

Data assimilation in reservoir management

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. dr. ir. J.T. Fokkema,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op maandag 19 januari 2009 om 15:00 uur

door

Joris Rombout ROMMELSE

ingenieur in de technische wiskunde
geboren te Haarlem

Dit proefschrift is goedgekeurd door de promotoren:

Prof. dr. ir. A.W. Heemink

Prof. dr. ir. J.D. Jansen

Samenstelling promotiecommissie:

Rector Magnificus,

voorzitter

Prof. dr. ir. J.D. Jansen,

Technische Universiteit Delft and

Shell International E&P, promotor

Prof. dr. ir. A.W. Heemink,

Technische Universiteit Delft, promotor

Prof. dr. A.C. Reynolds,

The University of Tulsa

Prof. dr. D.B. McLaughlin,

Massachusetts Institute of Technology

Prof. dr. ir. P.M.J. Van den Hof,

Technische Universiteit Delft

Prof. ir. C.P.J.W. Van Kruijsdijk,

Technische Universiteit Delft and Shell Canada

Dr. ir. C.B.M. Te Stroet,

TNO - NITG

The research for and printing of this thesis was financially supported by the Delphi consortium.

The printing of this thesis was financially supported by the J.E. Jurriaanse Stichting.

Cover design: Han Verhoeven

Paranymphs: Gijs Rommelse & Nanda Lambregts-Rommelse

An electronic version of this thesis is available from <http://www.library.tudelft.nl>.

ISBN: 978-90-8891-079-1

Keywords: petroleum reservoir management, history-matching, data assimilation, filters and variational methods, production and seismic data

Copyright © 2008 by J.R. Rommelse

All rights reserved. No part of the material protected by this copyright notice may be reproduced or utilised in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage and retrieval system, without written permission from the author.

Printed by Proefschriftmaken, Oosterwijk, The Netherlands

Contents

1	Introduction	1
1.1	The Exploration & Production industry	1
1.1.1	Energy and oil demand	1
1.1.2	Oil Field Development	3
1.1.3	Smart wells	3
1.1.4	Seismics	4
1.2	Closed-loop reservoir management	5
1.3	Research objectives and motivation	6
1.4	Thesis outline	7
2	Reservoir Simulation	9
2.1	Flow equations	9
2.1.1	From mass balance and Darcy to ODE	9
2.1.2	From ODE to PDE and back	12
2.1.3	Discretization in time	13
2.1.4	Well model	16
2.1.5	Simulating	17
2.1.6	More realistic physics	17
2.1.7	User input	18
2.1.8	Simulating in weak constraint or stochastic mode	19
2.2	Simulator sensitivities	21
2.2.1	Jacobians for non-linear solver	21
2.2.2	Influence of model parameters on system states	22
2.2.3	Jacobians for sensitivities	23
3	Data Assimilation	25
3.1	Introduction	25
3.2	Bayesian data assimilation	26
3.2.1	Bayes rule	26
3.2.2	Special cases	27
3.3	Variational data assimilation	28
3.3.1	Variational calculus	29
3.3.2	The reservoir simulator as a strong constraint	30

3.3.3	The reservoir simulator as a weak constraint	32
3.3.4	Regularization	32
3.3.5	Representer Method	35
3.3.6	Approximated gradient	38
3.4	Filtering	40
3.4.1	Classical Kalman filter for linear systems	40
3.4.2	Ensemble Kalman filter for non-linear systems	40
3.4.3	EnKF; towards a non-linear measurement update	44
3.4.4	Smoother	45
3.4.5	Adapting a filter to handle seismic data with time-correlated errors	46
3.4.6	Other low-order approximations	46
3.5	Hybrid method	47
3.5.1	Expectation maximization	47
3.5.2	Integrating a filter in a variational method	47
3.6	Relation between data assimilation methods	48

4	An efficient weak-constraint gradient-based parameter estimation algorithm using representer expansions¹	51
4.1	Introduction	52
4.1.1	Gradient-based parameter estimation	52
4.1.2	Model errors; strong and weak constraints	52
4.1.3	Notation	53
4.1.4	Representer Method	54
4.2	Gradient of the strong constraint minimization problem	55
4.2.1	Obtaining a gradient	55
4.2.2	Using the gradient	56
4.3	Gradient of the weak constraint minimization problem	56
4.3.1	Local minimizer	56
4.3.2	Basis functions	57
4.3.3	Representer equations	58
4.3.4	Representer Method as iterative minimizer	59
4.3.5	Obtaining a gradient	60
4.3.6	Computational issues in using the gradient	61
4.3.7	Choosing \mathbf{Q}	61
4.4	Numerical experiments: twin experiment	62
4.4.1	Inverted 5-spot	62
4.4.2	Reservoir simulator in weak or stochastic mode	62
4.4.3	Permeability reconstructed	66
4.4.4	Additional output from minimization process	71
4.4.5	Order reduction	71
4.5	Discussion	74

¹ This chapter is based on [Rommelse *et al.* (2007)], which was published as TUD-DIAM report 07-05 and submitted to SPE Journal

4.5.1	Strong constraint solver and the RM as post-processor	74
4.5.2	Variable time steps	74
4.5.3	Measure of success	75
4.5.4	Use of parameter representers to quantify the impact of measurements	75
4.5.5	Data selection	76
4.5.6	Regularization	76
4.5.7	Computational efficiency	76
4.6	Conclusion	77
4.6.1	Recapitulation	77
4.6.2	Conclusions	77
5	Comparison of the Ensemble Kalman Filter and a modified Representer Method for sensitivity to prior data²	81
5.1	Introduction	81
5.1.1	Strong and weak constraint reservoir simulation	82
5.1.2	Bayesian data-assimilation	83
5.2	Ensemble Filter	84
5.2.1	EnKF	84
5.2.2	ESRKF	85
5.2.3	Duplicated measurements	86
5.3	Representer Method	87
5.3.1	Objective function and derivatives	87
5.3.2	Local minimizer	89
5.3.3	Representer expansions	89
5.3.4	Representer equations	90
5.4	Numerical experiments	92
5.4.1	Correct prior and prior with exponentially decreasing correlation length	92
5.4.2	Filter results	94
5.4.3	Representer results	95
5.4.4	Alternative quantification of success	98
5.5	Conclusions	109
6	Variational estimation of permeability and model errors from 3D and 4D seismic data using model-driven regularization³	111
6.1	Introduction	112
6.1.1	Variational parameter estimation	112
6.1.2	Model errors; strong and weak constraints	113
6.1.3	Regularization	113
6.2	The VPERM method	113

² This chapter is based on [Rommelse *et al.* (2008a)], which was published as TUD-DIAM report 08-16 and submitted to Computational Geosciences

³ This chapter is based on [Rommelse *et al.* (2008b)], which was published as TUD-DIAM report 08-18 and submitted to Computational Geosciences

6.2.1	High dimensional gradient of the weak constraint problem	114
6.2.2	Low dimensional gradient of a strong constraint problem	115
6.2.3	Choosing the basis functions	116
6.2.4	Quadratic line search	117
6.3	The models	118
6.3.1	Weak constraint reservoir simulator	118
6.3.2	Rock-physics model	118
6.3.3	Measuring impedance with synthetic noise	120
6.4	Numerical experiments	121
6.5	Conclusions	133
7	Summary and conclusions	139
7.1	Summary	139
7.2	Conclusions	140
	Nomenclature	143
	Bibliography	145
A	Rock-physics derivatives	153
	Summary	155
	Samenvatting	157
	Acknowledgments	159
	About the author	161

Chapter 1

Introduction

1.1 The Exploration & Production industry

1.1.1 Energy and oil demand

Due to increasing prosperity, [EIA (2006)] predicts an increase in world energy consumption by 55% until 2030, Fig (1.1). Most of this energy will have to come from oil, gas and coal, Fig. (1.2). Sometimes it is suggested that there is not enough oil in the subsurface to meet this demand. This is true in terms of oil that is currently economically profitable. However, new technologies or increasing oil price, Fig. (1.3), make the concept "economically profitable" flexible. Due to this flexible definition, world oil reserves are still increasing, Fig. (1.4).

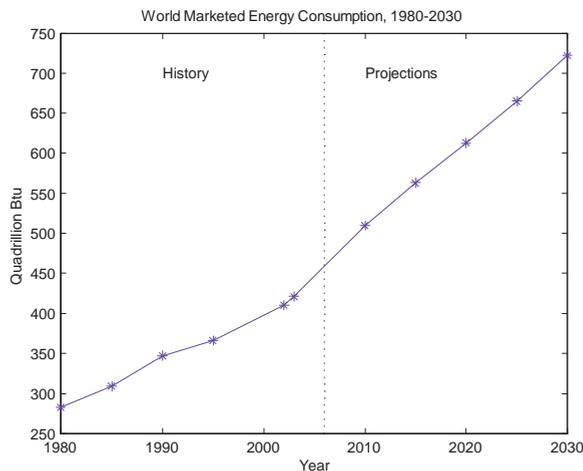


Figure 1.1: World energy consumption. Source: [EIA (2006)]

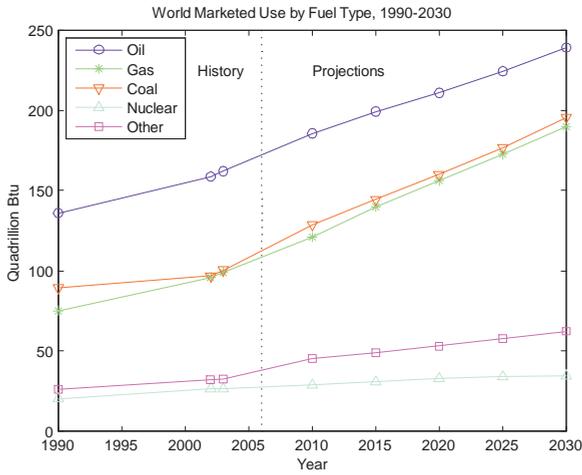


Figure 1.2: World energy use by fuel type. Source: [EIA (2006)]

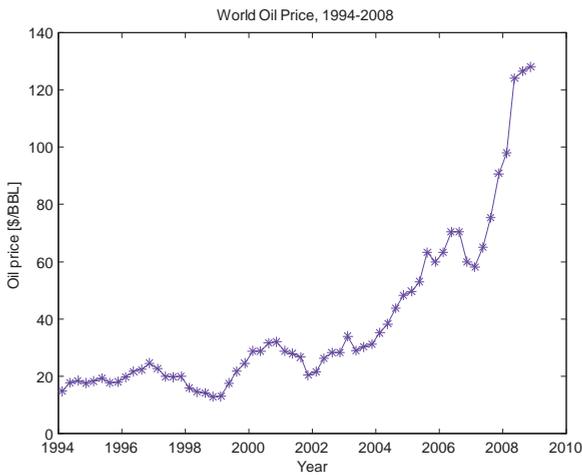


Figure 1.3: Oil price. Source: [EIA (2008)]

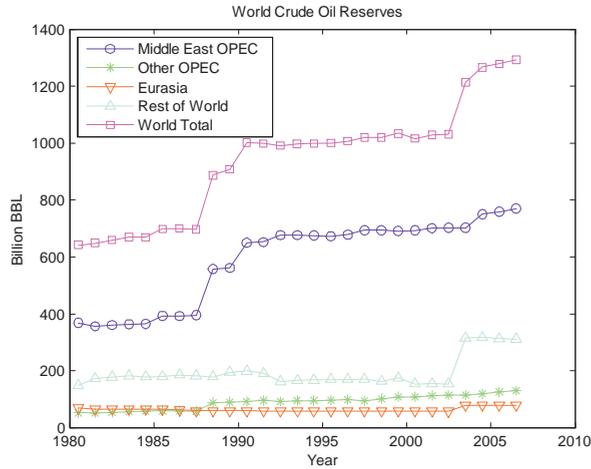


Figure 1.4: World crude oil reserves. Source: [EIA (2006)]

1.1.2 Oil Field Development

Conventional production from an oil field is categorized in three stages:

1. During primary recovery, the natural pressure of the reservoir pushes the hydrocarbons to the production wells. Wells can be stimulated by pumping or by gas lift. Also the flow through the reservoir near the wells can be improved by fracturing the near wellbore by injecting water.
2. As fluids are produced, the pressure in the reservoir decreases. In the secondary recovery stage, water or gas is (re)injected to push the hydrocarbons from the injection wells towards the production wells.
3. The tertiary recovery stage attempts to change the fluid properties. The viscosity of the oil can sometimes be decreased by injecting carbon dioxide or other gasses, or by heating the reservoir. Water may be blocked by turning it into gel with chemicals that are dissolved in injected oil.

1.1.3 Smart wells

Recently, new technology has been developed to better monitor and control the fluid flow through the reservoir. It has become possible to drill wells "around the corner" and to drill

wells off existing wells, creating a complex subsurface network of Smart Wells. Due to Inflow- or Interval Control Valves (ICV's) it is possible to shut-in a part of a well and still keep producing from deeper parts of the well. Down-hole sensors allow for nearly continuous monitoring of pressure, flow rate, fluid composition, temperature or even electro magnetic signals of wells and near-well areas.

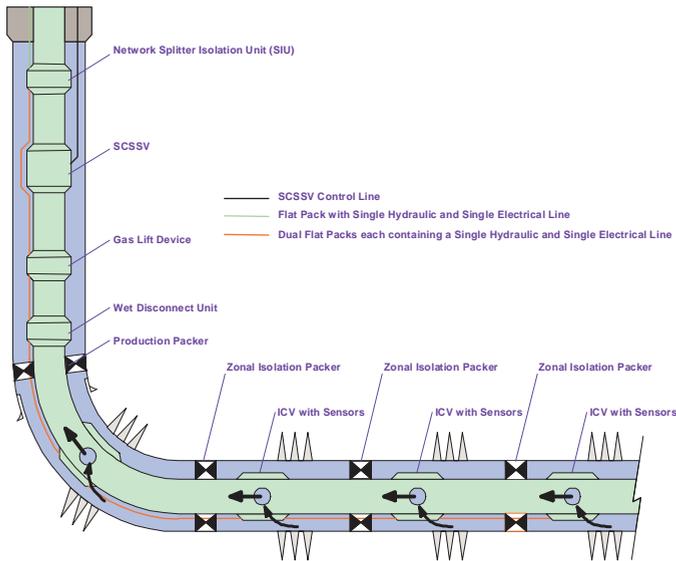


Figure 1.5: Smart Well

1.1.4 Seismics

Due to research in seismics and especially time-lapse seismics, it is now possible to track changes of fluid composition in the subsurface at locations away from the wells. The seismic process consists of three stages, acquiring seismic data, structural imaging and characterizing the subsurface [Berkhout (2004)]. In the acquisition stage, acoustic or elastic vibrations are generated by vibrator units, air guns or explosions. These vibrations travel through the earth as waves that get diffracted or reflected where the subsurface properties are discontinuous. The reflected waves are recorded at the surface. The second stage of seismic processing tries to reconstruct how these waves might have traveled through the subsurface from the data that were recorded at the surface. This gives information about the layering and the rock prop-

erties in the subsurface. This information can then be used in the reservoir characterization stage to better understand how water and hydrocarbons flow through the reservoir.

The research for this thesis touches the third stage of the seismic process; in chapter 6 synthetic seismic data are used in addition to production data to get a better understanding of two imaginary petroleum reservoirs.

1.2 Closed-loop reservoir management

The term closed loop reservoir management [Jansen *et al.* (2005)] is used when a data assimilation routine is present in addition to a reservoir simulator and an optimization routine (Fig. 1.6). Optimization can be done over the lifetime of the reservoir or on short-term. The optimal strategy can for example contain injection flow rates, bottomhole pressures in production wells or valve settings that maximize the Net Present Value. The optimal strategy can be applied to the reservoir simulator and to the real reservoir. Using a sensor model, it is possible to predict measurements. A discrepancy between the "predicted measurements" and the "measured measurements" may indicate that the simulator should be corrected. This is the aim of data assimilation. In state estimation, the output of the simulator, typically pressure and saturation values, is updated. Optionally the simulator's underlying parameters, like permeability or porosity, may be updated. This is called parameter estimation. Parameter estimation is more appropriate to reservoir management, since state estimation does not alter the simulator's ability to make future predictions. Traditionally the parameters of the reservoir simulator are history-matched a few times during the lifetime of the reservoir. However by integrating sensors in "smart" wells, it has become possible to have data available on a weekly or daily basis. When formulated in a general way, data assimilation algorithms provide a framework in which data from varying different sources, like sensors in "smart" wells or seismics, can together contribute to updating model parameters of different natures of uncertainty, like permeability, porosity or PVT data. "The loop is closed" when data assimilation routines are put in place to provide nearly continuous reservoir monitoring. From a systems engineering point of view, there are actually two closed loops when both an optimization routine and a data assimilation routine are manipulating the output of the reservoir simulator.

