which is used. A unique advantage of the streamline based method is that semi-analytic expressions for these sensitivities can be found. These semi-analytic expressions is motivated and analyzed in Chapter 5. In Chapter 6, a generalized travel-time method to solve the history matching problems is introduced. To test different methods for solving GTTI, some test cases are considered. Finally, Chapter 7 summarizes the results and suggests directions for future work.

Acknowledgments

First I want to thank my supervisor Inga Berre. Her enthusiasm for the subject has been an important encouragement in my work. She has also given me indispensable guidance and support during my master. I also want to express my gratitude towards Knut-Andreas Lie and Atgeirr Rasmussen at Sintef in Oslo for lending me the SAM Research Simulator for streamline simulations. I am also grateful to my fellow students at the departments of Mathematics. Especially Ingvild, Anders and Hanne for helpful comments and questions to my work and Christoffer for helping me with all Linux related problems. Finally, I want to thank my fiancee, Ingelin, for her support and care during my work.

Tor Harald Sandve
Bergen, May 2009

Contents

Chapter 1

Introduction

In mathematics you don't understand things.
You just get used to them.
Johann von Neumann

It is important for the petroleum industry to have an accurate description of the reservoir. A reservoir is described by its rock and fluid properties. With an accurate description of the reservoir, future oil and gas production can be predicted with the use of a reservoir simulator. The reservoir simulator solves the model equations governing the fluid flow in the reservoir, given a set of rock and fluid parameters and feasible boundary conditions. To obtain a good reservoir description, as much as possible of the available information needs to be incorporated into the geological model. The available information can be categorized into two categories: static and dynamic. Static data are time invariant data, such as well logs, core measurements and data from seismic. Time dependent data, such as pressure, saturation, flow rates, data from 4D seismic, and tracer responses are examples of dynamic data. The process of including dynamic data into the geological model based on static data is referred to as history matching. While static data can be incorporated relatively easy, with the help of geo-statistical model, the process of integrating dynamic data is much more complicated [2]. Traditionally the history matching is done by adjusting the reservoir parameters manually by a trial and error procedure. Manual history matching is time consuming and depends highly on the experience of the reservoirs engineers. In the past years significant development has been done in the field of automatic history matching. The most common approach for automatic history matching is based on minimizing an objective function. Typically the objective function is given as the sum of the misfit between the observed data d_j^{obs} and the calculated data d_j^{cal} over all data points $j = 1, ..., N_d$:

$$f = \sum_{j=1}^{N_d} w_j (d_j^{obs} - d_j^{cal})^2, \tag{1.1}$$

where w_j is a scalar that weighs the influence of each observation.

There are different techniques to minimize the objective function in (1.1). Commonly used methods as steepest descent, Gauss-Newton and Levenberg-Marquardt, see [13], uses the gradient of the objective function to find its minimum. The calculation of the gradients is usually the most computationally expensive part of the history matching algorithms. Effective algorithms that compute the gradients are therefore a crucial part of the minimization methods. There are minimization methods that do not need any calculation of the gradients, but they will not be considered here [11].

History matching is an example of an inverse problem. In a forward problem the goal is to compute a set of data given a set of parameters, which is in contrast to an inverse problem where the goal is to find parameters that best match the data. Typically the number of parameters to match in a history matching problem is higher than the number of data points available. This lead to an under determined problem, which do not have a unique solution. In addition to lack of

data, there will be errors in the measurements and numerical errors that make the problem more difficult to solve. To find the best possible solution, additional constraint must therefore be added to the objective function to regularize the problem.

Finite volume, finite element and streamline based fluid flow models can for example be used in an automatic history matching. One of the advantages of the streamline based method is that it is faster than conventional methods based on finite volume and finite elements. Especially for high resolution models, where the number of reservoir parameters can be several million, the streamline formulation gives benefits compared to the more traditional methods. Another huge advantage of the streamline formulation is that sensitivities of streamline related production responses can be computed semi-analytically along the streamlines. The sensitivities needed in the history matching problem can therefore be computed using a single forward run of the simulator.

Chapter 2

A reservoir model of fluid flow in porous medium

all models are wrong,
but some are useful
George E. P Box

A reservoir model consists of a set of differential equations that reflects the physical laws, and a set of fluid and rock parameters. In this chapter basic fluid and rock properties is defined. A general reservoir model based on the physical laws that governs fluid flow in a porous medium is also introduced. These models are in general difficult to solve, therefore simplified models for single-phase and two-phase flow will be presented. The presentation is mainly based on the book written by Bear [1].

2.1 Basic definitions

An oil or gas reservoir is a porous geologic formation that contains in its pore space, in addition to water, at least one hydrocarbon (oil or gas) in liquid or gaseous phase. Basic concepts and equations will also be valid for other types of reservoir with other types of fluids, still oil and gas reservoirs will be used as examples throughout this thesis.

A porous medium consists of two parts, the solid part and the empty part. The solid part is called the matrix and the empty part pore. It is a difficult task to exactly describe the complex structure of a porous medium on a microscopic level. To calculate the fluid flow within these complex structures for a whole reservoir is even more difficult, still microscopic models on either simplified structures or on small scale problems can be used to model fluid flow. Alternatively the processes may be described on a macroscopic level. To be able to describe a porous medium on a macroscopic level, different parameters that describes fluid and matrix properties in a porous medium is defined.

2.1.1 Fluid properties

On a macroscopic level, fluid is seen as a continuum and not as solid particles. The density of the fluid is defined as mass per unit volume. A relation between pressure, temperature and density is called an equation of state. Only isothermal processes will be discussed in this paper. An isothermal process is a process with constant temperature. Density, ρ can then be expressed as a function of pressure, p:

$$\rho = \rho(p) \tag{2.1}$$

For an incompressible fluid, the density is constant. A second property describing a fluid is the ability of a fluid to resist deformation. A fluid obeying Newton's law of viscosity, that is that the

share force per unit area is proportional to the local velocity gradient, is called a Newtonian fluid. This proportional constant is called the viscosity, μ of the fluid. All fluids considered in this paper are Newtonian. For isothermal processes the viscosity equation is a function of p,

$$\mu = \mu(p). \tag{2.2}$$

In this thesis both the density and the viscosity will be assumed to be constants.

2.1.2 Rock properties

The porosity of a porous medium is defined as the fraction of the pore space in the medium. To find the porosity, a medium needs to be pulverized to measure the matrix volume. This is not an optimal approach, not only because it destroys the sample, but also because the porosity found in this way, do not resemble the transport abilities of the medium. In a porous medium there are isolated pores and pore channels that do not reconnect to the pore. Since these pores do not contribute to the fluid flow, it is more convenient to define an effective porosity. The effective porosity of a medium is defined as:

$$\phi = \frac{V_{Pe}}{V_t}, \tag{2.3}$$

where V_t is the total volume of the medium and V_{Pe} is the effective pore volume without isolated and unconnected pores. When it is referred to porosity in this paper, it is referred to the effective porosity. An example of a porous rock is limestone. The porosity of limestone is approximately 1-10%. Sediments, such as clay, has a porosity as large as 45-55% [1].

In 1855 Henry Darcy did a series of experiments to investigate the ability of water to flow through different types of sand. For one dimensional flow he concluded that the flux, $\tilde{f}$ is proportional to the hydraulic gradient, $\tilde{J}$. The hydraulic gradient describes the direction of flow from high to low values of the total pressure and potential energies. For a horizontal case where the difference in water elevation is $h_1 - h_2$, and the length is l, the hydraulic gradient is equal to: $\tilde{J} = (h_2 - h_1)/l$. The result of these experiments is known as Darcy's law:

$$\tilde{f} = \tilde{K}\tilde{J}. \tag{2.4}$$

The proportional coefficient $\tilde{K}$ is called the hydraulic conductivity. Following Bear [1] the hydraulic conductivity can be expressed in terms of viscosity and density as

$$\tilde{K} = \frac{K\rho g}{\mu},$$

where K is called the permeability. The permeability is a measure of the ability of a medium to transmit fluids. The unit of permeability is Darcy. One Darcy is converted to SI-units, 1 Darcy $= 0.987 * 10^{-12} m^2$. Typical reservoir rocks, as sandstone and limestone, have a permeability between $10^{-1} - 10^4$ milli-Darcy(mD), while dolomite is less previous with permeability less than 10^{-1} mD. Other examples are gravel and sand with permeability between $10^2 - 10^8$ mD and granite with values as small as 10^{-5}mD [1]. In general permeability is both dependent on spatial location and direction of flow. Therefore it must be described by a second order tensor field. In this context a tensor field is an extension of the vector field. A vector field is a first order tensor. For a n dimensional space each point is associated by a vector determined by n components. For a second order tensor field, each point is associated by a matrix with $n \times n$ components. The permeability tensor is then, $\mathbf{K} = K_{ij}(x)$. A medium that is independent of spatial location is called homogeneous. The permeability tensor for a homogen medium is: $\mathbf{K} = K_{ij}$. If a medium is isotropic, it is independent of the direction of flow. For an isotropic and homogeneous medium the permeability is a constant, $\mathbf{K} = K\mathbf{I}$, where $\mathbf{I}$ is the identity matrix. Trough out this work, only isotropic mediums will be considered.

2.2 The continuity equation

Within a closed volume Ω, quantities such as mass and momentum are conserved. For a quantity $\tilde{m}$ the conservation law can be expressed by the following integral equation:

$$\int_\Omega \frac{\partial \tilde{m}}{\partial t} dV + \int_{\partial\Omega} \tilde{\mathbf{f}} \cdot \mathbf{n} dS = \int_\Omega q dV.$$

The conservation law says that the change in $\tilde{m}$ must come from a source or sink, q inside the volume or from fluxes, $\tilde{\mathbf{f}}$ through the surface of the volume. The outward unit normal vector $\mathbf{n}$ gives the direction of the surface $\partial\Omega$. The divergence theorem together with the fact that the conservation law is valid for an arbitrary volume Ω, leads to the continuity equation:

$$\frac{\partial \tilde{m}}{\partial t} + \nabla \cdot \tilde{\mathbf{f}} = q. \tag{2.5}$$

2.3 Single-phase flow

A typical oil reservoir consists not only of a single phase of oil, but also of water and gas. Still, single-phase models could be used in other types of reservoirs such as ground water reservoirs and dam reservoirs. Single-phase models can also be used for tracking pollution. Some chemical and radioactive pollution do not interact with other present phases and can therefore be modeled by single-phase models. Radioactive and chemical components can also be injected into the reservoir as tracers. Tracer concentration curves in production wells can then help reservoir engineers deciding about reservoir properties as layers, channels and how wells are connected. Examples of tracers used in reservoir characterization are NO_X and SO_4.

For a single-phase flow, the mass density, $\tilde{m}$ is given by: $\tilde{m} = \phi\rho$ and the flux density, $\tilde{\mathbf{f}}$ is given by $\tilde{\mathbf{f}} = \rho\mathbf{u}$. The velocity, $\mathbf{u}$ is not the actual particle velocity, but a macroscopic measure called the Darcy velocity. The Darcy velocity is defined as the flow rate (volume/time) divided by the cross section area along the flow path. With these expressions of the mass density and flux density, the conservation equation in (2.5) becomes:

$$\frac{\partial}{\partial t}(\phi\rho) + \nabla \cdot (\rho\mathbf{u}) = q. \tag{2.6}$$

For a incompressible fluid,

$$\nabla \cdot \mathbf{u} = 0 \tag{2.7}$$

away from the wells. Since $\nabla \cdot (\rho\mathbf{u}) = \mathbf{u} \cdot \nabla\rho + \rho\nabla \cdot \mathbf{u}$, equation (2.6) becomes

$$\phi\frac{\partial \rho}{\partial t} + \mathbf{u} \cdot \nabla\rho = q \tag{2.8}$$

for incompressible fluids. The Darcy velocity $\mathbf{u}$ is given by the generalized version of Darcy's law on differential form [1]:

$$\mathbf{u} = -\frac{\mathbf{K}}{\mu}(\nabla p - \rho\mathbf{g}). \tag{2.9}$$

In this generalized form of the Darcy's law, the flow can be either pressure driven or driven by gravitational forces. Together with the equation of state for density (2.1) and viscosity (2.2), the incompressible equation (2.7) and the continuity equation (2.8) forms a closed set of equation for single-phase flow in a porous medium. For a tracer that flows without interacting with other present phases, the continuity equation reads

$$\phi\frac{\partial C}{\partial t} + \mathbf{u} \cdot \nabla C = q, \tag{2.10}$$

where C is the concentration of the tracer.

2.4 A model for multi-phase flow

In general, a reservoir consists of multiple phases. In addition to already present oil and gas, water is often injected to increase oil recovery. Another way to increase oil recovery is to inject gas. The injected gas is miscible with the displaced fluid. Therefore a mixed zone will be created between the different phases. In this thesis only immiscible fluids will be discussed. Water and oil are examples of immiscible fluids. For an immiscible fluid, saturation, S_α of phase α is defined as the fraction of the void volume of the porous media:

$$S_\alpha = \frac{V_\alpha}{V_{Pe}}, \tag{2.11}$$

where V_α is the volume occupied by phase α and V_{Pe} is the effective pore volume. Assuming all pores to be filled with fluids, the total saturation at a point is always one,

$$\sum_\alpha S_\alpha = 1. \tag{2.12}$$

Forces acting between different medium are called surface tensions. These tensions bend the surfaces such that it is always true that the pressure on the concave side is larger than the pressure on the convex side. The fluid on the convex side is called wetting and the fluid on the concave side is called nonwetting. The difference in pressure between the phases is called the capillary pressure,

$$P_c = p_{nw} - p_w, \tag{2.13}$$

where p_{nw} is the pressure in the non-wetting phase and p_w is the pressure in the wetting phase. For idealized models it is possible to derive a expression for $P_c = P_c(S_w)$, where S_w is the saturation of the wetting phase w. For a general medium a relation between saturation of the wetting phase and the capillary pressure must be found experimentally. Only one phase of immiscible fluids can flow through a pore at a given time. Still, other present phases will reduce the pore space and therefore influence the permeability. A relative permeability is defined to decouple the effect of the present phases from the rock permeability. The effective permeability for wetting and non-wetting fluid is given by

$$\mathbf{k_w} = k_{rw}\mathbf{K}, \tag{2.14}$$

$$\mathbf{k_{nw}} = k_{rnw}\mathbf{K}, \tag{2.15}$$

where k_{rw} and k_{rnw} is the relative permeability for wetting and nonwetting fluids. To simplify notation, mobility is defined as:

$$\lambda_\alpha = k_{r\alpha}/\mu_\alpha, \tag{2.16}$$

where $k_{r\alpha}$ is the relative permeability and μ_α is the viscosity.

For a multi-phase and multi-dimensional flow, a generalization of Darcy's law for single phase flow given in (2.9) can be made. Several attempts have been made to prove the generalized Darcy's law from more fundamental laws of physics. By averaging the Navier-Stokes equations, Whitaker showed that Darcy's law could by motivated from the principle of conservation of momentum. Other examples can be found in [1]. Even though Darcy's generalized law cannot be proved, it is supported by experimental data, therefore it is assumed that

$$\mathbf{u}_\alpha = -\lambda_\alpha \mathbf{K}(\nabla p_\alpha - \rho_\alpha \mathbf{g}) \tag{2.17}$$

for each phase α [1]. Darcy's generalized law is an empiric law that tells that the velocity, $\mathbf{u}_\alpha$ of the flow in a porous media is proportional to the pressure gradient, ∇p_α and the gravitational force rate, $\rho_\alpha \mathbf{g}$. The proportional constant is divided into two parts. The first part is a tensor describing the rock permeability, $\mathbf{K}$. The second part is the relative mobility of the flow, λ_α. While the relative mobility is depending on the saturation and concentration of other present phases, the rock permeability is only depending on the rock.

By using the conserved mass density, $\tilde{m}_\alpha = \phi\rho_\alpha S_\alpha$ and the flux density, $\tilde{\mathbf{f}}_\alpha = \rho_\alpha \mathbf{u}_\alpha$, the following modified form of the continuity equation could be found:

$$\frac{\partial}{\partial t}(\phi\rho_\alpha S_\alpha) + \nabla \cdot (\rho_\alpha \mathbf{u}_\alpha) = q_\alpha. \tag{2.18}$$

The modified continuity equation (2.18) and Darcy's law (2.17) forms a coupled set of nonlinear differential equations. Together with the equation of state (2.1), the viscosity equation (2.2), the capillary pressure equation (2.13) and the assumptions that all pores are totally filled with fluids (2.12), these equations form a closed set of equations. Adding initial and boundary conditions, it is possible to solve these sets of equations, and therefore describe the fluid flow. Still the complexity of the equations and the geometry of the reservoir make this a difficult task. Analytic solutions exist only for simplified equations with simple boundaries. In practice numerical methods, for example based on finite differences or finite elements must be used to find a solution. Solving the full model equations for a multi-phase flow is computational intensive. To solve the flow equations efficiently, simplified models can be introduced. See Chen et al [19] for a guide to computational methods for multi-phase flow in porous medium. In this thesis a simplified model for two phase flow due to Buckley and Leverett will be used.

2.5 The Buckley-Leverett model for two phase flow

For a two phase flow of oil and water that is incompressible, immiscible, and with constant viscosity, and where the effect of gravity and capillary forces are neglected, Buckley and Leverett (1942) introduced a simplified fluid model for flow in a porous medium. For constant ρ and ϕ equation (2.18) can be rewritten as

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

The total velocity for a two face flow of oil, $\alpha = o$ and water, $\alpha = w$ is defined as: $\mathbf{u} = \mathbf{u_o} + \mathbf{u_w}$. Combining the equations from (2.19) for $\alpha = (o, w)$ gives

$$\frac{\partial}{\partial t}(S_o + S_w) + \nabla \cdot (\mathbf{u_o} + \mathbf{u_w}) = q, \tag{2.20}$$

where $q = \frac{q_o}{\rho_o} + \frac{q_w}{\rho_w}$. The fluids are assumed to fill the pore space completely therefore $S_o + S_w = 1$, and (2.20) is reduced to:

$$\nabla \cdot \mathbf{u} = q. \tag{2.21}$$

Neglecting capillary forces such that $P_c(S_w) = 0$ gives equal pressure for the oil and water phase: $p = p_o = p_w$. The following expressions can then be found from Darcy's law (2.17):

$$\mathbf{u}_w = -\lambda_w \mathbf{K}(\nabla p - \rho_w \mathbf{g}) \Rightarrow \nabla p = (\frac{-\mathbf{u}_w}{\lambda_w \mathbf{K}} + \rho_w \mathbf{g}), \tag{2.22}$$

$$\mathbf{u}_o = -\lambda_o \mathbf{K}(\nabla p - \rho_o \mathbf{g}) \Rightarrow \nabla p = (\frac{-\mathbf{u}_o}{\lambda_o \mathbf{K}} + \rho_o \mathbf{g}). \tag{2.23}$$

By combining (2.22), (2.23) and using that $\mathbf{u}_o = \mathbf{u} - \mathbf{u}_w$ the following expression can be found for $\mathbf{u}_w$:

$$\mathbf{u}_w = F_w \mathbf{u} + F_w \lambda_o \mathbf{K}\mathbf{g}(\rho_o - \rho_w),$$

here the fractional flow of water F_w is defined as:

$$F_w = \frac{\lambda_w}{\lambda_t}, \tag{2.24}$$

where λ_t is the total mobility given by $\lambda_t = \lambda_o + \lambda_w$. For incompressible fluid, $\nabla \cdot (F_w \mathbf{u}) = \mathbf{u} \cdot \nabla F_w$ since $\nabla \cdot \mathbf{u} = 0$ away from the wells. Finally, by neglecting gravity, the equation for water saturation becomes:

$$\phi\frac{\partial S_w}{\partial t} + \mathbf{u} \cdot \nabla F_w = \tilde{q}, \tag{2.25}$$

where $\tilde{q} = \frac{q_w}{\rho_w}$. By eliminating the w subscript, equation (2.21) and (2.25) becomes

$$\nabla \cdot \mathbf{u} = q, \tag{2.26}$$

and

$$\phi \frac{\partial S}{\partial t} + \mathbf{u} \cdot \nabla F = \tilde{q}. \tag{2.27}$$

The velocity, $\mathbf{u}$ is given by the Darcy's law (2.17) without the gravity term:

$$\mathbf{u} = -\mathbf{K}\lambda_t \nabla p. \tag{2.28}$$

The elliptic equation (2.26) is called the pressure equation, and the hyperbolic transport equation (2.27) is usually referred to as the Buckely-Leverett equation. Since the equations are dependent on each other they have to be solved simultaneously. For some special cases with simple initial and boundary conditions, it is possible to find analytic solutions. Some examples can be found in [2] and [19]. These analytic solutions can give valuable understanding to the problem. Still, for practical applications, a numerical approach is necessary.

In the Buckely-Leverett model the pressure equation and the saturation equation forms a set of coupled differential equations. One way to solve these equations is to discretize them using finite-differences and then solve them sequentially in time. The widely used reservoir simulator Eclipse is based on an IMPES approach. IMPES solves an implicit discretization of the pressure equations and an explicit discretization of the saturation equation sequentially in time. See [19] for details. Another approach is to use a finite element discretization. The reservoir simulator developed by Sintef, which is used in this master thesis, uses a multi-scale mixed finite element solver for the pressure equation (MsMFEM). A short description of MsMFEM based on work by Aarnes at al at Sintef [9] can be found in appendix A.

2.6 Summary and comments

In this chapter a mathematical model for fluid flow in a porous medium is introduced. Simplified equations for single- and two-phase flow are found and both analytic and numerical solution techniques are presented. The presentation in this chapter is general and is not restricted to a streamline formulation. In the next chapter further simplification of these equation will derived be introducing the concept of streamlines.

Chapter 3

Streamline based transport simulation

Make everything as simple as possible, but not simpler.
Albert Einstein

The computational speed is the most important reason for introducing a streamline based method for simulation of fluid transport. Past years development in reservoir characterization has made it possible to generate fine scale static models based on geologic and geophysical data. These fine scale static models can include million of cells, making a simulation based on conventional grid based methods time consuming. Even though streamline based methods have limitations, and therefore never can substitute conventional methods, they can help provide important knowledge about the reservoir. The strengths and limitations of streamline based methods will be discussed in detail in the end of this chapter, but first some basic concepts need to be introduced.

3.1 Basic definitions

3.1.1 Streamlines

Streamlines are instantaneous lines that are everywhere tangential to a velocity field. Streamlines can therefore be defined as the function $\mathbf{x}$ that solves the differential equation:

$$d\mathbf{x} \times \mathbf{u}(\mathbf{x}, t_0) = 0, \qquad \mathbf{x}(t_0) = \mathbf{x}_0, \tag{3.1}$$

where $\mathbf{u}$ is the velocity field. Note that streamlines are not time dependent. By knowing the initial position $\mathbf{x}_0$ and the initial velocity field $\mathbf{u}(\mathbf{x}, t_0)$, streamlines can be generated instantly for the whole region. This is in contrast to path lines. A path line is tangential to the velocity field at time t, and describes the physical trajectory of a particle through space and time. For steady velocity, streamlines will correspond to path lines, else streamlines represents the instantaneous velocity field, and not the path of a particle. If the velocity field is time dependent, different streamlines needs to be defined at each time of interest. Another important difference between path lines and streamlines is that streamlines never cross for divergence free flow. Therefore fluid transport can be calculated independently on each streamline.

3.1.2 A time-of-flight based coordinate system

A key concept of streamline methods is to use time-of-flight instead of arc-length to parameterize the streamlines. By introducing time-of-flight as a spatial coordinate variable, the effect of the geological heterogeneity is embedded into a time-of-flight based coordinate system. The other coordinate variables are the Streamfunction in 2D and the Bi-Streamfunction in 3D.

The Streamfunction for 2D flow

A 2D flow of an incompressible fluid can be described in Cartesian coordinates by a function Ψ defined such that:

$$u_x = \frac{\partial \Psi}{\partial y}, \tag{3.2}$$

$$u_y = -\frac{\partial \Psi}{\partial x}. \tag{3.3}$$

With this choice of function Ψ, the continuity equation for incompressible flow is automatically satisfied:

$$\nabla \cdot \mathbf{u} = \frac{\partial^2 \Psi}{\partial x \partial y} - \frac{\partial^2 \Psi}{\partial x \partial y} = 0.$$

For a 2D flow in the xy-plane $\mathbf{x} = (x, y, 0)$ and $\mathbf{u} = (u_x, u_y, 0)$. Therefore the definition of streamlines in (3.1) simplify to

$$\frac{dx}{u_x} - \frac{dy}{u_y} = 0. \tag{3.4}$$

Substituting (3.3) and (3.3) into (3.4) gives

$$\frac{\partial \Psi}{\partial x} dx + \frac{\partial \Psi}{\partial y} dy = 0,$$

which means that $d\Psi = 0$ along the streamlines and that

$$\Psi = \Psi(x, y) = \text{constant}, \tag{3.5}$$

is a solution of (3.4). The function Ψ is the Streamfunction for 2D flow. The Streamfunction and the streamlines are related such that the lines given by, $\Psi = \text{constant}$ are the streamlines.

Bi-Streamfunctions

The notion of Streamfunction can be generalized to 3D flow. For divergence free flow, it is shown by Bear [1] that the velocity in 3D can be described in terms of two scalar functions,

$$\mathbf{u} = \nabla \psi \times \nabla \chi.$$

Observe that by this definition the continuity equation is satisfied, since

$$\nabla \cdot \mathbf{u} = \nabla \cdot (\nabla \psi \times \nabla \chi) = 0,$$

for arbitrary scalar functions. The surfaces defined by

$$\psi = \psi(x, y, z) = \text{constant}, \tag{3.6}$$

$$\chi = \chi(x, y, z) = \text{constant}. \tag{3.7}$$

solves (3.1) [1], and are known as Bi-Streamfunctions in 3D. The intersection of these Bi-Streamfunctions defines streamlines, since the cross product $\nabla \psi \times \nabla \chi$ must lie in both of the surfaces. Even though Bi-Streamfunctions are sufficient to describe fluid flow, they are in general difficult to construct. Therefore Bi-Streamfunctions are seldom used directly, but they play an important role in constructing the time-of-flight based coordinate system.

time-of-flight

Time-of-flight, τ is the travel time of a passive tracer along a streamline [2]. Since the velocity varies along the streamline, the time-of-flight along the streamline given by Γ is

$$\tau = \int_\Gamma \frac{\phi(\mathbf{x}(\xi))}{\mid \mathbf{u}(\mathbf{x}(\xi)) \mid} d\xi, \tag{3.8}$$

where ξ is the arc-length. Alternatively

$$\mathbf{u} \cdot \nabla \tau = \phi. \tag{3.9}$$

The two definitions are equivalent since

$$\frac{\phi}{|\mathbf{u}|} = \frac{d\tau}{d\xi} = \frac{\mathbf{u}}{|\mathbf{u}|} \cdot \nabla \tau.$$

The first of these equalities is due to the fundamental theorem of calculus and the second is by the definition of directional derivatives. Since time-of-flight is given in the direction of the streamlines, they are orthogonal to the bi-streamfunctions. A new coordinate system with (τ, ψ, χ) as orthogonal bases, can then be constructed. The Jacobian of the transformation $(x, y, z) \mapsto (\tau, \psi, \chi)$ is

$$|\frac{\partial(\tau, \psi, \chi)}{\partial(x, y, z)}| = |(\nabla\psi \times \nabla\chi) \cdot \nabla\tau| = |\mathbf{u} \cdot \nabla\tau| = \phi. \tag{3.10}$$

By using the chain rule the Cartesian gradient operator can be given with streamline coordinates (τ, ψ, χ):

$$\nabla = (\nabla\tau)\frac{\partial}{\partial\tau} + (\nabla\psi)\frac{\partial}{\partial\psi} + (\nabla\chi)\frac{\partial}{\partial\chi}.$$

Since $\mathbf{u}$ is orthogonal to both $\nabla\psi$ and $\nabla\chi$ it follows that

$$\mathbf{u} \cdot \nabla = (\mathbf{u} \cdot \nabla\tau)\frac{\partial}{\partial\tau} = \phi\frac{\partial}{\partial\tau}. \tag{3.11}$$

Equation (3.10) and (3.11) will be used to simplify the model equations, but first more has to be said about the calculation of streamlines and time-of-flight. If the velocity field can be expressed in terms of a simple function, streamlines may be found analytically by solving the corresponding differential equations in (3.1). Usually, the velocity field is found numerically by solving the pressure equation. In this case a semi-analytic method due to Pollock is commonly used to trace streamlines.

3.1.3 Pollock's algorithm for tracing streamlines

Pollock's method is based on making a piecewise linear approximation of the velocities in each direction:

$$\begin{aligned} u_x &= u_{x_1} + c_x(x - x_1), \\ u_y &= u_{y_1} + c_y(y - y_1), \\ u_z &= u_{z_1} + c_z(z - z_1), \end{aligned} \tag{3.12}$$

where $(x_1, x_2, y_1, y_2, z_1, z_2)$ are the corners in a cell and $(u_{x_1}, u_{x_2}, u_{y_1}, u_{y_2}, u_{z_1}, u_{z_2})$ gives the fluid velocities on each of the six faces of the cell. See Figure 3.1. These face velocities, can be found be solving the pressure equation (2.26). The coefficients in equation (3.12) are given by the difference of the Darcy velocities on each of the cell faces:

$$c_x = (u_{x_2} - u_{x_1})/(x_2 - x_1),$$

$$c_y = (u_{y_2} - u_{y_1})/(y_2 - y_1),$$

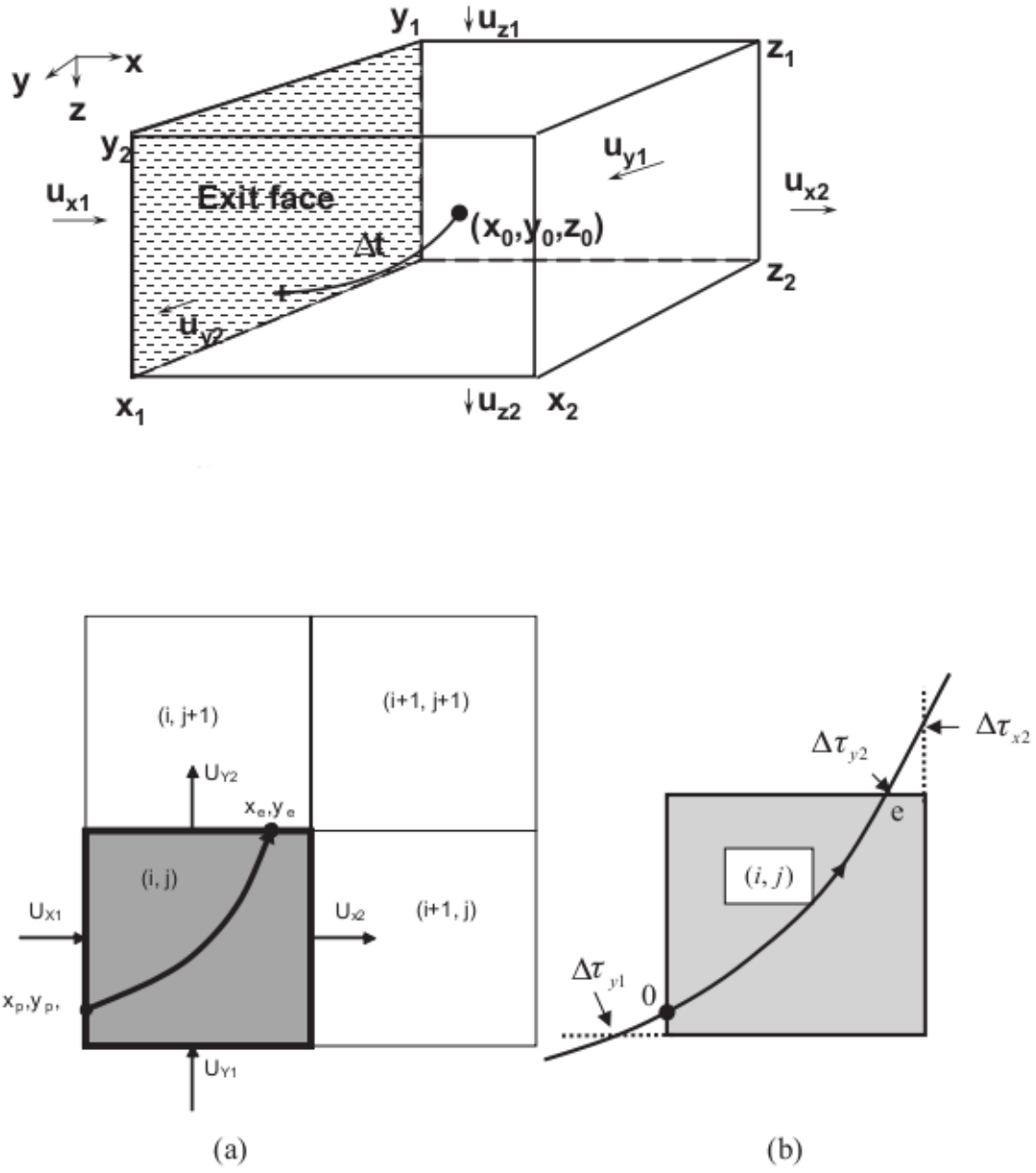

Figure 3.1: Illustration of the Pollock's algorithm in 2D, after [2].

$$c_z = (u_{z_2} - u_{z_1})/(z_2 - z_1).$$

The streamlines can be found by using the definition in (3.1) for each direction, that is:

$$\frac{d\tau}{\phi} = \frac{dx}{u_x} = \frac{dy}{u_y} = \frac{dz}{u_z}. \tag{3.13}$$

The first equality comes from the definition of time-of-flight in (3.9). For a particle that starts in $\mathbf{r}_0 = (x_0.y_0, z_0)$, integrating (3.13) for the linear velocities given in (3.12) gives the time-of-flight to each of the six faces:

$$\begin{aligned}
\Delta\tau_{x_i} &= \phi \int_{x_0}^{x_i} \frac{dx}{u_x} = \frac{1}{c_x} \ln\left(\frac{u_{x_i}}{u_{x_0}}\right) \\
\Delta\tau_{y_i} &= \phi \int_{y_0}^{y_i} \frac{dy}{u_y} = \frac{1}{c_y} \ln\left(\frac{u_{y_i}}{u_{y_0}}\right) \\
\Delta\tau_{z_i} &= \phi \int_{z_0}^{z_i} \frac{dz}{u_z} = \frac{1}{c_z} \ln\left(\frac{u_{z_i}}{u_{z_0}}\right)
\end{aligned} \tag{3.14}$$

where $i = (1, 2)$. In Pollock's method the time-of-flight is given as the minimum positive time-of-flight, that is:

$$\Delta\tau = \min_{\Delta\tau > 0} (\Delta\tau_{x1}, \Delta\tau_{x2}, \Delta\tau_{y1}, \Delta\tau_{y2}, \Delta\tau_{z1}, \Delta\tau_{z2}). \tag{3.15}$$

The exit coordinates can be found by rearranging (3.14):

$$\begin{aligned}
x &= x_0 + u_{x0} \left(\frac{\mathrm{e}^{c_x \Delta\tau/\phi} - 1}{c_x}\right) \\
y &= y_0 + u_{y0} \left(\frac{\mathrm{e}^{c_y \Delta\tau/\phi} - 1}{c_y}\right) \\
z &= z_0 + u_{z0} \left(\frac{\mathrm{e}^{c_z \Delta\tau/\phi} - 1}{c_z}\right)
\end{aligned} \tag{3.16}$$

Observe from equation (3.12) that, $\nabla \cdot \mathbf{u} = c_x + c_y + c_z$, which means that the discrete method conserves flux locally for an incompressible flow.