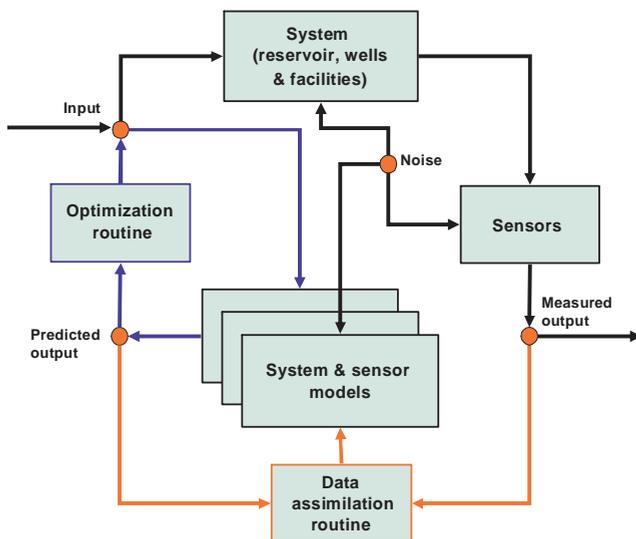


Figure 1.6: Closed-loop reservoir management

1.3 Research objectives and motivation

The objective of this thesis is formulated as:

Apply data assimilation techniques, invented and developed in other areas of research, to petroleum reservoir engineering, modify them to be better suited for their new application, and investigate how they can help to integrate both production data and seismic data to support decision-making in petroleum reservoir management.

The development of new hardware like smart wells and downhole sensors, opens up new possibilities to better produce hydrocarbon reservoirs. Smart wells allow for a much more flexible and dynamic way of operating facilities than conventional wells. However, without a better understanding of the subsurface, a smart well is just as "dumb" as a conventional well. New hardware allows for more flexibility in producing hydrocarbons from the reservoir, but new software is also needed to support the *decision-making* process. Improvements of sensors in the smart wells and advancements in seismic research produce large quantities of *production- and seismic data* that cannot just be used in conventional history-matching. Traditional history-matching is usually only performed at the re-development of a hydrocarbon reservoir; it does not take into account the uncertainties in the available data and involves applying ad hoc techniques. The availability of (much) higher frequency data makes it interesting to adopt *data assimilation techniques from other fields of research* like oceanography or atmospheric research to *petroleum reservoir engineering*. Applying data assimilation

techniques to a new area of research introduces new criteria to measure the performance of the data assimilation algorithms and the methods must be *modified for their new application*. With current technology, for example, we can not control the atmosphere or the oceans, but we do wish to manage subsurface fluid flows.

1.4 Thesis outline

This thesis was built around three articles. Chapter 4 proposes a modification of the Representer Method (RM) and shows the applicability by estimating permeability from production data. In chapter 5 the robustness of RM is compared with a modification of the Ensemble Kalman Filter (EnKF) with respect to errors in prior information. Chapter 6 introduces the VPERM method and illustrates the added value of assimilating seismic data in conjunction with production data. Since the articles were written to be published standalone in the open literature, they contain some overlap. They share, for example, similar introductions of basic variational algorithms and filters.

The articles are preceded by a general introduction, chapter 1, a discussion on petroleum reservoir simulation, chapter 2, and an introduction to data assimilation, chapter 3. Chapter 7 concludes the thesis.

Chapter 2

Reservoir Simulation

Section 2.1 discusses a numerical model that can be used to simulate the fluid flow through the pores of a porous reservoir rock in the subsurface. It is not meant to teach all the details about reservoir simulation; it only deals with physical processes that were considered in this research. A discussion on more complicated reservoir simulators is presented in section 2.1.6. Section 2.1.7 indicates what information a user must provide before the simulator can be run. Sections 2.1.6 and 2.1.7 indicate how simulators can produce wrong results and how much uncertainty must be dealt with by data assimilation methods.

When data are used to improve the reservoir simulator's forecasting capability, or when the simulator is used to calculate optimal production strategies, some methods rely on the sensitivity of the reservoir simulator's output with respect to certain model parameters or control parameters. Using numerical perturbations, as described in section 2.2, is an obvious, but computationally not very efficient way to calculate these. Analytically obtaining gradients in a way that is computationally feasible will be discussed later in section 3.3.

2.1 Flow equations

2.1.1 From mass balance and Darcy to ODE

No matter how many approximations are made to speed up a simulator or to ease its implementation, every reservoir simulator contains a mass balance principle and Darcy's law. If the reservoir is divided in arbitrarily shaped volumes or grid blocks, then the mass balance equation performs bookkeeping of the water and hydrocarbon masses in the fluid phases, and states that the accumulation of mass in grid block $\xi \in \aleph$ is equal to the mass that flows in from its neighboring grid blocks $\eta \in \aleph_\xi$ plus a source/sink term

$$\frac{\partial}{\partial t} \left(V^\xi \phi^\xi S_\alpha^\xi \rho_\alpha^\xi \right) = \sum_{\eta \in \aleph_\xi} Q_\alpha^{\xi\eta} \rho_\alpha^{\xi\eta} + q_\alpha^\xi, \quad (2.1)$$

where

- t [s]: time,
- \aleph [-]: set of all grid block indices,
- ξ [#]: index of grid block ξ ,
- V^ξ [m^3]: volume of grid block ξ ,
- ϕ^ξ [-]: porosity of grid block ξ ,

$S_\alpha^\xi [-]$: saturation of component/phase α in grid block ξ ,
 $\rho_\alpha^\xi [kg\ m^{-3}]$: density of component/phase α in grid block ξ ,
 $\eta [\#]$: index of neighbor of grid block ξ ,
 $\mathcal{N}_\xi [-]$: set of indices of neighbors of grid block ξ ,
 $Q_\alpha^{\xi\eta} [m^3\ s^{-1}]$: mass flux through $\xi\eta$ -interface,
 $\rho_\alpha^{\xi\eta} [kg\ m^{-3}]$: density of component/phase α on $\xi\eta$ -interface,
 $q_\alpha^\xi [kg\ s^{-1}]$: injection/production of component/phase α in grid block ξ .

Darcy's law

$$Q_\alpha^{\xi\eta} = \frac{k_\alpha^{\xi\eta} A^{\xi\eta} p_\alpha^\eta - p_\alpha^\xi}{\mu_\alpha^{\xi\eta} h^{\xi\eta}}, \quad (2.2)$$

relates these mass fluxes $Q_\alpha^{\xi\eta}$ through the $\xi\eta$ -interface with area $A^{\xi\eta} [m^2]$ to a pressure difference $p_\alpha^\eta - p_\alpha^\xi [Pa = kg\ m^{-1}\ s^{-2}]$ over distance $h^{\xi\eta} [m]$ using

$k_\alpha^{\xi\eta} [m^2]$: permeability of component/phase α on $\xi\eta$ -interface,
 $\mu_\alpha^{\xi\eta} [kg\ m^{-1}\ s^{-1}]$: viscosity of component/phase α on $\xi\eta$ -interface.

The permeability $k_\alpha^{\xi\eta}$ is usually split into a rock dependent permeability $k^{\xi\eta} [m^2]$ and a fluid dependent relative permeability $k_{r\alpha}^{\xi\eta} [-]$.

For a two-phase (water-oil) reservoir without gravity or capillary pressure, a simulator can be formulated as an ordinary differential equation (ODE)

$$\frac{d}{dt} (\mathbf{f}_1(\mathbf{x})) = \mathbf{f}_2(\mathbf{x}), \quad (2.3)$$

where the state vector $\mathbf{x} = \begin{bmatrix} \mathbf{p} \\ \mathbf{S} \end{bmatrix} = \begin{bmatrix} \{p^\xi\}_{\xi \in \mathcal{N}} \\ \{S_w^\xi\}_{\xi \in \mathcal{N}} \end{bmatrix}$ contains the pressures p^ξ and water saturations S_w^ξ of all grid blocks,

$$\mathbf{f}_1(\mathbf{x}) = \begin{bmatrix} \left\{ V^\xi \phi^\xi S^\xi \rho_w^\xi \right\}_{\xi \in \mathcal{N}} \\ \left\{ V^\xi \phi^\xi (1 - S^\xi) \rho_o^\xi \right\}_{\xi \in \mathcal{N}} \end{bmatrix} \quad (2.4)$$

and

$$\mathbf{f}_2(\mathbf{x}) = \begin{bmatrix} \left\{ q_w^\xi + \sum_{\eta \in \mathcal{N}_\xi} t_w^{\xi\eta} (p^\eta - p^\xi) \right\}_{\xi \in \mathcal{N}} \\ \left\{ q_o^\xi + \sum_{\eta \in \mathcal{N}_\xi} t_o^{\xi\eta} (p^\eta - p^\xi) \right\}_{\xi \in \mathcal{N}} \end{bmatrix}, \quad (2.5)$$

where the transmissibilities $t_w^{\xi\eta}$ and $t_o^{\xi\eta} [m\ s]$ are defined as

$$t_\alpha^{\xi\eta} = \frac{k^{\xi\eta} k_{r\alpha}^{\xi\eta} A^{\xi\eta} \rho_\alpha^{\xi\eta}}{\mu_\alpha^{\xi\eta} h^{\xi\eta}}. \quad (2.6)$$

The fluid-dependent part of the transmissibility is called the mobility, λ_α [$m^{-2} s$], and is defined as

$$\lambda_\alpha = \frac{k_{r\alpha} \rho_\alpha}{\mu_\alpha}. \quad (2.7)$$

Sometimes the rock-dependent permeability is included in the mobility, $\tilde{\lambda}_\alpha$ [s]:

$$\tilde{\lambda}_\alpha = \frac{k k_{r\alpha} \rho_\alpha}{\mu_\alpha}. \quad (2.8)$$

From grid block centers to interfaces

The state variables are usually defined at the grid block centers or are at least representative for the entire grid block. Evaluating \mathbf{f}_2 , Eq. (2.5), requires state-dependent fluid properties at the grid block interfaces. Two strategies exist; first the reservoir states are interpolated to the interfaces and then the fluid properties are calculated, or vice versa. Here the former method is used

$$t_\alpha^{\xi\eta} = t_\alpha^{\xi\eta} (p^{\xi\eta}, S^{\xi\eta}),$$

$$p^{\xi\eta} = \frac{p^\xi + p^\eta}{2}, \quad (2.9)$$

$$\begin{aligned} S^{\xi\eta} &= \begin{cases} S^\xi & \text{if } p^\xi > p^\eta \\ S^\eta & \text{if } p^\xi < p^\eta \\ \frac{S^\xi + S^\eta}{2} & \text{if } p^\xi = p^\eta \end{cases} = \\ &= S^\eta + (S^\xi - S^\eta) H(p^\xi - p^\eta) = \\ &= S^\xi + (S^\eta - S^\xi) H(p^\eta - p^\xi), \end{aligned} \quad (2.10)$$

with the Heaviside function $H(x)$ defined as

$$H(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x < 0 \\ \frac{1}{2} & \text{if } x = 0 \end{cases}. \quad (2.11)$$

Sometimes correlation of the permeability values at the grid block interfaces is modelled by defining permeability values at the grid block centers and interpolating them to the interfaces. Appropriate interpolation methods are under discussion [Plug *et al.* (2006)] and applicability of these methods depend on the configuration of the reservoir rock and the dimension of the fluid flow. Here the harmonic average is used

$$k^{\xi\eta} = \left(\frac{(k^\xi)^{-1} + (k^\eta)^{-1}}{2} \right)^{-1} = \frac{2k^\xi k^\eta}{k^\xi + k^\eta}. \quad (2.12)$$

2.1.2 From ODE to PDE and back

ODE to PDE

When the ODE

$$\frac{d}{dt} \left(V^\xi \phi^\xi S_\alpha^\xi \rho_\alpha^\xi \right) = \sum_{\eta \in \mathcal{N}_\xi} \frac{k_r^{\xi\eta} k_r^{\xi\eta} A^{\xi\eta} \rho_\alpha^{\xi\eta} p_\alpha^\eta - p_\alpha^\xi}{\mu_\alpha^{\xi\eta} h^{\xi\eta}} + q_\alpha^\xi, \quad (2.13)$$

is applied to an infinitely small volume $V = \Delta x \Delta y \Delta z$, then

$$\frac{d}{dt} (\phi S_\alpha \rho_\alpha) = \sum_{\gamma \in \{x^-, x^+, y^-, y^+, z^-, z^+\}} \left(\frac{k k_{r\alpha} \rho_\alpha}{\mu_\alpha} \frac{A}{V h} \right)_\gamma (p_\alpha^\gamma - p_\alpha) + \frac{q_\alpha}{V}, \quad (2.14)$$

where subscript γ refers to the interfaces of the small volume and superscript γ refers to neighboring small volumes, or equivalently

$$\frac{d}{dt} (\phi S_\alpha \rho_\alpha) = \sum_{\gamma \in \{x, y, z\}} \frac{\left(\frac{k k_{r\alpha} \rho_\alpha}{\mu_\alpha} \right)_{\gamma^+} \frac{p_\alpha^{\gamma^+} - p_\alpha}{\Delta \gamma} - \left(\frac{k k_{r\alpha} \rho_\alpha}{\mu_\alpha} \right)_{\gamma^-} \frac{p_\alpha - p_\alpha^{\gamma^-}}{\Delta \gamma}}{\Delta \gamma} + \frac{q_\alpha}{V}. \quad (2.15)$$

When Taylor series are substituted, then the partial differential equation (PDE)

$$\frac{\partial}{\partial t} (\phi S_\alpha \rho_\alpha) = \sum_{\gamma \in \{x, y, z\}} \frac{\partial}{\partial \gamma} \left(\frac{k k_{r\alpha} \rho_\alpha}{\mu_\alpha} \frac{\partial p_\alpha}{\partial \gamma} \right) + \tilde{q}_\alpha, \quad (2.16)$$

is obtained. Allowing for anisotropy in the permeability, the PDE becomes

$$\frac{\partial}{\partial t} (\phi S_\alpha \rho_\alpha) = \nabla \cdot \left(\mathbf{K} \frac{k_{r\alpha} \rho_\alpha}{\mu_\alpha} \nabla p_\alpha \right) + \tilde{q}_\alpha, \quad (2.17)$$

where \tilde{q}_α [$kg \ m^{-3} \ s^{-1}$] now denotes a mass injection/production density.

Discretization in space

Several methods can be used to discretize the PDE Eq. (2.17) in space and turn it back into an ODE. Eq. (2.3) is just one example, but different ODE's can be derived. The discretization methods can generally be categorized as finite difference [Strikwerda (2004)], finite volume [Versteeg and Malalasekra (1996)] or finite element methods [Zienkiewicz *et al.* (2005)]. In a finite difference discretization, the derivatives in the PDE are replaced by differences, which are usually calculated on a user-defined stencil or structured grid. Finite volume and finite element methods are more suitable for unstructured grids. In a finite element method, the state variables are defined at the vertices of the grid blocks. The user must specify as many basis functions as there are vertices. A solution is looked for as a weighted sum of these basis functions. In a finite volume method, the state variables are defined at the grid block cen-

ters and the ODE is obtained by integrating the PDE over every grid block. When the fluxes through the grid block interfaces are calculated from the state variables at the centers of the two adjacent grid blocks, the same ODE as in section 2.1.1 is obtained. However, improved performance, in particular reduced sensitivity to grid orientation, has been reported [Aavatsmark and Eigestad (2006)] when additional grid blocks are involved in the approximation of the fluxes.

Although the PDE is based on mass conservation, this property is not automatically preserved after spatial discretization. The ODE that follows after finite volume discretization is mass conservative by construction. More care must be taken when finite difference or finite element discretization is performed. Even when the ODE is mass conservative, this can still be destroyed by the time discretization or by the algebraic solver that solves the (non-linear) system that results after discretizing the ODE in time.

Needless to say that an ODE can be obtained by completely skipping section 2.1.2 altogether, as was done in this research. However, literature usually starts from a PDE formulation.

2.1.3 Discretization in time

No matter what spatial discretization is used, the time discretization is usually done by finite differences. Two choices have to be made; how many history states \mathbf{x} should be used to approximate $\frac{d}{dt}(\mathbf{f}_1(\mathbf{x}))$ of Eq. (2.3) and at what time(s) should $\mathbf{f}_2(\mathbf{x})$ be evaluated? Depending on these choices, totally different algebraic or numeric behavior can be expected from the simulator.

Euler; forward, backward and combinations

In an Euler discretization scheme, the new (or future) state \mathbf{x}_n and one history (or current) state \mathbf{x}_{n-1} is used to approximate the time derivatives. The result of the scheme, applied to Eq. (2.3), looks like

$$\frac{\mathbf{f}_1(\mathbf{x}_n) - \mathbf{f}_1(\mathbf{x}_{n-1})}{t_n - t_{n-1}} = (1 - \alpha) \mathbf{f}_2(\mathbf{x}_{n-1}) + \alpha \mathbf{f}_2(\mathbf{x}_n) \quad (2.18)$$

or

$$\frac{\mathbf{f}_1(\mathbf{x}_n) - \mathbf{f}_1(\mathbf{x}_{n-1})}{t_n - t_{n-1}} = \mathbf{f}_2((1 - \alpha) \mathbf{x}_{n-1} + \alpha \mathbf{x}_n), \quad (2.19)$$

where Euler forward ($\alpha = 0$) is fully explicit and Euler backward ($\alpha = 1$) is fully implicit. In case \mathbf{f}_1 is the identity function ($\mathbf{f}_1(\mathbf{x}) = \mathbf{x}$) and \mathbf{f}_2 is linear ($\mathbf{f}_2(\mathbf{x}) = \mathbf{A}\mathbf{x}$), Euler forward is stable when the time step $t_n - t_{n-1}$ is chosen small enough. Euler backward is unconditionally stable, but introduces more numerical diffusion, and is therefore less accurate. For arbitrary \mathbf{f}_1 and \mathbf{f}_2 , Euler backward is, in general, only conditionally stable, but the condition is less strict than for Euler forward. Therefore larger time steps can be taken at the expense of accuracy. The computational cost per time step is usually higher for Euler backward, since

the non-linearities \mathbf{f}_1 and \mathbf{f}_2 must be dealt with iteratively, possibly converging to the wrong solution. Euler forward only needs to deal with the non-linearities in \mathbf{f}_1 .

BDF

Due to the different time scales of the pressure- and saturation changes, the ODE may show stiff behavior [Hairer and Wanner (1996)]. Solving these kind of problems requires more advanced methods like Backward Differentiation Formulae (BDF). BDF are multistep extensions to Euler backward. For q steps, Eq. (2.3) is discretized as

$$\mathbf{f}_1(\mathbf{x}_n) = \sum_{i=1}^q \alpha_{n,i} \mathbf{f}_1(\mathbf{x}_{n-i}) + (t_n - t_{n-1}) \beta_n \mathbf{f}_2(\mathbf{x}_n). \quad (2.20)$$

The BDF coefficients are found by fitting a q -th order polynomial through \mathbf{f}_1 at the new state and the q previous states. The time derivative of \mathbf{f}_1 is then evaluated as the derivative of the polynomial, evaluated at t_n . For example, if $q = 1$, then the polynomial is

$$\mathbf{P}(t) = \frac{(t_n - t) \mathbf{f}_1(\mathbf{x}_{n-1}) + (t - t_{n-1}) \mathbf{f}_1(\mathbf{x}_n)}{t_n - t_{n-1}} \quad (2.21)$$

and its derivative

$$\mathbf{P}'(t) = \frac{\mathbf{f}_1(\mathbf{x}_n) - \mathbf{f}_1(\mathbf{x}_{n-1})}{t_n - t_{n-1}}, \quad (2.22)$$

which is the same as Euler backward, Eq. (2.18) with $\alpha = 1$. Values for the coefficients, for up to 5 steps, are shown in Tbl. (2.1) [Benner and Mena (2004) (one minus-sign incorrect)]; it is assumed that the step size is constant.

q	β	$\alpha_{n,1}$	$\alpha_{n,2}$	$\alpha_{n,3}$	$\alpha_{n,4}$	$\alpha_{n,5}$
1	1	1				
2	2/3	4/3	-1/3			
3	6/11	18/11	-9/11	2/11		
4	12/25	48/25	-36/25	16/25	-3/25	
5	60/137	300/137	-300/137	200/137	-75/137	+12/137

Table 2.1: BDF coefficients

BDF is the engine of the DASSL solver [Petzold (1983)], which turned out to be the fastest and most robust solver for the system described in section 2.1.1, out of a benchmark [Lioen and de Swart (1998)] of 6 solvers (DASSL, MEBDFDAE, PSIDE, RADAU, RADAU5, VODE).

IMPES

In an IMPES scheme (IMplicit Pressure Explicit Saturation), the ODE Eq. (2.3) is written as

$$\frac{d\mathbf{x}}{dt} = \mathbf{J}_{\mathbf{f}_1}^{-1} \left(\begin{bmatrix} \mathbf{T}_w & \mathbf{0} \\ \mathbf{T}_o & \mathbf{0} \end{bmatrix} \mathbf{x} + \mathbf{b} \right), \quad (2.23)$$

where $\mathbf{J}_{\mathbf{f}_1}^{-1}$ is the Jacobian of \mathbf{f}_1 with respect to \mathbf{x} and \mathbf{f}_2 is split into states and state-dependent fluid properties. When the state-dependent properties are evaluated explicitly and the states themselves are evaluated both implicitly (for pressures) and explicitly (for saturations), then a parabolic pressure equation and a hyperbolic saturation equation are obtained. The coefficients of the saturation equation depend on the solution of the pressure equation.

Although the implicit part of the IMPES equations is of much smaller dimension than the BDF equations, it must be solved (many) more times, since the time steps are linked to explicit equations and therefore (a lot) smaller. [Li *et al.* (2004)] claims that IMPES is not suitable for black-oil reservoir simulation. Modifications to IMPES have been made where the time step of the pressure equation is of significantly higher order than the time step of the saturation equation. The methods described in the following section take this a step further.

Streamlines

In a streamline simulator, the pressure equation is decoupled from the saturation equation(s) by an IMPES scheme. The solution from the pressure equation is used to make as many time steps for the saturation equation as possible [Datta-Gupta *et al.* (2001)]. Moreover, the saturation equation is not solved in the original 3D space. Instead, it is mapped onto streamlines, which are traced from the 3D pressure field, producing a set of 1D problems, all parameterized by a time-of-flight (TOF) coordinate. Modern streamline simulation rests on five key principles [Thiele (2001)]:

- tracing 3D streamlines and periodically updating them
- mapping of the mass conservation and Darcy equations onto streamlines
- solving 1D transport problems along streamlines
- operator splitting to account for gravity effects
- extension to compressible flow

[Thiele (2001)] also claims that "the speed and efficiency as well as the availability of new data make streamlines potentially the most significant tool for solving complex optimization problems related to history-matching and optimal well placements". This does not hold with the development of adjoint reservoir simulators, Eq. (3.19), which will be discussed in section 3.3.

2.1.4 Well model

Wells can be modelled by boundary conditions, but also by setting $q_w^\xi > 0$, $q_o^\xi = 0$ for injection or $q_w^\xi < 0$, $q_o^\xi < 0$ for production in Eq. (2.5) while enforcing no-flow boundary conditions. Usually wells can be operated under pressure or rate constraints. Too high pressures may cause damage to the reservoir (although in some applications this is in fact the goal) and surface facilities are only capable of handling a limited amount of production or injection fluids. The actual flow rate in a well is determined by the most stringent constraint. For example, the actual flow rate of a production well can firstly be limited by the amount of oil or water that can be processed by the surface facilities. As the pressure in the reservoir decreases, this may become the most stringent constraint. Therefore a switching mechanism must be built into a reservoir simulator.

Here only bottomhole pressure constraints and surface rate constraints are considered. The switching is left to the user, so only one type of constraint can be applied to a well at one point in time.

Rate constraint

In case the wellbore flow rate at the surface $Q [m^3 s^{-1}]$ is specified, the mass flow rate $q_\alpha^\xi [kg s^{-1}]$ to/from grid block ξ is calculated by dividing the mass over the grid volumes that are penetrated by the well, proportionally to the mobility Eq. (2.7), so

$$Q = \sum_\alpha \sum_\xi \frac{|q_\alpha^\xi|}{\rho_\alpha^{STC}}. \quad (2.24)$$

This means that

$$q_\alpha^\xi = -Q \rho_\alpha^{STC} \frac{\lambda_\alpha^\xi}{\lambda_t} \quad (2.25)$$

for production and

$$q_w^\xi = Q \rho_w^{STC} \sum_\alpha \frac{\lambda_\alpha^\xi}{\lambda_t}, \quad q_o^\xi = 0 \quad (2.26)$$

for injection, with the total mobility $\lambda_t [m^{-2} s]$ defined as

$$\lambda_t = \sum_\alpha \sum_\xi \lambda_\alpha^\xi. \quad (2.27)$$

Pressure constraint

The mass flow rate $q_\alpha^\xi [kg s^{-1}]$ to/from grid block ξ is proportional to the pressure difference between that grid block and the wellbore pressure $p_{wb} [kg m^{-1} s^{-2}]$ and the mobility [Peaceman (1977)]. For production of the α -phase, this is the mobility of that phase;

$$q_\alpha^\xi = \lambda_\alpha^\xi \omega^\xi (p_{wb}^\xi - p_\alpha^\xi), \quad (2.28)$$

for injection, it is the sum of the mobilities of all phases that have to be displaced from the grid block;

$$q_w^\xi = \left(\lambda_w^\xi + \lambda_o^\xi \right) \omega^\xi \left(p_{wb}^\xi - p_w^\xi \right) \quad , \quad q_o^\xi = 0. \quad (2.29)$$

The well factor or well index ω^ξ [m^3] is taken constant in this research. More realistic choices may depend on the direction of the well, the angle that is open to flow, an effective permeability, the well radius and a skin factor [Peaceman (1977), Wheeler (1988)].

2.1.5 Simulating

Performing one time step comes down to finding \mathbf{x}_n that satisfies

$$\mathbf{f}(\mathbf{x}_n, \mathbf{x}_{n-1}, \dots, \mathbf{x}_{n-q}, \boldsymbol{\theta}) = \mathbf{0}, \quad (2.30)$$

given q history states $\{\mathbf{x}_{n-1}, \dots, \mathbf{x}_{n-q}\}$ and a set of model parameters $\boldsymbol{\theta}$ (for example permeability values in all grid blocks). From Eq. (2.18), it follows that for Euler backward

$$\mathbf{f}(\mathbf{x}_n, \mathbf{x}_{n-1}, \boldsymbol{\theta}) = \mathbf{f}_1(\mathbf{x}_n, \boldsymbol{\theta}) - (t_n - t_{n-1}) \mathbf{f}_2(\mathbf{x}_n, \boldsymbol{\theta}) - \mathbf{f}_1(\mathbf{x}_{n-1}, \boldsymbol{\theta}). \quad (2.31)$$

A robust simulator needs to contain a time step regulator. The time step must be chosen as large as possible to minimize computational costs. Simultaneously, the time step is bounded by accuracy and stability constraints. Moreover, Eq. (2.30) may have non-unique roots, possibly including physically unrealistic ones (negative pressure, saturation outside $[0, 1]$). These constraints can be explicitly checked for every solution of Eq. (2.30). When the solution is not satisfactory, another attempt can be made with a smaller time step. The time step can then be increased again slightly after every successful step. The decrement factor should be significantly larger than the increment factor.

2.1.6 More realistic physics

Section 2.1.1 describes the reservoir simulator that was used for this research. Significant physical phenomena were neglected:

- Fluid miscibility; the fluid components can be defined as the phases that are present at standard or surface conditions. At reservoir conditions, components can exist in other phases. For example, in a Black Oil model [Aziz and Settari (1979)], the gas component can be present in both the gas phase and the oil phase. Adding this to Eq. (2.17) gives a new PDE for the gas component

$$\frac{\partial}{\partial t} (\phi S_o b_o R_{go} + \phi S_g b_g) = \nabla \cdot \mathbf{K} \left(\frac{k_{ro} b_o}{\mu_o} R_{go} \nabla p_o + \frac{k_{rg} b_g}{\mu_g} \nabla p_g \right) + b_o R_{go} \hat{q}_o + b_g \hat{q}_g, \quad (2.32)$$

where b_α [–] is the reciprocal formation volume factor, expressing how the volume of a phase changes and gets split into the separate components if the volume were moved

from reservoir conditions to standard/surface conditions,

$$\frac{1}{b_\alpha} = B_\alpha = \frac{\rho_\alpha^{STC} + \sum_{\beta \neq \alpha} R_{\beta\alpha} \rho_\beta^{STC}}{\rho_\alpha}, \quad (2.33)$$

and $R_{\beta\alpha} [-]$ is the volume fraction of component β in the α -phase. The production terms $\hat{q}_\alpha [s^{-1}]$ are in volume of liquid at standard conditions per volume of grid block and per second. In a fully compositional model, all components can exist in all phases.

- Gravity effects; besides pressure differences, also density differences can cause fluid flow. This can be modelled by substituting $\nabla p_\alpha - \rho_\alpha g \nabla D$ for ∇p_α in Eq. (2.17), where $D [m]$ stands for depth.
- Capillary pressure; capillary effects [Leverett (1941), Morrow (1970)] are caused by the fact that on the pore scale the flow behavior of the fluids are affected by adhesive forces between rock and fluid molecules, rather than just cohesive forces within the fluids. The fluid pressures are different because the fluids interact differently with the rock, so $p_{c\alpha\beta} = p_\alpha - p_\beta \neq 0$. The capillary pressure does not just depend on the fluid saturations, but also on their time derivatives:
 - * Imbibition; the wetting fluid displaces the non-wetting fluid
 - * Drainage; the non-wetting fluid displaces the wetting fluid
- Appropriate well model; in section 2.1.4 a constant well factor ω^ξ is used. More physically realistic models can be found in [Peaceman (1977), Wheeler (1988)].
- Higher order effects; Darcy's law for anisotropic porous media can be derived from the Navier-Stokes equation by using a formal averaging procedure [Neuman (1977)]. This only holds for an incompressible homogeneous Newtonian fluid moving slowly through a rigid porous medium with uniform porosity under isothermal and steady state conditions. In other cases the applicability of Darcy's law is questionable and adding higher order terms may be appropriate.
- Temperature; fluids behave differently at different temperatures. To accurately setup the fluid flow equations, the fluid properties must be evaluated at the correct local temperature. An extra equation, based on conservation of energy, must then be introduced to model the temperature changes over time.

2.1.7 User input

Running a reservoir simulation can be done after

- the user has specified the grid. A grid specification consists of a list of grid blocks with their volumes, porosity values and links to the neighboring grid blocks. For every link between two grid blocks, the area of the interface and the distances from the grid block centers to the interface must be specified. When gravity effects are taken into account, also the orientation of the interface must be specified, for example by a vector that is normal to the interface. Permeability values must be specified for all grid block interfaces, or they must be specified at the grid block centers and interpolated to the interfaces using the center-interface distances.
- a set of wells is added. For every well an operating constraint must be specified as well as a list of grid blocks that are penetrated by the well.
- the fluid properties (density, viscosity, gas-oil ratio, formation volume factor, relative permeability, capillary pressure) are specified as functions of the pressure and saturation state variables. Some examples are shown in Fig. (2.1). When they are defined as analytical functions, also the derivatives must be defined as analytical functions. In case they are specified as lookup tables, new approximate lookup tables for the derivatives can be automatically generated.