3.2 The transport equation along a streamline

3.2.1 Single-phase flow

The continuity equation of tracer flow given in (2.10) can be mapped from the Cartesian coordinates to a time-of-flight based coordinate system. The transport equation for tracer flow along the streamline is then given by

$$\frac{\partial C}{\partial t} + \frac{\partial C}{\partial \tau} = q. \tag{3.17}$$

This comes directly from putting (3.11) into the continuity equation for a single-phase flow from Chapter 2. This equation has an analytic solution. Observe that equation (3.17) is just the definition of the directional derivative in the direction $(t, \tau) = (1, 1)$, therefore the solution must be constant along each characteristic, $t - \tau =$ constant. For initial condition, $C(0, \tau) = C_0(\tau)$, the tracer concentration is given by

$$C(t, \tau) = C_0(t - \tau). \tag{3.18}$$

For a constant injecting rate C_0 and zero initial tracer concentration, the solution becomes

$$C(t, \tau) = \begin{cases} 0, & t < \tau, \\ C_0, & t > \tau. \end{cases}$$

In [6] tracer data is used for history matching. An advantage of using tracer data for history matching is that analytic solutions can be found. Tracer data will however not be used for history matching in this theses. Instead water cut data from a two-phase flow will be used.

3.2.2 Two-phase flow

As with the single-phase flow problem, the introduction of a time-of-flight based coordinate system reduces the Buckely-Leverett equation to simple 1D transport equations along the streamlines. Applying equation (3.11) to the Buckely-Leverett equation (2.27) gives

$$\phi\frac{\partial S}{\partial t} + \phi\frac{\partial F(S)}{\partial \tau} = \tilde{q},$$

along the streamlines. Since each streamline is traced between wells there will be no sink or source term, $\tilde{q} = 0$. Dividing the equation on ϕ simplifies it further to:

$$\frac{\partial S}{\partial t} + \frac{\partial F(S)}{\partial \tau} = 0. \tag{3.19}$$

The 1D transport equation given in (3.19) describes changes in water saturation along the streamline. For a Buckley-Leverett displacement it is also possible to find analytic solutions. Solving the differential equation in (3.19) with initial water saturation S_0 and constant injecting state S_{inj}, corresponds to solving a Riemann problem with initial data

$$S(0,\tau) = \begin{cases} S_{inj}, & \tau < 0, \\ S_0, & \tau \geq 0. \end{cases} \tag{3.20}$$

Assuming $S_{inj} > S_0$, the solution for $F''(S) > 0$ will be [4]

$$S(t,\tau) = \begin{cases} S_{inj}, & \tau < \sigma t, \\ S_0, & \tau \geq \sigma t, \end{cases} \tag{3.21}$$

where σ is the shock speed given by the Rankine-Hugoniot condition

$$\sigma = \frac{F(S_0) - F(S_{inj})}{S_0 - S_{inj}}. \tag{3.22}$$

For $F''(S) > 0$, the saturation will form a shock with height S_{inj} that travels with shock speed σ. The waterfront will reach the producer at $t = \tau/\sigma$. For $F''(S) < 0$ the solution becomes[4]

$$S(t,\tau) = \begin{cases} S_{inj}, & \tau < F'(S_{inj})t, \\ (F')^{-1}(\tau/t), & F'(S_{inj})t \leq \tau \leq F'(S_0)t, \\ S_0, & \tau > F'(S_0)t \end{cases}$$

For $F''(S) < 0$ there will be no shock, instead a rear fraction wave will form a continuous solution between the different states S_{inj} and S_0. If F is both convex and concave the upper concave envelope of F must be constructed, that is $\tilde{F}$. The solution will then be

$$S(t,\tau) = \begin{cases} S_{inj}, & \tau < \tilde{F}'(S_{inj})t, \\ (F')^{-1}(\tau/t), & \tilde{F}'(S_{inj})t \leq \tau \leq \tilde{F}'(S_0)t, \\ S_0, & \tau > \tilde{F}'(S_0)t. \end{cases}$$

For more general displacements where the injecting problem is not a simple Riemann problem, numerical methods must be used. An often used numerical method called the Upwind method for solving the transport equation is based on taking the one-sided approximation of the derivative in space and a backward Euler step in time. An disadvantage with the upwind method is that numerical diffusion is introduced [10]. Therefore a better method that does not introduce numerical diffusion will be used. This method is called front tracking.

3.2.3 Front Tracking

The main idea behind front tracking is to approximate the initial saturation by piecewise constant functions, and the flux function by piecewise linear functions, and then solve the Riemann problems associated with each of the initial discontinuities. By tracking the fronts and solving new Riemann problems whenever fronts collides, a global solution can be found. Following Holden and Risebro in [8], the procedure for solving the one-dimensional transport equation (3.19) will be:

1. Approximate F by a continuous, piecewise linear flux function F^δ.
2. Approximate initial data $S(0,\tau)$ by a piecewise constant function $S^\eta(0,\tau)$.
3. Solve the initial value problem

$$\frac{\partial S}{\partial t} + \frac{\partial F^\delta(S)}{\partial \tau} = 0, \qquad S|_{t=0} = S^\eta(0,\tau),$$

   exactly for each η and δ. Denote the solution $S_{\delta,\eta}$.
4. As F^δ and $S^\eta(0,\tau)$ approach F and $S(0,\tau)$, respectively, the approximate solution $S_{\delta,\eta}$ will converge to S, the solution of (3.19).

Since the front tracking method solves the Riemann problems exactly, the method do not need to take into account the CFL criteria [8], and can therefore use larger time steps. This method does not introduce numerical diffusion and will therefore preserve the frontal structure of the solution.

3.2.4 Mappings between Streamlines and the pressure grid

To get the saturation in each cell, the saturation along each streamline must be mapped back to the underlying grid. The initial saturation, S_{inj} used for solving equation (3.19) can be obtained by picking the values from the pressure grid.

$$S_{l,i} = S_i$$

Note that for the Riemann problem given in (3.20), S_{inj} for streamline l is equal to $S_{l,i} = S_i$, where i is the cell number where the streamline starts. To map the saturation from the streamlines back to the pressure grid, a volumetric average can be used. Each streamline l is considered as a representation of a stream tube with cross section A. The volumetric flux, q_l is defined such that $q_l = |\mathbf{u}(\xi)|A(\xi)$. The pore volume associated with streamline l is

$$V_l = \int_\Gamma \phi(\xi)A(\xi)\mathrm{d}\xi = q_l \int_\Gamma \frac{\phi(\xi)}{|\mathbf{u}(\xi)|}\mathrm{d}\xi = q_l\tau_l.$$

The volume associated with a streamline going through cell i is $V_{l,i} = q_l\Delta\tau_{l,i}$. The saturation in cell i is given as the average of all streamlines going through that cell:

$$S_i = \frac{\sum_l S_{l,i}V_{l,i}}{\sum_l V_{l,i}}. \tag{3.23}$$

All streamlines are traced from equally distant points at the faces of cells where the injector is placed. The streamlines will then carry approximately the same amount of fluid; that is, q_l=q/N_{sl}, where q is the total well rate and N_{sl} is the total number of streamlines. For large and heterogeneous reservoir streamlines will not go through all cells. These untraced cells can either be ignored or alternatively additional streamlines can be traced both forward to a producer and backward to an injector. Since untraced cells often lie in regions that contribute little to the well production they will be ignored.

A disadvantage of the mapping in 3.23 is that it does not conserve mass, and may lead to incorrect saturation and production curves. Stenerud et al. [17] therefore suggest an alternative

method, based on the observation that $\Delta\tau_{l,i}$ may not be a good approximation to the average time-of-flight over the cross section of the associated stream tubes. To guarantee global conservation of mass, a factor $\alpha_i = V_i / \sum_l V_{l,i}$ is used to scale $\Delta\tau_{l,i}$ in cell i. The factor α guarantees global conservation of mass for the mapping from grid to streamlines,

$$V^w_{grid} = \sum_i V_i S_i = \sum_{l,i} \alpha_i V_{l,i} S_i = \sum_{l,i} \alpha_i V_{l,i} S_{l,i} = V^w_{sl},$$

where V^w_{sl} and V^w_{grid} are the total volumes for water on the streamline grid and pressure grid, respectively. The same is also valid for the mapping from streamlines to grid. By forcing mass conservation, local errors in the saturation profile are introduced. This is because the scaling factor α corresponds to a local stretching or shrinking of the streamline grid. But as demonstrated by Stenerud et al. [17], the global properties of the solution are better. The water breakthrough time at a producer, can be used as data in a history matching procedure. Therefore to get an accurate breakthrough time, this scaling is only applied after the water breakthrough. The effect of this mapping is showed in one of the examples that follow.

3.3 Examples

In the two first chapter of this thesis, the theory behind streamline simulation has been introduced. To illustrate some of the practical aspects of streamline simulation some examples will be considered. In all examples and through out this thesis the SAM Research Simulator version 1.0 from Sintef is used. More details about the reservoir simulator can for instance be found in [16] and [17].

As the first example, a reservoir with 10 grid cells in each dimension is considered. All cells are $1m \times 1m$ in size and have constant porosity 1. Three different rock permeability fields is used to illustrate the effect of the permeability has on the solution, these is referred to as Case 1, 2 and 3, respectively. The initial permeability is given in Figure 3.2. Initially all cells are filled with oil with density $1kg/m^3$ and viscosity $2centipose$. Water, with density $1kg/m^3$ and viscosity $1centipose$, is injected with a constant rate $1.0m^2/day$ in the lower right corner, grid cell $(10, 10)$. A producer is placed in the upper left corner, $(1, 1)$. The relative permeability of water and oil is given by: $k_{rw}(S_w) = S_w^2$ and $k_{ro}(S_w) = (1 - S_w)^2$. By the definition of mobility (2.16); $\lambda_w = k_{rw}/\mu_w = S_w^2$ and $\lambda_o = k_{ro}/\mu_o = (1 - S_w)^2/2$. The fractional flow of water is therefore $F(S_w) = \lambda_w/(\lambda_o + \lambda_w) = 2S_w^2/(1 - 2S_w + 3S_w^2)$.

The results of the simulation for these examples are plotted in different figures. In all simulations, 10 steps are taken, each with step size 10 and the number of streamlines used are 100. Water saturation for different times are given in Figure 3.3. For Case 1 with homogeneous permeability, the water is flowing radially out from the injecting well. The plot in the middle of Figure 3.4 shows that the water reaches the producer after approximately 50 days. In Case 2 and Case 3 the water saturates the reservoir almost radially, but the effects of the difference in permeability field are clearly seen. The water cut profile for the different cases shows no significant differences, apart from the arrival time. In Case 2 and 3 water arrive after 40 days of injection. Even thought the permeability fields are totally different the production data is not. This is because the wells are forced to produce with a constant rate. To obtain constant production the pressure at the well is adjusted. In more realistic problems the production of the wells depends on the reservoir pressure. The reservoir simulator used in this thesis uses only simple wells that inject and produce with constant rates.

The second example is used to illustrate how streamlines and time-of-flight are dependent on the heterogeneity of the permeability field. The Sintef simulator uses the method of Pollock to trace streamlines. The Figures in 3.5 clearly show how the streamlines are dependent of the permeability. The structure in the permeability field in Case 2 is clearly seen in the streamlines in Figure 3.5. In all these simulations a total of 100 streamlines are used. Observe that in the corners away from the wells there are almost no streamlines. Also in the middle of the reservoir the number of streamlines is fewer than expected. This is due to the regular two dimensional

Cartesian grid that is used. Streamlines are traced from equally distant points at the faces of cell $(10, 10)$, where the injector is placed [16]. Since no streamlines are traced from the corners, the center for the region is under represented by streamlines. The time-of-flight in the producer is seen in Figure 3.5.

The Sintef simulator uses Front tracking to solve the transport equation. The third example compares the numerical solution obtained from the Sintef simulator with an analytic solution. Since F is either convex nor concave, the convex envelope of $F(S)$ must be constructed. That is the straight line from $(0, 0)$ that intersect $F(S)$, but do not cross $F(S)$. The slope of the straight line is the shock speed. The slope of the straight line from $(0, 0)$ that intersect $F(S)$, but do not cross is 1.366 for the fractional flow given above. Therefore the shock speed, $\sigma = 1.366$. The analytic and the numerical solution from the Sintef Simulator for streamline 1 and 25 is compared in Figure 3.6. Streamline 1 is in the middle, and streamline 25 is the first from right. The numerical solution does not match the analytic solution very well, especially for streamline 25. Since the front tracking method is semi-analytic it should solve the saturation equation exactly for this simple example. The difference between the numerical and analytic solution is due to the coarse Cartesian grid. The effects of the grid can be seen as small jumps along the saturation profile. In Figure 3.7 the solution can be seen for a finer 100×100 grid. Observe that the grid effects are less obvious for this case, both for streamline 1 and 25. To be able to get a saturation profile on grid cell i the mapping in (3.23) are used. An illustrate of what happens if too few streamlines are traced in the simulation, can be seen in Figure 3.9, but observe that there are no major difference in the water cut in Figure 3.8. This is due to the α-factor introduced in the mapping. All these examples show that there are different aspects such as number of streamlines and grid cells that must be kept in mind when using a streamline simulator. More about spatial errors and convergence is found in the textbook by Datta-Gupta and King [2].

3.4 Summary and comments

In this chapter the basic concepts of streamline based flow simulation have been introduced. The main key for the streamline simulation is the introduction of time-of-flight. With time-of-flight the heterogeneity of the reservoir is embedded into the coordinate system and the multi-dimensional transport equation is decoupled into one-dimensional problems. This gives significant speed up compared to traditional methods. For advection dominated flow the pressure equation does not need to be updated at every time step. This is also a huge advantage since the pressure equation often is the most computational expensive part of the simulation. Dispersion will however be present in the flow equations if the effect of capillary forces is included in the model. The dispersion term will then introduce transverse fluid flow and therefore violate the important assumption above. The same cross streamline flow will also be present if the fluid is more compressible. Dispersion can however be handled by updating the streamlines more often, but then the computationally expensive pressure equation most be solved equivalently often. Effects from gravitation, changing well condition, etc can also be included in the model [2].

For an incompressible advection dominated fluid flow, streamline methods are optimal, but for more complex models other methods such as IMPES are better to use. Another disadvantage with a streamline based method is the inaccurate mapping between the streamlines and the grid cells. The examples in Section 3.3 showed how grid size and number of streamlines affects the solution accuracy. Still much work remains to be done to get a better understanding of error estimates and convergence. For uncertainty analysis and history matching where a large number of realizations of the model are needed, streamline simulation has shown its greatest potential. Streamline simulation will therefore be used to solve the history matching problem given in the introduction. But first some more has to be said about the nature of inverse problems.

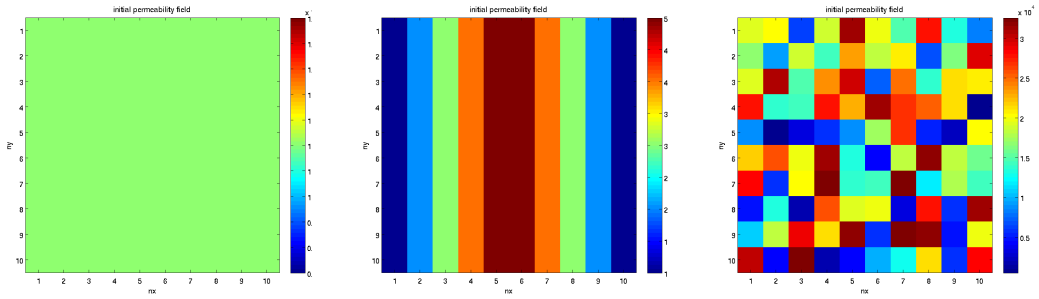

Figure 3.2: Initial permeability field for Case 1 (left), 2 (center) and 3 (right)

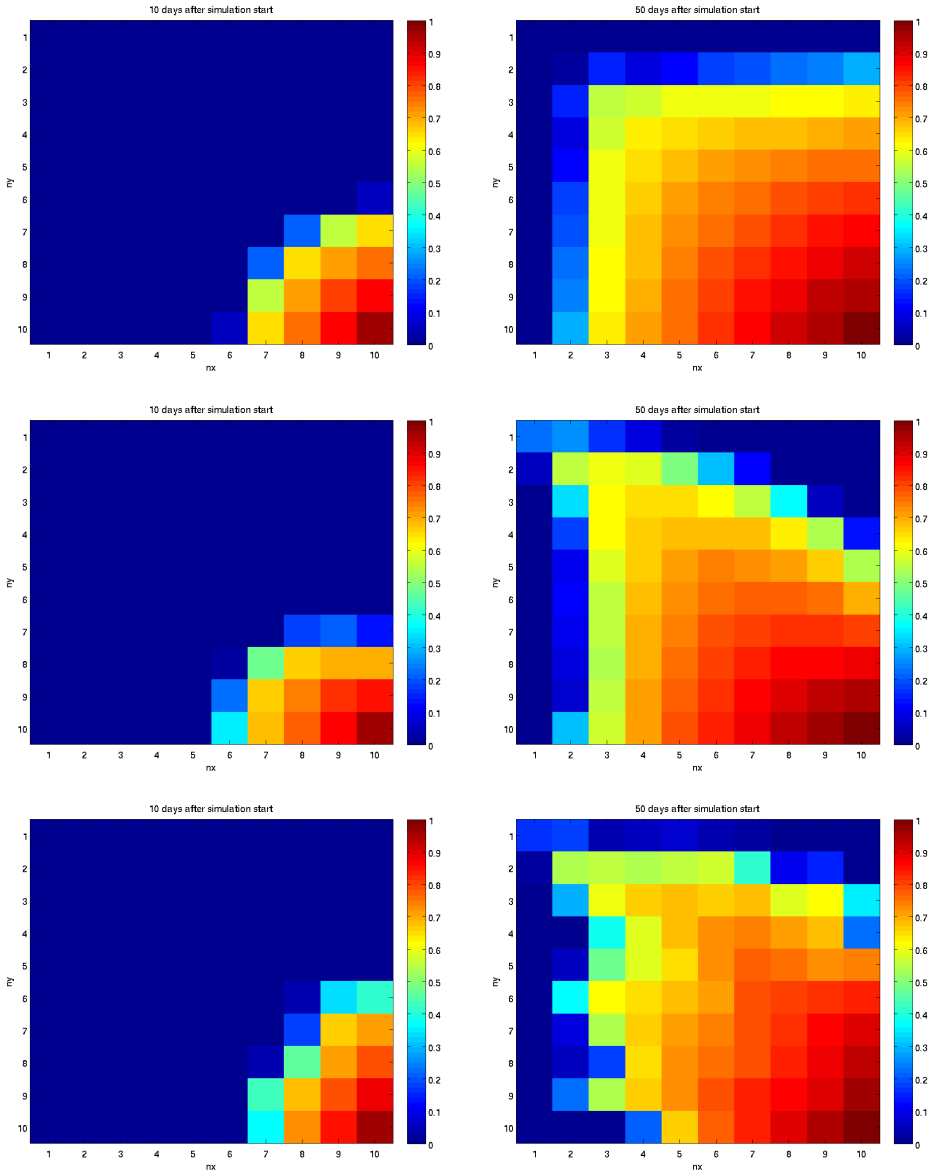

Figure 3.3: Water saturation after 10 (left) and 50 (right) days for case 1 (top), 2 (middle) and 3 (bottom).

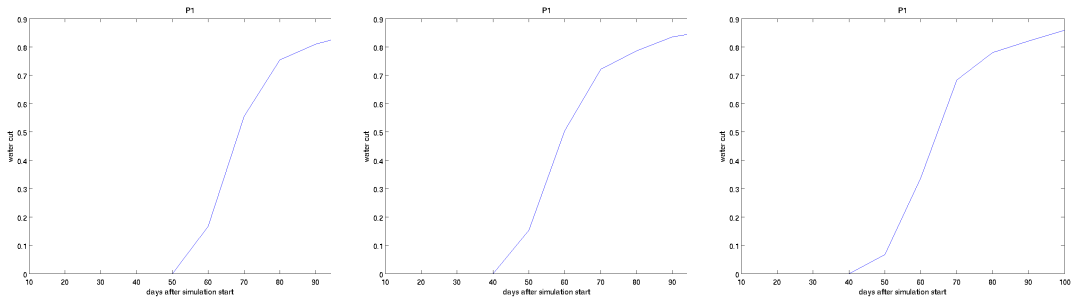

Figure 3.4: Watercut at producer for Case 1 (left), 2 (center) and 3 (right).

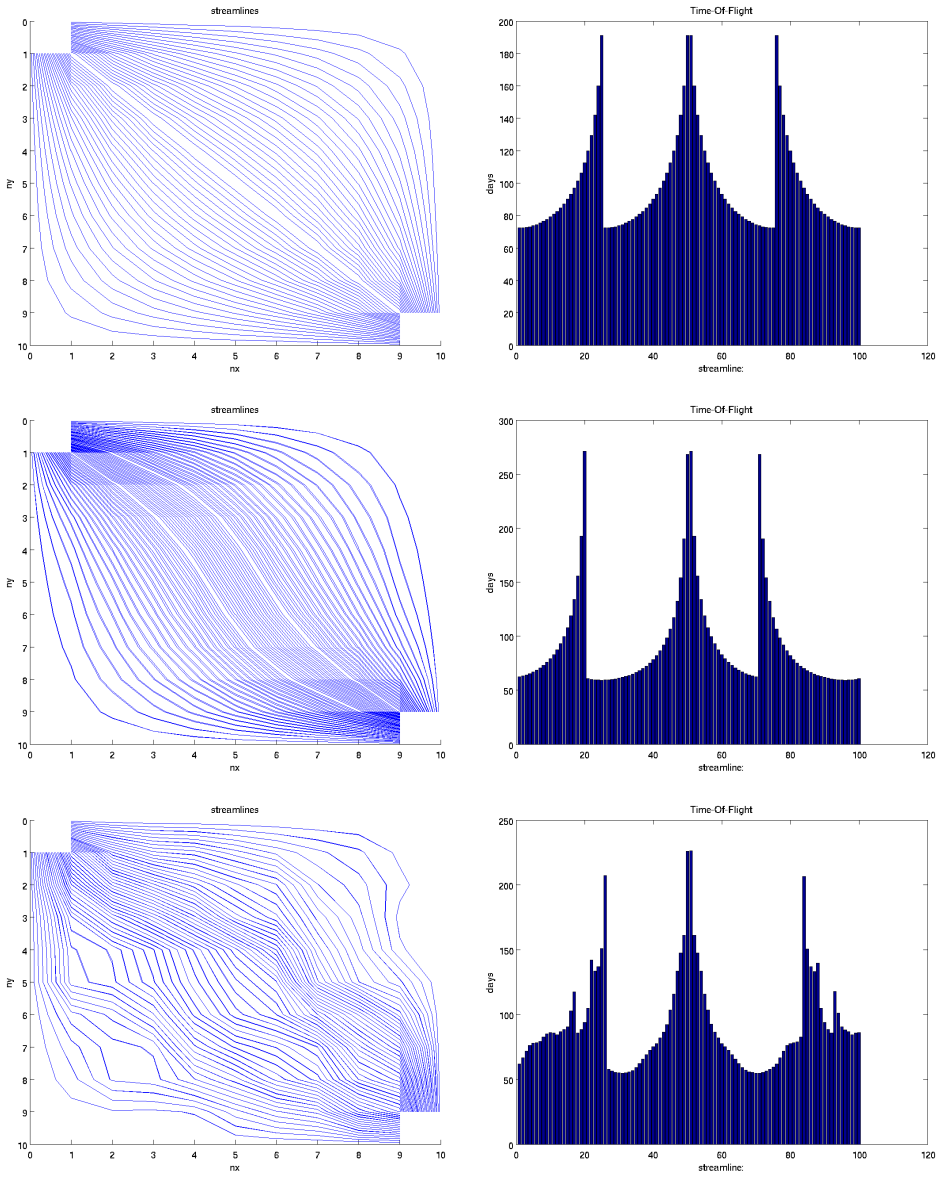

Figure 3.5: Streamlines (left) and time-of-flight (right) for Case 1 (top), 2 (middle) and 3 (bottom).

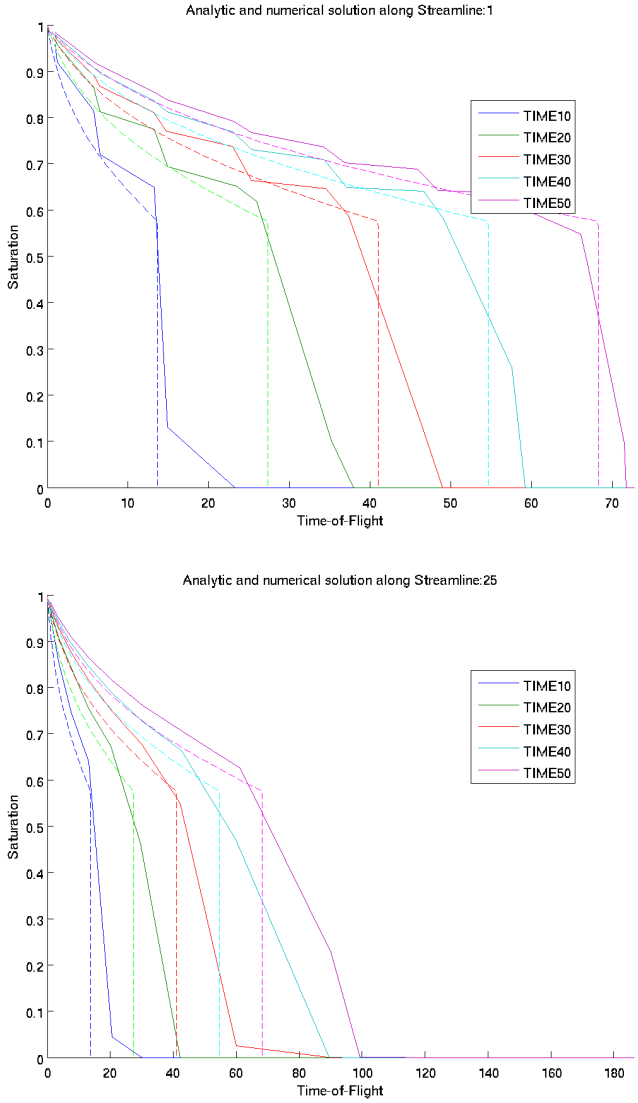

Figure 3.6: Comparing analytic (dotted) and numerical solution (solid) for a 10×10 grid for streamline 1 (top) and streamline 25 (bottom).

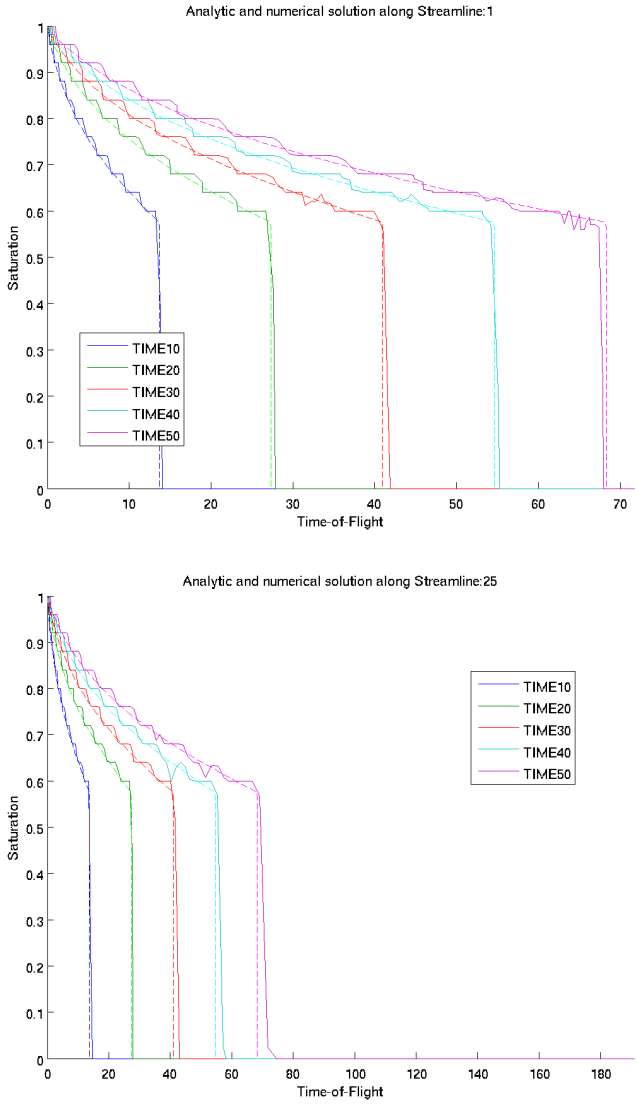

Figure 3.7: Comparing analytic (dotted) and numerical solution (solid) for a 100×100 grid for streamline 1 (top) and streamline 25 (bottom).

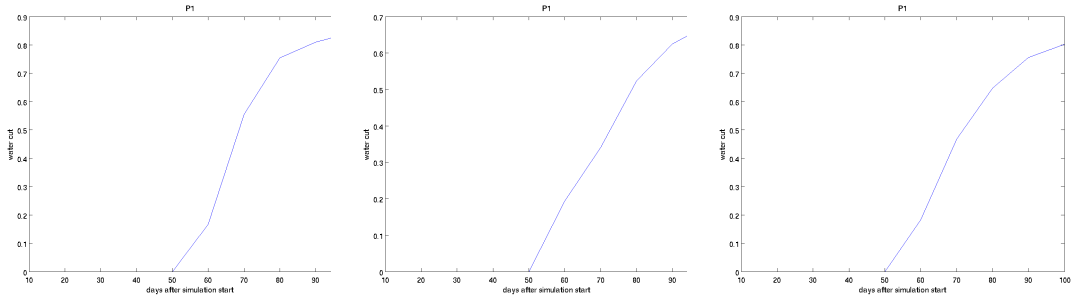

Figure 3.8: Water cut at producer for Case 1 for 10×10 (left), 100×100 (center), both with 100 streamline, and 100×100 with 200 streamlines (right).

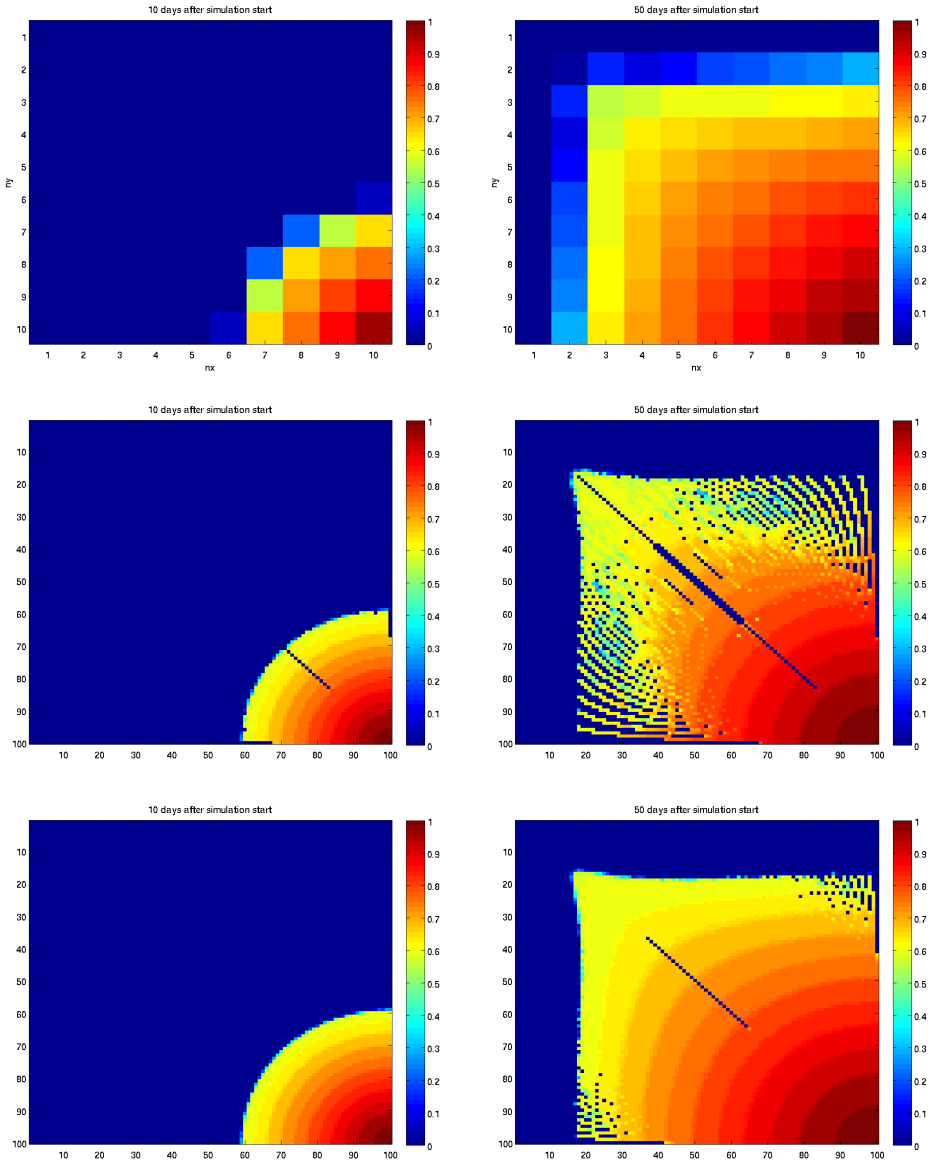

Figure 3.9: Water Saturation after 10 (left) and 50 (right) days for Case 1 with 10×10 (top), 100×100 (middle), both with 100 streamlines and 100×100, with 200 streamlines (bottom).

Chapter 4

Inverse Problems

Inverse problems are a field
in which one is called upon to reconstruct the cow
from the hamburger meat, so to speak
Andreas Mandelis

Since the days of Newton it has been widely accepted that the physical world is fully determined through physical laws. This makes it possible to predict the future. The problem of predicting the future is a forward problem. In mathematical language a forward problem is:

$$\mathbf{g}(\mathbf{m}) = \mathbf{d}, \tag{4.1}$$

where $\mathbf{m}$ is the model parameters, $\mathbf{d}$ is the collected data set, and $\mathbf{g}$ is the operator that reflects the physical laws that relates the model parameters to the data set. A model is a set of parameters that describes the physical system or as in our case, reservoir parameters as permeability and porosity. For a linear problem, equation (4.1) could be written as a matrix equation: $\mathbf{Gm} = \mathbf{d}$, where $\mathbf{G} \in \mathbb{R}^{m \times n}$, $\mathbf{m} \in \mathbb{R}^m$ and $\mathbf{d} \in \mathbb{R}^n$. The forward problem defined in (4.1) is only valid in a theoretical world. Even thought the fundamental physical laws governing the nature are adequately understood, simplification and assumptions needs to be done to actually solve the problem. This will result in errors in the prediction. In the real world, there always exists error in the measurements in addition to error in the prediction. This will introduce errors in the predicted data, hence $\mathbf{d} = \mathbf{d}_t + \eta$, where η is the data error.

For a forward model problem the target is to find $\mathbf{d}$ for a given $\mathbf{m}$. But frequently scientists and engineers meet problems where the model $\mathbf{m}$ is unknown. To determine the model they then collect data and solve the corresponding inverse problem, that is to find $\mathbf{m}$, for a given $\mathbf{d}$. The history matching problem introduced in the introduction, where reservoir engineers want to use historical production data to better characterize the reservoir, is an example of an inverse problem. There are two main approaches to formulate the theory of inverse problems, the deterministic and the stochastic. The main focus of this presentation will be on the deterministic approach. Still some references will be made to the stochastic formulation. More on the stochastic approach can be found in [13]. Solution strategies based on the deterministic approach for inverse problems is presented later in this chapter, but first some general concepts of inverse problems is introduced. While a forward problem has a unique solution if the model and physics is perfectly known, this is not the case for the inverse problem. For a given forward operator, a set of data cannot be matched by any model or, alternatively, by infinitely many models. Therefore, a careful analysis of the well-posedness of the inverse problem needs to be done.

4.0.1 Well-posedness

A problem is said to be well-posed if a stable and unique solution exists. To better understand the well-posedness of inverse problems, some linear and discrete problems will be discussed. For a

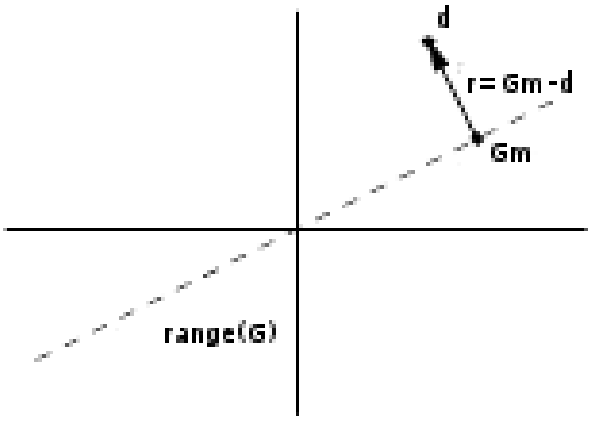

Figure 4.1: Geometrical interpretation of least squares

linear and discrete case the forward operator is a matrix. The condition and rank of this matrix will determine the well-posedness of the problem. In all cases the measurement error is assumed to be zero.

First let $m = n$, that means the dimension of the model and data space is equal and $\mathbf{G}$ is a square matrix. If $\mathbf{G}$ is of full rank, that is rank($\mathbf{G}$) $= n$, then there exists a unique solution $\mathbf{m} = \mathbf{G}^{-1}\mathbf{d}$, since the inverse of a nonsingular and square matrix always exists and is unique.

The case when $m > n$ is called over-determined since there are more data than the model can match. For a over-determined system of full rank, the solution of the matrix equation will be unique, but not exact since null($\mathbf{G}\mathbf{G}^T$) is nontrivial. Therefore, instead of finding the exact solution, a solution that minimizes an objective function is found. Without any regularization, the objective function, $f(\mathbf{m})$ is defined as:

$$f(\mathbf{m}) = \| \mathbf{r} \|^2 \tag{4.2}$$

in some appropriate norm. The residual $\mathbf{r} = \mathbf{d} - \mathbf{G}\mathbf{m}$. A model that minimizes the objective function (4.2) in the 2-norm is called a least squares solution. If r_i is the i-th element of the residual, the discrete 2-norm of the residual is defined such that:

$$\| \mathbf{r} \|_2^2 = \sum_{i=1}^{n} r_i^2.$$

The least square solution has an intuitive geometric interpretation, see Figure 4.1. The orthogonal projection of $\mathbf{d}$ onto range($\mathbf{G}$) is $\mathbf{G}\mathbf{m} - \mathbf{d}$, and therefore each column of $\mathbf{G}$ is orthogonal to $\mathbf{G}\mathbf{m} - \mathbf{d}$. Thus $\mathbf{G}^T(\mathbf{G}\mathbf{m} - \mathbf{d}) = \mathbf{0}$ or $\mathbf{G}^T\mathbf{G}\mathbf{m} = \mathbf{G}^T\mathbf{d}$. This system of equations is called the normal equations. If $\mathbf{G}$ has full rank $(\mathbf{G}^T\mathbf{G})^{-1}$ always exists [13]. The least square solution

$$\mathbf{m}_{L_2} = (\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{d}. \tag{4.3}$$

For normal distributed errors, the least squares solution is unbiased and also the maximum likelihood solution [13].

The case when $m < n$ is called under-determined. An under-determined system will have an exact solution, but it will not be unique. Since $m < n$, the null space of $\mathbf{G}$, null($\mathbf{G}$) is nontrivial. Therefore, all models $\mathbf{m}_0 \in$ null($\mathbf{G}$) is solutions of $\mathbf{G}\mathbf{m}_0 = 0$, and also any combination of these solutions is a solution, resulting in infinitely many solutions of the inverse problem. This is typically the case for the history matching problem, since the number of data points $\mathbf{d}$ often are lower than the number of parameters $\mathbf{m}$ to match. Even if it is possible to find an exact solution

to the inverse problem, this is not what is aimed for. There is error both in the model and in the data, therefore a solution of the inverse problem is the model $\mathbf{m}$ that minimizes the objective function defined in (4.2). The minimization problem in (4.2) is unstable, which means that a small change in the measurements can lead to an enormous change in the estimated model [13]. The minimization problem in (4.2) often is ill-conditioned. A linear problem is ill-conditioned when the condition number of the matrix is large. For a matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ of full rank the condition number, $\kappa(\mathbf{A})$ is given by:

$$\kappa(\mathbf{A}) = \frac{\sigma_1}{\sigma_n}, \tag{4.4}$$

where σ_1 and σ_n are the largest and smallest eigenvalue of $\mathbf{A}$, receptively [10]. To stabilize the solution additional constrains can be added to the problem, this is referred to as regularization.

4.0.2 Regularization

One way to regularize the solution, is to add constraints that punish large variations of the model, $\| \mathbf{m} \|_2 < \epsilon$ for some given ϵ, to the objective function. An alternatively is to seek a solution that minimizes the 2-norm of the model among those solutions that fits the data good enough:

$$\min_{\mathbf{m}} \| \mathbf{m} \|_2^2 \quad \text{for} \quad \| \mathbf{d} - \mathbf{Gm} \|_2^2 \leq \delta, \tag{4.5}$$

for some given parameter δ. This regularization is called a zero order Tikhonovs regularization. A damped least squares formulation of the Tikhonovs regularization can be found by applying the method of Lagrange multipliers to (4.5):

$$\min_{\mathbf{m}} \| \mathbf{d} - \mathbf{Gm} \|_2^2 + \alpha^2 \| \mathbf{m} \|_2^2, \tag{4.6}$$

where α is a regularization parameter [13]. The regularization parameter must be chosen with care since all regularization introduces bias to the solution. One way to chose the regularization parameter is by the L-curve criteria [5]. Plotting the values of $\| \mathbf{m} \|_2$ versus $\| \mathbf{d} - \mathbf{Gm} \|_2$ on a log-log scale usually gives a characteristic L shape. The idea is to pick α such that both $\| \mathbf{m} \|_2$ and $\| \mathbf{d} - \mathbf{Gm} \|_2$ is minimized, in other words find the solution that is closest to the corner in the L-curve. Another method to find the regularization parameter is called Generalized Cross Validation (GCV). For more details about both the L-curve method and the GCV method see [13]. Prior knowledge of the model and data error variance can also be used to determine the regularization parameter. Often it is desired to minimize the norm of the model derivatives. If $\mathbf{L} \in \mathbb{R}^{n\times n}$ is the finite difference approximation of the derivatives, a higher-order Tikhonovs regularization is

$$\min_{\mathbf{m}} \| \mathbf{d} - \mathbf{Gm} \|_2^2 + \alpha^2 \| \mathbf{Lm} \|_2^2 . \tag{4.7}$$

In a zero-order Tikhonovs regularization $\mathbf{L} = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix. A first order Tikhonovs regularization will penalize solutions that are rough in its first derivatives. In one dimensions, the first order derivatives is approximated by

$$\mathbf{L} = \begin{bmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & \cdots & & \\ & & -1 & 1 & \\ & & & -1 & 1 \end{bmatrix}.$$

In two or three dimensions the corresponding finite difference approximation of the derivative must be used. An alternative way to regularize the inverse problem is to penalize large variation from a prior model, $\mathbf{m_p}$. This way to regularize is closely related to Bayesian inversion. For a linear model $\mathbf{Gm} = \mathbf{d}$, with Gaussian error distribution and Gaussian prior model parameter distribution the maximum a posterior (MAP) solution can be found by

$$\min_{\mathbf{m}} \| \mathbf{d} - \mathbf{Gm} \|_{C_D^{-1}}^2 + \| \mathbf{m} - \mathbf{m_p} \|_{C_M^{-1}}^2 \tag{4.8}$$

where $\| \mathbf{d}-\mathbf{Gm} \|^2_{C_D^{-1}} = (\mathbf{Gm}-\mathbf{d})^T\mathbf{C}_D^{-1}(\mathbf{Gm}-\mathbf{d})$ and $\| \mathbf{m}-\mathbf{m_p} \|^2_{C_M^{-1}} = (\mathbf{m}-\mathbf{m_p})^T\mathbf{C}_M^{-1}(\mathbf{m}-\mathbf{m_p})$ for the model and data covariance matrix given by $\mathbf{C}_M$ and $\mathbf{C}_D$, respectively. In the special case where $\mathbf{C}_M = (1/\alpha)^2\mathbf{I}$ and $\mathbf{C}_D = \mathbf{I}$ equation (4.8) simplifies to

$$\min_{\mathbf{m}} \| \mathbf{d}-\mathbf{Gm} \|_2^2 + \alpha^2 \| \mathbf{m}-\mathbf{m_p} \|_2^2, \tag{4.9}$$

which is the zero order non-standard Tikhonovs regularization [13]. In other words, prior knowledge acts as a regularization term with increasing weight as the belief in the prior model decreases. In history matching problems, geologists and geophysicists have made a prior model $\mathbf{m_p}$, based on known geological structure, well core tests and seismic data etc. Their predicted uncertainty of the prior model and error can therefore be used to determine the regularization terms in the regularized inverse problem.

To reduce the number of degrees of freedom will also regularize the inverse problem. For a discrete case this is the same as reducing the resolution of the model space and therefore reduce the rank-deficiency of matrix $\mathbf{G}$. All regularization will introduce bias and error to the solution. Still a stable biased solution is preferred to an unstable unbiased solution. More on additional regularization techniques can be found in [13].

4.0.3 Solving an inverse problem

The least square solution of the damped least square problem (4.7) can by rewritten as an augmenting system of equations

$$\min_{\mathbf{m}} \left\| \begin{pmatrix} \mathbf{G} \\ \alpha\mathbf{L} \end{pmatrix} \mathbf{m} - \begin{pmatrix} \mathbf{d} \\ \mathbf{0} \end{pmatrix} \right\|_2^2.$$

For $\alpha > 0$, $\begin{pmatrix} \mathbf{G} \\ \alpha\mathbf{L} \end{pmatrix}$ has full rank and can be solved by the method of normal equations [13],

$$\mathbf{m} = (\mathbf{G}^T\mathbf{G} + \alpha^2\mathbf{L}^T\mathbf{L})^{-1}\mathbf{G}^T\mathbf{d}.$$

Unfortunately the normal equations are typically unstable for ill-conditioned problems involving close fits [14]. A more stable way than solving the least square problem via the Normal equations, is to solve it via QR factorization [14]. Since $\begin{pmatrix} \mathbf{G} \\ \alpha\mathbf{L} \end{pmatrix}$ is a sparse matrix an iterative method is to be preferred since it requires less flops and storage. The conjugate gradient method can be applied directly to the normal equation since $\mathbf{G}^T\mathbf{G}$ and also $\mathbf{G}^T\mathbf{G} + \alpha^2\mathbf{L}^T\mathbf{L}$ is symmetric and positive definite [14]. To further stabilize the solution the conjugate gradient method can be used via a QR factorization. The LSQR algorithm combines the conjugate gradient method with the QR factorization. Both the LSQR and the QR factorization algorithms are implemented in Matlab and is therefore not investigated further here. For details about the LSQR algorithm, convergence and speed, we refer to [12]. The QR factorization and conjugate gradient methods is fully presented by Trefethen in [14]. The LSQR algorithm will be used to solve the history matching problem.

4.1 Nonlinear inverse problems

In the previous discussion only linear problems is considered. The equations that describe fluid flow in a porous medium are nonlinear. The nonlinearity of the equations adds additional challenges to the inverse problems. The least squares solution of a nonlinear inverse problem is the minimum of the objective function given by:

$$f(\mathbf{m}) = \| \mathbf{r}(\mathbf{m}) \|_2^2, \tag{4.10}$$

where the residual, $\mathbf{r}(\mathbf{m}) = \mathbf{d} - \mathbf{g}(\mathbf{m})$ [11]. Note that here $\mathbf{g}(\mathbf{m})$ is a multi-dimensional operator, and not a matrix. As for the linear case a regularization term can be added to the objective function. Regularization of non linear inverse problems will be considered later. One approach

that solves the nonlinear inverse problems is to use a local Taylor approximation of the objective function iteratively. This leads to a local solution, a point at which the objective function is smaller than all surrounding points. For a linear problem the local solution is always the global solution. The same is true for all convex problems, but for a general nonlinear problem the local solution is not always the global solution. To find the global solution of a non-convex problem is usually difficult. The global minimum of the objective function for an non-convex problem can be found by applying local minimization sequentially for different sub-domains [11]. Global minimization problems are complex and time consuming and is not the focus of this thesis. All algorithms for solving nonlinear least squares problems presented here are local solvers.

4.1.1 Newton's method

Most of the algorithms that solve nonlinear least squares problems are iterative methods. Given a initial point $\mathbf{m}^0$ the algorithms generate a sequence of iterates $\{\mathbf{m}^k\}$ until either, no more progress can be made, or the desired accuracy is reached. The nonlinear objective function can be approximated by its Taylor's polynomial. If f is twice continuously differentiable, the second order Taylor approximation of $f(\mathbf{m}^k + \Delta\mathbf{m})$, where $\Delta\mathbf{m}$ is a small perturbation of $\mathbf{m}$, is

$$f(\mathbf{m}^k + \Delta\mathbf{m}) \approx f(\mathbf{m^k}) + \Delta\mathbf{m}^T\nabla f(\mathbf{m^k}) + \frac{1}{2}\Delta\mathbf{m}^T\nabla^2 f(\mathbf{m^k})\Delta\mathbf{m},$$

where $\nabla f(\mathbf{m^k})$ and $\nabla^2 f(\mathbf{m^k})$ is the gradient and the Hessian of $f(\mathbf{m^k})$. Similarly the gradient of $f(\mathbf{m}^k + \Delta\mathbf{m})$ can be approximated around $\mathbf{m}^k$,

$$\nabla f(\mathbf{m}^k + \Delta\mathbf{m}) \approx \nabla f(\mathbf{m}^k) + \Delta\mathbf{m}^T\nabla^2 f(\mathbf{m}^k).$$

For $\mathbf{m}^*$ to be a local minimizer of $f(\mathbf{m})$, the gradient must be equal to zero, $\nabla f(\mathbf{m}^*) = \mathbf{0}$. A local minimum of the objective function can therefore be found by solving

$$\nabla^2 f(\mathbf{m}^k)\Delta\mathbf{m} = -\nabla f(\mathbf{m}^k). \tag{4.11}$$

iteratively until convergence. The next iterate is updated such that, $\mathbf{m}^{k+1} = \mathbf{m}^k + \Delta\mathbf{m}$. This method is called Newton's method. Newton's method is a second order method since it makes use of the second order derivatives. Newton's method will converge quadratically to $\mathbf{m}^*$ if $f(\mathbf{m})$ is continuously differentiable in the neighborhood of $\mathbf{m}^*$, and there is a constant C such that $\parallel \nabla^2 f(\mathbf{m}) - \nabla^2 f(\tilde{\mathbf{m}}) \parallel_2 \leq C \parallel \mathbf{m} - \tilde{\mathbf{m}} \parallel_2$ for every $\tilde{\mathbf{m}}$ in the neighborhood, and $\nabla^2 f(\mathbf{m})$ is positive definite, and $\mathbf{m}^0$ is close enough to $\mathbf{m}^*$ [13]. A real symmetric matrix $\mathbf{A}$ is positive definite if $\mathbf{z^T A z} > 0$ for all real nonzero vectors $\mathbf{z}$ [13].

For least squares problems the Jacobian and Hessian of $f(\mathbf{m^k})$ can be expressed in terms of the Jacobian and Hessian of $\mathbf{g}(\mathbf{m})$:

$$\nabla f(\mathbf{m}) = \mathbf{J}(\mathbf{m})^T\mathbf{r}(\mathbf{m}), \tag{4.12}$$

$$\nabla^2 f(\mathbf{m}) = \mathbf{J}(\mathbf{m})^T\mathbf{J}(\mathbf{m}) + \sum_{j=1}^{m} r_j(\mathbf{m})\nabla^2 r_j(\mathbf{m}), \tag{4.13}$$

where r_j is the jth component of $\mathbf{r}$ [13]. The Jacobian is given by:

$$\mathbf{J}(\mathbf{m}) = \begin{bmatrix} \frac{\partial g_1(\mathbf{m})}{\partial m_1} & \cdots & \frac{\partial g_1(\mathbf{m})}{\partial m_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial g_m(\mathbf{m})}{\partial m_1} & \cdots & \frac{\partial g_m(\mathbf{m})}{\partial m_n} \end{bmatrix}. \tag{4.14}$$

In Newton's method, the Hessian of $f(\mathbf{m})$ must be calculated in each iteration. To find explicit formulas for the Hessian is often not possible. Still, the Hessian can be found by using finite difference approximation to the derivatives, but this is often computationally expensive. In the mid 1950s a new class of nonlinear optimization methods were developed using an approximation to the Hessian instead of the Hessian itself. These methods are called Quasi-Newton methods. One such Quasi-Newton method is called the Gauss-Newton (GN) method.

4.1.2 The Gauss-Newton method

In the Gauss-Newton method the last part of the Hessian in equation (4.13) is omitted such that $\nabla^2 f(\mathbf{m}) \approx \mathbf{J}(\mathbf{m})^T\mathbf{J}(\mathbf{m})$. With this approximation of the Hessian and the gradient given in (4.13), equation (4.11) becomes:

$$\mathbf{J}(\mathbf{m}^k)^T\mathbf{J}(\mathbf{m}^k)\Delta\mathbf{m} = -\mathbf{J}(\mathbf{m}^k)^T\mathbf{r}(\mathbf{m}^k). \tag{4.15}$$

The Gauss-Newton method solves (4.10) by sequentially solving (4.15) until $\mathbf{r}(\mathbf{m}^{k+1}) < \delta$, where δ is some convergence criterion. Note that $\mathbf{J}(\mathbf{m})^T\mathbf{J}(\mathbf{m})$ always is positive definite, so for a almost linear problem where the initial model is close to the true model the Gauss-Newton algorithm converges rapidly. There is however a main drawback with the Gauss-Newton method. If $\mathbf{J}(\mathbf{m})$ is rank-deficient, the method is ill-posed.

As for the linear case given in (4.7), a regularization term can be added to the Gauss-Newton method to make it well-posed. A damped least squares formulation of a Tikhonovs regularized nonlinear objective function is:

$$f(\mathbf{m}) = \| \mathbf{d} - \mathbf{g}(\mathbf{m}) \|_2^2 + \alpha^2 \| \mathbf{Lm} \|_2^2 . \tag{4.16}$$

The Jacobian of the regularized objective function is:

$$\mathbf{K}(\mathbf{m}) = \begin{bmatrix} \mathbf{J}(\mathbf{m}) \\ \alpha\mathbf{L} \end{bmatrix}.$$

This gives the following regularized Gauss-Newton algorithm for solving (4.16):

$$(\mathbf{J}(\mathbf{m}^k)^T\mathbf{J}(\mathbf{m}^k) + \alpha^2\mathbf{L}^T\mathbf{L})\Delta\mathbf{m} = \mathbf{J}(\mathbf{m}^k)^T\mathbf{r}(\mathbf{m}^k) - \alpha^2\mathbf{L}^T\mathbf{L}\mathbf{m}^k. \tag{4.17}$$

One of the advantages of the Gauss-Newton method is that the evaluation of the approimation of the Hessian is almost free once the Jacobian is obtained. Another advantage is that the algorithm in (4.15) is similar to the normal equations (4.3). The Gauss-Newton method can therefore be reformulated as a linear problem with the Jacobian as the forward operator:

$$\min_{\Delta\mathbf{m}} \| \mathbf{J}(\mathbf{m}^k)\Delta\mathbf{m} - \mathbf{r}(\mathbf{m}^k) \|_2^2 .$$

As for the linear case, regularization can be added to the objective function

$$f(\boldsymbol{\Delta}\mathbf{m}) = \| \mathbf{J}(\mathbf{m}^k)\Delta\mathbf{m} - \mathbf{r}(\mathbf{m})^{\mathbf{k}} \|_2^2 + \alpha^2 \| \mathbf{L}\Delta\mathbf{m} \|_2^2, \tag{4.18}$$

where α is the regularization parameter. An advantage of this formulation is that the regularization parameter can be found using tools available for linear problems as the L-curve method and the general cross validation method. The minimum of the objective function in (4.18) is then the least squares solution to the following augmented linear system:

$$\begin{pmatrix} \mathbf{J}(\mathbf{m^k}) \\ \alpha\mathbf{L} \end{pmatrix} \Delta\mathbf{m} = \begin{pmatrix} \mathbf{r} \\ \mathbf{0} \end{pmatrix}. \tag{4.19}$$

A solution to a nonlinear inverse problem can be found by solving the linear system in (4.19) iteratively with LSQR. A pseudo algorithm for an iterative LSQR method is:

1. Let $\mathbf{m^0} = \mathbf{m_p}$, where $\mathbf{m_p}$ is the prior model.
2. Solve (4.19) with LSQR.
3. Update the model $\mathbf{m^{k+1}} = \mathbf{m}^k + \Delta\mathbf{m}$.
4. Repeat step 2-3 as long as $f(m^{k+1}) > f(m^k)$ or some other convergence criterion is reached.

For problems where the eigenvalues of $\mathbf{J^T J} \gg \sum_{j=1}^{m} r_j(\mathbf{m})\nabla^2 r_j(\mathbf{m})$, the Gauss-Newton method and the iterative LSQR method works as good as the Newton's method. This is the case when the residual, $\mathbf{r}(\mathbf{m})$ is small or $g(\mathbf{m})$ is almost linear [11]. For large residual problems the second part of the Hessian is too important to ignore, therefore the Hessian or a better approximation of the Hessian should be used if possible. There are more sophisticated methods to approximate the Hessian than what the Gauss-Newton method uses. For a more complete introduction to optimization and algorithms for solving optimization problems, see [11]. These methods will not be considered in this thesis since the Hessian can be approximated practically at no cost using the streamline formulation. To better the convergence properties, the Gauss-Newton method and the Newton method can be extended by including line search, trust region and scaling strategies.

4.1.3 Line search, trust region and scaling strategies

In a line search strategy, the next iterative $\mathbf{m}^{k+1} = \mathbf{m}^k + \gamma^k \Delta\mathbf{m}$, where $\Delta\mathbf{m}$ is the direction and γ^k is an optimal distance. The optimal distance is the distance such that $f(\mathbf{m}^{k+1})$ is minimized. The unit step, $\gamma^k = 1$ is a natural step length to the Gauss-Newton and Newton's method. Still the distance may be adjusted when it does not reduce the objective function satisfactory. A satisfactory reduction is guaranteed by the Wolfe conditions:

$$\begin{aligned} f(\mathbf{m}^k + \gamma^k \Delta\mathbf{m}) &\leq f(\mathbf{m}^k) + c_1 \gamma^k \nabla f(\mathbf{m}^k)^T \Delta\mathbf{m} \\ \nabla f(\mathbf{m}^k + \gamma^k \Delta\mathbf{m})^T \Delta\mathbf{m} &\geq c_2 \nabla f(\mathbf{m}^k)^T \Delta\mathbf{m}, \end{aligned} \tag{4.20}$$

for constants $0 < c_1 < c_2 < 1$. The first condition guaranties the upper bound and the second the lower bound of the step length. There are different strategies for adjusting the step length. Since each evaluation of the objective function requires at least one evaluation of the forward model, a good enough distance rather than the optimal distance is searched for. Different strategies, based on quadratic or cubic interpolation of known function and derivative values of the objective function, that adjusts the step length can be found in [11]. If the calculated step-size do not satisfy the Wolfe conditions in its first try, the step size is further restricted until the Wolfe conditions is satisfied or a maximum number of guess is reached. In a line search strategy the direction of the update is calculated and then the optimal distance that minimizes the objective function is found. An alternative strategy is to fix the distance first, and then calculate the optimal direction to move. This alternative strategy is called a trust region strategy.

A trust region strategy is based on constructing a trust region where the solution is searched within. The trust region sub-problem can be formulated as:

$$\min_{\mathbf{m}} f(\mathbf{m}) \qquad \text{such that } ||\mathbf{m}|| \leq \Delta_k \tag{4.21}$$

The trust region radius Δ_k is chosen based on the performance of the previous iterations. Different algorithms that computes the trust region radius is given in [11].

A Gauss-Newton method with a trust-region strategy is called a Levenberg-Marquardt method (LM). In the Levenberg-Marquardt method a positive term $\lambda\mathbf{I^T I}$ is added to (4.15):

$$(\mathbf{J}(\mathbf{m}^k)^T \mathbf{J}(\mathbf{m}^k) + \lambda^k \mathbf{I}^T \mathbf{I})\Delta\mathbf{m} = -\mathbf{J}(\mathbf{m}^k)^T \mathbf{r}(\mathbf{m}^k). \tag{4.22}$$

The λ parameter is closely related to the trust region radius since it always exists a $\lambda > 0$ such that equation (4.21) and equation (4.22) is satisfied [11]. The $\lambda\mathbf{I}^T\mathbf{I}$ term is not a Tikhonovs regularization even thought it looks like it, since $\lambda\mathbf{I}^T\mathbf{I}$ is added in the algorithm and not directly to the objective function. Note that while the regularization parameter α is fixed during all iteration, λ is adjusted during the iterations. If there is good agreement between the change in the model and the change in the objective function, it is safe to expand the trust region so λ is reduced. Similarly λ is increased if the agreement is bad [11]. A large λ will bias the search direction towards the steepest descent direction [11]:

$$\Delta\mathbf{m} = -\mathbf{J}(\mathbf{m}^k)^T \mathbf{r}(\mathbf{m}^k) \tag{4.23}$$

The steepest descent direction will always converge if small enough steps are taken, but for difficult problems the convergence can be extremely slow [11].

A trust region strategy can also be used together with Newton's method. An advantage of Newton's method with trust region is that the Hessian matrix do not need to be positive definite. If a positive term $\lambda \mathbf{I}^T \mathbf{I}$ is added such that $\mathbf{J}(\mathbf{m})^T \mathbf{J}(\mathbf{m}) + \sum_{j=1}^{m} r_j(\mathbf{m}) \nabla^2 r_j(\mathbf{m}) + \lambda \mathbf{I}^T \mathbf{I}$ is positive definite, Newton's method with trust region will converge. Different techniques for choosing λ can be found in [11].