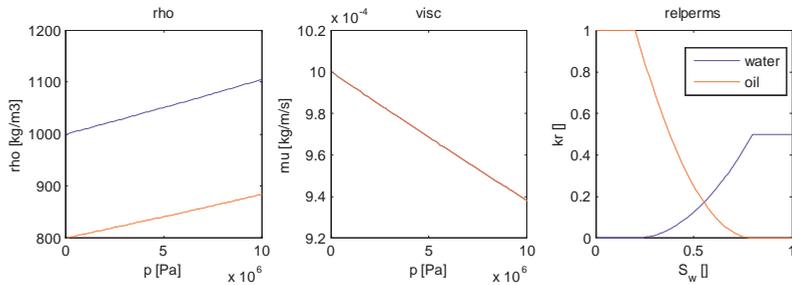


Figure 2.1: Density, viscosity and relative permeability of water and oil.

2.1.8 Simulating in weak constraint or stochastic mode

Due to unmodelled physics or numerical approximations, a reservoir simulator is never able to produce results that perfectly match the true reservoir state, even when the correct model parameters θ were known. These imperfections can all be lumped together and modelled using additional parameters ε_n . These model errors can be sampled from an appropriate probability distribution, or they can be prescribed by a variational method, as will be described

in section 3.3.3. In the former case, the additional parameters ε_n are stochastic variables. In the latter case, the simulator is used as a weak constraint in a minimization problem. Without these model errors, the simulator is called a strong constraint in such a minimization problem.

After every time step the oil and water mass that is present in a grid block is wrongly predicted. This is corrected by artificially injecting or producing extra mass into/from the grid block. Eq. (2.31) can be extended to

$$\tilde{\mathbf{f}}(\mathbf{x}_n, \mathbf{x}_{n-1}, \boldsymbol{\theta}, \varepsilon_n) = \mathbf{f}_1(\mathbf{x}_n, \boldsymbol{\theta}) - (t_n - t_{n-1}) \mathbf{f}_2(\mathbf{x}_n, \boldsymbol{\theta}) - \mathbf{f}_1(\mathbf{x}_{n-1}, \boldsymbol{\theta}) + \mathbf{f}_3(\mathbf{x}_n, \boldsymbol{\theta}, \varepsilon_n), \quad (2.34)$$

where $\mathbf{f}_3 < 0$ stands for extra injection and $\mathbf{f}_3 > 0$ for extra production. Adding too much mass gives physically unrealistic values for the state variables, for example very high pressures. Subtracting too much mass gives physically impossible results, for example negative pressures or saturations outside $[0, 1]$. This is modelled by constraining \mathbf{f}_3 with an upper bound that is calculated from \mathbf{f}_1 :

$$\mathbf{f}_3(\mathbf{x}_n, \boldsymbol{\theta}, \varepsilon_n) = \min \{ \mathbf{f}_1(\mathbf{x}_n, \boldsymbol{\theta}), (t_n - t_{n-1}) \varepsilon_n \}. \quad (2.35)$$

This gives ε_n the dimension $[kg s^{-1}]$. Eq. (2.35) can also be written using the Heaviside function and regularized in the same way as Eq. (2.10) in section 2.2

$$\begin{aligned} \min \{a, b\} &= aH(b - a) + bH(a - b) = \\ &= a + (b - a)H(a - b) = \\ &= b + (a - b)H(b - a). \end{aligned} \quad (2.36)$$

The stochastic reservoir simulator used in this research is therefore denoted by

$$\tilde{\mathbf{f}}(\mathbf{x}_n, \mathbf{x}_{n-1}, \boldsymbol{\theta}, \varepsilon_n) = \mathbf{f}(\mathbf{x}_n, \mathbf{x}_{n-1}, \boldsymbol{\theta}) + \mathbf{f}_1(\mathbf{x}_n, \boldsymbol{\theta}) - H(\tilde{\boldsymbol{\varepsilon}}_n) \otimes \tilde{\boldsymbol{\varepsilon}}_n, \quad (2.37)$$

where

$$\tilde{\boldsymbol{\varepsilon}}_n = \mathbf{f}_1(\mathbf{x}_n, \boldsymbol{\theta}) - (t_n - t_{n-1}) \varepsilon_n, \quad (2.38)$$

and \otimes stands for element-wise multiplication. The corresponding Jacobians are given by

$$\frac{\partial (H(\tilde{\boldsymbol{\varepsilon}}_n) \otimes \tilde{\boldsymbol{\varepsilon}}_n)}{\partial \tilde{\boldsymbol{\varepsilon}}_n} = \text{diag}(H(\tilde{\boldsymbol{\varepsilon}}_n) + \delta(\tilde{\boldsymbol{\varepsilon}}_n) \otimes \tilde{\boldsymbol{\varepsilon}}_n), \quad (2.39)$$

$$\mathbf{J}_{\tilde{\mathbf{f}}^{\mathbf{x}_n}} = \mathbf{J}_{\mathbf{f}^{\mathbf{x}_n}} + (\mathbf{I} - \text{diag}(H(\tilde{\boldsymbol{\varepsilon}}_n) + \delta(\tilde{\boldsymbol{\varepsilon}}_n) \otimes \tilde{\boldsymbol{\varepsilon}}_n)) \mathbf{J}_{\mathbf{f}_1^{\mathbf{x}_n}}, \quad (2.40)$$

$$\mathbf{J}_{\tilde{\mathbf{f}}^{\theta}} = \mathbf{J}_{\mathbf{f}^{\theta}} + (\mathbf{I} - \text{diag}(H(\tilde{\boldsymbol{\varepsilon}}_n) + \delta(\tilde{\boldsymbol{\varepsilon}}_n) \otimes \tilde{\boldsymbol{\varepsilon}}_n)) \mathbf{J}_{\mathbf{f}_1^{\theta}}, \quad (2.41)$$

$$\mathbf{J}_{\tilde{\mathbf{f}}^{\varepsilon_n}} = (t_n - t_{n-1}) \text{diag}(H(\tilde{\boldsymbol{\varepsilon}}_n) + \delta(\tilde{\boldsymbol{\varepsilon}}_n) \otimes \tilde{\boldsymbol{\varepsilon}}_n). \quad (2.42)$$

2.2 Simulator sensitivities

2.2.1 Jacobians for non-linear solver

Most non-linear solvers depend on, or can be speeded up by, using the Jacobian with respect to the state variables $\mathbf{f}_{\mathbf{x}}$. For Eq. (2.31) this Jacobian can easily be constructed from the Jacobians of \mathbf{f}_1 and \mathbf{f}_2 :

$$\mathbf{J}_{\mathbf{f}_{\mathbf{x}}}(\mathbf{x}_n, \mathbf{x}_{n-1}, \boldsymbol{\theta}) = \mathbf{J}_{\mathbf{f}_1}(\mathbf{x}_n, \boldsymbol{\theta}) - (t_n - t_{n-1}) \mathbf{J}_{\mathbf{f}_2}(\mathbf{x}_n, \boldsymbol{\theta}). \quad (2.43)$$

These can be calculated from

$$\frac{\partial \mathbf{f}_1^{w\xi}}{\partial p^\xi} = V^\xi \phi^\xi S^\xi \frac{\partial \rho_w^\xi}{\partial p^\xi}, \quad \frac{\partial \mathbf{f}_1^{o\xi}}{\partial p^\xi} = V^\xi \phi^\xi (1 - S^\xi) \frac{\partial \rho_o^\xi}{\partial p^\xi}, \quad (2.44)$$

$$\frac{\partial \mathbf{f}_1^{\alpha\xi}}{\partial p^\zeta} = 0 \quad \forall \zeta \neq \xi, \quad (2.45)$$

$$\frac{\partial \mathbf{f}_1^{w\xi}}{\partial S^\xi} = V^\xi \phi^\xi \rho_w^\xi, \quad \frac{\partial \mathbf{f}_1^{o\xi}}{\partial S^\xi} = -V^\xi \phi^\xi \rho_o^\xi, \quad \frac{\partial \mathbf{f}_1^{\alpha\xi}}{\partial S^\zeta} = 0 \quad \forall \zeta \neq \xi, \quad (2.46)$$

$$\frac{\partial \mathbf{f}_2^{\alpha\xi}}{\partial p^\xi} = \frac{\partial q_\alpha^\xi}{\partial p^\xi} + \sum_{\eta \in \mathbb{N}_\xi} \left((p^\eta - p^\xi) \frac{\partial t_\alpha^{\xi\eta}}{\partial p^\xi} - t_\alpha^{\xi\eta} \right), \quad (2.47)$$

$$\frac{\partial \mathbf{f}_2^{\alpha\xi}}{\partial p^\eta} = (p^\eta - p^\xi) \frac{\partial t_\alpha^{\xi\eta}}{\partial p^\eta} + t_\alpha^{\xi\eta}, \quad (2.48)$$

$$\frac{\partial \mathbf{f}_2^{\alpha\xi}}{\partial S^\xi} = \frac{\partial q_\alpha^\xi}{\partial S^\xi} + \sum_{\eta \in \mathbb{N}_\xi} (p^\eta - p^\xi) \frac{\partial t_\alpha^{\xi\eta}}{\partial S^\xi}, \quad (2.49)$$

$$\frac{\partial \mathbf{f}_2^{\alpha\xi}}{\partial S^\eta} = (p^\eta - p^\xi) \frac{\partial t_\alpha^{\xi\eta}}{\partial S^\eta}, \quad (2.50)$$

$$\frac{\partial \mathbf{f}_2^{\alpha\xi}}{\partial p^\zeta} = \frac{\partial q_\alpha^\xi}{\partial p^\zeta}, \quad \frac{\partial \mathbf{f}_2^{\alpha\xi}}{\partial S^\zeta} = \frac{\partial q_\alpha^\xi}{\partial S^\zeta}, \quad (2.51)$$

$$\frac{\partial t_\alpha^{\xi\eta}}{\partial p^{\xi\eta}} = t_\alpha^{\xi\eta} \left(\frac{1}{\rho_\alpha^{\xi\eta}} \frac{\partial \rho_\alpha^{\xi\eta}}{\partial p^{\xi\eta}} - \frac{1}{\mu_\alpha^{\xi\eta}} \frac{\partial \mu_\alpha^{\xi\eta}}{\partial p^{\xi\eta}} \right), \quad (2.52)$$

$$\frac{\partial t_\alpha^{\xi\eta}}{\partial S^{\xi\eta}} = \frac{k_\alpha^{\xi\eta} A^{\xi\eta}}{h^{\xi\eta}} \frac{\rho_\alpha^{\xi\eta}}{\mu_\alpha^{\xi\eta}} \frac{\partial k_{r\alpha}^{\xi\eta}}{\partial S^{\xi\eta}} = \frac{t_\alpha^{\xi\eta}}{k_{r\alpha}^{\xi\eta}} \frac{\partial k_{r\alpha}^{\xi\eta}}{\partial S^{\xi\eta}}, \quad (2.53)$$

$$\frac{\partial t_\alpha^{\xi\eta}}{\partial p^\zeta} = \frac{\partial t_\alpha^{\xi\eta}}{\partial p^{\xi\eta}} \frac{\partial p^{\xi\eta}}{\partial p^\zeta} + \frac{\partial t_\alpha^{\xi\eta}}{\partial S^{\xi\eta}} \frac{\partial S^{\xi\eta}}{\partial p^\zeta}. \quad (2.54)$$

$$\frac{\partial t_\alpha^{\xi\eta}}{\partial S^\zeta} = \frac{\partial t_\alpha^{\xi\eta}}{\partial S^{\xi\eta}} \frac{\partial S^{\xi\eta}}{\partial S^\zeta} \quad (2.55)$$

Calculating these properties is as easy as evaluating \mathbf{f}_1 and \mathbf{f}_2 themselves, except for three issues:

- Making sure the derivatives end up at the correct places in the Jacobian requires quite some bookkeeping, especially for unstructured grids.
- The derivatives of the fluid properties with respect to the state variables must be available. In case these properties are available in the form of analytical functions, additional functions must be supplied. In case the fluid properties are available in the form of lookup tables, new tables can be created by interpolation.
- The term $\frac{\partial p^{\xi\eta}}{\partial p^\zeta}$ in Eq. (2.54) can be calculated from Eq. (2.9). The terms $\frac{\partial S^{\xi\eta}}{\partial p^\zeta}$ and $\frac{\partial S^{\xi\eta}}{\partial S^\zeta}$ in Eq. (2.54) and Eq. (2.55) cannot be calculated from Eq. (2.10), since the Heaviside function is not differentiable. As a regularization method the Heaviside function can be smoothed resulting in a differentiable function. Alternatively, the derivative of the Heaviside function, a Dirac delta function, can be neglected.

The derivatives for injection/production $\frac{\partial q_\alpha^\xi}{\partial p^\zeta}$, $\frac{\partial q_\alpha^\xi}{\partial S^\zeta}$, $\frac{\partial q_\alpha^\xi}{\partial p^\zeta}$ and $\frac{\partial q_\alpha^\xi}{\partial S^\zeta}$ can be calculated using

$$\frac{\partial}{\partial x^\eta} \left(\frac{\lambda_\alpha^\xi}{\lambda_t} \right) = \frac{1}{\lambda_t} \frac{\partial \lambda_\alpha^\xi}{\partial x_\xi} \delta_{\xi\eta} - \frac{\lambda_\alpha^\xi}{\lambda_t^2} \sum_\alpha \frac{\partial \lambda_\alpha^\eta}{\partial x^\eta} \quad (2.56)$$

for rate constraints and Eq. (2.52) and Eq. (2.53) with mobility at the grid block centers substituted for transmissibility at the grid block interfaces for pressure constraints.

2.2.2 Influence of model parameters on system states

While \mathbf{x}_n is calculated from \mathbf{x}_{n-1} by $\mathbf{f}(\mathbf{x}_n, \mathbf{x}_{n-1}, \boldsymbol{\theta}) = 0$ (in case $q = 1$), the sensitivity matrix $\frac{d\mathbf{x}_n}{d\boldsymbol{\theta}}$ can also be updated from $\frac{d\mathbf{x}_{n-1}}{d\boldsymbol{\theta}}$ simultaneously by

$$\frac{d\mathbf{f}}{d\boldsymbol{\theta}}(\mathbf{x}_n, \mathbf{x}_{n-1}, \boldsymbol{\theta}) = \mathbf{0}, \quad (2.57)$$

$$\frac{\partial \mathbf{f}}{\partial \mathbf{x}_n}(\mathbf{x}_n, \mathbf{x}_{n-1}, \boldsymbol{\theta}) \frac{d\mathbf{x}_n}{d\boldsymbol{\theta}} + \frac{\partial \mathbf{f}}{\partial \mathbf{x}_{n-1}}(\mathbf{x}_n, \mathbf{x}_{n-1}, \boldsymbol{\theta}) \frac{d\mathbf{x}_{n-1}}{d\boldsymbol{\theta}} + \frac{\partial \mathbf{f}}{\partial \boldsymbol{\theta}}(\mathbf{x}_n, \mathbf{x}_{n-1}, \boldsymbol{\theta}) = \mathbf{0}. \quad (2.58)$$

Substituting Eq. (2.31) gives

$$\begin{aligned} & (\mathbf{J}_{\mathbf{f}_1^x}(\mathbf{x}_n, \boldsymbol{\theta}) - (t_n - t_{n-1}) \mathbf{J}_{\mathbf{f}_2^x}(\mathbf{x}_n, \boldsymbol{\theta})) \frac{d\mathbf{x}_n}{d\boldsymbol{\theta}} \\ &= \mathbf{J}_{\mathbf{f}_1^x}(\mathbf{x}_{n-1}, \boldsymbol{\theta}) \frac{d\mathbf{x}_{n-1}}{d\boldsymbol{\theta}} + (t_n - t_{n-1}) \mathbf{J}_{\mathbf{f}_2^\theta}(\mathbf{x}_n, \boldsymbol{\theta}) \\ & \quad + \mathbf{J}_{\mathbf{f}_1^\theta}(\mathbf{x}_{n-1}, \boldsymbol{\theta}) - \mathbf{J}_{\mathbf{f}_1^\theta}(\mathbf{x}_n, \boldsymbol{\theta}). \end{aligned} \quad (2.59)$$

The sensitivity matrix can be used to calculate the gradient of some objective function $J(\mathbf{x}_1, \dots, \mathbf{x}_n)$ w.r.t. the parameters:

$$\frac{d}{d\boldsymbol{\theta}} J(\mathbf{x}_1, \dots, \mathbf{x}_n) = \frac{\partial J}{\partial \mathbf{x}_1} \frac{d\mathbf{x}_1}{d\boldsymbol{\theta}} + \dots + \frac{\partial J}{\partial \mathbf{x}_n} \frac{d\mathbf{x}_n}{d\boldsymbol{\theta}}, \quad (2.60)$$

or

$$\left(\frac{dJ(\mathbf{x}_1, \dots, \mathbf{x}_n)}{d\boldsymbol{\theta}} \right)^T = \left(\frac{d\mathbf{x}_1}{d\boldsymbol{\theta}} \right)^T \left(\frac{\partial J}{\partial \mathbf{x}_1} \right)^T + \dots + \left(\frac{d\mathbf{x}_n}{d\boldsymbol{\theta}} \right)^T \left(\frac{\partial J}{\partial \mathbf{x}_n} \right)^T. \quad (2.61)$$

This is computationally expensive because it requires doing as many simulations as there are parameters in $\boldsymbol{\theta}$ with a linearized reservoir simulator. A computationally much more efficient way to calculate this gradient will be described in section 3.3.2.