Another important extension to the least squares algorithms is scaling. The range of the variables in least squares problems can be huge, if such variation is ignored, the algorithms may give solutions of poor quality. To better the convergence the problems are scaled. One way to scale the problem is to use a ellipsoidal trust region instead of a spherical trust region. The trust region sub-problem then becomes:

$$\min_{\mathbf{m}} f(\mathbf{m}) \qquad \text{such that} \quad ||\mathbf{D}^k \mathbf{m}|| \leq \Delta_k,$$

where $\mathbf{D^k}$ is a diagonal matrix where the diagonal elements are the principal axes of the trust region. The corresponding Levenberg-Marquardt method is

$$(\mathbf{J}(\mathbf{m}^k)^T \mathbf{J}(\mathbf{m}^k) + \lambda^k (\mathbf{D}^k)^T \mathbf{D}^k) \Delta \mathbf{m} = -\mathbf{J}(\mathbf{m}^k)^T \mathbf{r}(\mathbf{m}^k). \tag{4.24}$$

One way to chose the diagonal elements of $\mathbf{D}^k$ is to chose them such that it match the diagonal of the Hessian [11]. Scaling have significant effect on the rate of convergence. Some scaling techniques for History Matching problems are compared in [18].

4.2 Summary and comments

In this chapter the basic topics related to solving both linear and nonlinear inverse problems is presented. For more details see [13] and [11]. The theory in this chapter is valid for general inverse problems and is not related to the streamline formulation presented in the previous chapter. Still the streamline formulation gives huge benefits, also in solving inverse problems. Computation of the Jacobian plays an important role in all the methods presented for solving nonlinear inverse problems and is often the most time consuming part of the method. In the next chapter different techniques of computing the Jacobian and Hessian is presented. Using the streamline formulation and time-of-flight, the Jacobian and the Hessian matrix for reservoir responses related to time-of-flight can be expressed analytically. This a huge advantage of the streamline model compared to more standard models of fluid flow in porous medium.

Chapter 5

Streamline based calculation of sensitivities

Computers are incredibly fast, accurate, and stupid.
Human beings are incredibly slow, inaccurate, and brilliant.
Together they are powerful beyond imagination.
Albert Einstein

Derivatives of reservoir responses with respect to reservoir parameters are called sensitivities. The first order sensitivity is the Jacobian required in the solution algorithms for solving nonlinear inverse problems described in the previous chapter. The Hessian is similarly related to the second order sensitivity. In this chapter a streamline based method for calculating the sensitivities is introduced. A benefit of the streamline based approach is that sensitivities can be obtained semi-analytically along the streamlines. This makes streamline based methods optimal regarding speed for computation of sensitivities for high-resolution geological models. An important assumption in all streamline based computation of sensitivities is that streamlines are assumed to not shift during perturbation. To understand how this assumption affects the calculation is therefore an important question that must and is addressed in this chapter.

5.1 A finite difference approximation of the sensitivities

Derivatives measure the sensitivity of a function to a infinitesimal change in the values of the variables. The idea of the finite difference method is to make small but finite perturbations of the variables and examine the resulting difference in the function value. If g_k is the k th element of a vector function $\mathbf{g} : \mathbb{R}^n \rightarrow \mathbb{R}^m$, and m_i the ith component of vector $\mathbf{m}$, the elements of the Jacobian can be approximated by the forward-difference formula

$$\frac{\partial g_k(\mathbf{m})}{\partial m_i} \approx \frac{g_k(\mathbf{m} + \epsilon e_i) - g_k(\mathbf{m})}{\epsilon}, \tag{5.1}$$

where ϵ is a small positive scalar and e_i is the ith unit vector. If $\epsilon \rightarrow 0$ the forward difference formula is the definition of the Jacobian. This formula requires $(n + 1) * m$ evaluations of the function $\mathbf{g}$. In Newton's method the Hessian matrix is needed in each of the iteration. The Hessian matrix can be found be taking the finite difference formula on the gradient of the function, this process requires a total of $(n+1)*m$ evaluations of the gradient of the function. Taken into account that there can be millions of parameters in a high-resolution model and that the iterative methods for solving nonlinear inverse problems presented in the previous chapter needs several iterations to converge, a faster method for computing the Jacobian will have significant effect on the total solution time. More sophisticated methods must therefore be used to find the Jacobian and the Hessian. The Jacobian matrix is often sparse, a method that make use of the sparsity of the

Jacobian can reduce the number of function evaluation considerable, still the number of function evaluation can be too large for high-resolution models. There are other methods for computing the sensitivities such as the adjoint method that requires less function evaluation [15]. Still, by using the streamline formulation from Chapter 3, it is possible to find semi-analytic expressions for sensitivities of production responses that are related to the time-of-flight. These semi-analytic expressions will only require one function evaluation.

5.2 Semi-analytic expressions for the sensitivities

5.2.1 The time-of-flight sensitivity

As time-of-flight was the key idea behind the streamline based simulation, it also plays a major role in the streamline based computation of the sensitivities. Having the time-of-flight sensitivity, sensitivities that can be related to the time-of-flight sensitivity can be found. Recalling the definition of time-of-flight in (3.8)

$$\tau = \int_{\Psi} s(\mathbf{x}) dr,$$

where s is the slowness function. The slowness function is

$$s(\mathbf{x}) = \frac{\phi(\mathbf{x})}{\mid \mathbf{u}(\mathbf{x}) \mid}, \tag{5.2}$$

where $\mathbf{u}$ is the Darcy's velocity given by Darcy's law (2.28). For constant reservoir parameters m_i in grid block i, the sensitivity of the time-of-flight along a streamline following Γ_i is

$$\frac{\partial \tau}{\partial m_i} = \int_{\Gamma_i} \frac{\partial s(\mathbf{x})}{\partial m_i} dr. \tag{5.3}$$

The first order sensitivity of the slowness function with respect to permeability is:

$$\frac{\partial s}{\partial K} \approx \frac{\partial}{\partial K}\left(\frac{\phi}{K\lambda_t \mid \nabla p \mid}\right) = \frac{-\phi}{K^2\lambda_t \mid \nabla p \mid} = -\frac{s}{K}, \tag{5.4}$$

and the second order sensitivity is:

$$\frac{\partial^2 s}{\partial K^2} \approx \frac{\partial}{\partial K}\left(\frac{-\phi}{K^2\lambda_t \mid \nabla p \mid}\right) = \frac{2\phi}{K^3\lambda_t \mid \nabla p \mid} = \frac{2s}{K^2}. \tag{5.5}$$

These expressions for the sensitivities are only approximation, since a perturbation in the permeability will generate change in pressure and hence a corresponding shift in the streamlines. Following Datta-Gupta in [2], the consequence of these approximations for the total variation of the velocity due to a small perturbation $\delta K_i = \epsilon K_i$ can be divided into two parts:

$$\delta \mathbf{u_i} = \epsilon \mathbf{u_i} - \lambda_t K_i \nabla \delta p_i, \tag{5.6}$$

where δp_i is the corresponding shift in the pressure in cell i. The first part of (5.6) corresponds to changes in the velocity along the streamlines. The second part corresponds to the effect of shifting streamlines. For an incompressible flow

$$\nabla \delta \mathbf{u_i} = \nabla(\epsilon \mathbf{u_i} - \lambda_t K_i \nabla \delta p_i) = 0.$$

Therefore δp_i satisfies the Laplace equation:

$$\nabla \cdot (\lambda_t K_i \nabla \delta p_i) = \nabla \cdot (\epsilon \mathbf{u_i}) = \phi \frac{\partial \epsilon}{\partial \tau}, \tag{5.7}$$

where the last equality in (5.7) is due to equation (3.11). The change in the velocity that is not along the streamlines will therefore decay by a factor $1/R$ for two dimensions and $1/R^2$ for three

dimensions, where R is the distance from the shifted permeability [2]. This effect can be seen in the examples in Section 5.3. . Similar expressions for the sensitivities can be found with respect to parameters such as porosity, relative permeability, mobility, well rates etc. [15]. If the definition of the time-of-flight sensitivity in equation (5.3) is combined with the expression for the sensitivity of the slowness function with respect to permeability in (5.4), an expression for the first order time-of-flight sensitivity with respect to permeability is found. The time-of-flight sensitivity with respect to the permeability K_i in cell i will then be:

$$\frac{\partial \tau}{\partial K_i} = \int_{\Gamma_i} -\frac{s}{K_i} dr = -\frac{\Delta \tau_i}{K_i}, \tag{5.8}$$

where $\Delta \tau_i$ is the increase of time-of-flight over cell i. An expression for the second order sensitivity if found if (5.5) is used instead of (5.4):

$$\frac{\partial^2 \tau}{\partial K_i^2} = \int_{\Gamma_i} \frac{2s}{K_i^2} dr = \frac{2\Delta \tau_i}{K_i^2}. \tag{5.9}$$

By relating the sensitivities of the travel-time and production data to the time-of-flight sensitivities, semi-analytical expression for the sensitivities can be found for single- and two-phase flow.

5.2.2 Sensitivities for single-phase flow

Time-of-flight measures the time it takes a passive tracer to propagate from one point in the reservoir to another. For a general quantity, the same time is referred to as travel-time. The travel-time sensitivities for a passive tracer will therefore be equal to the time-of-flight sensitivities. Assuming that the streamlines do not shift, the sensitivities of the tracer concentration can be found by differentiating equation (3.18) using the chain rule:

$$\frac{\partial C}{\partial m_i} = \frac{\partial C}{\partial \tau} \frac{\partial \tau}{\partial m_i} = -\frac{\partial C_0}{\partial \tau} \frac{\partial \tau}{\partial m_i}. \tag{5.10}$$

The same expression can be found without assuming that streamlines do not shift, by using a first order approximation of the sensitivities [2]. As with the time-of-flight sensitivities, a less smooth sensitivity is expected using the above expression. Expression for the sensitivities with respect to permeability can then be found by using equation (5.8).

5.2.3 Sensitivities for two-phase flow

For a Buckley-Leverett displacement satisfying equation (3.19), where the water injection rate is constant, the travel-time of the water front is related to the time-of-flight by $t = \tau/\sigma$, where σ is the shock speed (3.21). Assuming that the streamlines do not shift, the travel-time sensitivities are:

$$\frac{\partial t}{\partial m_i} = \frac{1}{\sigma} \frac{\partial \tau}{\partial m_i}. \tag{5.11}$$

Since the sensitivities are calculated along one dimensional streamlines that do not shift, the expressions for the second order sensitivities will be similar to the first order sensitivities. The second order sensitivity of the travel-time is:

$$\frac{\partial^2 t}{\partial m_i^2} = \frac{1}{\sigma} \frac{\partial^2 \tau}{\partial m_i^2}. \tag{5.12}$$

The second order sensitivity with respect to permeability can then be found by using equation (5.9). Expressions for the sensitivities for the fractional flow can be found by assuming that the streamlines do not shift and then using the chain rule

$$\frac{\partial F(S)}{\partial m_i} = \frac{\partial F(S)}{\partial \tau} \frac{\partial \tau}{\partial m_i},$$

where $\frac{\partial F(S)}{\partial \tau}$ can be found analytically using (3.2.2) or numerically using a finite difference approximation to the fractional flow at at given time.

5.2.4 Average sensitivities at the wells

All the semi-analytic expressions for the sensitivity derived so fare are along individual streamlines. Since streamlines are assumed to not shift during perturbation each of the individual sensitivity of a streamline is independent of all the other streamlines. To map the sensitivities from streamlines to the pressure grid a flux weighed average is used. In history matching problems, data is usually obtained in the production wells. The sensitivities of the production response at the well is therefore of special interest. The travel-time sensitivities at well k is given as the flux weighed average of all the travel-time sensitivities given in (5.11) corresponding to streamlines $l = 1, ..., N_{sl}$, that is connected to the well:

$$\frac{\partial t_k}{\partial m_i} = \frac{1}{\mathbf{q}} \sum_{l=1}^{N_{sl}} q_l \frac{\partial t_{k,l}}{\partial m_i}, \tag{5.13}$$

where N_{sl} is the total number of streamlines connected to the well, and $\mathbf{q}$ is the total flux in the well. The second order sensitivity of the average travel-time is given by:

$$\frac{\partial^2 t_k}{\partial m_i \partial m_\imath} = \begin{cases} \frac{\partial^2 t_k}{\partial m_i^2} & \imath = i \\ 0 & \imath \neq i, \end{cases} \tag{5.14}$$

where the diagonal term is given by the flux weighed average of the travel-time sensitivity given in (5.12):

$$\frac{\partial t_k}{\partial m_i^2} = \frac{1}{\mathbf{q}} \sum_{l=1}^{N_{sl}} q_l \frac{\partial^2 t_{k,l}}{\partial m_i^2}.$$

The non-diagonal element of the second order sensitivity is assumed to be zero since the streamlines is assumed to not shift during perturbation.

5.3 Semi-analytic vs finite difference approximation of the Jacobian and Hessian

To compare the semi-analytic streamline based approximation with the finite difference approximation, two different approximations of the sensitivities with respect to permeability values at all grid-cells are compared. First the sensitivity of the time-of-flight at the well with respect to permeability is compared to investigate the effect of the assumptions that the streamlines do not shift when the permeability is perturbed. To investigate the effects of the summations that maps from streamlines to grid cells, the sensitivity of the average travel-time at the well with respect to permeability is also compared. Both the sensitivities are with respect to permeability. In both cases the first and the second order sensitivities will be compared. The reservoir has $11X11$ grid cells each of size 10. The permeability are equal to 100 in all cells. In the lower right corner a well injects water at rate 5, and in the upper left corner a well produces at the same rate. A total of 20 simulation steps is made, each with time step equal to 20. The permeability will be shifted 0.01 in each grid cell, that is $\epsilon = 0.01$ in (5.1). For the finite difference approximation to be a good approximation of the sensitivity ϵ must be close to zero. The streamline simulator do not sense changes less than 0.01, this makes the finite difference approximation of the sensitivity less accurate.

The finite difference method does not assume that the streamlines do not shift when the permeability is shifted. From the analysis in (5.2.1), the sensitivities obtained with finite difference are expected to show diffusive effects compared to the streamline based semi-analytic sensitivities. Figure 5.3 shows the sensitivity of the time-of-flight at the producer with respect to permeability for streamline 1 at time 10 for both the semi-analytic and finite difference approximation. The finite difference approximation is smoothed out with a factor $1/R$ compared to the semi-analytic approximation as expected. Figure 5.3 shows the same effect for streamline 25. Since the streamlines are assumed to not shift when the permeability is changed, the time-of-flight at producer for

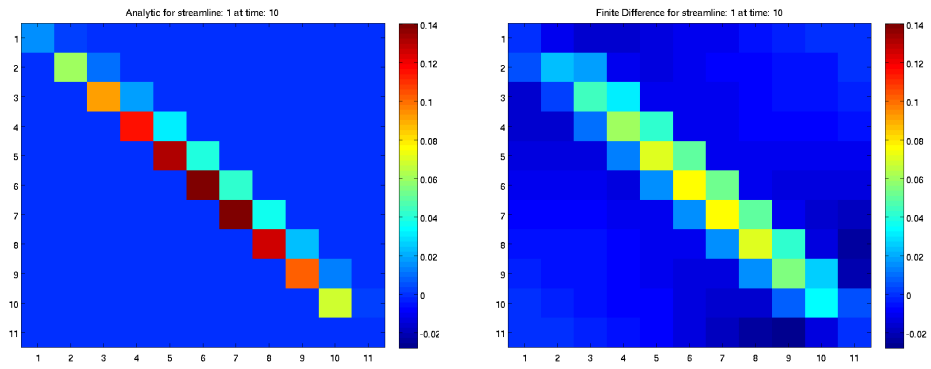

Figure 5.1: Semi-analytic (left) and finite difference (right) approximations of the time-of-flight sensitivities with respect to permeability for streamline 1 at time 10.

streamline 1 is only sensitive to changes in the cells it goes through. In the finite difference approximation the streamlines do shift and therefore the time-of-flight at the producer for the streamlines are sensitive to changes in all cells. The streamline based semi-analytic sensitivities will therefore be less smoothed out than the sensitivities obtained from the finite difference method. The same effect is shown in Figure 5.3. This plots show the sensitivities for the $5th$ column and $5th$ row of the grid. Note also that all the semi-analytic sensitivities are positive. Since the semi-analytic approximation of the sensitivity is computed independently along each streamline, a positive change in the permeability in one grid cell will lead to a positive change in the time-of-flight at the well for all streamlines that goes through this grid cell, and no change in all the streamlines that do not go through that grid cell. In a finite difference approximation the sensitivity are not computed independently along each streamline. A positive change in the permeability may therefore lead to a negative change in the time-of-flight for a streamline that do not go through the grid cell that is changed. In Figure 5.3 the sensitivity of the average arrival time at time 1, 10 and 100 are plotted. The streamline based sensitivity of the average arrival-time is of a greater magnitude than the sensitivity computed from finite differences. This is expected since only positive sensitivities are summed in the streamline based method. Figure 5.3 shows the diagonal elements of the second order sensitivities of the average arrival-time. The finite difference approximation of the second order sensitivities are based on a perturbation in the semi-analytic streamline based first order sensitivities. The non-diagonal elements of column 49, that corresponds to a finite difference approximation of (5.14) with $\imath = 49$, is plotted together with its streamline based semi-analytic approximation in Figure 5.3. The finite difference approximation of the second order sensitivity shows the same tendency as the first order sensitivities. The magnitude of the sensitivity decays as it moves from the shifted grid cell. The same results can be seen for all the mixed finite sensitivities. For small perturbations of the permeability, all mixed partial sensitivities can be assumed to be zero.

5.4 Summary and comments

In this chapter different semi-analytic expressions for computation of sensitivities have been found by relating them to the time-of-flight sensitivities. Underlying all the expressions for the sensitivities is the major assumption that streamlines do not shift when the permeability is shifted. Both theoretical and computational studies indicate that this assumption is valid for small shifts in the permeability. So far the sensitivity of travel-time and production data for single- and two-phase flow has been accounted for. In similarly ways semi-analytical expressions for water cut and oil/gas ratio can be found for three-phase flow [2].

Pressure data has not been accounted for in this presentation. A huge benefit of pressure

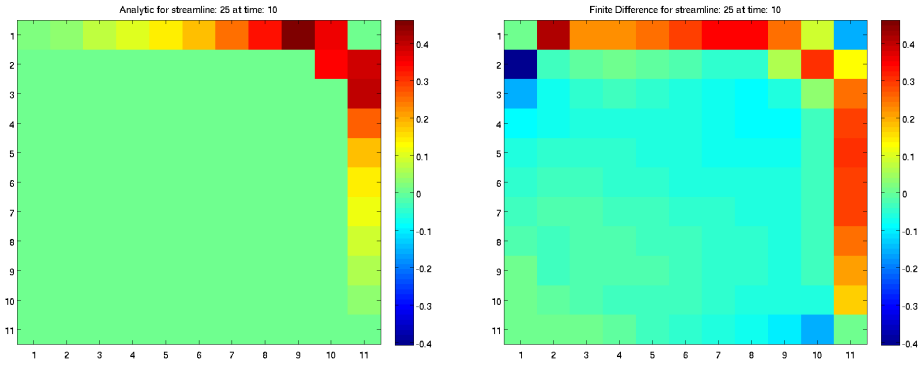

Figure 5.2: Semi-analytic (left) and finite difference (right) approximations of the time-of-flight sensitivities with respect to permeability for streamline 25 at time 10.

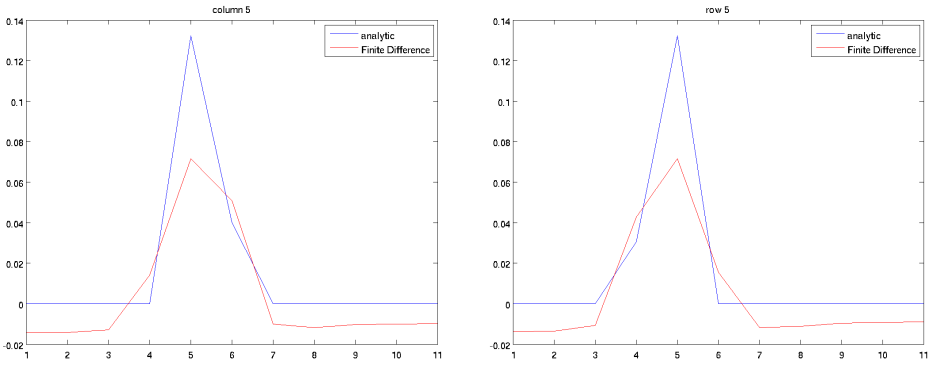

Figure 5.3: Column 5 (left) and row 5 (right) of the time-of-flight sensitivity with respect to permeability for streamline 1 at time 10.

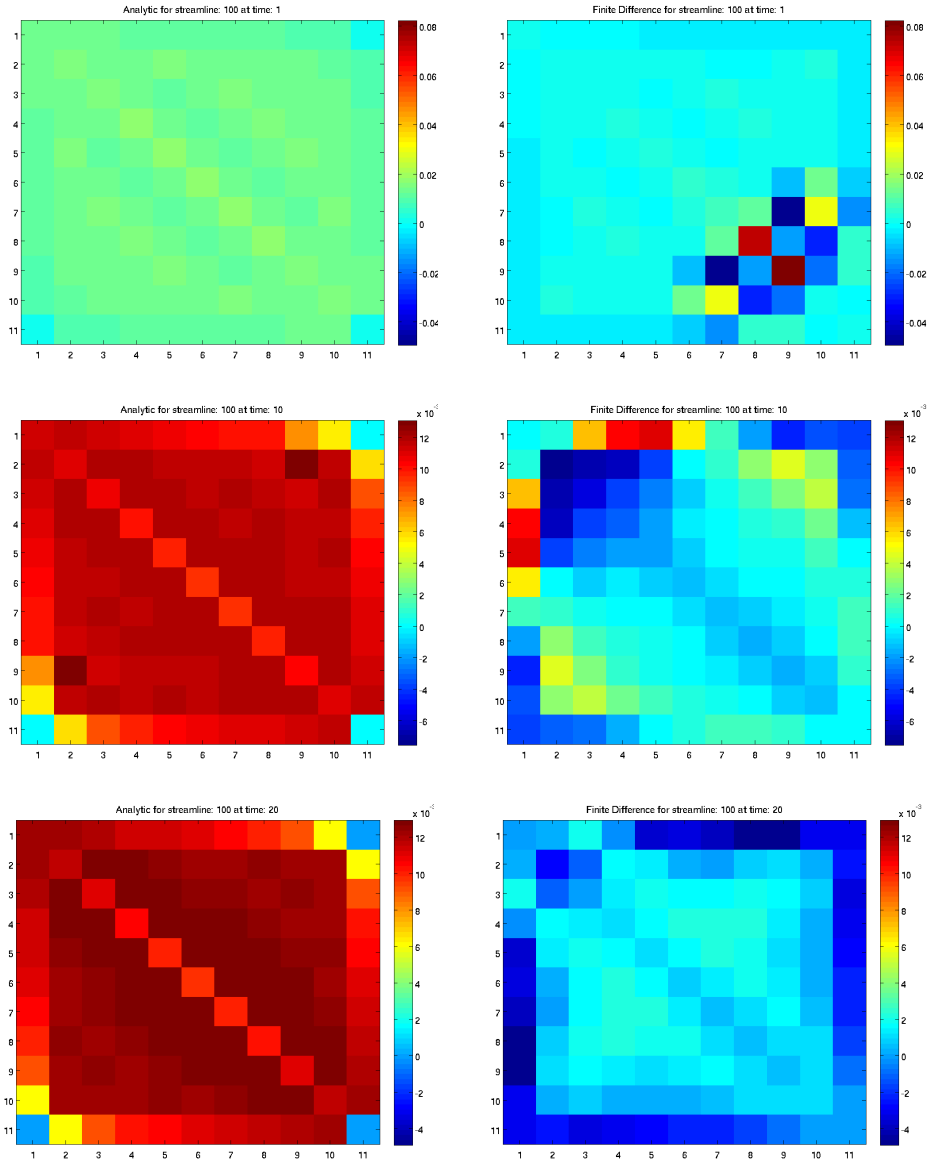

Figure 5.4: Semi-analytic (left) and finite difference (right) approximations of the average arrival-time sensitivities with respect to permeability at time 1 (top), 10 (middle) and 20 (bottom).

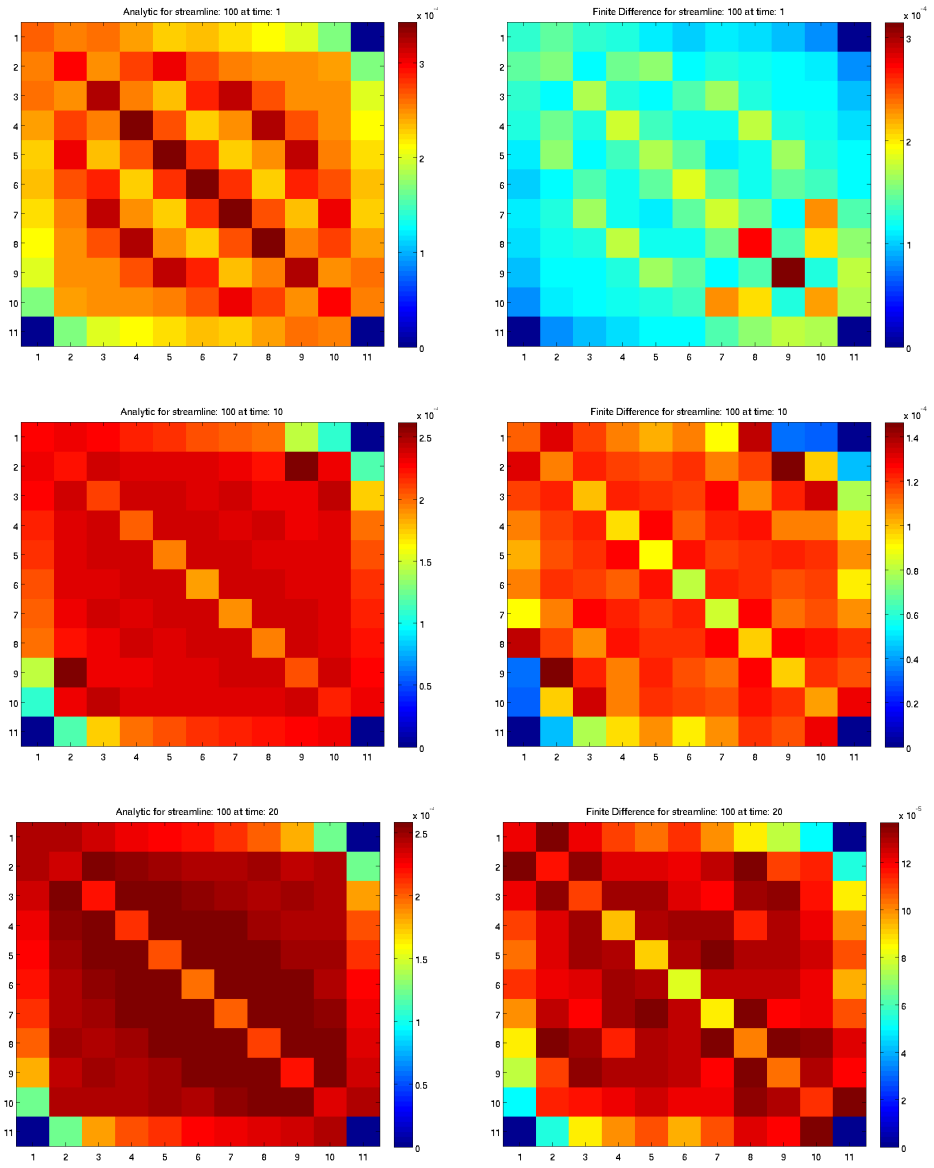

Figure 5.5: Semi-analytic (left) and finite difference (right) approximations of the second order average arrival-time sensitivities with respect to permeability at time 1 (top), 10 (middle) and 20 (bottom).

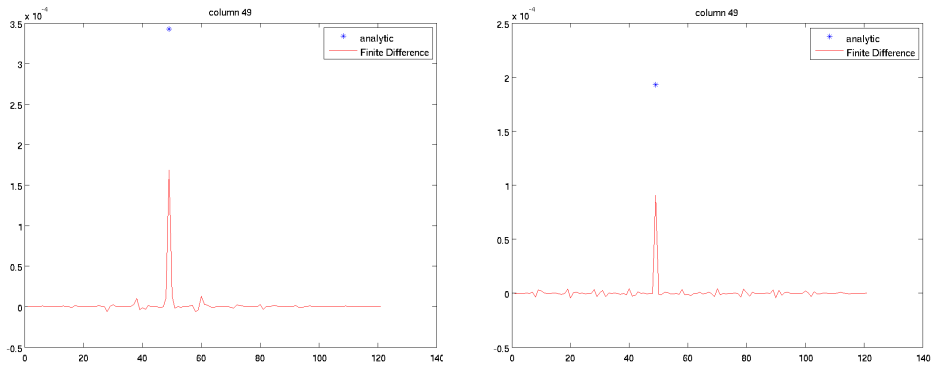

Figure 5.6: Column 49 of the second order sensitivity of the average arrival-time with respect to permeability at time 1 (top) and 10 (bottom).

data is that the pressure front propagates with an infinite speed. While the water front can use years before it reaches the producer, the pressure response from injecting wells are observed instantly in the surrounding wells. By defining a diffusive time-of-flight, the arrival-time of pressure interference can be related to the diffusive time-of-flight. More on how to integrate pressure data with the use of streamline formulation can be found in [2].

In this work only sensitivities with respect to permeability are considered. Derivatives with respect to porosity, relative permeability, mobility, well rates and the magnitude of the pressure gradient is given in [15].

The streamline approach to compute sensitivities is optimal in the sense that it only needs one forward simulation. Compared to a finite difference approximation of the sensitivities the potential to save computational time is huge. The streamline based computation of the sensitivities is not restricted to a streamline simulator. In [7] streamline sensitivities are used together with a standard forward simulator based on finite difference to speed up the computation of the sensitivities.

Still much work remains in the streamline based computation of the sensitivity. First of all more reservoir parameters and responses need to be related to the time-of-flight sensitivities. As important is it to understand the errors done in the approximations of the sensitivity.

Chapter 6

History Matching

Far better an approximate answer to the right question,
which is often vague,
than an exact answer to the wrong question,
which can always be made precise.
John W. Tukey

The goal of history matching is to use dynamic data from reservoir responses to characterize the reservoir parameters. Different types of reservoir responses are used for history matching. One type of data used in history matching is the amplitude of production data such as pressure data, flow rates, and fractional flows. Examples of reservoir parameters that can be determined in the history matching procedure are permeability, relative permeability and porosity.

History matching is an inverse problem. The goal is to find reservoir parameters such that the mismatch between the simulated reservoir responses, $d_k^{cal}(t_{k,j})$ and the historical reservoir responses, $d_k^{obs}(t_{k,j})$ is minimized for all time, $t_{k,j}$, where the time index $j = 1, ..., N_d$ and the well index $k = 1, ..., N_w$. In a least squares formulation the history matching problem is to find the reservoir parameters such that

$$\sum_{k=1}^{N_w}\sum_{j=1}^{N_d}(d_k^{obs}(t_{k,j}) - d_k^{cal}(t_{k,j}))^2 \tag{6.1}$$

is minimized for all k and j.

Some of the challenges related to history matching is due to the non-linearity in the governing fluid flow equations in the porous mediums. From Chapter 3, it is known that the relation between the reservoir parameters and the reservoir responses is a set of nonlinear differential equations. In general it is not possible to find an analytic solution to such a set of equations. Therefore $\mathbf{d}_k^{cal}$ is picked from the results of a reservoir simulation. To find $\mathbf{d}_k^{cal}$ is therefore as time consuming as one simulation with the simulator. Since the inverse problem is non-linear a iterative method must be used to minimize the objective function. The Jacobian must therefore be calculated in each of the iterations. For Newton's method also the Hessian is needed. The calculation of the Jacobian and Hessian typically involves several function evaluations and is therefore the most time consuming part of the minimization. The second main challenge in history matching is therefore the computational time. For large problems with millions of parameters, one simulation can take several hours, even if a fast computer is used. A method that solves the inverse problem with as few as possible simulation runs is therefore preferable. In addition the inverse problem is ill-posed. In history matching data often are available only in the wells. The number of data to match is therefore fewer than the number of parameters to determined. The inverse problem is therefore highly under determined and no unique solution exists.

The challenge related to the ill-posedness of history matching is solved with regularization. Often prior information of the reservoir is known via static data from well-cores, seismic, etc.

Prior knowledge of the reservoir is used to regularize the problem as it is formulated in (4.9). Regularization will make the problem well-posed such that a unique solution to the regularized problem exists. Still it is important to keep in mind that the solution to the regularized problem is biased and depends on the chosen regularization parameters.

To speed up the simulation a streamline based simulator is used. A huge advantage with the streamline formulation is that a semi-analytic approximation of both the Jacobian and the Hessian can be found for data that can be related to the time-of-flight variable. With the streamline based formulation of the sensitivities, an iterative nonlinear solver will use only one simulation for each of the iterations.

To make the history matching problem less nonlinear, production data at a given reference time such as the peak production time or the first arrival time of water or tracers can be matched. This is called travel-time inversion. If t_k^{obs} is the observed reference time at well k and t_k^{cal} is the calculated time, the travel-time inversion finds the reservoir parameters that minimize

$$\sum_{k=1}^{N_w}(t_k^{obs} - t_k^{cal})^2,$$

for all k. Travel-time and amplitude inversion is illustrated in Figure 6.1. While the inverse problem associated with matching production data amplitudes is fully nonlinear, the travel-time inversion has quasi-linear properties well known in the context of seismic inversion [6]. This makes travel-time inversion more stable and robust than the amplitude matching. Still travel-time inversion only uses production data at one given time. To include all data in the amplitude matching, while keeping the robustness of the travel-time inversion, the amplitude matching and travel-time matching is merged into the so-called generalized travel-time inversion.

6.1 The generalized travel-time inversion

The main idea behind the generalized travel-time inversion (GTTI) is to combine production data matching with travel-time matching. According to [6] GTTI preserves the quasi-linear property of the travel-time method while making full use of the production data. The GTTI solves the history matching problem in two steps. First it finds the time-shift Δt_k that minimize

$$f(\Delta t_k) = \sum_{j=1}^{N_d}(d_k^{obs}(t_{k,j}) - d_k^{cal}(t_{k,j} + \Delta t_k))^2 \tag{6.2}$$

for all k. A cubic interpolation is used to compute $d^{cal}(t_{k,j}+\Delta t_k)$ based on known calculated data. Additional reservoir simulations is therefore not needed to minimize the least squares problem in (6.2). The one-dimensional least squares problem in (6.2) can be solved by systematically shifting the calculated production curve until the objective function is minimized. For example Matlabs lsqnonlin can be used to minimize the objective function in (6.2). Figure 6.1 shows an example of how the calculated production curves are shifted to maximize its correlation to the observed production curve. The second step is to minimize

$$\sum_{k=1}^{N_w}\Delta t_k^2 \tag{6.3}$$

The least squares problem in (6.3) is solved with one of the algorithms for solving nonlinear least squares problems presented in Chapter 4.

The main advantage of the generalized travel-time inversion is that the relation between the time-shift and the reservoir parameters is less non-linear than the direct relation between the amplitudes of the production data [6]. Since the one-dimensional least squares problem in (6.2) can be solved without any need of sensitivities, only the time-shift sensitivities with respect to the reservoir parameters are needed. Time dependent data such as production rates, water cut and

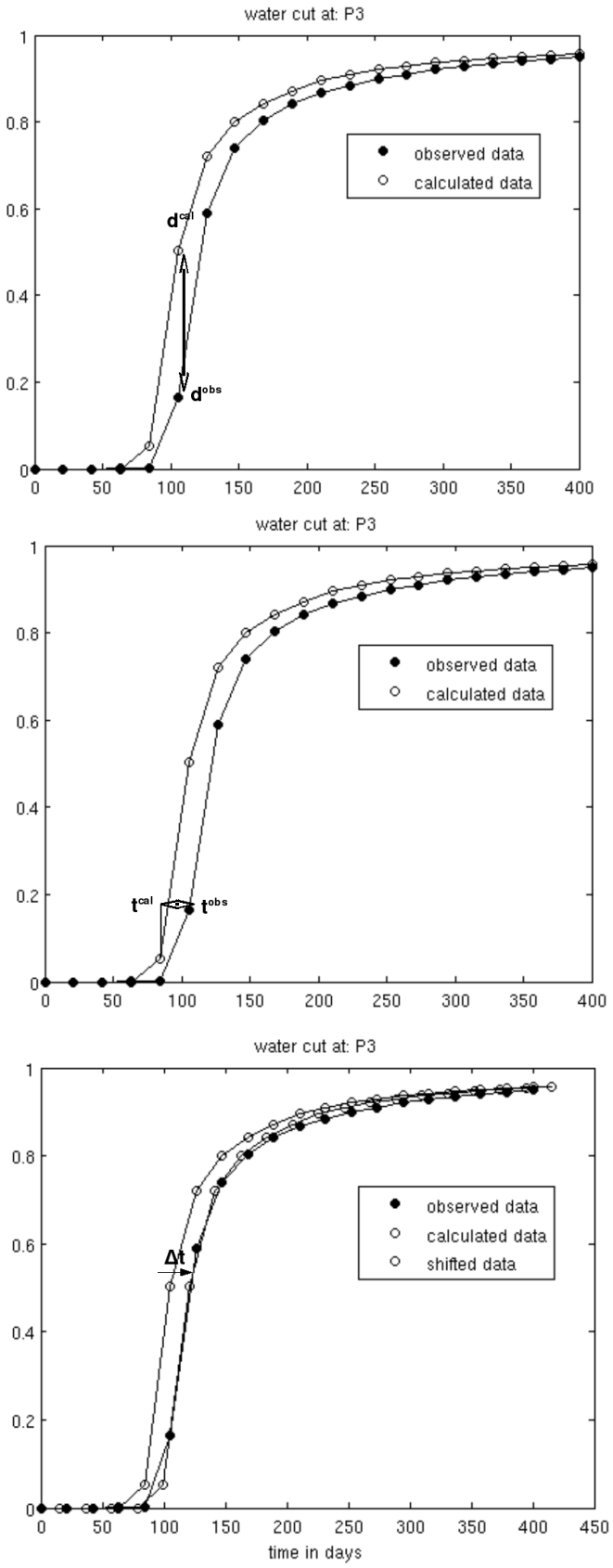

Figure 6.1: Illustration of amplitude inversion (top), travel-time inversion (middle) and GTTI (bottom)

tracer data can be matched with GTTI without need of additional sensitivity calculations. With GTTI more reservoir responses can therefore be match even thought semi-analytic expressions that relates the sensitivities of the reservoir responses to the time-of-flight sensitivities is not known. Another advantage of the time-shift sensitivity is that it is independent of the number of data points. For large problems this could lead to a considerable savings in computational time in the inversion algorithms.

6.1.1 The time-shift sensitivities

The Time-shift sensitivity with respect to the reservoir parameter, $\mathbf{m}$ is defined as an average of the travel-time sensitivities given in (5.13) for each well k

$$\frac{\partial \Delta t_k}{\partial m_i} = -\frac{1}{N_d} \sum_{j=1}^{N_d} \frac{\partial t_{k,j}}{\partial m_i}, \tag{6.4}$$

where N_d is the total number of data points. The minus sign in (6.4) is by convention. The second order sensitivity is similarly given by averaging (5.14):

$$\frac{\partial^2 \Delta t_k}{\partial m_i \partial m_\imath} = -\frac{1}{N_d} \sum_{j=1}^{N_d} \frac{\partial^2 t_{k,j}}{\partial m_i \partial m_\imath}, \tag{6.5}$$

The definitions of the sensitivities in (6.4) and (6.5) are approximations. The Figure in 6.2 compares the semi-analytic first and second order sensitivities with a finite difference based calculation. This comparesing is similar to what is done in Chapter 5. As the figures indicate the difference in the semi-analytic and finite difference approximation is large. One reason for the large variation is the large shift in the permeability in the finite difference approximation. The no shift assumption in the semi-analytic approximations is discussed in the previous chapter and will not be discussed further here. For small shifts in the permeability the effect of streamlines that shift is small on the sensitivity. Therefore the streamline based semi-analytic approximation is a decent approximation of the sensitivities. As in the previous chapter all mixed partial sensitivities are assumed to be zero. It is therefore only the diagonal elements of the Hessian matrix that is plotted in Figure 6.2.

6.1.2 An automatic history matching procedure

Given an initial prior model based on static data, the goal of the history matching is to update the initial model such that the mismatch between the observed and calculated production response is minimized. The main steps of an automatic streamline based GTTI history matching algorithm are: (i) Run the streamline simulator to obtain the desired production responses; (ii) Calculate the time-shift by finding the minimum of (6.2); (iii) Calculate the streamline based semi-analytic time-shift sensitivities with respect to a given reservoir parameter; (iv) Solve (6.3) to updated the reservoir parameters. These steps are repeated until a desired history match is achieved. One stop criteria is to stop the iterations when the objective function reaches a given value. An alternative is to stop the iterations if the reduction in each step is less than a given tolerance, or the number of iterations reaches a maximum given value. Since the purpose of this thesis is to examine different minimization algorithms the iterations will be repeated as long as the objective function is reduced. By not restricting the maximum number of iterations, the long term behavior of the method can be studied. An outline of the automatic history matching procedure is given in a flowchart in Figure 6.3

6.2 Solution methods

The nonlinear least squares problem stated in equation (6.3) can be solved by one of the algorithms that solves nonlinear least squares problem in Chapter 4. The performance of some of these methods will be studied through some examples.

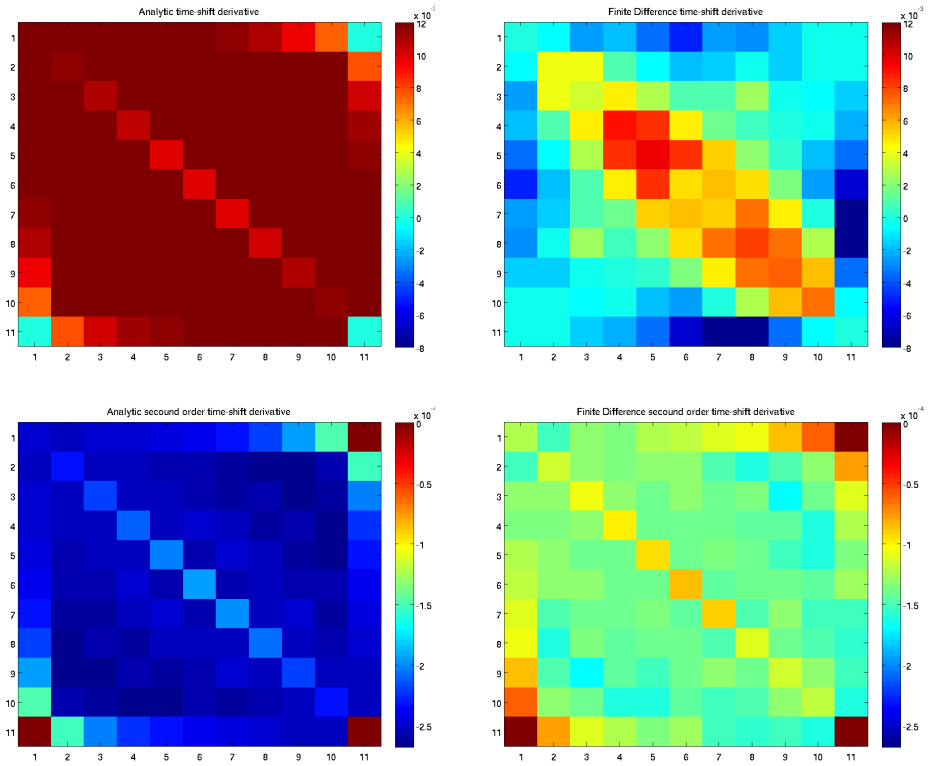

Figure 6.2: Semi-analytic (left) and finite difference (right), first (top) and second (bottom) order time-shift sensitivity.

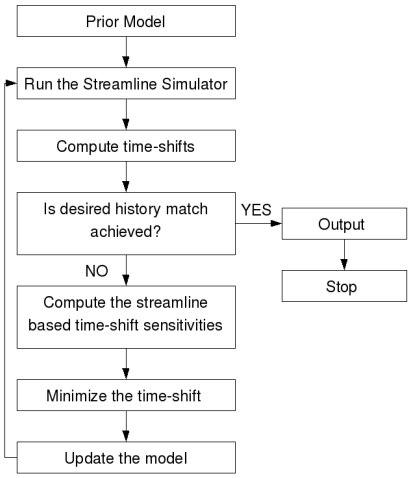

Figure 6.3: A flowchart for automatic history matching

6.2.1 Gauss-Newton vs Newton

A standard way to minimize the objective function in (6.3) is to solve the augmented linear system in (4.19) with LSQR. Both synthetic and field case examples can be found in [2, 6, 16, 17, 20]. This method only makes use of the first order sensitivities. Since it is possible to compute the Hessian matrix without making any new forward runs, Newton's method is a good alternative option for solving the history matching problem. In the context of history matching, the initial model based on static data, is close to the true model, only small adjustments to the model is therefore needed. In this case using LSQR on the linear system in (4.19) gives good results. Solving the linear system in (4.19) corresponds to the Gauss-Newton method. In the Gauss-Newton method, the last part of the Hessian matrix is omitted such that $\nabla^2 f(\mathbf{m}) \approx \mathbf{J}(\mathbf{m})^T \mathbf{J}(\mathbf{m})$. For a almost linear problem where the prior model is close to the true model this is a good approximation. For large residual cases a better approximation is needed. One alternative is to use a more sophisticated quasi-Newton method as BFGS [11], but since the Hessian matrix can be calculated analytically using the streamline formulation, the original Newton method is a better alternative. In theory, the Gauss-Newton method shows slower convergence than the Newton method, if the residuals in the solution are large [11]. The Newton method is therefore expected to show faster convergence than the Gauss-Newton method for large residual cases.

In the Gauss-Newton method, the residual and the Jacobian are needed. The residual is the time-shift, $\mathbf{r} = \Delta\mathbf{t}$ and the Jacobian is the semi-analytic approximation of the time-shift sensitivity, $\mathbf{J}(\mathbf{m}) = \frac{\partial \mathbf{\Delta t}}{\partial \mathbf{m}}$. The Newton method also involves calculation of the Hessian matrix in addition to the residual and the Jacobian. The Hessian matrix for the least squares problem is

$$\nabla^2 f(\mathbf{m}) = \mathbf{J}(\mathbf{m})^T \mathbf{J}(\mathbf{m}) + \sum_{k=1}^{m} r_k(\mathbf{m}) \nabla^2 r_k(\mathbf{m}), \tag{6.6}$$

where $r_k(\mathbf{m})$ and $\nabla^2 r_k(\mathbf{m})$ is the residual and the Hessian in well k, respectively. The Hessian is given by: $\nabla^2 r_k(\mathbf{m}) = \frac{\partial^2 \Delta t_k}{\partial m_i \partial m_i}$. The non-diagonal element of the Hessian is zero since streamlines are assumed to not shift during perturbation. For Newton's method to converge, the Hessian matrix must be positive definite [13]. To assure positive definiteness a small term, λ is added to the diagonal. The λ term will bias the search direction towards the steepest descent direction and must therefore be kept as small as possible. One simple way to chose λ is to put it equal the largest diagonal element of the Hessian matrix. If a large λ must be used to assure positive definiteness, the modified Newton's method will show slower convergence than the Gauss-Newton method. Some synthetic examples will be shown to compare the capability of the Gauss-Newton method (GN) and the Newton method (NM).

6.2.2 Regularization, line-search and trust-regions strategies

In Chapter 4, there is a small discussion about regularization, line-search, trust-region and scaling strategies. The effect of these strategies will also be tested. In the line-search strategy (LS) a cubic interpolant is used to predict the reduction in the objective function. The constants $c_1 = 1.0^{-4}$ and $c_2 = 0.9$ are used in the Wolfe conditions as given in (4.20). These relatively loose restrictions on the time-steps have shown good performance for Newton and Gauss-Newton types of problems [11]. The line-search algorithm is restricted to make only 5 guesses of the step-length, since each guess requires one time consuming run with the streamline simulator. However, in most cases a step-length satisfying the Wolfe conditions is obtained in the first guess. An ellipsoidal trust-region will be used for the trust-region strategy in the Levenberg-Marquardt method (LM) as given in (4.24). The axis of the ellipsoidal is given as the absolute value of the diagonal of the Hessian matrix. An automatic Tikhonovs regularization where the regularization terms are given via the L-curve criteria will also be considered. This method will be referred to as Lcurve.

6.3 Synthetic example cases

In all the examples a two dimensional, two-phase flow of oil and water is considered. The fluid properties is as in the examples in section 3.3. Synthetic water cut data at the well is included to update the permeability in each cell. Since the objective of these test cases is to compare the performance of different minimization techniques, no error is assumed in the dynamic data. The porosity is constantly equal to 0.3 in all the examples. Initially the reservoir is saturated with oil. Five different cases is considered.

6.3.1 Case 1

The first case is a 11×11 reservoir, where each cell has size $10m \times 10m$. Synthetic data is obtained from the true permeability field. No prior information of the permeability is assumed to be known, so the prior permeability field is assumed to be a constant equal the average of the true permeability field. The prior permeability is plotted visa via the true permeability in Figure 6.4. In the center of the reservoir one injector (I1) starts to inject water immediately with a constant rate of $20m^2/day$. In each of the corners, the production wells P1, P2, P3 and P4 produces at a constant production rate of $5m^2/day$. A total of 20 time-steps is taken, each with a length of 20 days. The number of streamlines used is 250.

The objective function for the time-shifts given in (6.3) is shown in Figure 6.5. During the first iterations, while the iterate still is far from the solution, NM and GN show nearly the same convergence rate, but as the iterate approach the solution NM shows faster convergence than the GN method. In the first iterations NM needs to add a small term to the diagonal of the Hessian to assure positive definiteness. This term bias the Newton direction towards the steepest decent direction and hence slows the convergence.

In GNLS and NMLS a line-search strategy is used together with the GN and NM. For both the cases the line-search strategy is able to improve the convergence of the objective function. Especially NMLS shows a fast convergence rate, and is able to reduce the objective function value to less than 1.0^{-5}.

The trust-region strategy is also able to improve the performance of the method. The LM method shows a smaller convergence rate than the GN method in the beginning but is able to reduce the objective function more. In LM a positive term is added to the diagonal of the approximated Hessian, for large such terms the convergence rate will be slower. Still the trust-region strategy is more stable and LM is therefore able to converge in situations where the Gauss-Newton method is not. In LM the trust-region is adjusted during the iterations, a similar adjustment of the regularization terms is done in Lcurve.

The Gauss Newton method with an automatic Tikhonovs regularization determined by the L-curve criteria, Lcurve, shows a slower convergence rate than the GN method without the regularization. The regularization terms will linearize the problem similar to what the LM method did. Therefore slower convergence is expected. Slower convergence is the cost to pay to obtain a more stable and regularized solution. The regularization parameter automatically chosen by the L-curve method is plotted in Figure 6.8. As the iterate moves closer to the true solution, less regularization is used as expected.

Our original problem is to minimize the production misfit. The behavior of the objective function defined in (6.2) is plotted in Figure 6.6. All the methods are able to convergence to reasonable small values of the objective function for production data. LM shows the best performance both that it converges with the least number of iteration and also to the smallest value of the objective function. The plot for the GNLS method shows some strange effects. The reason for this is that all the simulation runs are plotted, also the ones that is rejected since they do not satisfies the Wolfe conditions. Due to this the number of iterations for the GNLS and NMLS are larger in the plot of the production data misfit than in the time-shift misfit plot in Figure 6.5.

The figures in 6.9-6.12 shows the water cut in the production wells for the initial, observed and updated permeability fields for all the different methods. In well P1, P2 and P3 all methods are

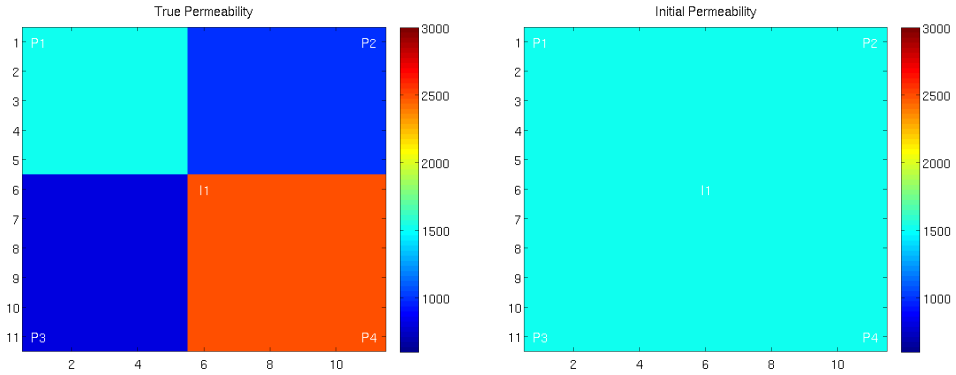

Figure 6.4: True (left) and initial (right) permeability in Case 1

able the match the water cut data. But a closer look at P4 shows that GN and GNLS is not able to match the water cut data in the well as good as the other methods.

The objective in history matching is to use dynamic data to characterize the reservoir. To illustrate the ability of the different methods to obtain a permeability field that resemble the true permeability with the help of water cut data in the wells, the norm of the relative error is plotted in 6.7. The updated permeability fields for the different methods is plotted in Figure 6.13. Both the GN and NM show promising decay in the error on early iteration, but are not able to stop the iterations at the right time. The updated permeability for GN shows that the method gives too high values to the permeability in the lower right corner of the reservoir. NM is able to characterize the true permeability field in the lower right corner. Still in the lower left corner NM is not able to reduce the value of the permeability as good as LM.

The pressure in the wells are plotted in Figure 6.14. These figures show that the pressure in the well is adjusted to maintain a fixed production rates in the well. In well P1, the pressure is almost zero. In P2 and P3 the well pressure is negative such that more oil and water is produced. In P4, which is connected to I1 with a high permeable zone, the pressure is positive such that less oil and water is produced than expected. As these figures indicate the reservoir responses are sensitive to the pressure in the wells. The updated permeability field with NM is not able to match the pressure data in the well. The other methods show approximately the same poor performance. This is expected since pressure data is not used in the history matching. A better characterization of the reservoir is possible if pressure data is included in the history matching. The streamline simulator used in this thesis only uses simple wells with fixed production rates. Since only water cut data is used to characterize the permeability in the reservoir, rate controlled wells is not optimal. Still since all methods are using the same simulator, the comparing of the methods is valid.

The streamline effects can be seen in all the updated permeability fields. The effect of fewer traced streamlines in the center and on the edges is also clearly seen in the entire updated permeability field. This effect is reduced, if more complex cell geometries and more streamlines are used in the simulations.

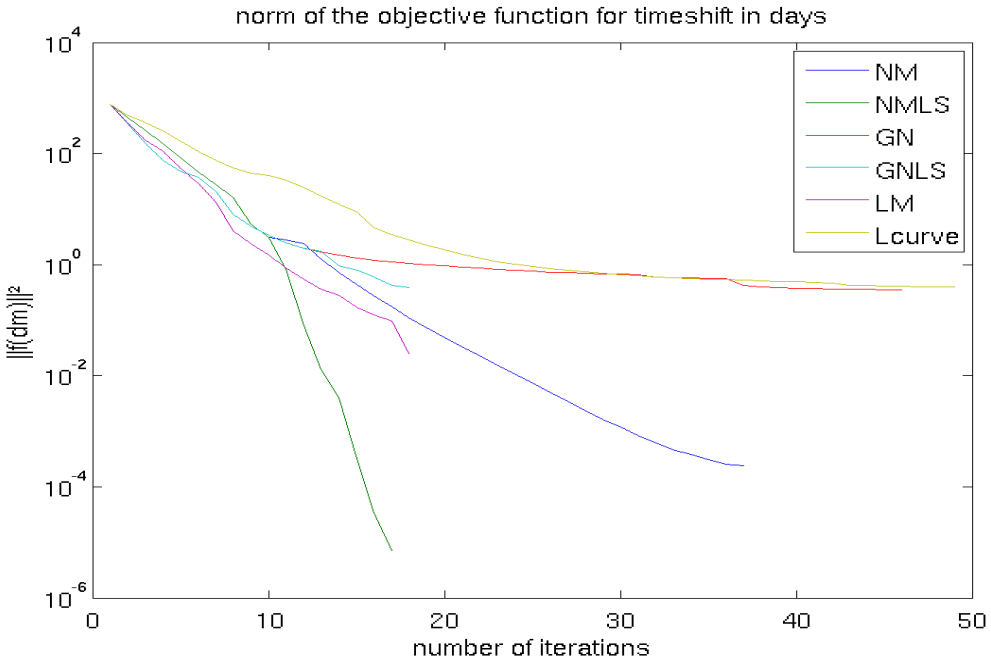

Figure 6.5: Norm of objective function of time-shifts in Case 1

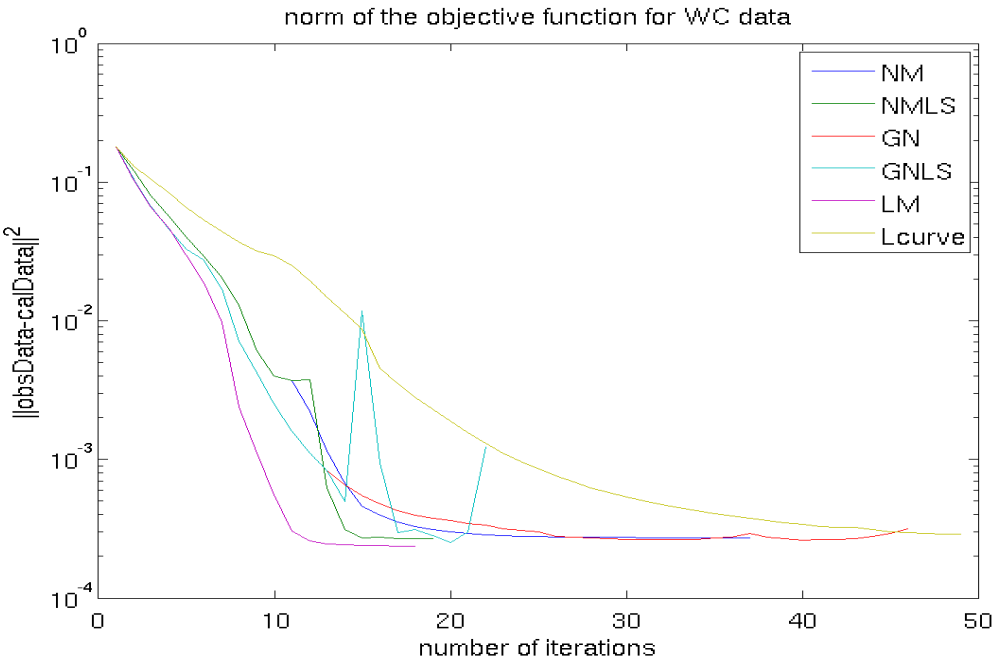

Figure 6.6: Norm of the objective function of water cut data in Case 1

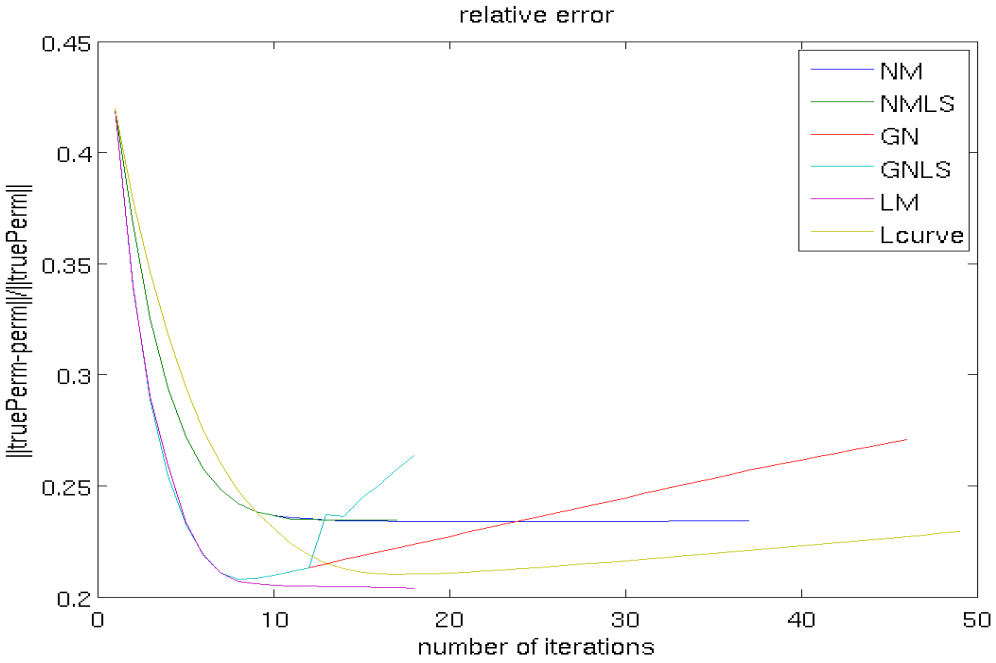

Figure 6.7: Norm of relative error in Case 1

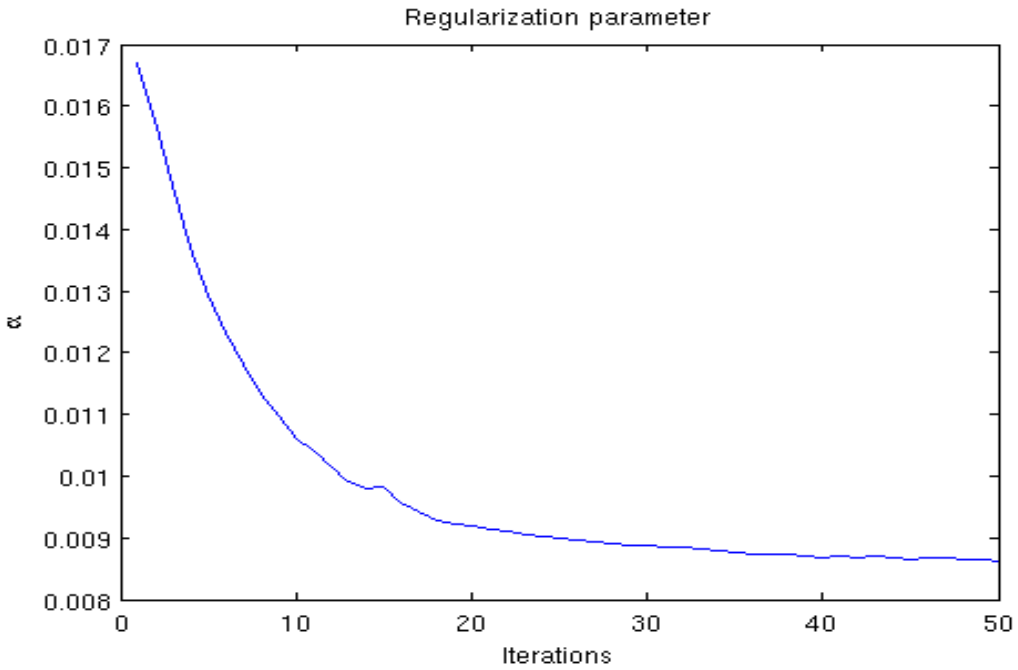

Figure 6.8: Regularization term given by L-curve criteria in Case 1

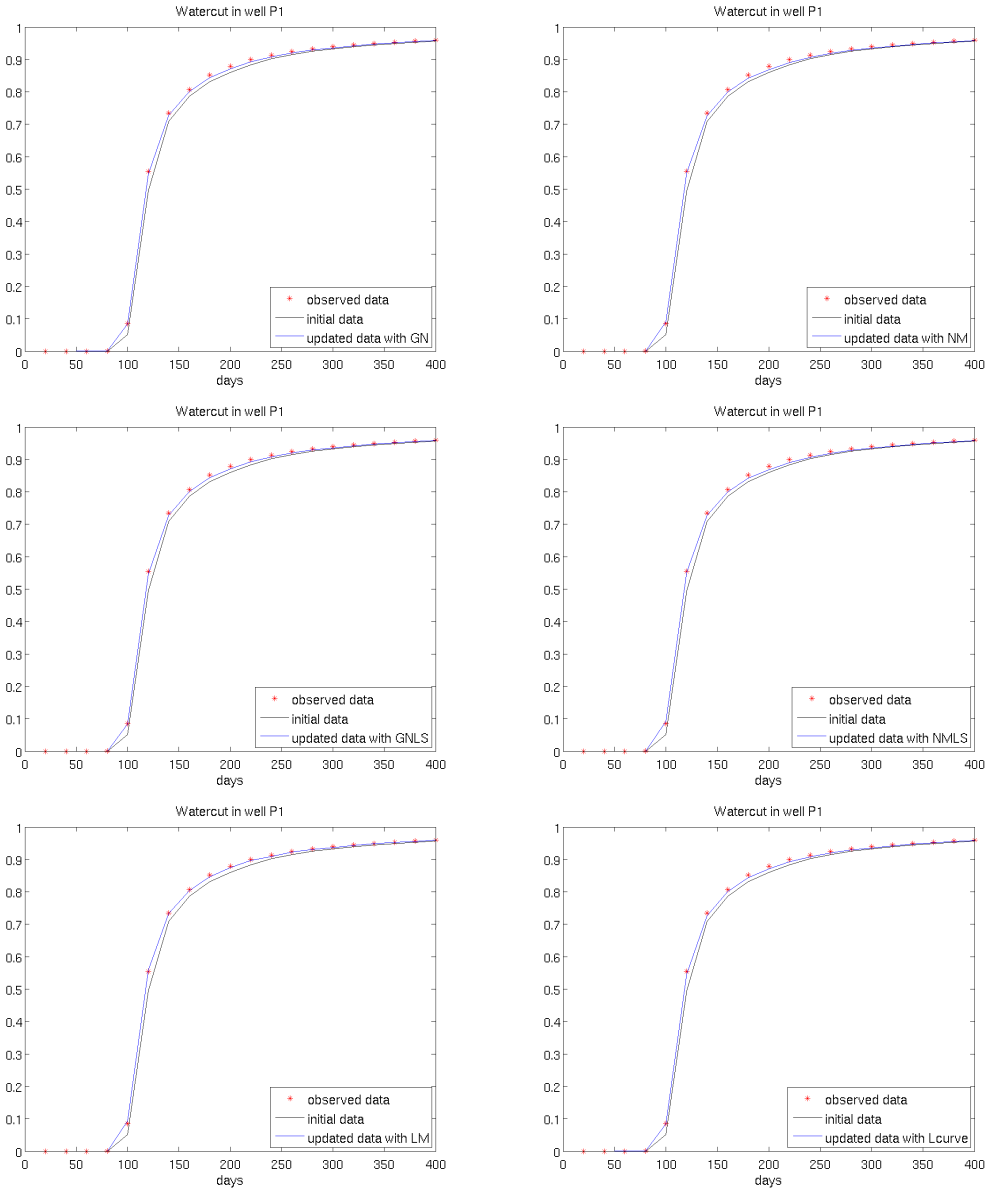

Figure 6.9: Observed (*), initial (black) and updated (blue) water cut in well P1 for GN (top left), NM (top right), GNLS (middle left), NMLS (middle right), LM (bottom left) and Lcurve (bottom right) in Case 1