2.2.3 Jacobians for sensitivities

Updating the sensitivity matrix $\frac{d\mathbf{x}_n}{d\boldsymbol{\theta}}$, requires the availability of the Jacobians $\mathbf{J}_{\mathbf{f}_1^\theta}$ and $\mathbf{J}_{\mathbf{f}_2^\theta}$ (Eq. (2.59)).

- The Jacobian of \mathbf{f}_1 with respect to porosity values ϕ can be evaluated by substituting $\phi^\xi = 1$ in Eq. (2.4) and doing some bookkeeping to make sure the derivatives end up in the correct columns of $\mathbf{J}_{\mathbf{f}_1^\theta}$.
- The Jacobian of \mathbf{f}_2 with respect to permeability values at the grid block interfaces $k^{\xi\eta}$ can be evaluated by substituting $k^{\xi\eta} = 1$ in Eq. (2.6) and Eq. (2.5).
- The Jacobian of \mathbf{f}_2 with respect to permeability values at the grid block centers can be found by multiplying the result with the derivatives of Eq. (2.12), according to the chain rule for differentiation. These are

$$\frac{\partial k^{\xi\eta}}{\partial k^\xi} = 2 \left(\frac{k^\eta}{k^\xi + k^\eta} \right)^2. \quad (2.62)$$

- The derivatives with respect to the logarithm of permeability values can be obtained by multiplying with the permeability values, according to

$$\frac{dy}{d \ln x} = \frac{dy}{dx} \frac{dx}{d \ln x} = \frac{dy}{dx} \frac{1}{\frac{d \ln x}{dx}} = x \frac{dy}{dx}. \quad (2.63)$$

Chapter 3

Data Assimilation

3.1 Introduction

The aim of data assimilation is to improve numerical models by adding measurement information. In case of petroleum engineering, the model might be the combination of a reservoir simulator, a rock-physics model and a wave propagation package. Measurements can originate from geology, seismics, petrophysics, down-hole sensors and surface facilities. The models are updated by estimating their parameters. These parameters can for example be the porosity values of all grid blocks, the permeability values on the grid block interfaces, the elastic moduli of the reservoir rock or the parameters of a parameterized capillary pressure curve.

In a probabilistic setting, the likelihood of the model parameters given the measurements is maximized. The numerical model is used as a weak constraint, since it is assumed that besides the uncertainty in the model parameters, there is an additional source of uncertainty, the model errors. Without these model errors, the model is assumed to be perfect and it is used as a strong constraint. In practice the problem is often reduced to a least squares problem by assuming Gaussian error statistics, resulting in a variety of related data assimilation algorithms. For linear systems they solve the same least squares problem; for non-linear systems, like multiphase flow in porous media, they have their own peculiarities and utilization. This chapter gives an overview of data assimilation algorithms and shows how they are related. Numerical examples are shown in chapters 4, 5 and 6 to illustrate the applicability of the methods by estimating permeability values.

Bayes rule, section 3.2.1, can be seen as the basis from which many data assimilation routines are derived. However, most data assimilation algorithms can also be formulated independently from Bayes rule.

In general, two families of derived methods can be distinguished: variational methods and filters. In variational methods, the sensitivity of the data-mismatch between predicted data and measured data over the lifetime of the reservoir with respect to model parameters is used to get a better estimate of the parameters. This is repeated until convergence. Filters work locally in time; measurements are assimilated whenever they become available without recomputing the reservoir history. However, this is done in such a way that some kind of optimality criterion is honoured, ensuring consistency with previously assimilated data.

In the context of variational methods, regularization methods will be discussed. Regularization is performed to decrease the number of degrees of freedom of the data-mismatch objective function. This also reduces the number of local minima of the objective and increases the chance that gradient-based optimization algorithms find a local minimum that

is closer to the global minimum. Usually basis functions are chosen that map a coefficient space to the parameter space of much higher dimension. Optimal coefficients are searched for rather than the parameters themselves. Optionally the basis functions may be updated adaptively, as is done by the Representer Method [Bennett (2002)] or by Binary Levelset functions [Nielsen (2006)].

The Kalman filter [Kalman (1960), Gelb (1974), Welch and Bishop (1995)] for state and parameter estimation of linear systems will be introduced. To deal with non-linear systems that arise from petroleum simulation applications, the Ensemble Kalman filter [Evensen (2003)] can be applied. Also an improvement is discussed to handle non-linear measurement operators. Data that are correlated in time, like seismic measurements, can be assimilated using a smoother [Cohn *et al.* (1994)] or by a modified filter with a double ensemble size (section 3.4.5).

3.2 Bayesian data assimilation

3.2.1 Bayes rule

Reservoir simulation can be embedded in a stochastic or probabilistic framework. In that case the reservoir state variables (pressures and saturations in all grid blocks) do not have deterministic values, but are described by a multivariate probability distribution (or density) function (PDF). The stochastic nature of the state variables is caused by the uncertainty in the initial states, the uncertainty in the model parameters (permeability, porosity, etc.) and the fact that the reservoir simulator is imperfect (gravity or capillary effects were not modelled or 3 components were modelled where 5 would have been more appropriate). The uncertainty in the measurements are caused by two effects; sensors try to monitor a stochastic quantity and are subject to influences that might damage them or otherwise corrupt the data.

Bayes rule [Bayes (1763)]

$$P(\mathbf{x}|\mathbf{y}) = \frac{P(\mathbf{y}|\mathbf{x})P(\mathbf{x})}{P(\mathbf{y})}, \quad P(\mathbf{x}|\mathbf{y}) \propto P(\mathbf{y}|\mathbf{x})P(\mathbf{x}) = P(\mathbf{x}, \mathbf{y}), \quad (3.1)$$

states that the *posterior* PDF $P(\mathbf{x}|\mathbf{y})$ (the probability of the reservoir states and parameters given the data) is proportional to the *prior* PDF $P(\mathbf{x})$ (the probability of the reservoir states and parameters) multiplied by the *likelihood* $P(\mathbf{y}|\mathbf{x})$ of the data given the reservoir states.

The \mathbf{x} of Eq. (3.1) can be interpreted in two ways:

1. \mathbf{x} contains the reservoir state variables at all times, augmented with the uncertain static model parameters.

2. \mathbf{x} contains all uncertain parameters at $t = t_0$, including state variables and static model parameters. In this case, the state variables at $t \neq t_0$ are not shown in Eq. (3.1).

In the former case, the prior can be calculated by running a modified reservoir simulator on the initial reservoir states. The likelihood of the data can then be calculated by inserting the results into a forward predicting sensor model. In the latter case, the prior is defined at $t = t_0$ and the likelihood is affected by both the reservoir simulator and the sensor model. No inverse sensor model is needed; the inversion is done by Bayes rule.

Bayes rule can only be used analytically for very simple PDF's, reservoir models (1 phase) and sensor models (only pressure measurements). In other cases the PDF's must be approximated numerically or sampled by a random number generator. To represent multi-modal or otherwise complex PDF's, many samples are needed, making Bayesian data assimilation very computationally intensive or infeasible.

3.2.2 Special cases

Some properties of the multivariate Gaussian probability density/distribution are:

$$p(\mathbf{x}) = \frac{1}{\sqrt{|2\pi\Sigma|}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})} \quad , \quad P(\mathbf{x}) = \int_{(-\infty, \dots, -\infty)}^{\mathbf{x}} p(\mathbf{y}) d\mathbf{y}, \quad (3.2)$$

with mean $\boldsymbol{\mu}$ and covariance Σ

- If \mathbf{x} is Gaussian, then $\mathbf{Ax} + \mathbf{b}$ is Gaussian
- If (\mathbf{x}, \mathbf{y}) is jointly Gaussian, then $\mathbf{Ax} + \mathbf{By} + \mathbf{c}$ is Gaussian
- If (\mathbf{x}, \mathbf{y}) is jointly Gaussian, then $\mathbf{x}|\mathbf{y}$ and $\mathbf{y}|\mathbf{x}$ are Gaussian

Consequently:

- If the prior is Gaussian and the simulator and sensor model are linear, then the posterior is Gaussian
- Hence, it suffices to only compute the mean and covariance of the posterior instead of the full posterior
- Let $(\boldsymbol{\theta}, \boldsymbol{\varepsilon}, \mathbf{y})$ be jointly Gaussian, with mean $(\boldsymbol{\mu}_\theta, \mathbf{0}, \boldsymbol{\mu}_y)$ and covariance

$$\begin{bmatrix} \mathbf{P}_\theta & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_\varepsilon & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{P}_y \end{bmatrix}. \quad (3.3)$$

θ contains the model parameters, ε contains the model errors that are caused by unmodelled physics and numerical errors and \mathbf{y} contains the measurements. Then the posterior probability density is proportional to the joint probability density, or

$$p(\theta, \varepsilon | \mathbf{y}) \propto e^{-J}, \quad (3.4)$$

$$J = (\mathbf{y} - \boldsymbol{\mu}_y)^T \mathbf{P}_y^{-1} (\mathbf{y} - \boldsymbol{\mu}_y) + (\theta - \boldsymbol{\mu}_\theta)^T \mathbf{P}_\theta^{-1} (\theta - \boldsymbol{\mu}_\theta) + \varepsilon^T \mathbf{P}_\varepsilon^{-1} \varepsilon. \quad (3.5)$$

Finding the maximum a posteriori estimate (MAP) can be done by minimizing J .

3.3 Variational data assimilation

Even though Eq. (3.5) does not have any physical or probabilistic interpretation when the prior is not Gaussian or the reservoir simulator or sensor model is not linear, it still is the basis for variational data assimilation methods. The quadratic objective, Eq. (3.5), could have been derived from Bayes rule. In that case \mathbf{P}_y , \mathbf{P}_θ and \mathbf{P}_ε can be seen as scaling matrices that represent how accurate the sensors are and how much trust is given to the initial model parameters ($\boldsymbol{\mu}_\theta$) and the model itself. \mathbf{y} contains the "measured measurements" and $\boldsymbol{\mu}_y$ contains the "predicted measurements", which are related to the model parameters θ through the reservoir simulator and the sensor model. The calculated physics are allowed to deviate from the reservoir model, since the reservoir model does not correctly model the true physical phenomena. This deviation is modelled by ε . Thus, $\theta - \boldsymbol{\mu}_\theta$ and ε both model errors in the reservoir simulator; $\theta - \boldsymbol{\mu}_\theta$ represents a quantification of errors in the modelled physics and ε represents a quantification of errors in the unmodelled physics, or a superposition of physical phenomena where the individual effects can not be quantified. Mathematically, the term $\varepsilon \mathbf{P}_\varepsilon^{-1} \varepsilon$ can be augmented to $(\theta - \boldsymbol{\mu}_\theta)^T \mathbf{P}_\theta^{-1} (\theta - \boldsymbol{\mu}_\theta)$ by introducing $\boldsymbol{\mu}_\varepsilon = \mathbf{0}$.

The idea of variational data assimilation is that the first (or higher) order variation of the objective J w.r.t. the parameters θ is used to update these parameters. Usually the initial estimate of θ is equal to $\boldsymbol{\mu}_\theta$. The second term at the right-hand side of Eq. (3.5) can be deleted, but often it is needed to regularize the minimization or keep θ within a physically meaningful range.

Section 3.3.1 discusses the computational resources involved in calculating the gradient of J w.r.t. θ and introduces the term "adjoint reservoir state". Explicitly allowing the model errors ε to be not equal to $\mathbf{0}$ (using the reservoir simulator as a weak constraint) introduces an extra obstacle in obtaining a gradient. This effect is explained in sections 3.3.2 and 3.3.3. Section 3.3.4 describes regularization methods that decrease the dimension of the parameter space and reduce the number of local minima of the objective function. The Representer Method was designed as a regularization method that deals with weak constraints and is explained in section 3.3.5.

3.3.1 Variational calculus

Constrained optimization of a static system

Imagine that the objective $J = J(\mathbf{y}, \boldsymbol{\theta})$ needs to be minimized w.r.t. $(\boldsymbol{\theta}, \mathbf{y})$, where \mathbf{y} is related to $\boldsymbol{\theta}$ by the static system $\mathbf{f}(\mathbf{y}, \boldsymbol{\theta}) = \mathbf{0}$. The gradient $\frac{dJ}{d\boldsymbol{\theta}}$ can be obtained by applying the chain rule for differentiation (indicated by subscripts):

$$\frac{dJ}{d\boldsymbol{\theta}} = J_{\boldsymbol{\theta}} + J_{\mathbf{y}}\mathbf{y}_{\boldsymbol{\theta}}. \quad (3.6)$$

The Jacobian $\mathbf{y}_{\boldsymbol{\theta}}$ originates from differentiating the static system \mathbf{f} :

$$\mathbf{f}_{\boldsymbol{\theta}} + \mathbf{f}_{\mathbf{y}}\mathbf{y}_{\boldsymbol{\theta}} = \mathbf{0} \quad \Rightarrow \quad \mathbf{y}_{\boldsymbol{\theta}} = -\mathbf{f}_{\mathbf{y}}^{-1}\mathbf{f}_{\boldsymbol{\theta}}. \quad (3.7)$$

This is known as forward sensitivity analysis, and requires solving $|\boldsymbol{\theta}|$ linear systems of order $|\mathbf{y}|$, where $|\cdot|$ stands for counting the number of elements in a vector.

It can also be done by just solving 1 linear system of order $|\mathbf{y}|$. In the forward sensitivity analysis, first $\mathbf{y}_{\boldsymbol{\theta}}$ is calculated and then the rows are summed by $J_{\mathbf{y}}$. In the backward or "adjoint" sensitivity analysis, a linear combination of the rows of $\mathbf{y}_{\boldsymbol{\theta}}$ is calculated without explicitly calculating $\mathbf{y}_{\boldsymbol{\theta}}$ itself. In order to do so, Eq. (3.7) is first multiplied by (yet undetermined) weighting factors $\boldsymbol{\lambda}$.

$$\boldsymbol{\lambda}^T \mathbf{f}_{\boldsymbol{\theta}} + \boldsymbol{\lambda}^T \mathbf{f}_{\mathbf{y}}\mathbf{y}_{\boldsymbol{\theta}} = 0. \quad (3.8)$$

The weighting factors $\boldsymbol{\lambda}$ can now be determined by requiring that the summation of the rows of $\mathbf{y}_{\boldsymbol{\theta}}$ is the same for both the forward and the adjoint sensitivity analysis, thus (compare second terms of Eq. (3.6) and Eq. (3.8))

$$J_{\mathbf{y}} = \boldsymbol{\lambda}^T \mathbf{f}_{\mathbf{y}} \quad \Rightarrow \quad \boldsymbol{\lambda} = \mathbf{f}_{\mathbf{y}}^{-T} J_{\mathbf{y}}^T. \quad (3.9)$$

Now the gradient can be determined by substituting Eq. (3.8) and Eq. (3.9) into Eq. (3.6):

$$\frac{dJ}{d\boldsymbol{\theta}} = J_{\boldsymbol{\theta}} - \boldsymbol{\lambda}^T \mathbf{f}_{\boldsymbol{\theta}}. \quad (3.10)$$

This method is equivalent to minimizing the alternative objective $\tilde{J} = J(\mathbf{y}, \boldsymbol{\theta}) + \boldsymbol{\lambda}^T \mathbf{f}(\mathbf{y}, \boldsymbol{\theta})$ w.r.t. $(\boldsymbol{\theta}, \mathbf{y}, \boldsymbol{\lambda})$ without extra explicit constraints; the constraint is implicitly present in the objective function.

Constrained optimization of a dynamic system

Consider an objective of the form $J = J(\mathbf{y}_{t_n})$ which has to be minimized w.r.t. \mathbf{y}_{t_0} under the constraint that $\mathbf{y}_{t_i} = \mathbf{f}(\mathbf{y}_{t_{i-1}})$. By applying the chain rule:

$$\frac{dJ}{d\mathbf{y}_{t_0}} = \frac{dJ}{d\mathbf{y}_{t_n}} \frac{d\mathbf{f}(\mathbf{y}_{t_{n-1}})}{d\mathbf{y}_{t_{n-1}}} \dots \frac{d\mathbf{f}(\mathbf{y}_{t_0})}{d\mathbf{y}_{t_0}}. \quad (3.11)$$

If Eq. (3.11) is evaluated from t_0 to t_n , then every time step requires a matrix-matrix product and at t_n one matrix-vector product is needed. Computer memory can be saved if Eq. (3.11) is evaluated from t_n to t_0 , because only matrix-vector products are needed. When \mathbf{f} represents a reservoir simulator with an implicit time discretization scheme, then the above should be read with "matrix-matrix product" replaced by "matrix-inverse-matrix product" and "matrix-vector product" replaced by "solving a linear system". Evaluating from t_n to t_0 not only saves computer memory but also a lot of computation time. Moreover, if the interest is $\frac{dJ}{d\theta}$ rather than $\frac{dJ}{dy_{t_0}}$ and the constraint is of the form $\mathbf{y}_{t_i} = \mathbf{f}(\mathbf{y}_{t_{i-1}}, \theta)$, the choice to evaluate in inverse time becomes even more apparent.

Obtaining a gradient

A numerical gradient can be obtained by perturbing all parameters, each time running the reservoir simulator and evaluating the data-mismatch objective function. An analytical gradient can be obtained by forward or adjoint sensitivity analysis as discussed above. It is computationally much more efficient to introduce adjoint reservoir states and modify the reservoir simulator to compute these adjoint variables in inverse time. Calculating a numerical gradient requires almost the same amount of computer resources as forward sensitivity analysis.

3.3.2 The reservoir simulator as a strong constraint

In variational data assimilation, the non-linear reservoir simulator

$$\mathbf{x}_{t_0} = \mathbf{x}_0(\theta) \quad , \quad \mathbf{f}(\mathbf{x}_{t_j}, \mathbf{x}_{t_{j-1}}, \theta) = \varepsilon_j, \quad (3.12)$$

with model errors ε_j is added to the objective function, Eq. (3.5), by a Lagrange multiplier or adjoint state λ

$$\begin{aligned} J = & (\mathbf{y} - \boldsymbol{\mu}_y)^T \mathbf{P}_y^{-1} (\mathbf{y} - \boldsymbol{\mu}_y) + (\boldsymbol{\theta} - \boldsymbol{\mu}_\theta)^T \mathbf{P}_\theta^{-1} (\boldsymbol{\theta} - \boldsymbol{\mu}_\theta) + \\ & + \sum_{j=1}^M \boldsymbol{\varepsilon}_j^T \mathbf{P}_{\varepsilon_j}^{-1} \boldsymbol{\varepsilon}_j + 2 \sum_{j=1}^M \boldsymbol{\lambda}_j^T (\mathbf{f}(\mathbf{x}_{t_j}, \mathbf{x}_{t_{j-1}}, \theta) - \boldsymbol{\varepsilon}_j). \end{aligned} \quad (3.13)$$

In order to find a minimum of the original objective function Eq. (3.5) w.r.t. the model constraints, it suffices to look for a saddle point of the modified objective function Eq. (3.13). A necessary condition for a saddle point of J is that the first order variations of J w.r.t. $\boldsymbol{\varepsilon}_j$, $\boldsymbol{\lambda}_j$, \mathbf{x}_{t_j} and $\boldsymbol{\theta}$ equal zero. These variations are

$$\frac{\partial J}{\partial \boldsymbol{\varepsilon}_j} = 2 \left(\mathbf{P}_{\varepsilon_j}^{-1} \boldsymbol{\varepsilon}_j - \boldsymbol{\lambda}_j \right)^T, \quad (3.14)$$

$$\frac{\partial J}{\partial \boldsymbol{\lambda}_j} = 2 (\mathbf{f}(\mathbf{x}_{t_j}, \mathbf{x}_{t_{j-1}}, \boldsymbol{\theta}) - \boldsymbol{\varepsilon}_j)^T, \quad (3.15)$$

$$\frac{\partial J}{\partial \mathbf{x}_{t_j}} = -2 (\mathbf{y} - \boldsymbol{\mu}_y)^T \mathbf{P}_y^{-1} \frac{\partial \boldsymbol{\mu}_y}{\partial \mathbf{x}_{t_j}} + 2 \boldsymbol{\lambda}_j^T \frac{\partial \mathbf{f}(\mathbf{x}_{t_j}, \mathbf{x}_{t_{j-1}}, \boldsymbol{\theta})}{\partial \mathbf{x}_{t_j}} + 2 \boldsymbol{\lambda}_{j+1}^T \frac{\partial \mathbf{f}(\mathbf{x}_{t_{j+1}}, \mathbf{x}_{t_j}, \boldsymbol{\theta})}{\partial \mathbf{x}_{t_j}}, \quad (3.16)$$

$$\frac{\partial J}{\partial \boldsymbol{\theta}} = -2 (\mathbf{y} - \boldsymbol{\mu}_y)^T \mathbf{P}_y^{-1} \frac{\partial \boldsymbol{\mu}_y}{\partial \boldsymbol{\theta}} + 2 (\boldsymbol{\theta} - \boldsymbol{\mu}_\theta)^T \mathbf{P}_\theta^{-1} + 2 \sum_{j=1}^M \boldsymbol{\lambda}_j^T \frac{\partial \mathbf{f}(\mathbf{x}_{t_j}, \mathbf{x}_{t_{j-1}}, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}, \quad (3.17)$$

respectively. The first term of Eq. (3.17) is only unequal to zero if model parameters can be directly measured, for example permeability of core samples after drilling of new wells. Even then the term can be made equal to zero by adding this new data to the prior $\boldsymbol{\mu}_\theta$ and \mathbf{P}_θ of the second term.

From these, the forward system

$$\mathbf{x}_{t_0} = \mathbf{x}_0(\boldsymbol{\theta}), \quad \mathbf{f}(\mathbf{x}_{t_j}, \mathbf{x}_{t_{j-1}}, \boldsymbol{\theta}) = \boldsymbol{\varepsilon}_j = \mathbf{P}_{\varepsilon_j} \boldsymbol{\lambda}_j, \quad (3.18)$$

and adjoint system

$$\begin{aligned} \boldsymbol{\lambda}_{M+1} &= \mathbf{0} \\ \left(\frac{\partial \mathbf{f}(\mathbf{x}_{t_j}, \mathbf{x}_{t_{j-1}}, \boldsymbol{\theta})}{\partial \mathbf{x}_{t_j}} \right)^T \boldsymbol{\lambda}_j &= - \left(\frac{\partial \mathbf{f}(\mathbf{x}_{t_{j+1}}, \mathbf{x}_{t_j}, \boldsymbol{\theta})}{\partial \mathbf{x}_{t_j}} \right)^T \boldsymbol{\lambda}_{j+1} + \left(\frac{\partial \boldsymbol{\mu}_y}{\partial \mathbf{x}_{t_j}} \right)^T \mathbf{P}_y^{-1} (\mathbf{y} - \boldsymbol{\mu}_y), \end{aligned} \quad (3.19)$$

can be derived, as well as the parameter equation

$$\boldsymbol{\theta} = \boldsymbol{\mu}_\theta - \mathbf{P}_\theta \sum_{j=1}^M \left(\frac{\partial \mathbf{f}(\mathbf{x}_{t_j}, \mathbf{x}_{t_{j-1}}, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right)^T \boldsymbol{\lambda}_j \quad (3.20)$$

and the desired gradient

$$\left(\frac{\partial J}{\partial \boldsymbol{\theta}} \right)^T = 2 \mathbf{P}_\theta^{-1} (\boldsymbol{\theta} - \boldsymbol{\mu}_\theta) + 2 \sum_{j=1}^M \left(\frac{\partial \mathbf{f}(\mathbf{x}_{t_j}, \mathbf{x}_{t_{j-1}}, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right)^T \boldsymbol{\lambda}_j. \quad (3.21)$$

If $\mathbf{P}_{\varepsilon_j}$ is explicitly set to zero, the reservoir simulator is used as a strong constraint. This can be done if the model is assumed to be perfect, for example if data are synthesized with the same model that is used to do data assimilation in a twin experiment or if the user has really no clue what value to use for $\mathbf{P}_{\varepsilon_j}$. In that case the following iterative algorithm can be used:

- Make an initial guess for $\boldsymbol{\theta}$; usually $\boldsymbol{\theta} = \boldsymbol{\mu}_\theta$.
- Run the forward model, Eq. (3.18), to obtain the states \mathbf{x}_{t_j} .
- Run the adjoint system backward in time, Eq. (3.19), to obtain the adjoint states $\boldsymbol{\lambda}_j$.

- Improve the parameter estimates using Eq. (3.20) or supply the gradient Eq. (3.21) to some advanced gradient-based search algorithm. Eq. (3.20) can also be used when the model parameters linearly enter the model. In that case, the algorithm is finished in one iteration.
- Loop until satisfied.

3.3.3 The reservoir simulator as a weak constraint

In case of weak constraint variational data assimilation, the algorithm mentioned in the previous section cannot be used directly, since the forward and backward simulations become coupled. It can however be used if the model errors ε_j are initialized by zero. In later iterations, the model errors for the forward simulation can be calculated from the adjoint simulation of previous iteration. Alternatively, an extra loop can be included, where Eq. (3.18) and Eq. (3.19) are used sequentially until convergence, before the step is made to the parameter update Eq. (3.20) or Eq. (3.21). Without this extra loop, the algorithm is equivalent to augmenting ε_j to θ :

$$\tilde{\theta} = (\theta, \varepsilon_1, \dots, \varepsilon_M) \quad , \quad \tilde{\mathbf{f}}(\mathbf{x}_{t_j}, \mathbf{x}_{t_{j-1}}, \tilde{\theta}) = \mathbf{f}(\mathbf{x}_{t_j}, \mathbf{x}_{t_{j-1}}, \theta) - \varepsilon_j. \quad (3.22)$$

The Representer Method, as described in section 3.3.5, is an alternative method that also regularizes (see section 3.3.4) the minimization process. Moreover, it gives information about the "value of data" as a bonus.

3.3.4 Regularization

When "smart" wells are equipped with downhole gauges, data from these gauges might indicate that the path between two wells is more permeable than predicted by the reservoir simulator. This can be corrected by increasing the permeability of the grid blocks that are in the flow path. However, there are many ways to make such a correction if there are more than one grid block in the flow path; the permeability in grid blocks A and B can both be increased a little, or the permeability in grid block A can be increased a lot while it is kept constant in grid block B. This corresponds to multiple local minima in the data-misfit objective function. Regularization is the subjective process of choosing a solution out of many possible solutions. This choice is not made after calculating all solutions and evaluating their quality, but it consists of a set of rules that hopefully guide the search algorithm towards a solution of preference.

Decomposition into basis functions

The guiding rules are usually represented by basis functions. The basis functions are used to (linearly or non-linearly) map a coefficient space to the parameter space of much higher dimension. Sometimes the deviations of the parameters from the prior are used rather than the parameters themselves, so for a linear mapping $\boldsymbol{\theta} = \mathbf{R}_\theta \mathbf{b}$ or

$$\boldsymbol{\theta} - \boldsymbol{\mu}_\theta = \mathbf{R}_\theta \mathbf{b}, \quad (3.23)$$

where the columns of \mathbf{R}_θ contain the basis functions and \mathbf{b} contains the coefficients (the parameters of the lower order minimization problem). Equations Eq. (3.18), Eq. (3.19) and Eq. (3.21) do not need to be modified, or in other words, the adjoint reservoir simulator remains unchanged. Instead a translation layer must be implemented in between the adjoint simulator and the minimization algorithm. The layer translates coarse scale parameters into fine scale parameters Eq. (3.23), and translates the fine scale gradient into a coarse scale gradient,

$$\left(\frac{\partial J}{\partial \mathbf{b}} \right)^T = \mathbf{R}_\theta^T \left(\frac{\partial J}{\partial \boldsymbol{\theta}} \right)^T. \quad (3.24)$$

After initialization of the fine scale parameters, the coarse scale parameters need to be initialized by inverting Eq. (3.23). The best (in terms of minimal Euclidian distance) coarse scale initialization can be found by projecting $\boldsymbol{\theta} - \boldsymbol{\mu}_\theta$ onto the space spanned by the basis functions \mathbf{R}_θ , so

$$\mathbf{b}_0 = (\mathbf{R}_\theta^T \mathbf{R}_\theta)^{-1} \mathbf{R}_\theta^T (\boldsymbol{\theta}_0 - \boldsymbol{\mu}_\theta) = \mathbf{R}_\theta^\dagger (\boldsymbol{\theta}_0 - \boldsymbol{\mu}_\theta). \quad (3.25)$$

Choosing the basis functions

Piecewise constant functions

Based on the prior information $\boldsymbol{\mu}_\theta$, a histogram can be derived. Parameters that have similar values can be grouped together and averaged to obtain a characteristic parameter of that group. Individual parameters of the group are then overwritten by the characteristic group parameter. The basis function of every group contains ones for fine scale parameters that are assigned to the group and zeros at all other positions; it is a Heaviside function. The coarse scale parameters are equal to the groups characteristic parameters.

Example

If $\boldsymbol{\mu}_\theta = (1, 2, 5)$ and regularization is done by reducing the size of the parameter vector from three to two elements by a piecewise constant function, then a possible choice would be to group the first two parameters together

$$\tilde{\boldsymbol{\theta}}_0 = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{2}(1+2) \\ 5 \end{bmatrix}. \quad (3.26)$$

Using these basis functions, it is not possible to reconstruct the original parameters ($\tilde{\theta}_0 \neq \mu_\theta$). Optionally, the columns of \mathbf{R}_θ may be normalized to produce an orthogonal matrix. The scaling factors that are needed to do this must then also be incorporated in the coarse scale parameters \mathbf{b} .

Detecting dominant features from a training image

In general, the prior does not just consist of μ_θ but also \mathbf{P}_θ . Maybe even higher order moments or a full non-Gaussian probability distribution is known. Extra constraints can be added to this prior. For example, the probability distribution of permeability values can be modified when core samples become available after drilling of wells. This can be used to sample realizations of the parameter vector. These realizations can be put as columns in a matrix to form a large database. Theoretically, this matrix contains the same information as the original probability function if infinitely many realizations are used. The database can be compressed with methods like Proper Orthogonal Decomposition (POD) [Smaoui and Garrouch (1997), Heijn *et al.* (2004)]. This results in a set of basis functions and a measure of their dominance in the database in terms of energy. This energy measure provides a criterion to select a useful number of basis functions. This feature is not available when the parameter vector is decomposed in piecewise constant functions. [Jafarpour and McLaughlin (2007)] claims that a cosine transform is better suited to parameterize a permeability field than the principal components that are obtained from a database with POD. When the prior only consists of μ_θ , this can also be used to generate a database. Small portions of μ_θ can be taken randomly or deterministically and added as columns to the database. [Strebelle (2002)]. Regression methods can also be used [Draper and Smith (1998)].

Updating the basis functions

Basis functions that come from a POD-like method represent some kind of digitized probability distribution. These functions are based on the prior but are also reasonably adequate in representing the posterior. Heaviside basis functions can be constructed to reasonably represent the prior, but they are in general not suited to represent the posterior. The basis functions need to be updated iteratively in order to obtain a good estimate of the parameters. Two steps can be used:

1. Keeping the basis functions fixed, improve the coarse scale parameters
2. Keeping the coarse scale parameters fixed, update the basis functions

Step 1 is present in all variational data assimilation algorithms, step 2 is optional. [Lien *et al.* (2006)] starts out with very few Heaviside basis functions. After step 1 has converged, the small scale gradient obtained from the adjoint system is used to evaluate how every basis function can be cut into two smaller scale basis functions. In [Nielsen (2006)] the basis functions are present only implicitly and the number of basis functions necessarily needs to

be equal to a power of 2 due to the representation. This can easily be extended to an arbitrary number of basis functions [Rommelse *et al.* (2004)]. Updating a basis function comes down to deciding whether a small scale parameter should stay in its coarse parameter group or move to another group. This can be naively done by running two reservoir simulations and putting the small scale parameter in the group that gives the lowest data-misfit. In [Nielsen (2006)] the decision is made based on the fine scale gradient obtained by the adjoint system.