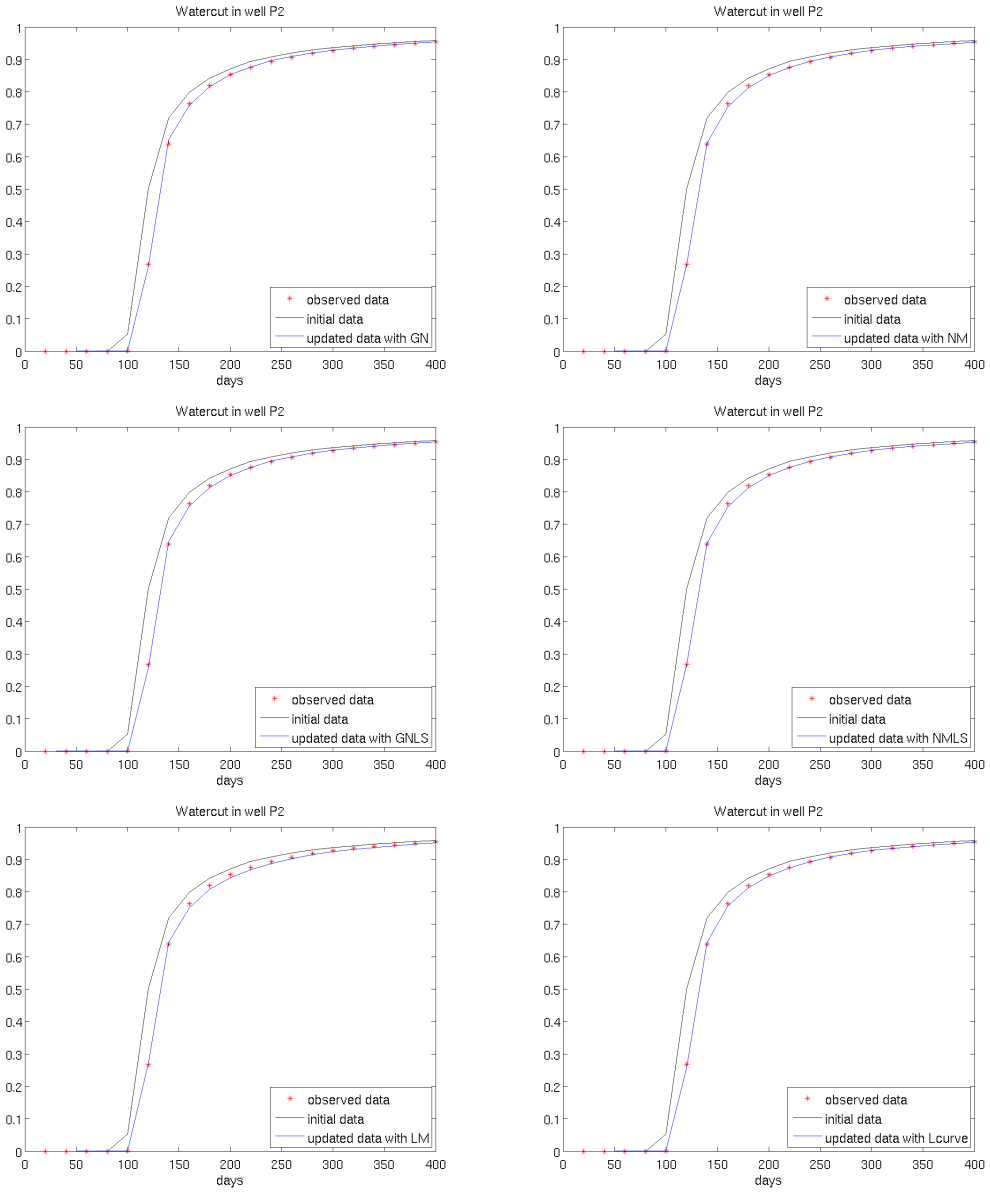

Figure 6.10: Observed (*), initial (black) and updated (blue) water cut in well P2 for GN (top left), NM (top right), GNLS (middle left), NMLS (middle right), LM (bottom left) and Lcurve (bottom right) in Case 1

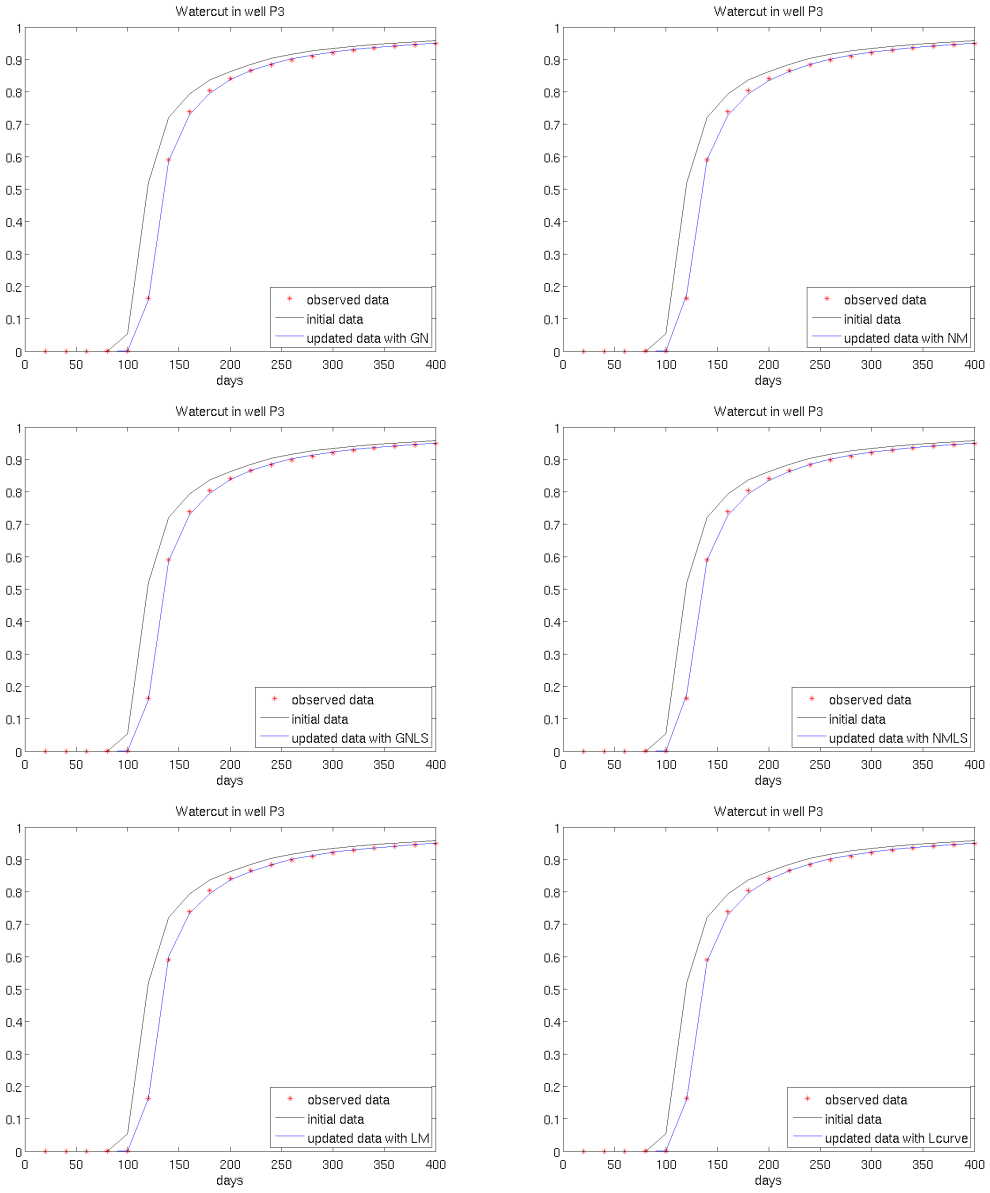

Figure 6.11: Observed (*), initial (black) and updated (blue) water cut in well P3 for GN (top left), NM (top right), GNLS (middle left), NMLS (middle right), LM (bottom left) and Lcurve (bottom right) in Case 1

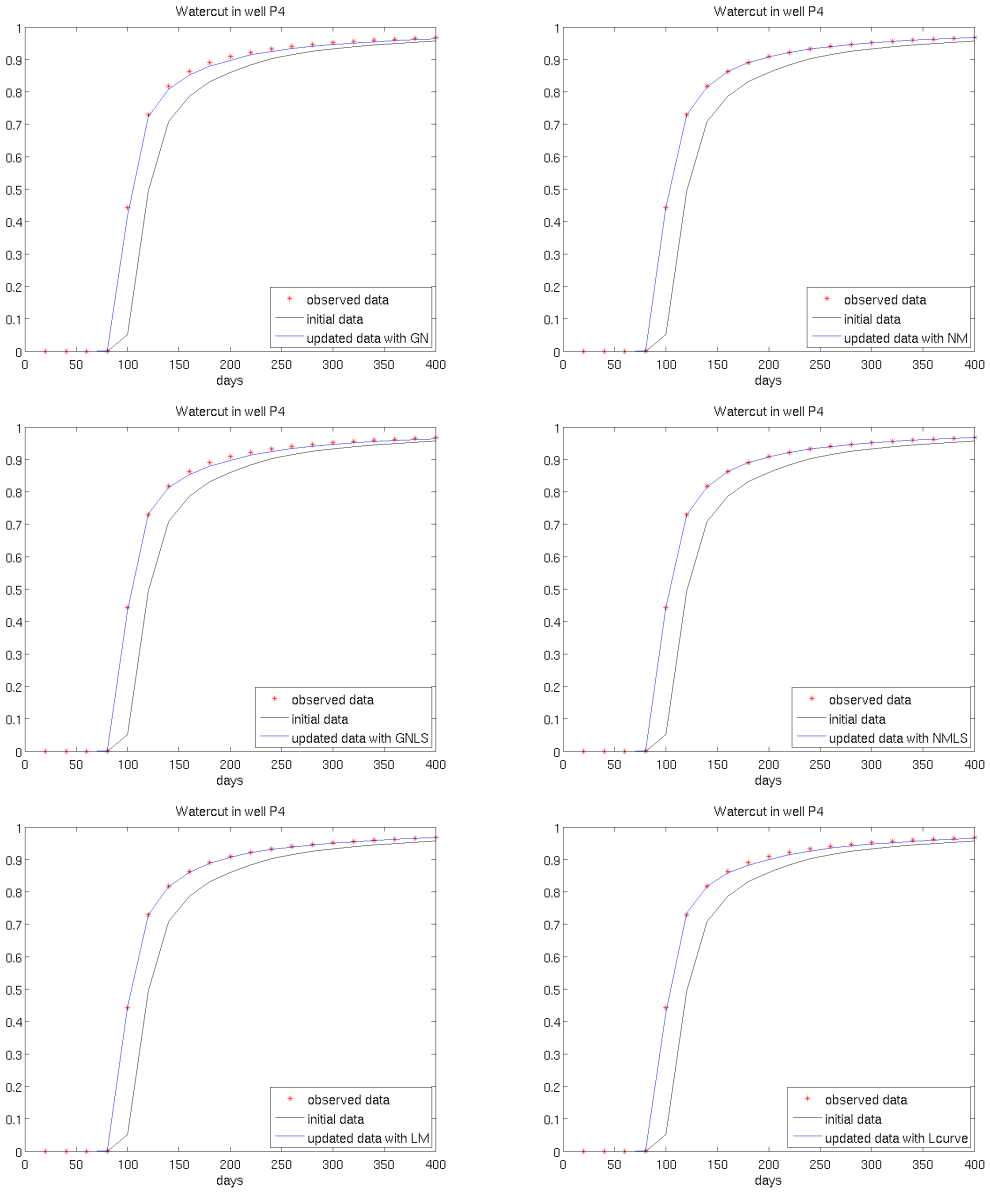

Figure 6.12: Observed (*), initial (black) and updated (blue) water cut in well P4 for GN (top left), NM (top right), GNLS (middle left), NMLS (middle right), LM (bottom left) and Lcurve (bottom right) in Case 1

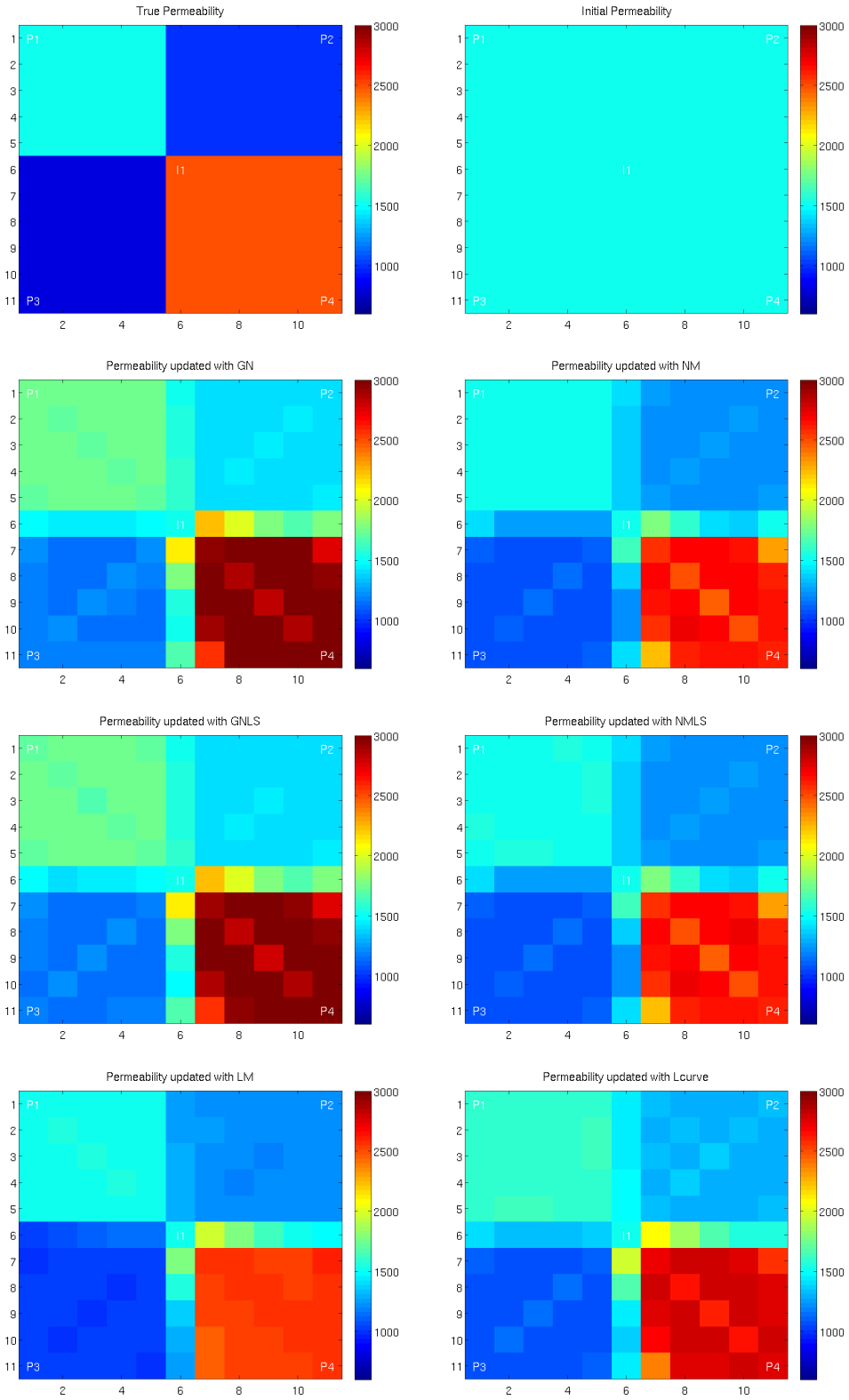

Figure 6.13: Updated permeability for GN (top left), NM (top right), GNLS (middle left), NMLS (middle right), LM (bottom left) and Lcurve (bottom right) in Case 1

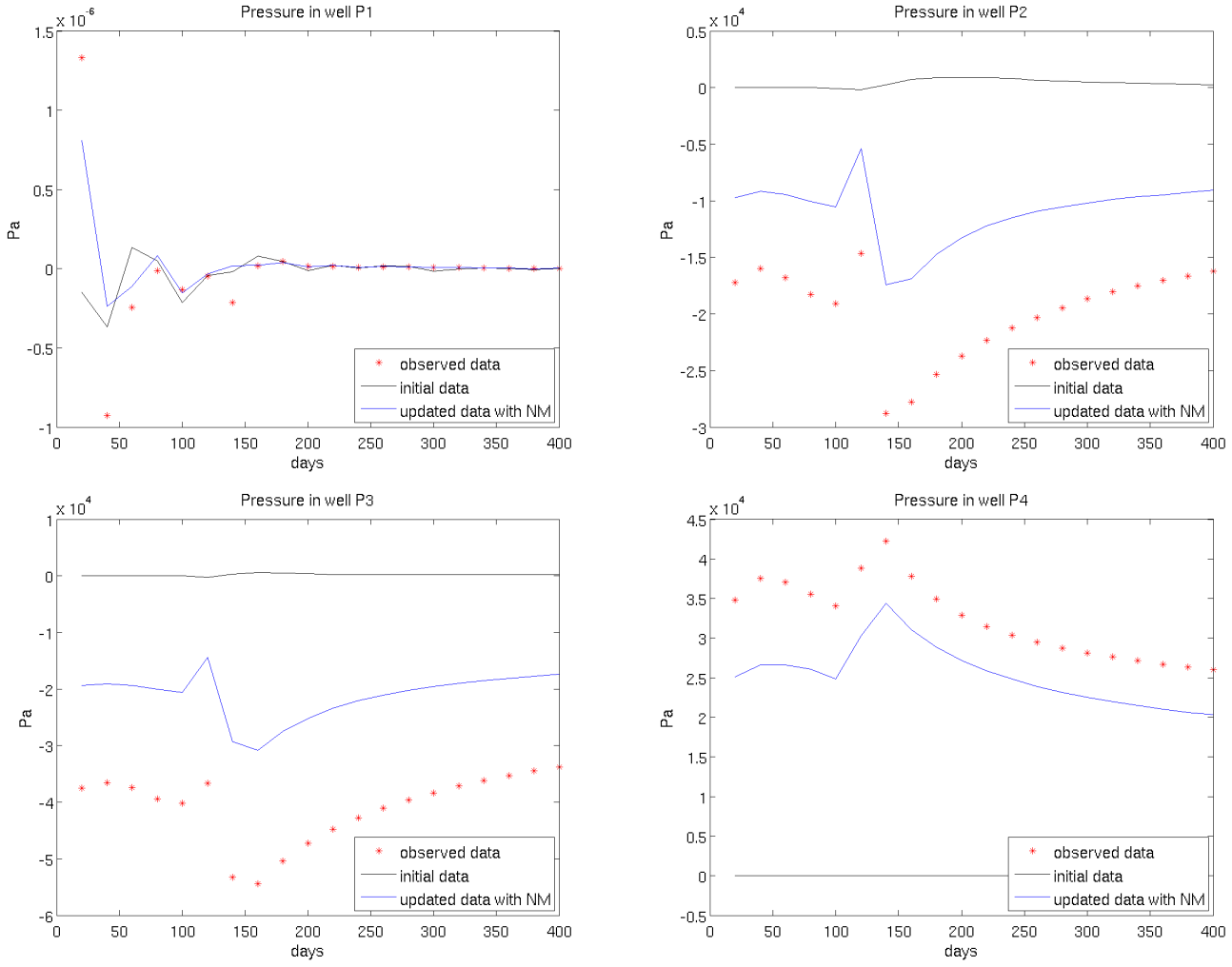

Figure 6.14: Observed (*), initial (black) and updated pressure data with NM (blue) in well P1 (top left), P2 (top right), P3 (bottom left) and P4 (bottom right) in Case 1.

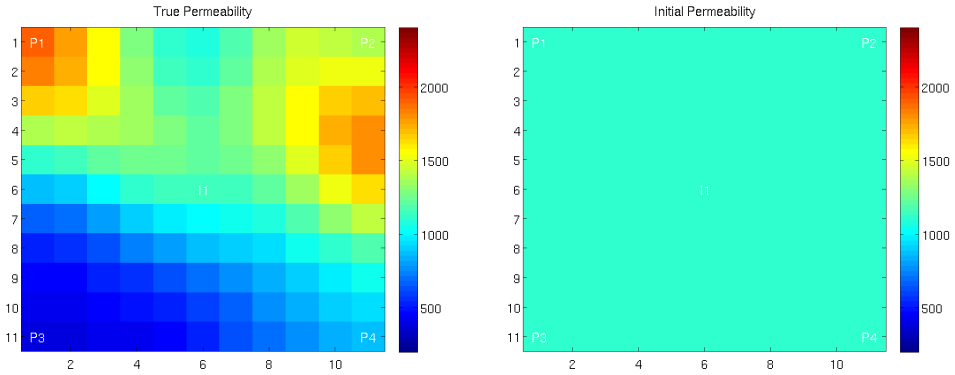

Figure 6.15: True (left) and initial (right) permeability in Case 2

6.3.2 Case 2

The true permeability field used in the first case is made for the purpose of illustrating the performance of the different methods. In the next cases less regular true permeability field will be used to obtain synthetic data used for the history matching. First the 11×11 reservoir from Case 1 will be considered but with a different true permeability field. No prior information of the structure of the reservoir is assumed to be known so the initial permeability field is a constant. The true and prior permeability is shown in Figure 6.15. All other input parameters are similar to Case 1.

In Case 2, NM gives a better reduction of the objective function than GN and Lcurve. See Figure 6.16. Still the extended Gauss-Newton methods, GNLS and LM show more or less the same reduction of the objective function as NM. Both GNLS and NMLS are able to reduce the objective function faster than the other methods. However as Figure 6.17 indicates, the LS-strategy needs to take some extra simulations to be able to compute step-lengths satisfying the Wolfe conditions. Figure 6.18 shows the relative error of the updated permeability and Figure 6.19 shows the updated permeability for the different methods. None of the methods are able to characterize the structure of the true permeability field. This is expected since all the data are available in wells. The local structure of the reservoir can therefore not be found with only dynamic data in the wells. In a history matching problem, some of the structures of the reservoir are assumed to be known. Static or dynamic data from seismic surveys are not restricted to wells and can therefore be used to characterize the structure of the permeability field. In the next case a prior permeability field is used as a initial guess in the history matching algorithms.

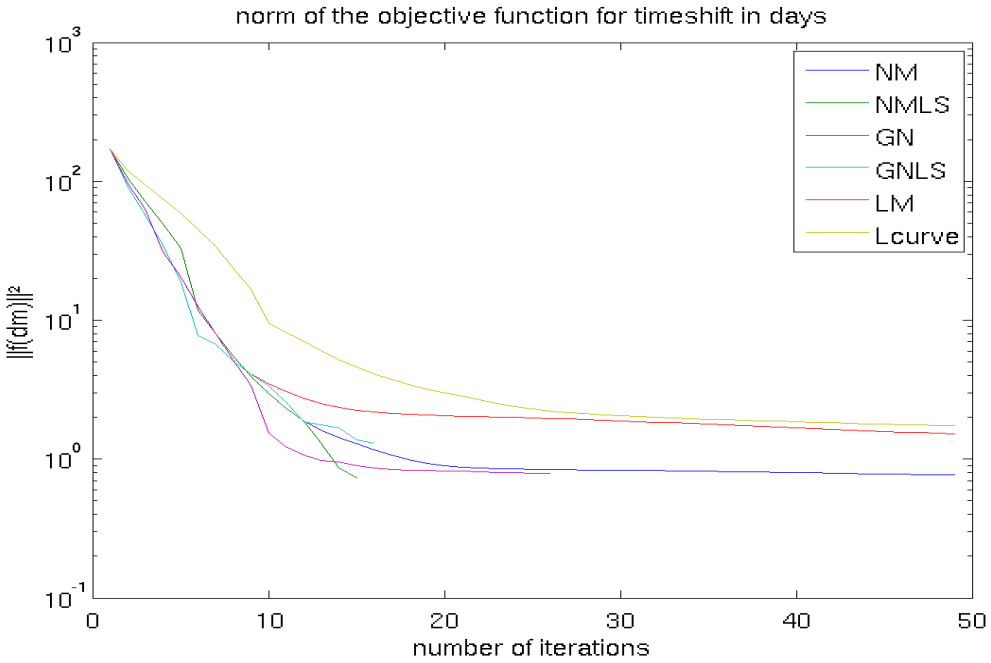

Figure 6.16: Norm of objective function of time-shifts in Case 2

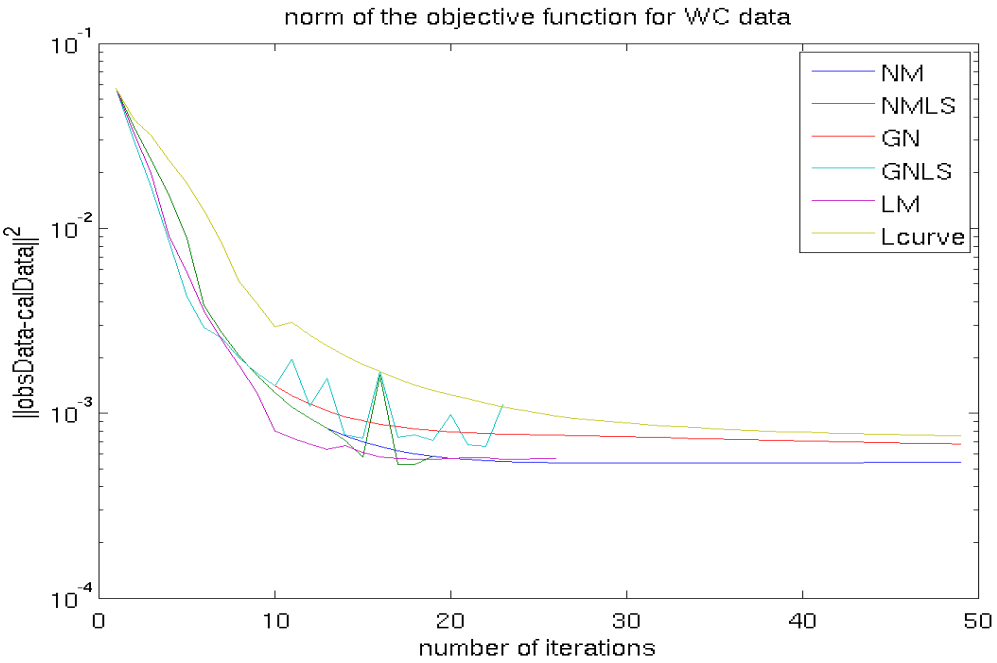

Figure 6.17: Norm of the objective function of water cut data in Case 2

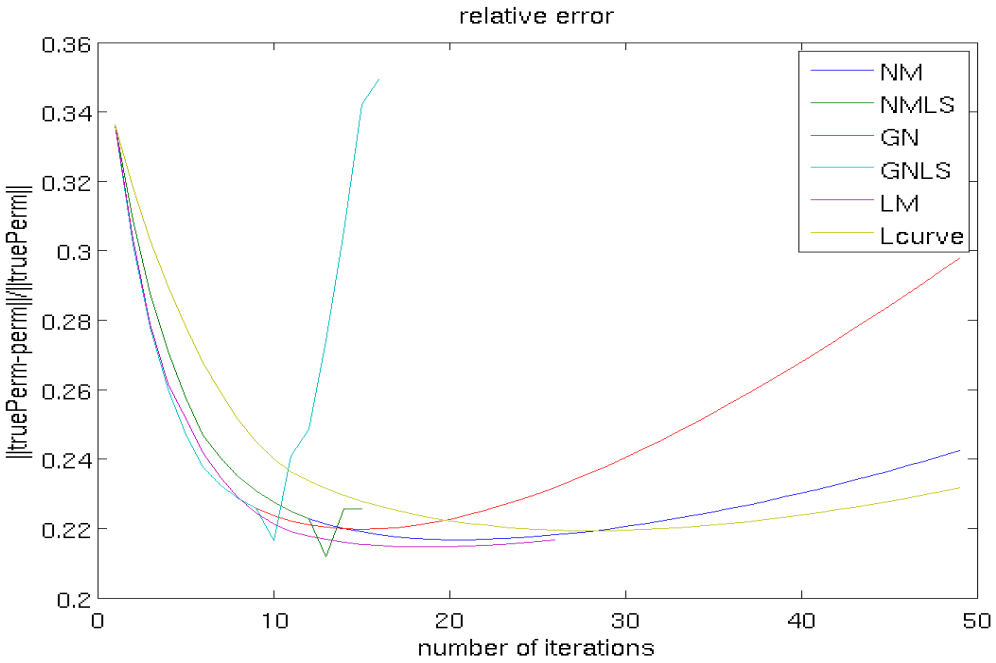

Figure 6.18: Norm of relative error in Case 2

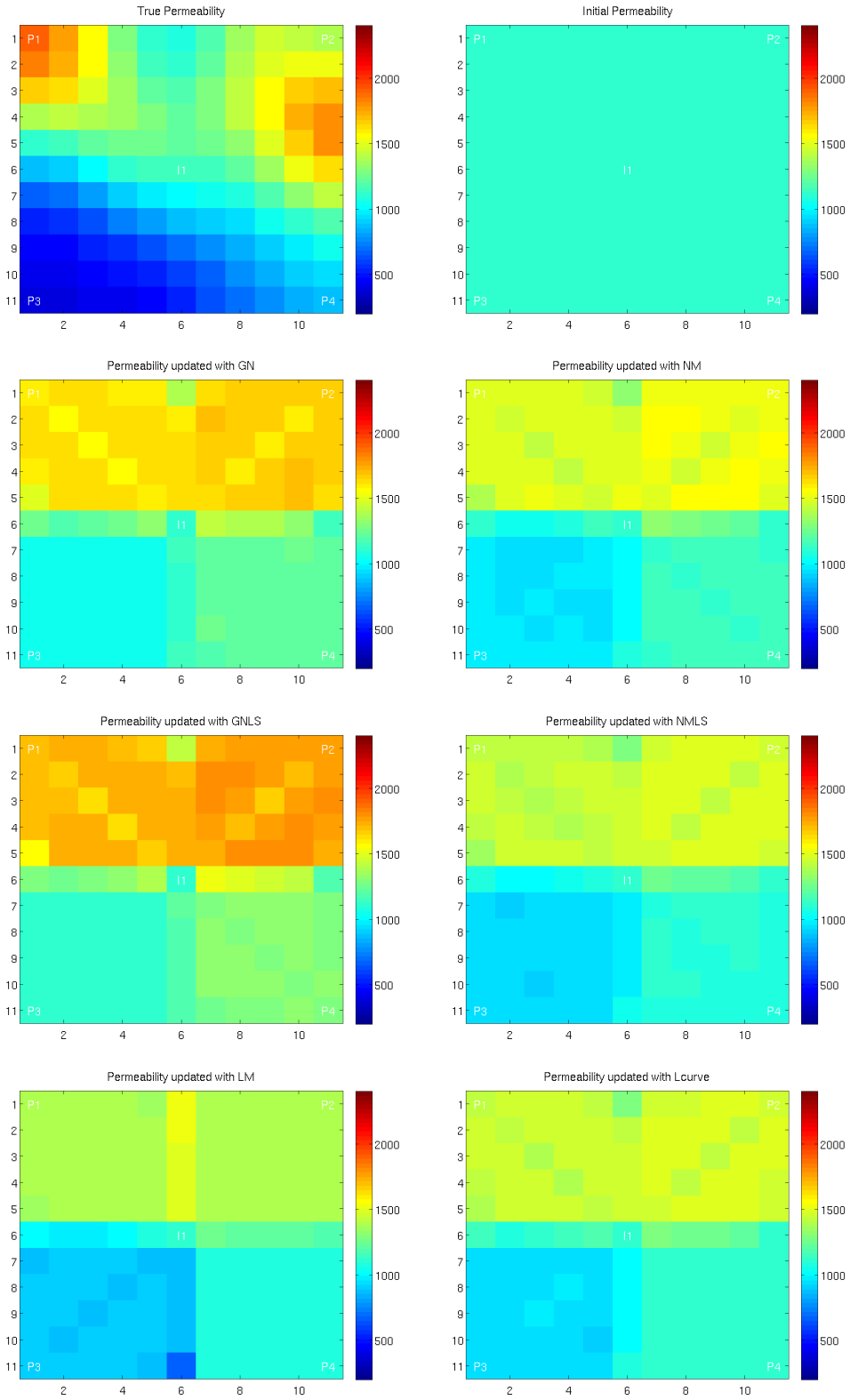

Figure 6.19: Updated permeability for GN (top left), NM (top right), GNLS (middle left), NMLS (middle right), LM (bottom left) and Lcurve (bottom right) in Case 2

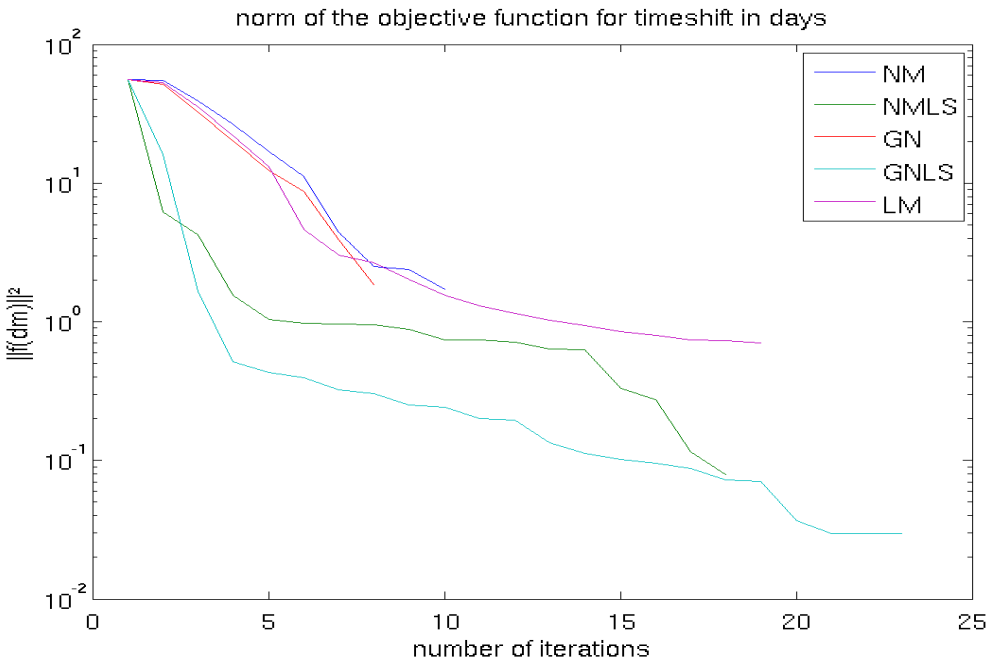

Figure 6.20: Norm of objective function of time-shift in Case 3

6.3.3 Case 3

In Case 3 a permeability field which shows some of the same structures as the true permeability field is used as a prior guess in the algorithms. All the other parameters and the true permeability field is the same as in Case 2. The prior permeability field is plotted together with the true permeability field in Figure 6.23. The prior permeability is used to regularize the problem as formulated in (4.9). The regularization parameter, $\alpha = 10^{-4}$ is used in for all the methods. Since a regularization parameter is fixed prior to the iterations, Lcurve with L-curve criteria is not tested in Case 3.

In Case 3 NM, GN and LM show approximately the same performance. Both in reducing the objective function for the time-shift and the water cut data. See Figure 6.20 and 6.21. The LS-strategy is able to improve the reduction of the objective function of time-shift considerably. Especially NMLS performs well for Case 3. But a closer look at Figure 6.21 shows that both GNLS and NMLS needs to make several iterations to find a step-length satisfying the Wolfe conditions. If these extra iterations are taken into account, the different methods show almost the same reduction rate in the objective function.

In Case 3 regularization is added to punish large variations from the prior model. In this way some of the structure of the prior field is preserved in the updated permeability fields. If the updated permeability fields for Case 2 in Figure 6.19 is compared with the updated permeability fields for Case 3 in 6.23, it shows that the updated permeability fields in Case 3 are more similar to the true permeability field. The plot of relative error in Figure 6.22 shows a decreasing error for all the methods. Especially GNLS and NMLS show a fast reduction of the relative error in the first couple of iterations.

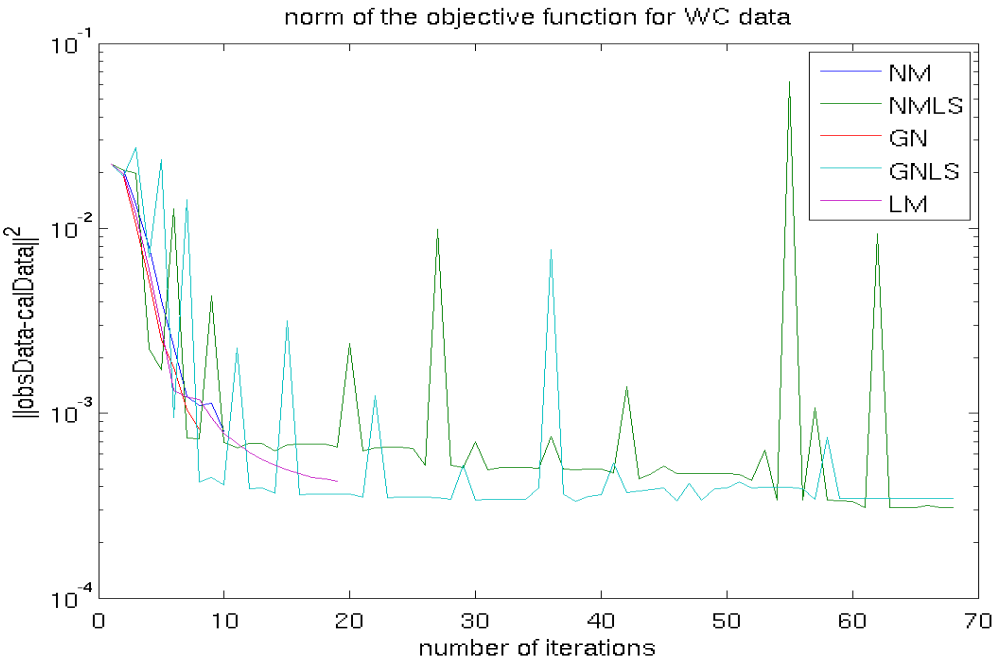

Figure 6.21: Norm of the objective function of water cut data in Case 3

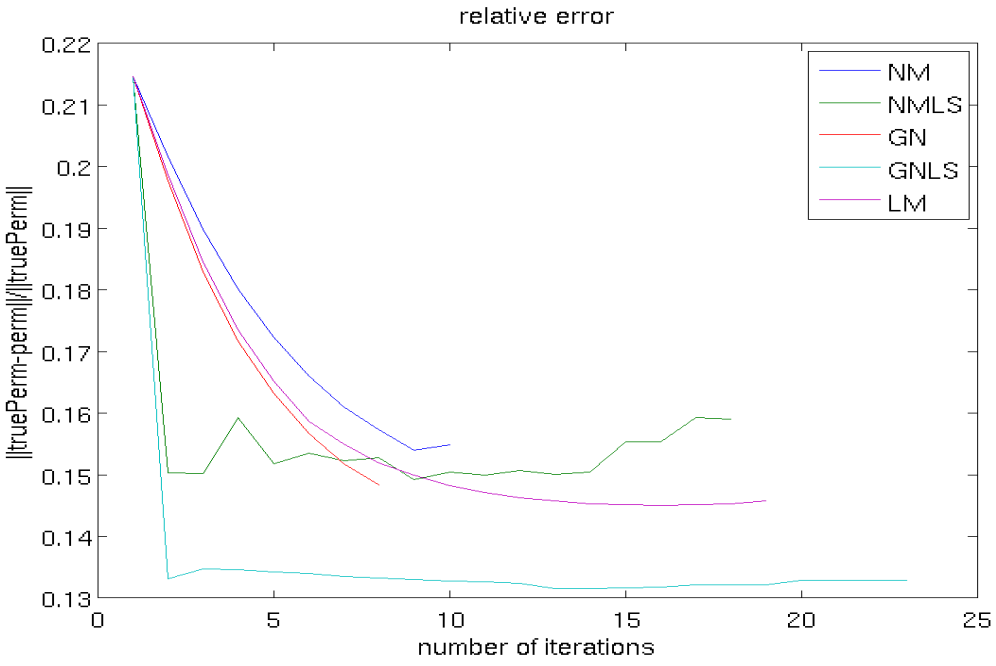

Figure 6.22: Norm of relative error in Case 3

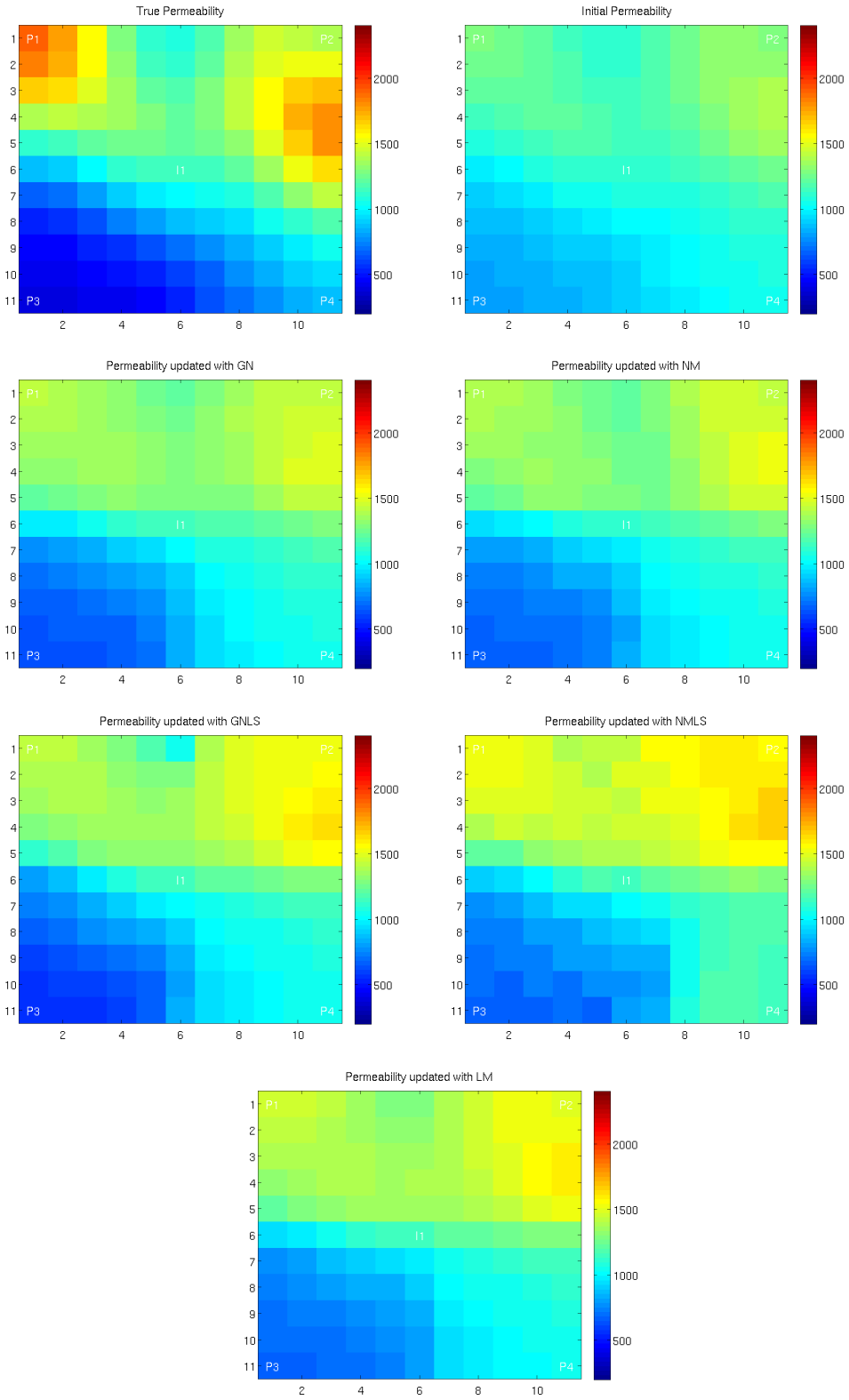

Figure 6.23: Updated permeability for GN (top left), NM (top right), GNLS (middle left), NMLS (middle right), LM (bottom left) and Lcurve (bottom right) in Case 3

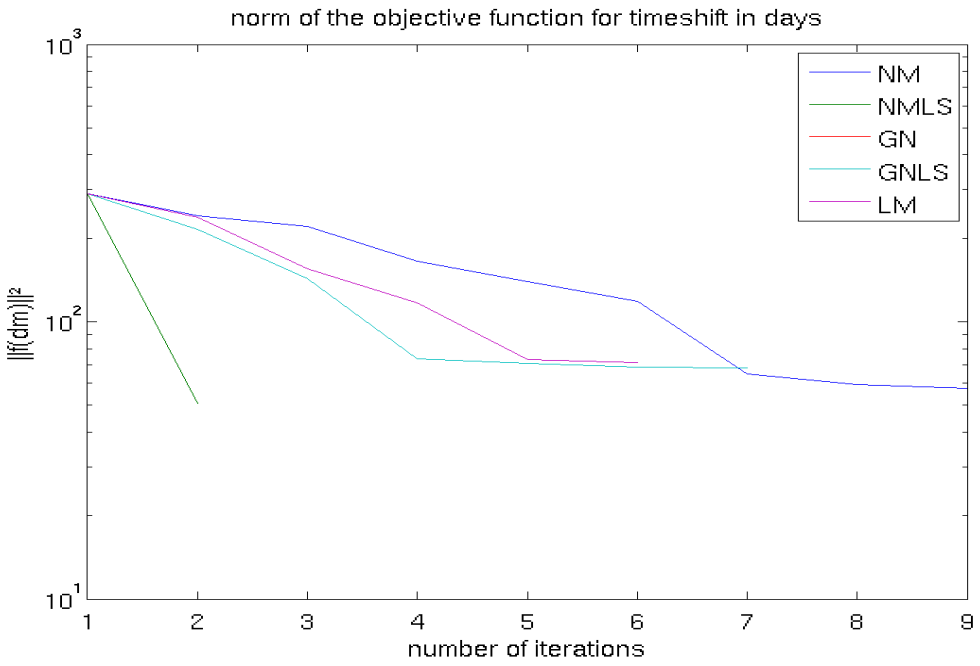

Figure 6.24: Norm of objective function of time-shift in Case 4

6.3.4 Case 4

In the three first cases a coarse grid with only 11×11 grid cells is used. One of the advantages with streamline simulation is that fine scale models can be used. In the last two cases the number of grid cells is extended to 41×41, each with size $1m \times 1m$. All the other fluid and rock parameters is as in the previous cases. As in Case 3, a prior permeability field is assumed to be known and used to regularize the problem. The regularization parameter used is the same as in Case 3, $\alpha = 10^{-5}$. A total number of 500 streamlines is used. The prior and true permeability field are plotted in the top of Figure 6.27.

As Figure 6.24 indicates none of the methods are able to reduce the objective function significantly. The largest reduction in the objective function of time-shift is done by NMLS. The objective function for water cut data is plotted in Figure 6.25, relative error in 6.26 and the updated permeability fields in Figure 6.27. Some change can be observed in the updated permeability fields. In the lower right corner, especially the NMLS is able to characterize the low permeable region. Still none of the methods is able to characterize the low permeable zone that separates well P3 from I1. The effect of the high permeable regions in the lower left corner is smeared out in the whole region where the streamlines from well I1 to P2 is traced. The GN and NM show almost the same performance in Case 4.

Case 4 emphasizes what was seen in Case 3. As the problem becomes more ill-posed, more regularization is needed, a larger positive term is added to the Hessian to make sure it is positive definite, and the trust-region is extended. The consequence is the same. A positive term is added to the diagonal of the Hessian matrix such that the search direction is biased towards the steepest decent direction. Since the steepest decent method has a slower convergence rate than both the Gauss-Newton and Newton's method a slower convergence is observed. The line-search strategy is able to speed up the convergence in Case 4. Since it can adjust the step length to obtain a decent reduction in the objective function even thought the search direction is not optimal.

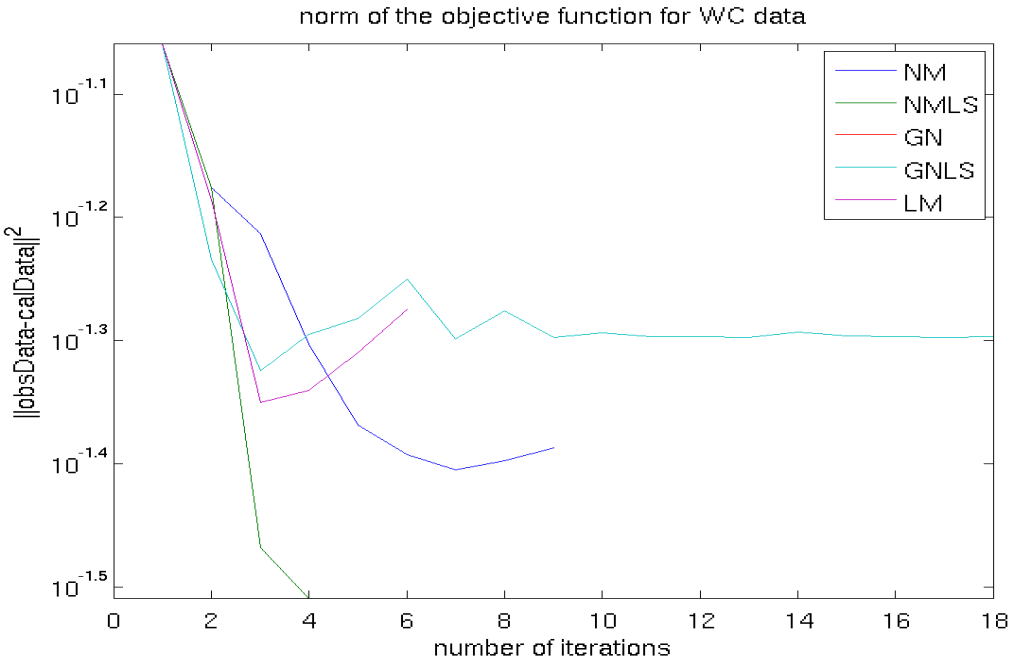

Figure 6.25: Norm of the objective function of water cut data in Case 4

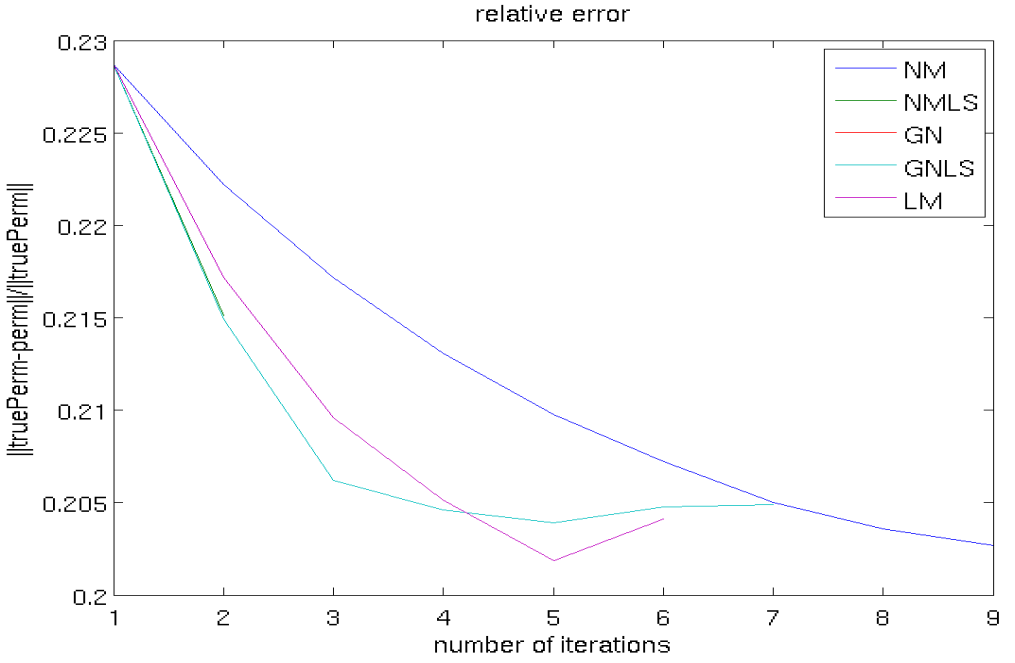

Figure 6.26: Norm of relative error in Case 4

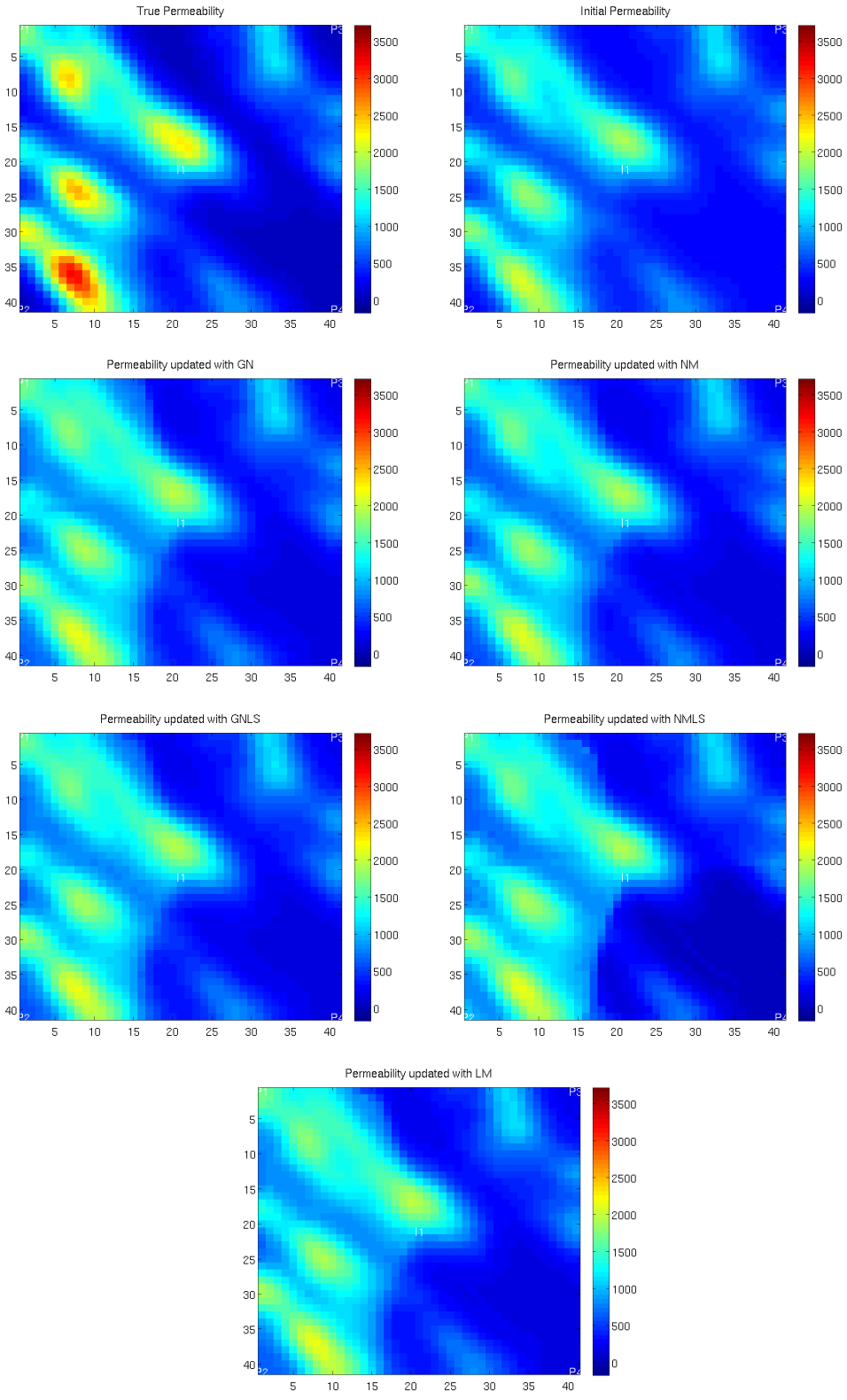

Figure 6.27: Updated permeability for GN (top left), NM (top right), GNLS (middle left), NMLS (middle right) and LM (bottom center) in Case 4

6.3.5 Case 5

In the last case the ability to characterize a high permable zone from the injection well, I1 to the production well, P3 is tested. As in the previous cases, only water cut data from the wells is used in the history matching. The true and initial permeability field is given in 6.31. Other input parameters are as in Case 4.

The norm of the objective function of time-shift and production data are plotted in Figure 6.28 and 6.29. As this figure indicates GN shows the best reduction in the objective function, followed by LM and NM. The line search strategy fails both for GN and NM and is not able to reduce the objective function any further. Figure 6.30 shows that the relative error are increasing for all the methods. The updated permeability field in 6.31 shows that all the methods are able to characterize some of the high permeable zone from the injection well to well P3. As expected the region is smeared out since data only are available in the wells. The low permeable zone in the lower right corner is however not detected by any of the methods.

Water cut and pressure data is plotted for P2, P3 and P4 for GN and NM in 6.32-6.37. There is almost no difference between the initial and the true permeability field in the lower left corner. Both GN and NM is therefore able to match the water cut data in P2 as expected. Still a closer look at the pressure data in the same well shows that the pressure in the updated permeability field is further from the pressure data for the true permeability field than for the initial field. In P3 both GN and NM is able to match the water cut data better. Especially the water cut data after 50 days shows a good match. Also the pressure data match is better in P3 for both the methods. That is however not the situation in P4. In P4 as in P2 the pressure match after the update is worse than before the update. If pressure data is included in the history matching or if pressure is used to control the wells, a better characterization is expected. The match of water cut data is improved with both GN and NM. As these figures indicates the difference between the performance of GN and NM is not to large. The second order information of the sensitivities is not able to improve the history matching in Case 5 significantly. The line-search strategy also fails in Case 5 and is not able to improve the performance of the method. During the first iterations the step-length is not adjusted. One reason for this is the loose Wolfe conditions. With more strict Wolfe conditions, and a larger number of allowed step-length guesses, a better reduction of the objective function is expected. Still the additional time required to do so, will not justify the expected improvements.