Non-linear parameterizations

Sometimes it is more appropriate to apply basis functions to the parameters of interest after some non-linear transformation. For example, in reservoir simulation applications it has become common practice to apply basis functions to the logarithm of reservoir grid block permeabilities, rather than the reservoir permeabilities themselves. [Jafarpour and McLaughlin (2007)] proposes a different non-linear parameterization, using the Discrete Cosine Transform.

3.3.5 Representer Method

The essence of the Representer Method (RM) [Bennett (2002), Baird and Dawson (2005), Valstar (2001), Valstar *et al.* (2004), Przybysz (2004)] is threefold:

- The results of the reservoir simulator, Eq. (3.18), are needed to calculate the adjoint states by Eq. (3.19). However, the model errors of the simulator depend on the adjoint states through $\varepsilon_j = \mathbf{P}_{\varepsilon_j} \lambda_j$ (section 3.3.3). These equations are decoupled by the Representer Method.
- Non-linear data assimilation is notoriously ill-posed; several (or many) different parameter sets exist that minimize the squared data-misfit. Trying to determine more details than there are present in the data is called over-fitting. Regularization (section 3.3.4) is the subjective process of choosing one solution out of many solutions. Usually regularization is done by mapping a low order space of basis functions to the original high order parameter space. Instead of estimating the original parameters, the parameters of this mapping are estimated. The subjective nature of regularization is because the user has to choose the basis functions, for example by POD [Smaoui and Garrouch (1997), Heijn *et al.* (2004)]. In the Representer Method, the number of basis functions is chosen equal to the number of measurements. Over-fitting is avoided, because the level of detail that is looked for in the solution is equal to the level of detail in the measurement data. Unlike most methods, the actual form of the basis functions in the Representer Method is not chosen by the user, but is obtained by the simulator, Eq. (3.18), and the adjoint simulator, Eq. (3.19).
- After convergence of the Representer Method a parameter estimate is found that (locally) minimizes the data-misfit objective function. Moreover, every basis function provides in-

formation on the impact and influence in space and time of the measurement corresponding to that basis function. The RM therefore has the potential to quantify the usefulness of data. This might help to design optimal measuring strategies. When, where and what should be measured in order to better understand reservoirs in an early stage of production? How is that going to affect the production strategies? How can measurement design be included in lifetime production optimization?

This section describes the RM as it was formulated in the literature that is referenced at the top of this section. Modifications to RM will be proposed in chapter 4 and used throughout this thesis.

Not only the parameters, but also the forward and adjoint reservoir states are decomposed into basis functions, called representer functions or representers. The deviation of the parameters from the prior ($\theta - \mu_\theta$) and the model errors $\varepsilon_j = \mathbf{P}_{\varepsilon_j} \lambda_j$ are developed around zero, whereas the forward states are developed around the states of previous iteration. In order to use the same representer coefficients \mathbf{b} , an extra correction term $\mathbf{x}_{t_j}^{corr}$ must be introduced:

$$\begin{aligned}\boldsymbol{\theta} &= \boldsymbol{\mu}_\theta + \mathbf{R}_\theta \mathbf{b}, \\ \boldsymbol{\lambda}_j &= \mathbf{R}_{\lambda_j} \mathbf{b}, \\ \mathbf{x}_{t_j} &= \mathbf{x}_{t_j}^f + \mathbf{x}_{t_j}^{corr} + \mathbf{R}_{\mathbf{x}_{t_j}} \mathbf{b}.\end{aligned}\quad (3.27)$$

Adjoint representers

The representer coefficients can be defined as

$$\mathbf{b} = \mathbf{P}_y^{-1} (\mathbf{y} - \boldsymbol{\mu}_y), \quad (3.28)$$

or equivalently

$$\mathbf{b} = \mathbf{P}_y^{-1} (\mathbf{y} - (\mathbf{y}^f + \mathbf{y}^{corr} + \overline{\mathbf{R}}_{\mathbf{x}} \mathbf{b})) \quad , \quad (\mathbf{P}_y + \overline{\mathbf{R}}_{\mathbf{x}}) \mathbf{b} = \mathbf{y} - (\mathbf{y}^f + \mathbf{y}^{corr}), \quad (3.29)$$

where $\overline{\mathbf{R}}_{\mathbf{x}}$ is a concatenation of $\overline{\mathbf{R}}_{\mathbf{x}_{t_j}}$ for all time steps and $\overline{\mathbf{R}}_{\mathbf{x}_{t_j}}$ only contains the rows of $\mathbf{R}_{\mathbf{x}_{t_j}}$ at positions where the state vector can be directly measured. $\overline{\mathbf{R}}_{\mathbf{x}}$ is a square matrix, since there are as many basis functions as there are measurements. Following this definition, substitution of the adjoint representer expansions Eq. (3.27) into the adjoint system Eq. (3.19) gives a system for updating the adjoint representers:

$$\begin{aligned}\mathbf{R}_{\lambda_{M+1}} &= \mathbf{0} \\ \left(\frac{\partial \mathbf{f}(\mathbf{x}_{t_j}, \mathbf{x}_{t_{j-1}}, \boldsymbol{\theta})}{\partial \mathbf{x}_{t_j}} \right)^T \mathbf{R}_{\lambda_j} &= - \left(\frac{\partial \mathbf{f}(\mathbf{x}_{t_{j+1}}, \mathbf{x}_{t_j}, \boldsymbol{\theta})}{\partial \mathbf{x}_{t_j}} \right)^T \mathbf{R}_{\lambda_{j+1}} + \left(\frac{\partial \boldsymbol{\mu}_y}{\partial \mathbf{x}_{t_j}} \right)^T.\end{aligned}\quad (3.30)$$

Parameter representers

Substitution of adjoint and parameter representer expansions Eq. (3.27) into the parameter equation Eq. (3.20) gives a system for updating the parameter representers:

$$\mathbf{R}_\theta = -\mathbf{P}_\theta \sum_{j=1}^M \left(\frac{\partial \mathbf{f}(\mathbf{x}_{t_j}, \mathbf{x}_{t_{j-1}}, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right)^T \mathbf{R}_{\lambda_j}. \quad (3.31)$$

State representer

A system for updating the state representer is found by perturbing the forward system Eq. (3.18), linearization and then substituting the unperturbed forward system:

$$\frac{\partial f(\mathbf{x}_{t_j}, \mathbf{x}_{t_{j-1}}, \theta)}{\partial \mathbf{x}_{t_j}} \mathbf{R}_{\mathbf{x}_{t_j}} + \frac{\partial f(\mathbf{x}_{t_j}, \mathbf{x}_{t_{j-1}}, \theta)}{\partial \mathbf{x}_{t_{j-1}}} \mathbf{R}_{\mathbf{x}_{t_{j-1}}} + \frac{\partial f(\mathbf{x}_{t_j}, \mathbf{x}_{t_{j-1}}, \theta)}{\partial \theta} \mathbf{R}_\theta - \mathbf{P}_{\varepsilon_j} \mathbf{R}_{\lambda_j} = \mathbf{0} \quad (3.32)$$

Correction term

In addition to steps 1 and 2 of section 3.3.4, there is an additional third step:

1. Keeping the basis functions fixed, improve the coarse scale parameters
2. Keeping the coarse scale parameters fixed, update the basis functions
3. Whenever parameters are changed, different state predictions should be expected and the linearization of the state representer around the forward prediction should be updated accordingly by modifying the correction term $\mathbf{x}_{t_j}^{corr}$.

The criterion for step 3 is that the state representer decomposition should be consistent for the old $(\eta - 1)$ and the new (η) iteration, so $\mathbf{x}_{t_j}^{corr}(\eta)$ is calculated from Eq. (3.27) by:

$$\mathbf{x}_{t_j}(\eta - 1) = \mathbf{x}_{t_j}^f(\eta) + \mathbf{x}_{t_j}^{corr}(\eta) + [\mathbf{R}_{\mathbf{x}_{t_j}} \mathbf{b}](\eta) \quad (3.33)$$

The term $[\mathbf{R}_{\mathbf{x}_{t_j}} \mathbf{b}](\eta) = \gamma_{t_j}$ is approximated by multiplying Eq. (3.32) by \mathbf{b} and substituting $[\mathbf{R}_\theta \mathbf{b}](\eta - 1) = \psi$ and $[\mathbf{R}_{\lambda_j} \mathbf{b}](\eta - 1) = \phi_j$ for $[\mathbf{R}_\theta \mathbf{b}](\eta)$ and $[\mathbf{R}_{\lambda_j} \mathbf{b}](\eta)$:

$$\frac{\partial f(\mathbf{x}_{t_j}, \mathbf{x}_{t_{j-1}}, \theta)}{\partial \mathbf{x}_{t_j}} \gamma_{t_j} + \frac{\partial f(\mathbf{x}_{t_j}, \mathbf{x}_{t_{j-1}}, \theta)}{\partial \mathbf{x}_{t_{j-1}}} \gamma_{t_{j-1}} + \frac{\partial f(\mathbf{x}_{t_j}, \mathbf{x}_{t_{j-1}}, \theta)}{\partial \theta} \psi = \mathbf{P}_{\varepsilon_j} \phi_j \quad (3.34)$$

Flow chart

The computations involved in the representer method for solving the inverse parameter estimation problem are shown in the following flow chart:

1. Initialize with $\lambda = \mathbf{0}$ and $\theta = \mu_\theta$
2. Solve the forward model, Eq. (3.18)
3. For each measurement
 - * calculate the adjoint representer, Eq. (3.30)
 - * calculate the parameter representer, Eq. (3.31)

- * calculate the state variable representer, Eq. (3.32)
- 4. Calculate correction term, Eq.(3.33), Eq. (3.34)
- 5. Calculate representer coefficients, Eq. (3.29)
- 6. Update adjoint and parameters, Eq. (3.27)
- 7. Go to 2 if convergence criterion is not met

3.3.6 Approximated gradient

A reservoir simulator implements differential equations that are discretized in time and space. Whenever the discretization scheme is changed, the adjoint reservoir simulator needs to be modified for consistency as well. Often reservoir simulators contain highly complicated numerical routines for the time integration that are optimized for computational efficiency or stability. These solvers usually do not come with a consistent adjoint solver. It is possible to also use the forward solvers for the adjoint computation. In that case, the adjoint system is not derived from the time-discretized reservoir flow equations but from the time-continuous formulation. The time-continuous forward and adjoint systems are discretized by the numerical routines independently from each other. An approximated gradient is obtained, instead of the true gradient. Using this gradient in a gradient-based search algorithm is not necessarily worse than using the true gradient [Cao *et al.* (2002)].

The time-continuous space-discrete version of a reservoir simulator can be written as

$$\mathbf{x}_{t_0} = \mathbf{x}_0(\boldsymbol{\theta}) \quad , \quad \frac{\partial \mathbf{f}_1(\mathbf{x}(t), \boldsymbol{\theta})}{\partial t} = \mathbf{f}_2(\mathbf{x}(t), \boldsymbol{\theta}) + \boldsymbol{\varepsilon}(t), \quad (3.35)$$

which is not as general as Eq. (3.18). \mathbf{f}_1 models the accumulation of water/oil mass in each grid block and \mathbf{f}_2 models the flow of liquids due to pressure changes and production/injection. For one liquid phase, Eq. (3.35) can be solved by an ODE-solver (ordinary differential equation), requiring the first order derivatives of \mathbf{f}_1 and \mathbf{f}_2 . A more realistic reservoir simulator with more phases requires higher order derivatives or a DAE-solver (differential algebraic equation). The system must then be formulated as

$$\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{x}}_2 \\ \dot{\mathbf{x}}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{f}_2(\mathbf{x}_1, \boldsymbol{\theta}) + \boldsymbol{\varepsilon} \\ \mathbf{f}_1(\mathbf{x}_1, \boldsymbol{\theta}) - \mathbf{x}_2 \end{bmatrix} \quad (3.36)$$

(using $\mathbf{x}_1 = \mathbf{x}$, $\mathbf{x}_2 = \mathbf{f}_1(\mathbf{x}_1, \boldsymbol{\theta})$), where $\dot{\cdot}$ denotes a time derivative. The system Jacobian equals

$$\mathbf{J} = \begin{bmatrix} \mathbf{0} & \mathbf{J}_{\mathbf{f}_2^x} \\ -\mathbf{I} & \mathbf{J}_{\mathbf{f}_1^x} \end{bmatrix}. \quad (3.37)$$

The time-continuous adjoint system is the tangent linearization of the forward system Eq. (3.36)

$$\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \dot{\lambda}_2 \\ \dot{\lambda}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{J}_{\mathbf{f}_2^x} \\ -\mathbf{I} & \mathbf{J}_{\mathbf{f}_1^x} \end{bmatrix}^T \begin{bmatrix} \lambda_2 \\ \lambda_1 \end{bmatrix}, \quad (3.38)$$

where $\dot{\cdot}$ denotes a time derivative in inverse time. It can be derived by setting the derivatives of the objective function

$$\tilde{J} = J + 2 \int_{t_0}^{t_{end}} \lambda^T \left(\frac{\partial \mathbf{f}_1(\mathbf{x}(t), \boldsymbol{\theta})}{\partial t} - \mathbf{f}_2(\mathbf{x}(t), \boldsymbol{\theta}) - \boldsymbol{\varepsilon} \right) dt, \quad (3.39)$$

or

$$\begin{aligned} \tilde{J} &= J + 2\lambda^T(t_{end}) \mathbf{f}_1(\mathbf{x}_1(t_{end}), \boldsymbol{\theta}) - 2\lambda^T(t_0) \mathbf{f}_1(\mathbf{x}_1(t_0), \boldsymbol{\theta}) \\ &\quad - 2 \int_{t_0}^{t_{end}} \left(\dot{\lambda}^T \mathbf{f}_1 + \lambda^T (\mathbf{f}_2 + \boldsymbol{\varepsilon}) \right) dt, \end{aligned} \quad (3.40)$$

w.r.t. the reservoir states $\mathbf{x}(t)$ equal to zero (and substituting $\lambda_1 = \dot{\lambda}$, $\lambda_2 = -\lambda$). This also gives the magnitude of the discontinuities of the adjoint states at times when measurements are available;

$$2\mathbf{J}_{\mathbf{f}_1^x}^T(\mathbf{x}_1, \boldsymbol{\theta}) \left(\lambda^{after} - \lambda^{before} \right) = -\frac{\partial J}{\partial \mathbf{x}_1}, \quad (3.41)$$

$$\left(\mathbf{J}_{\mathbf{f}_1^x}^T(\mathbf{x}_1, \boldsymbol{\theta}) \lambda \right)^{after} = \left(\mathbf{J}_{\mathbf{f}_1^x}^T(\mathbf{x}_1, \boldsymbol{\theta}) \lambda \right)^{before} + \frac{1}{2} \frac{\partial J}{\partial \mathbf{x}_1}. \quad (3.42)$$

In general the reservoir states will be continuous (so $\mathbf{J}_{\mathbf{f}_1^x}^{after} = \mathbf{J}_{\mathbf{f}_1^x}^{before}$), but not when calculated by a filter [section 3.5.2]. The relation between $\boldsymbol{\varepsilon}$ and λ remains unchanged ($\boldsymbol{\varepsilon} = \mathbf{P}_\varepsilon \lambda$) and the gradient that is passed to the search algorithm is calculated by

$$\left(\frac{\partial J}{\partial \boldsymbol{\theta}} \right)^T = 2\mathbf{P}_\theta^{-1} (\boldsymbol{\theta} - \boldsymbol{\mu}_\theta) - 2 \left(\mathbf{J}_{\mathbf{f}_1^x} \frac{d\mathbf{x}_0}{d\boldsymbol{\theta}} + \mathbf{J}_{\mathbf{f}_1^\theta} \right) \lambda(t_0) - 2 \int_{t_0}^{t_{end}} \left(\mathbf{J}_{\mathbf{f}_1^\theta}^T \dot{\lambda} + \mathbf{J}_{\mathbf{f}_2^\theta}^T \lambda \right) dt. \quad (3.43)$$