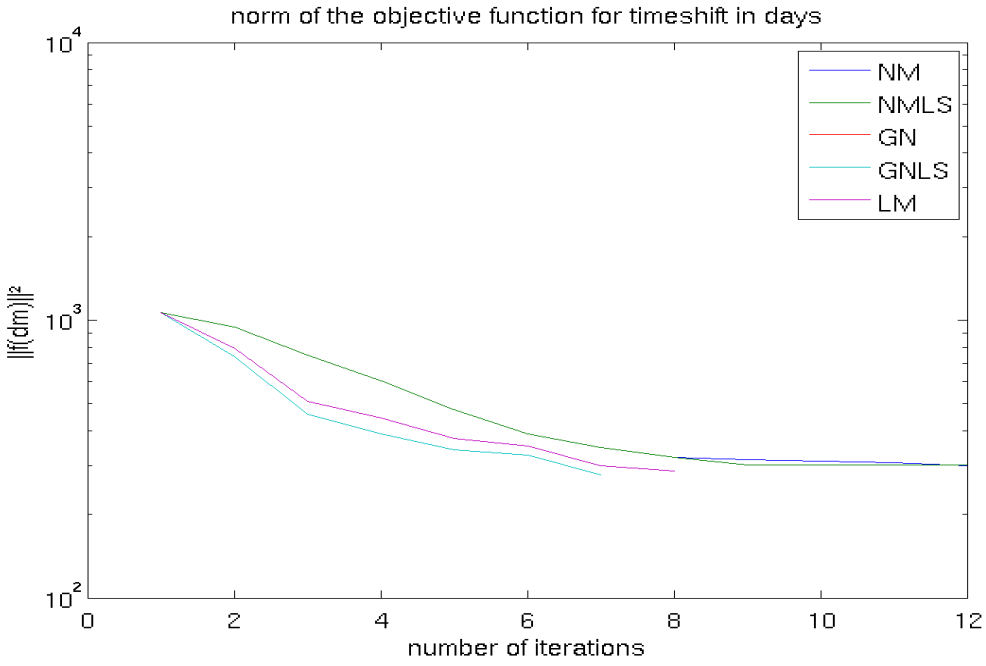

Figure 6.28: Norm of objective function of time-shift in Case 5

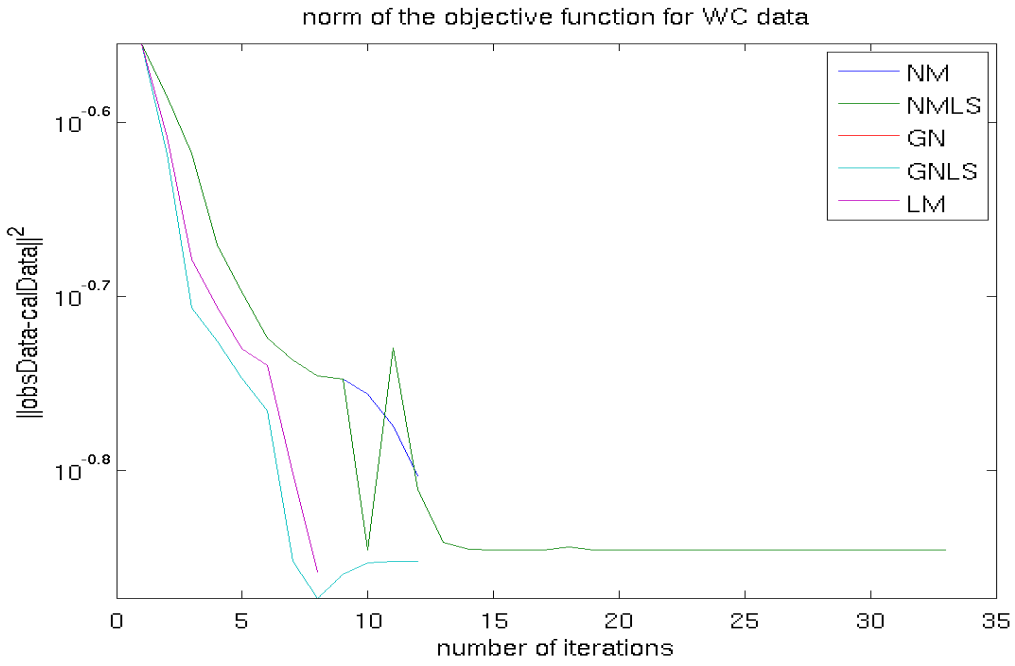

Figure 6.29: Norm of the objective function of water cut data in Case 5

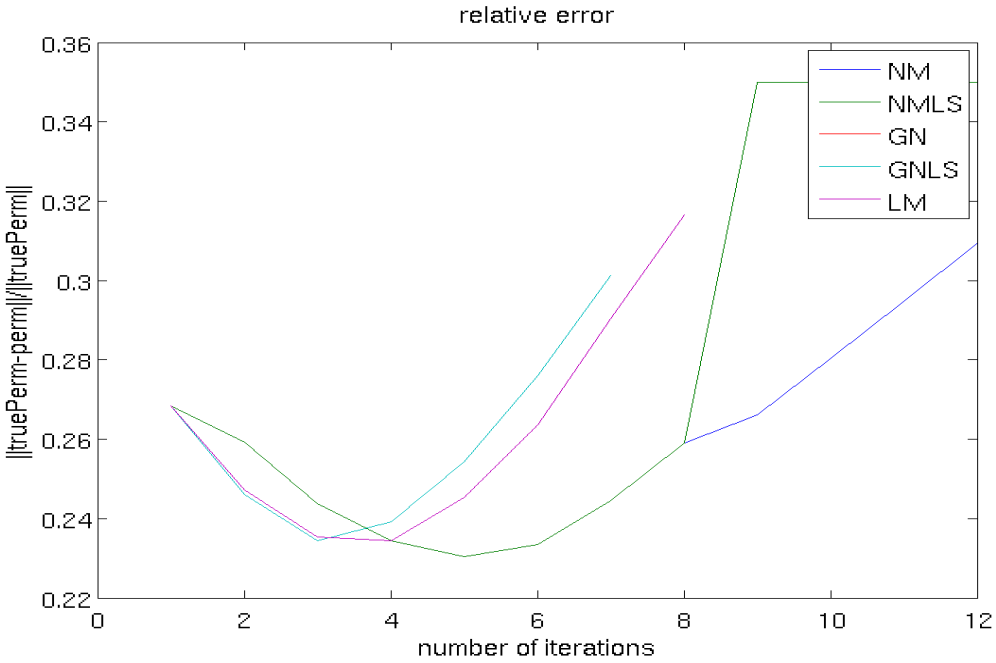

Figure 6.30: Norm of relative error in Case 5

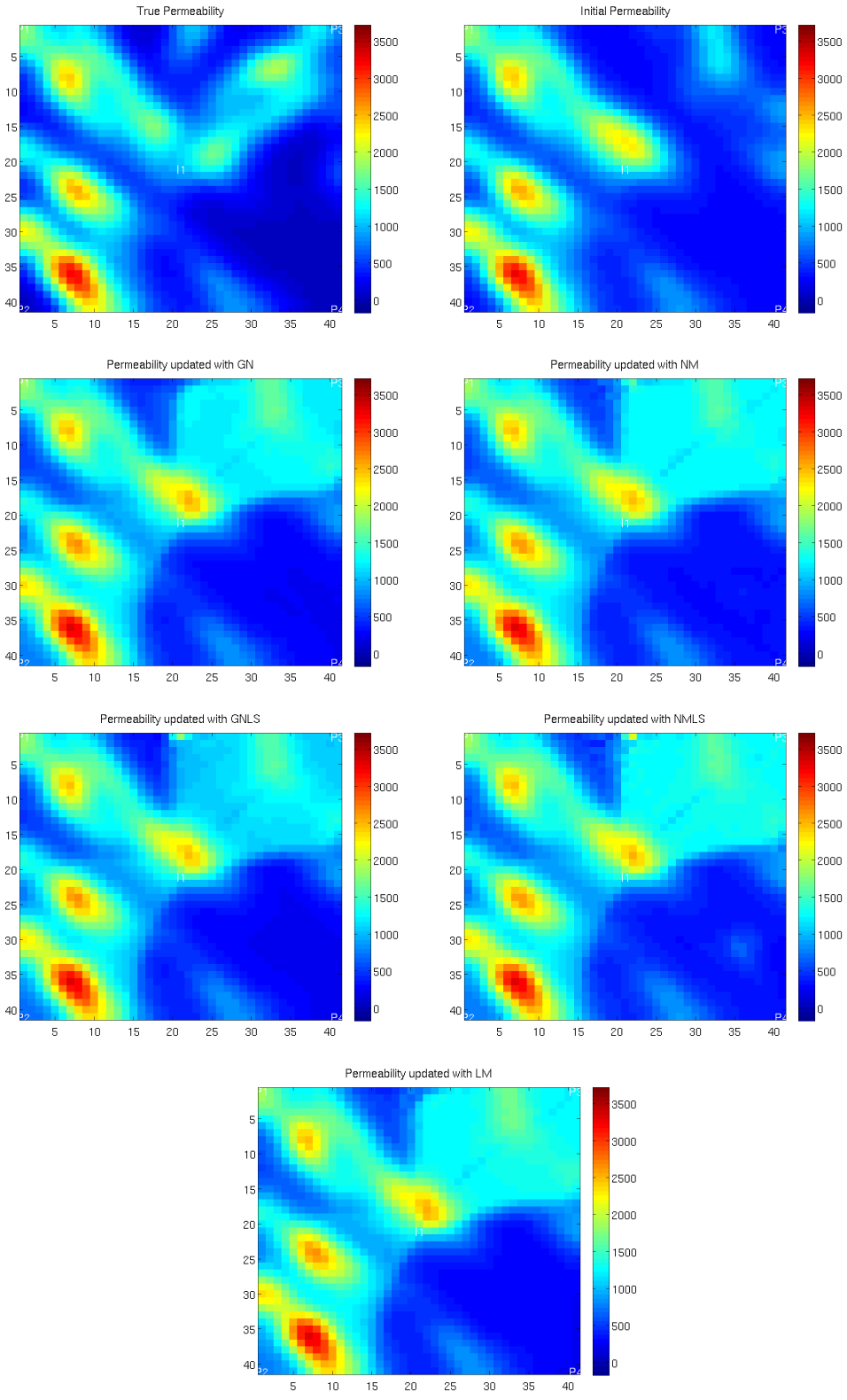

Figure 6.31: Updated permeability for GN (top left), NM (top right), GNLS (middle left), NMLS (middle right) and LM (bottom center) in Case 5

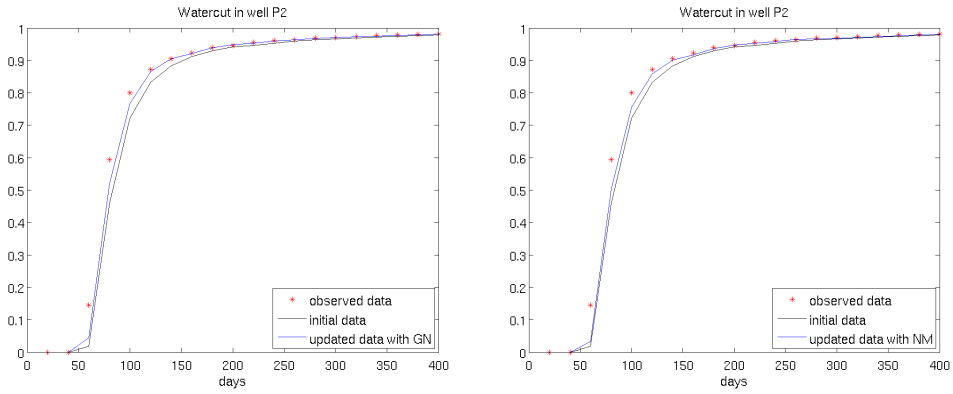

Figure 6.32: Observe(*), initial (black) and updated (blue) water cut in well P2 for GN (left) and NM (right) in Case 5.

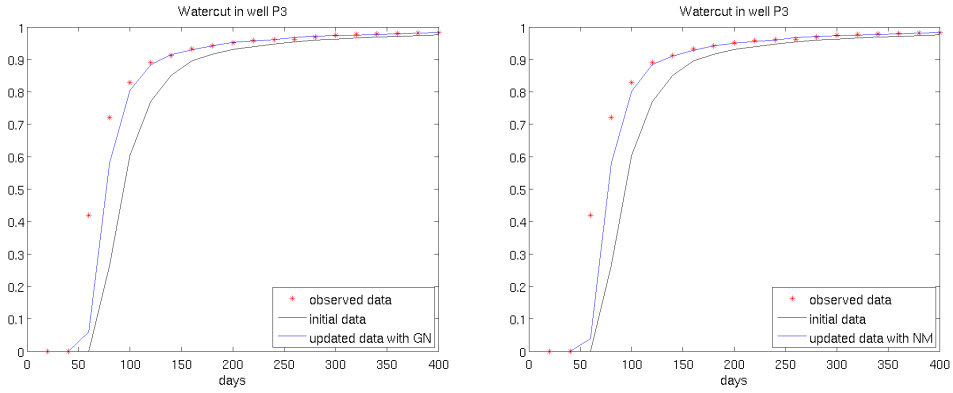

Figure 6.33: Observe(*), initial (black) and updated (blue) water cut in well P3 for GN (left) and NM (right) in Case 5.

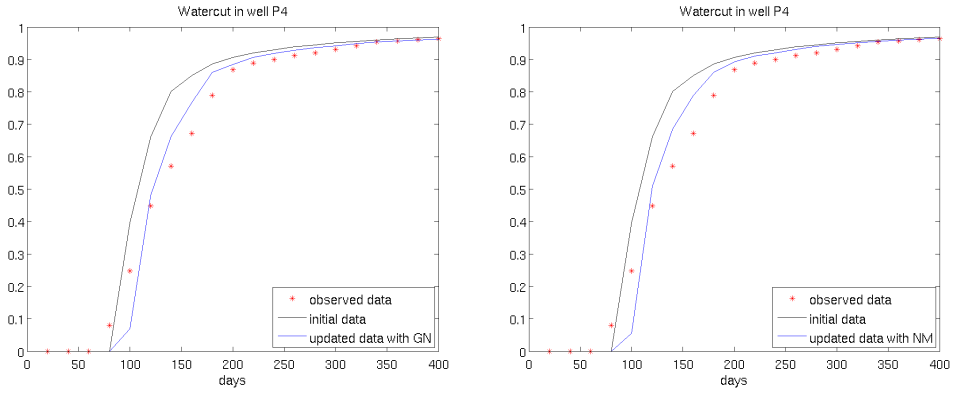

Figure 6.34: Observe(*), initial (black) and updated (blue) water cut in well P4 for GN (left) and NM (right) in Case 5.

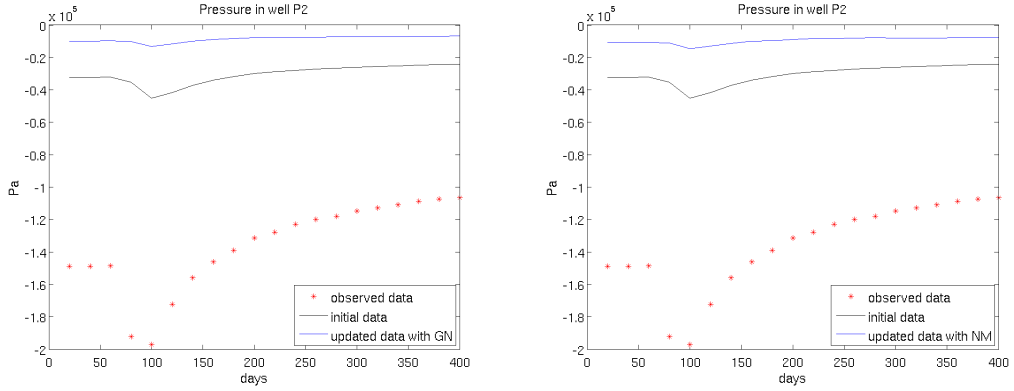

Figure 6.35: Observe(*), initial (black) and updated (blue) pressure in well P2 for GN (left) and NM (right) in Case 5.

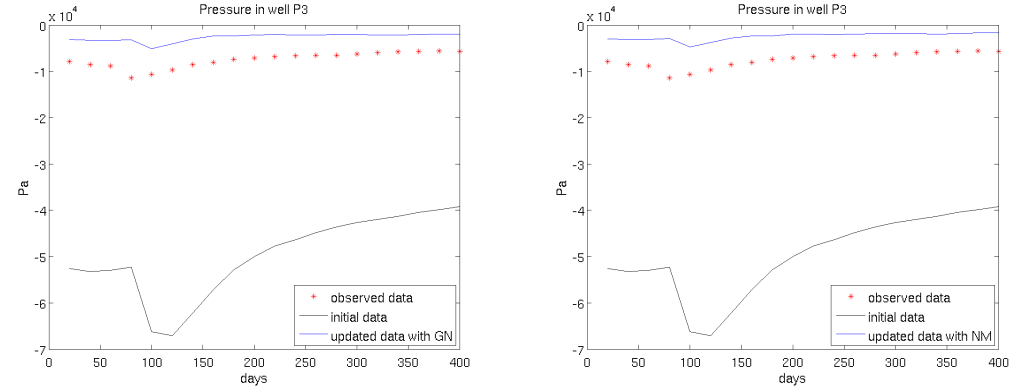

Figure 6.36: Observe(*), initial (black) and updated (blue) pressure in well P3 for GN (left) and NM (right) in Case 5.

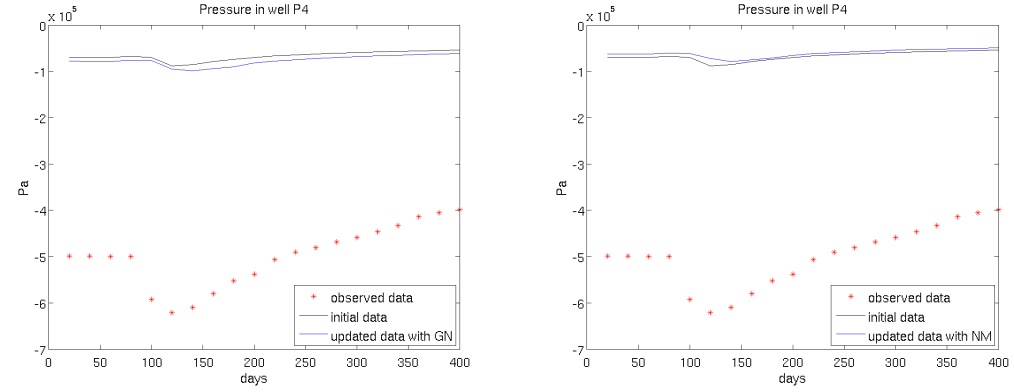

Figure 6.37: Observe(*), initial (black) and updated (blue) pressure in well P4 for GN (left) and NM (right) in Case 5.

6.4 Summary and comments

The goal of history matching is to include dynamic data in the reservoir characterization. Streamline models are well suited for history matching for two reasons. The first reason is the speed and the second is the fact that sensitivities can be calculated along the streamline analytically. Using the generalized travel-time inversion water cut data can be used to characterize the reservoir model. The quasi-linearity of the generalized travel-time inversion makes the method more suitable for integrating dynamic data in the models than the more traditional amplitude matching. Generalized travel-time inversion solves the history matching problem in two steps. First it finds the time-shift that minimizes the production data. The second step is to find the reservoir parameters that best match the time-shift. This minimization problem is nonlinear and in general ill-posed. Different methods and strategies is used to solve the minimization problem. Newton's method, where second order information of the sensitivities is included, shows good performance on problems where no regularization is required. But as the history matching problem gets more ill-posed such that regularization is needed, the improved performance of Newton's method compared to the original Gauss-Newton method is less obvious. For Newton's method to converge the Hessian must be positive definite, to assure positive definiteness, a small and positive term is added to the diagonal. For ill-posed problems regularization will add a similar term to the diagonal. These terms will bias the direction of the iterations toward the steepest decent direction which has a slower convergence rate than the Newton direction. The second order information of the sensitivities added in the Newton algorithms will therefore not contribute significantly to the reduction of the objective function for highly ill-posed problems.

Also different line-search and trust-region strategies are suggested to speed up and better the convergence ability of the methods. The line-search strategy is able to reduce the objective function significantly. The trust-region strategy in Levenberg-Marquardt also showed promising performance both in the reduction of the objective function and in characterizing the true permeability field.

In theory a better approximation of the Hessian gives a faster reduction of the objective function, but in practice the ill-posedness of the history matching problem makes the second order information of the sensitivities less relevant compared to the regularization terms that needs to be added to the diagonal of the Hessian to obtain a stable solution.

Chapter 7

Summary, recommendations and further work

As far as the laws of mathematics refer to reality, they are not certain;
and as far as they are certain, they do not refer to reality.
Albert Einstein

The main objective of this master thesis has been to investigate the streamline based history matching procedure developed by Datta-Gupta and others. Some of their work is gathered in [2]. The two main focus of the investigation has been the streamline based sensitivity calculation and the minimization of the objective function based on the time-shift in the generalized travel-time method.

The streamline based calculation of sensitivities is optimal in the sense that semi-analytic expressions of sensitivities of reservoir responses can be obtained as long as they can be related to the time-of-flight variable. With semi-analytic expressions these sensitivities can be calculated at practically no additional cost. This is a huge advantage since calculation of the sensitivities usually are the most time consuming part of the inversion algorithms. An important assumption underlying all streamline based calculation of sensitivities is that streamlines are assumed to not shift during perturbation. For small perturbations this assumptions holds. The streamline based expressions therefore gives an approximation of the sensitivities. Another advantage with the streamline based approximation of the sensitivities is that second order sensitivities can be found along the streamlines. These second order sensitivities can also be used to improve the minimization of the objective function in the history matching procedure.

The standard way to minimize the objective function based on the time-shift is to solve a reformulated version of the Gauss-Newton method with LSQR. This method only uses first order information of the sensitivities. Since second order sensitivities can be found at practically no additional cost, a second order method is suggested to improve the convergence speed of the inversion algorithm. Examples shows that for cases where no regularization is used, Newton's method shows promising performance, but as the problems becomes more ill-posed, the standard first order method shows approximately the same performance. Newton's method has slow convergence in early iterations since a small and positive term is added to the diagonal of the Hessian to make sure that the Hessian is positive definite. These added terms, bias the search direction towards the steepest decent direction given in, which has a slower convergence rate than the Gauss-Newton method.

To better the convergence properties different additional strategies where suggested. The line-search strategy is able to reduce the objective function considerably. But for more ill-posed problems where regularization is used, the line-search strategy is not able to characterize the reservoir very well.

The trust-region strategy also shows promising results. There are two reasons for the good performance of Levenberg-Marquardt. First of all Levenberg-Marquardt is a stable and a robust

method. For well-posed problems large steps are taken, but as the problem becomes more ill-posed the trust-region is extended such that smaller steps are used. The result is a method which is as fast as possible, and still is able to handle ill-posed problems. The second reason for the good performance of Levenberg-Marquardt is that the problem is scaled. Scaling both better the condition of the linear system and also weights the regularization parameters such that a better control of the effect of the regularization is possible.

The L-curve criterion for an automatic determination of the regularization parameter is also tested. The idea was that with an automatic determined regularization parameter, the regularization parameter is reduced as the iterate moves closer to the solution. The reason for suggesting this strategy is that the objective function is expected to be more linear as the iterate moves closer to the solution. Therefore the need of regularization is less in the neighborhood of the solution. In Case 1, regularization with parameters determined by the L-curve criteria shows good results, but as the problems gets more ill-posed the method is not able to show the same good performance.

7.1 Recommendations and further work

To extend the utilization of streamline based history matching the streamline simulator needs to be further developed to be able to simulate more realistic reservoirs. Extensions to include pressure controlled wells, complex cell geometries and multi-phase flow is considered in the textbook by Datta-Gupta and King [2]. The textbook also shows how to include effects of capillary forces and compressibility of fluids to the streamline simulator.

In this work only water cut data has been used for history matching. To include other types of production data, analytic expressions of the sensitivities of these production data is needed. Expressions for oil/gas ratio and tracer data can be found in [2].

Pressure data propagates with an infinite speed. This is a huge benefit with pressure data compared to production data. To include pressure data in streamline based history matching has therefore been a major area of studies these past years. One way to include pressure data is to define a diffusive time-of-flight. The arrival time of the pressure interference can then be related to the diffusive time-of-flight. Some of this work is summarized in [2].

Permeability is the only reservoir parameter that is estimated in this thesis. To be able to update other reservoir properties they have to be related to the time-of-flight variable. Semi-analytic expressions for the derivatives of the reservoir responses with respect to porosity, relative permeability, mobility and well rates is given in [15].

The various examples cases shows the difficulty of solving inverse problems and in particularly under determined problems. Regularization is needed to stabilize the solution, but the regularization will bias the solution. The regularization parameters are often given by the uncertainty of the data and the model in history matching problems. These regularization terms are fixed during the iterations. For faster convergence these regularization terms can be adjusted during the iterations. The automatic L-curve criterion for finding the regularization parameters is suggested. Only Tikhonovs regularization is considered in this thesis. Other regularization methods exist. If the reservoir parameters are reparametrized by a linear expansion of basis functions, the numbers of degrees of freedom may be reduced so that the problem becomes well-posed. Regularization with a linear expansion of basis functions will not bias the search direction of the algorithm towards the steepest decent direction and will therefore not slow down the convergence. A combination of Newton's method with regularization based on reparametrization may therefore give good results.

Another suggestion for further work is to include a trust-region strategy to the Newton's method. An advantage of the trust-region method is that the Hessian do not need to be positive definite for the method to converge. Instead the trust-region radius can be constructed such that the modified Hessian always is positive definite [11].

Since the objective of these test cases is to compare the performance of different minimization techniques, no error is assumed in the dynamic data. Dynamic data used in history matching problems always contains errors. A study of the effect of adding noise to the dynamic data is therefore important. If error is added to the dynamic data the objective function is not expected

to converge to zero. The residual will therefore not be equal to zero in the solution. For large residual cases, the Hessian approximation of Gauss-Newton is not valid. The Newton method is therefore expected to show better performance for large residual cases.

One of the advantages of streamline simulation is that a fine grid with millions of parameters can be used in the simulation. Since the focus of this thesis has been to test different minimization techniques, only relatively coarse grids are used. With a coarse grid the properties of each minimization method is more visible, still for a final validation of the methods a full scale field test with real production data is recommended.

The final conclusion is that even thought additional strategies to better the convergence properties of the minimization methods shows promising performance on test cases where no regularization is required, the complexity of a general history matching problem, unable the additional strategies to improve the performance of the method for more realistic and ill-posed problems.

Appendix A

A multi-scale mixed finite element formulation(MsMFEM)

The are several reasons for using a MsMFEM formulation. First MsMFEM has a finite element formulation. Finite element methods are optimal for solving elliptic equations in complex domains, because of the flexibility of the finite elements. Also important physical principle as mass balances can be guaranteed locally by the special formulation of these finite elements. An introduction to finite element theory can be found in [3]. The reason for using a multi-scale method is to be able to solve the pressure equation on a course grid, without losing fine scale geological information. The fine scale information of the velocity field is represented by different basis functions.

To obtain the mixed variational formulation of (2.26) and (2.28), we formally multiply the first equation (2.26) with a vector function $\mathbf{w} \in U$ and the second (2.28) with a function $l \in V$. Then we integrate each of the equations over the domain $\Omega \in \mathbb{R}^d$ where d is the dimension:

$$\int_\Omega l \nabla \cdot \mathbf{u} \mathrm{dx} = \int_\Omega q l \mathrm{dx}, \qquad \forall l \in V,$$

$$\int_\Omega \mathbf{u} \cdot (\lambda_t K)^{-1} \mathbf{w} \mathrm{dx} - \int_\Omega \nabla p \cdot \mathbf{w} \mathrm{dx} = 0, \qquad \forall \mathbf{w} \ \in U.$$

The functional spaces U and V are representing solution spaces for velocity and pressure, respectively. For no flow across the boundary $\partial\Omega$ and incompressible fluid, U is chosen to be the following Hilbert space:

$$H_0^{\mathrm{div}}(\Omega) = \{\mathbf{u} : \mathbf{u} \in (L^2(\Omega))^d, \nabla \cdot \mathbf{u} \in L^2(\Omega) \text{ and } \mathbf{u} \cdot \mathbf{n} = 0 \text{ on } \partial\Omega\},$$

where the function space $L^2(\Omega)$ consist of all square integrable function over Ω in the sense of Lebesgue. The solution space $V = L^2(\Omega)$. By choosing $\mathbf{\Psi_i}$ as bases for space U, an approximation of the velocity can be formulated in terms of these basis functions, that is: $\mathbf{u} = \sum \hat{u}_i \mathbf{\Psi_i}$. Similarly $p = \sum \hat{p}_k \Phi_k$, for bases Φ_k for pressure space V. The coefficients $\hat{u}_i$ and $\hat{p}_k$ can be found be solving the following linear system.

$$\begin{bmatrix} \mathbf{B} & \mathbf{C} \\ \mathbf{C}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{u}} \\ -\hat{\mathbf{p}} \end{bmatrix} = \begin{bmatrix} \mathbf{o} \\ \mathbf{q} \end{bmatrix}$$

where $\mathbf{B} = b_{ij}$, $\mathbf{C} = c_{ij}$ and $\mathbf{q} = q_k$ are given by:

$$b_{ij} = \int_\Omega \Psi_i (\lambda_t K)^{-1} \Psi_j \mathrm{dx},$$

$$c_{ij} = \int_\Omega \Phi_k \nabla \cdot \Psi_j \mathrm{dx},$$

$$q_k = \int_\Omega \Phi_k q \mathrm{dx}.$$

To formulate a multi-scale method, a fine and course grid partitioning of Ω, K_m and T_i, respectively, must be defined. The pressure space can then be approximated by piecewise constant functions. Thus $\hat{V} = \mathrm{span}(\Phi_i)$ where

$$\Phi_i(x) = \begin{cases} 1, & x \in T_i, \\ 0, & \text{otherwise.} \end{cases}$$

In contrast to the standard finite-element formulations, where also the velocity space is approximated by some low-order piecewise polynomials, in this multi-scale method the basis functions $\{\Psi_i\}$ are adaptive to the local properties of the pressure equation. The bases $\Psi_{ij} = \Psi_i \Psi_j$, for the interface $\Gamma_{ij} = \partial T_i \bigcap T_j$, can then be obtained by solving the pressure equation on all sub domains $\Omega_{ij} = T_i \cup \Gamma_{ij} \cup T_j$:

$$\Psi_{ij} = \lambda_t K \nabla \Phi_{ij}, \quad \nabla \cdot \Psi_{ij} = \begin{cases} w_i(x), & x \in T_i, \\ -w_j(x), & x \in T_j, \end{cases}$$

with $\Psi \cdot \mathbf{n} = 0$ on $\partial\Omega_{ij}$, where $\mathbf{n}$ is the outward unit normal to $\partial\Omega_{ij}$. The total mobility $\lambda_t = \lambda_t(S)$ is given on the underlying fine grid K_m. The weight function $w_i(x)$ is a normalized source distribution function, and it can be defined in many ways. Aarnes suggest the following definition:

$$w_i(x) = \begin{cases} q(x) / \int_{T_i} q(\xi) \mathrm{d}\xi & \text{if } \int_{T_i} q(x) dx \neq 0, \\ \mathrm{trace}(K(x)) / (d \int_{T_i} \sigma(\xi) \mathrm{d}\xi), & \text{otherwise.} \end{cases}$$

, where d is the number of space dimensions. With this choice of weight function, the high-permeable directions are weighted more than low permeable directions. According to numerical testing done by Aarnes this gives good results also for non-heterogeneous permeability field.

Nomenclature

α	fluid phase
α	regularization parameter
Δt_k	time-shift at well k
δ, ϵ	small positive numbers
Δ_k	trust-region radius
Γ	the path of the streamline
γ	step length
$\kappa(\mathbf{A})$	condition number of a matrix $\mathbf{A}$ defined in (4.4)
λ	trust-region parameter
λ_α	mobility of phase α defined in (2.16)
μ	viscosity in centipose
ϕ	porosity defined in (2.3)
Ψ	Streamfunction in 2D defined in (3.5)
ψ, χ	Bi-Streamfunction in 3D defined in (3.7)
ρ	density in kg/m^3
σ	shockspeed defined in (3.22)
τ	time-of-flight defined in (3.8)
ξ	arc-length
C	tracer concentration
d_j^{cal}	calculated data j
d_j^{obs}	observed data j
F	fractional flow of water defined in (2.24)
f	objective function
g	forward operator
$i, \imath$	cell indexes
k	well index

k_w	relative permeability for the wetting phase defined in (2.14)
k_{nw}	relative permeability for the non-wetting phase defined in (2.15)
m	size of model space
n	size of data space
N_d	number of data points
N_w	number of wells
N_{sl}	number of streamlines
p	pressure in Pa
P_c	capillary pressure
q	flow rate from source or sink in m^2/day
S	saturation of water defined in (2.11)
$s(\mathbf{x})$	slowness function defined in (5.2)
t	time
t_k^{cal}	calculated reference time at well k
t_k^{obs}	observed reference time at well k
w_j^{obs}	weight of data j
$\mathbf{A}^T$	transposed of matrix $\mathbf{A}$
$\mathbf{C_D}$	data covariance matrix
$\mathbf{C_M}$	model covariance matrix
$\mathbf{d}$	data vector
$\mathbf{D}^k$	diagonal scaling matrix
$\mathbf{G}$	linear forward operator
$\mathbf{g}$	gravitational acceleration constant
$\mathbf{H}$	Hessian of $\mathbf{g}(\mathbf{m})$
$\mathbf{I}$	identity matrix
$\mathbf{J}$	Jacobian of $\mathbf{g}(\mathbf{m})$ defined in (4.14)
$\mathbf{K}$	permeability in millidarcy (mD)
$\mathbf{L}$	finite difference approximation of the derivative
$\mathbf{m_p}$	prior model
$\mathbf{m}$	model parameters
$\mathbf{r}$	residual vector
$\mathbf{u}$	the Darcy velocity in m/day
$\mathbf{x}$	position vector

Bibliography

[1] Jacob Bear. *Dynamics of fluids in porous media.* American Elseviser Publishing Company, Inc, 1972. ISBN 0-444-00114-X.

[2] Akhil Datta-Gupta and Michael J.King. *Streamline Simulation: Theory and Practice.* Society of Petrolem Engineers, 2007. ISBN 978-1-55563-111-6.

[3] D.Braess. *Finite elements.* Cambridge U. Press, 1997. ISBN 9780521588348.

[4] Lawrence C. Evans. *Partial Differential Equations.* American Mathematical Society, 1997. ISBN 0-8218-0772-2.

[5] Per Chrisian Hansen and Dianne Prost O'Leary. The use of the l-curve in the regularization of discrete ill-posed problems. *SIAM J.SCI COMPUT.*, 1993.

[6] Akhil Datta-Gupta Hao Chang and Zhong He. A comparison of travel-time and amplitude matching for field-scale production-data integration: Sensitivity,nonlinearity, and practical implication. *SPE 84570*, 2005.

[7] Zhong He Hao Chang, Arun Kharghoria and Akhil Datta-Gupta. Fast history matching of finite-difference models using streamline-derived sensitivities. *SPE 89447*, 2005.

[8] Helge Holden and Nils Henrik Risebro. *Front Tracking for Hyperbolic Conervation Laws.* Springer, 2002. ISBN 3-540-43289-2.

[9] V.Kippe J.E Aarnes and K.-A.Lie. A hierarchical multiscale method for two-phase flow based upon mixed finite elements and nonuniform coarse grids. *Multiscale Modeling and Simulation*, 2006.

[10] Randall J. LeVeque. Finite Difference Methods for Differential Equations, 1998.

[11] Jorge Nocedal and Stephen J. Wright. *Numerical optimization.* Springer, 1999. ISBN 0-387-98793-2.

[12] Christofher C Paige and Micheal A Sauders. Lsqr: An algorithm for spare linear equations and spares least squares. *ACM Transactions on Mathematical Software*, 1982.

[13] Brian Borchers Richard C. Aster and Clifford H. Thurber. *Parameter Estimation and Inverse Problems.* American Elseviser Publishing Company, Inc, 2004. ISBN 0-12-065604-3.

[14] Lloyd N. Trefethen and David Bau. *Numerical Linear Algebra.* Society for Industrial and Applied Mathematics, 1997. ISBN 0-89871-361-7.

[15] V.R.Stenerud. *Streamline-based history matching.* PhD thesis, Norwegian University of Science and Technology, 2007.

[16] K.-A.Lie V.R.Stenerud, V.Kippe and A.Datta-Gupta. Adaptive Multiscale Streamline Simulation and Inversion for High-Resolution Geomodels. *Society of Petroleum Engineers*, 2007.

[17] K.-A.Lie V.R.Stenerud, V.Kippe and A.Datta-Gupta. Multiscale-Streamline Simulation and dynamic data integration for High-Resolution subsurface models. 2007.

[18] Fengjun Zhang and Albert C. Reynolds. Optimization algorithms for automatic history matching of production data. *E48*, 2002.

[19] Guanren Huan Zhangxin Chen and Yuanle Ma. *Computational Methods for Multiphase Flows in Porous Media*. Society for Industrial and Applied Mathematics, 2006. ISBN 0-89871-606-3.

[20] Akhil Datta-Gupta Zhong He and Seongsik Yoon. Streamline-based production data integration under changing field condition. *SPE 71333*, 2001.