3.4 Filtering

3.4.1 Classical Kalman filter for linear systems

The classical Kalman filter is based on Bayes rule Eq. (3.1) and the fact that statistically independent measurements can be assimilated sequentially

$$P(\mathbf{x}|\mathbf{y}_1, \mathbf{y}_2) \propto P(\mathbf{y}_2|\mathbf{x})P(\mathbf{y}_1|\mathbf{x})P(\mathbf{x}). \quad (3.44)$$

It works as follows:

- At $t = t_0$ no model data are yet available and the posterior probability of θ is equal to the Gaussian prior with mean μ_θ and covariance \mathbf{P}_θ . The expectation of this pdf, μ_θ , results in zero when substituted into Eq. (3.5), since the first and last term are already zero at $t = t_0$.
- If the numerical model is linear, it can be written as $\mathbf{x}_{t_j} = \mathbf{A}\mathbf{x}_{t_{j-1}} + \mathbf{B}\mathbf{u} + \mathbf{F}\varepsilon_j$, where \mathbf{x} contains the reservoir states augmented with the estimated parameters and \mathbf{u} and ε_j are deterministic and stochastic forcings respectively. If there are no measurements available as time increases, it suffices to update \mathbf{x} and \mathbf{P}_x by

$$\mathbf{x} \leftarrow \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \quad , \quad \mathbf{P}_x \leftarrow \mathbf{A}\mathbf{P}_x\mathbf{A}^T + \mathbf{F}\mathbf{P}_{\varepsilon_j}\mathbf{F}^T. \quad (3.45)$$

- If the measurement operator is linear ($\mathbf{y} = \mathbf{C}\mathbf{x}$) and measurements errors are assumed to be Gaussian with zero mean and known covariance \mathbf{P}_y , then \mathbf{x} can be calculated from a statistically optimal weighing between \mathbf{x} and \mathbf{y} :

$$\mathbf{x} \leftarrow \mathbf{x} + \mathbf{K}(\mathbf{y} - \mathbf{C}\mathbf{x}) \quad , \quad \mathbf{K} = \mathbf{P}_x\mathbf{C}^T(\mathbf{C}\mathbf{P}_x\mathbf{C}^T + \mathbf{P}_y)^{-1}. \quad (3.46)$$

Here $\mathbf{C}\mathbf{x}$ are the "predicted measurements" and \mathbf{y} contains the "measured measurements". The measurement update causes discontinuities of the state variables (augmented with model parameters) at points in time where measurements are available.

3.4.2 Ensemble Kalman filter for non-linear systems

Both the time update and the measurement update of the Kalman filter assume that the state variables augmented with the model parameters are Gaussian. This Gaussianity is lost when the prior is not Gaussian or when either the reservoir model or the measurement operator is non-linear. The measurement operator $\mathbf{y} = \mathbf{g}(\mathbf{x})$ can always be written in linear form by state augmentation,

$$\tilde{\mathbf{x}} := \begin{bmatrix} \mathbf{x} \\ \mathbf{g}(\mathbf{x}) \end{bmatrix} \quad , \quad \tilde{\mathbf{g}}(\tilde{\mathbf{x}}) := \begin{bmatrix} \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{g}(\mathbf{x}) \end{bmatrix} = \tilde{\mathbf{C}}\tilde{\mathbf{x}}, \quad (3.47)$$

however, this only shifts the source of non-Gaussianity in the measurement update instead of solving it. All sub-optimal approximations of the Kalman filter that were designed to deal with nonlinearities calculate the first and second order moments of the posterior, even though these are not representative for a probability that is possibly much richer than a multivariate Gaussian. In the Extended Kalman filter (EKF) the reservoir flow equations are linearized and the Jacobians are substituted into Eq. (3.45) for \mathbf{A} and \mathbf{F} . For a system of the form

$$\begin{bmatrix} \mathbf{x}_{t_j} \\ \boldsymbol{\theta} \end{bmatrix} = \mathbf{f} \left(\begin{bmatrix} \mathbf{x}_{t_{j-1}} \\ \boldsymbol{\theta} \end{bmatrix} \right) + \boldsymbol{\varepsilon}_j = \left[\tilde{\mathbf{f}} \left(\mathbf{x}_{t_{j-1}}, \boldsymbol{\theta} \right) \right] + \boldsymbol{\varepsilon}_j, \quad (3.48)$$

these Jacobians are

$$\mathbf{A} = \begin{bmatrix} \frac{\partial \tilde{\mathbf{f}}}{\partial \mathbf{x}} \left(\mathbf{x}_{t_{j-1}}, \boldsymbol{\theta} \right) & \frac{\partial \tilde{\mathbf{f}}}{\partial \boldsymbol{\theta}} \left(\mathbf{x}_{t_{j-1}}, \boldsymbol{\theta} \right) \\ \mathbf{0} & \mathbf{I} \end{bmatrix}, \quad \mathbf{F} = \mathbf{I}. \quad (3.49)$$

The same is done with the measurement operator and Eq. (3.46) for the measurement update, or $\tilde{\mathbf{C}}$ (Eq. (3.47)) is used. The first moment of the posterior calculated by the EKF is biased. The Ensemble Kalman Filter (EnKF) corrects this.

The Ensemble Kalman Filter (EnKF) [Evensen (2003)] was designed to deal with non-linear models or simulators that do not have \mathbf{A} and \mathbf{F} from Eq. (3.45) explicitly available. The covariance matrix \mathbf{P}_x is decomposed as

$$\mathbf{P}_x = \frac{\mathbf{L}\mathbf{L}^T}{n-1}, \quad (3.50)$$

where the columns of \mathbf{L} represent statistical deviations from the mean reservoir state. To update \mathbf{P}_x correctly requires as many reservoir simulations as the dimension of the state vector (#grid blocks * #phases * ...). The EnKF (under)estimates \mathbf{P}_x by randomly sampling a lower number of columns for \mathbf{L} , typically 100. Alternatives are available where the sampling is based on eigendecompositions. Also hybrid methods exist that combine deterministic and random sampling. When the mean reservoir state $\bar{\mathbf{x}}^f$ is added to the deviations $\tilde{\mathbf{x}}_i^f$, an ensemble of reservoir state vectors is obtained. These are independently propagated by the non-linear model for the time update. The measurement update for the EnKF can be written as

$$\begin{aligned} \bar{\mathbf{x}}^a &= \bar{\mathbf{x}}^f + \mathbf{K} (\bar{\mathbf{y}}^o - \mathbf{C}\bar{\mathbf{x}}^f) & , & \quad \tilde{\mathbf{x}}_i^a = \tilde{\mathbf{x}}_i^f + \mathbf{K} (\tilde{\mathbf{y}}_i^o - \mathbf{C}\tilde{\mathbf{x}}_i^f), & \quad (3.51) \\ \mathbf{K} &= \mathbf{L}\tilde{\mathbf{L}}^T \left(\tilde{\mathbf{L}}\tilde{\mathbf{L}}^T + (n-1)\mathbf{P}_y \right)^{-1} & , & \quad \tilde{\mathbf{L}} = \mathbf{C}\mathbf{L}. \end{aligned}$$

The application

- All ensemble members are updated independently from each other by a reservoir simulator. The model errors are sampled randomly. Optionally the model parameters may be updated simultaneously by some appropriate stochastic process.

- When a measurement becomes available, all ensemble members are augmented with the model parameters that are to be estimated, as well as the non-linear "forecasted measurements". The matrix \mathbf{L} is formed by putting all these augmented ensemble members as columns in a matrix.
- The matrix $\tilde{\mathbf{L}}$ is created by deleting the rows of \mathbf{L} that correspond to quantities that can not directly be measured. This can vary over the different assimilation points in time, making the assimilation step very flexible to different kinds of sensors with their corresponding frequencies.
- The Kalman gain matrix is calculated. If the sensors are (nearly) independent, then \mathbf{P}_y is (close to) a diagonal matrix and easily invertible. Calculating the full inverse of $\tilde{\mathbf{L}}\tilde{\mathbf{L}}^T + (n-1)\mathbf{P}_y$ may result in numerical problems when different sensors have very different accuracies, or when there is a large range in the state error (co)variance. Using Woodbury's [Woodbury (1950)] equations for low-rank updating of matrix inverses partially takes care of that.
- Using the Kalman gain, the observation is assimilated into every ensemble member separately. In order to do so, an ensemble of observations is created from one observation by randomly sampling from the probability distribution of the sensor error.

Extrapolating measurements to reservoir states and parameters

$\tilde{\mathbf{L}} = \mathbf{C}\mathbf{L}$, Eq. (3.51), means that \mathbf{C} operating on \mathbf{L} cuts away certain rows, creating $\tilde{\mathbf{L}}$. The inverse is non-unique, but a pseudo-inverse exists; $\mathbf{L} \approx \mathbf{C}^T\tilde{\mathbf{L}}$. In other words, a pseudo-inverse \mathbf{L} can be recreated by adding zero-rows to $\tilde{\mathbf{L}}$ at places where rows were cut away. These zeros were not present in the original \mathbf{L} . Using the pseudo-inverse $\mathbf{C}^{-1} \approx \mathbf{C}^T$, the Kalman gain matrix Eq. (3.51) can be written as

$$\mathbf{K} \approx \mathbf{C}^T\tilde{\mathbf{L}}\tilde{\mathbf{L}}^T \left(\tilde{\mathbf{L}}\tilde{\mathbf{L}}^T + (n-1)\mathbf{P}_y \right)^{-1}. \quad (3.52)$$

Now assuming that the error covariance structure of the observations is equal to the error covariance structure of the state variables that can be directly measured ($(n-1)\mathbf{P}_y = \alpha\tilde{\mathbf{L}}\tilde{\mathbf{L}}^T$), the Kalman gain becomes

$$\mathbf{K} = \frac{\mathbf{C}^T}{1+\alpha}, \quad (3.53)$$

meaning that the measurement update becomes

$$\mathbf{x}_o^a = \frac{\alpha}{1+\alpha}\mathbf{x}_o^f + \frac{1}{1+\alpha}\mathbf{y} \quad , \quad \mathbf{x}_{no}^a = \mathbf{x}_{no}^f, \quad (3.54)$$

or in other words, for as far as the states than can be directly measured, the analyzed values are weighted averages between the forecasted values and the observations.

It can be concluded that there are two mechanisms responsible for extrapolating observations to parameters that can not be directly measured:

1. The error covariance of the observations differs significantly from the error covariance of the observable states ($(n-1) \mathbf{P}_y \neq \alpha \tilde{\mathbf{L}} \tilde{\mathbf{L}}^T$).
2. \mathbf{L} is not formed by adding zero-rows to $\tilde{\mathbf{L}}$, but from the original state data, including the covariance between the state variables that can be directly measured and the state variables that cannot be directly measured.

Ensemble Square Root Kalman Filter

Correlated measurements

From Eq. (3.51) it follows that the reservoir state error covariance is updated by

$$\begin{aligned} \mathbf{P}_x^a &= \overline{\tilde{\mathbf{x}}_i^a \tilde{\mathbf{x}}_i^{aT}} = \overline{\left((\mathbf{I} - \mathbf{K}\mathbf{C}) \tilde{\mathbf{x}}_i^f + \mathbf{K} \tilde{\mathbf{y}}_i^o \right) \left((\mathbf{I} - \mathbf{K}\mathbf{C}) \tilde{\mathbf{x}}_i^f + \mathbf{K} \tilde{\mathbf{y}}_i^o \right)^T} = \quad (3.55) \\ &= (\mathbf{I} - \mathbf{K}\mathbf{C}) \overline{\tilde{\mathbf{x}}_i^f \tilde{\mathbf{x}}_i^{fT}} (\mathbf{I} - \mathbf{K}\mathbf{C})^T + \overline{\mathbf{K} \tilde{\mathbf{y}}_i^o \tilde{\mathbf{y}}_i^{oT}} \mathbf{K} = \\ &= (\mathbf{I} - \mathbf{K}\mathbf{C}) \mathbf{P}_x^f (\mathbf{I} - \mathbf{K}\mathbf{C})^T + \mathbf{K} \mathbf{P}_y \mathbf{K}^T. \end{aligned}$$

If no noise would be added to the observations, then \mathbf{P}^a would be underestimated by the first term, while the second term would be missing. The idea of the Ensemble Square Root Kalman Filter (ESRKF) is to update the mean and deviations of the state without adding noise to the observations,

$$\begin{cases} \bar{\mathbf{x}}^a = \bar{\mathbf{x}}^f + \mathbf{K} (\bar{\mathbf{y}}^o - \mathbf{C} \bar{\mathbf{x}}^f) \\ \tilde{\mathbf{x}}_i^a = \tilde{\mathbf{x}}_i^f - \tilde{\mathbf{K}} \mathbf{C} \tilde{\mathbf{x}}_i^f \end{cases}, \quad (3.56)$$

in such a way that the underestimation of the state error covariance is compensated for. Obviously, $\tilde{\mathbf{K}}$ must be chosen such that

$$(\mathbf{I} - \tilde{\mathbf{K}}\mathbf{C}) \mathbf{P}_x^f (\mathbf{I} - \tilde{\mathbf{K}}\mathbf{C})^T = (\mathbf{I} - \mathbf{K}\mathbf{C}) \mathbf{P}_x^f (\mathbf{I} - \mathbf{K}\mathbf{C})^T + \mathbf{K} \mathbf{P}_y \mathbf{K}^T. \quad (3.57)$$

Whitaker and Hamill [Whitaker and Hamill (2002)] have shown that

$$\tilde{\mathbf{K}} = \mathbf{P}_x^f \mathbf{C}^T \left(\sqrt{\left(\mathbf{C} \mathbf{P}_x^f \mathbf{C}^T + \mathbf{P}_y \right)^{-1}} \right)^T \left(\sqrt{\mathbf{C} \mathbf{P}_x^f \mathbf{C}^T + \mathbf{P}_y} + \sqrt{\mathbf{P}_y} \right)^{-1}, \quad (3.58)$$

hence the name SQUARE ROOT filter. These square roots are non-unique and can be calculated by Cholesky or SVD. Because the square roots are non-unique, choices must be made. [Evensen (2004)] claims that it is necessary to make sure there is some variation in these consecutive choices.

Uncorrelated measurements

When the measurements are uncorrelated (\mathbf{P}_y is a diagonal matrix), they can be assimilated sequentially. Thus, without loss of generality it can be assumed that \mathbf{C}^T is equal to a unit vector ($\mathbf{C}^T = \mathbf{e}_j$) and \mathbf{P}_y is scalar. Now

$$\tilde{\mathbf{K}} = \frac{\mathbf{K}}{1 + \sqrt{\frac{\mathbf{P}_y}{(\mathbf{C}\mathbf{P}_x^f\mathbf{C}^T + \mathbf{P}_y)^{-1}}}}, \quad (3.59)$$

and updating the deviations can be done by $\mathbf{L} := \mathbf{L} - \tilde{\mathbf{K}}\tilde{\mathbf{L}}$

3.4.3 EnKF; towards a non-linear measurement update

Whenever measurements are available, there is a jump in the analyzed states. When the current estimate is close to the true value, then this jump will be small. When the measurements are very inaccurate, the analyzed state will be much closer to the forecasted state than the observation, so the jump will also be small. In case of accurate measurements, the jump may be large. Many large jumps cause filter divergence or at best poor estimates, because the magnitude of the jumps is overestimated by the linear measurement update. For numerical stability it is often better to have many inaccurate measurements available than a few accurate ones. This section attempts to duplicate an accurate measurement into several less accurate measurements.

Uncorrelated measurements

In 1D (uncorrelated measurements are assimilated sequentially) two Gaussian stochastic variables, a reservoir state and an observation, with parameters (μ_f, σ_f) and (μ_o, σ_o) are combined by a Kalman update to a new Gaussian with parameters (μ_a, σ_a) , where

$$\sigma_a^2 = \frac{\sigma_f^2 \sigma_o^2}{\sigma_f^2 + \sigma_o^2}, \quad \mu_a = \frac{\sigma_o^2 \mu_f + \sigma_f^2 \mu_o}{\sigma_f^2 + \sigma_o^2}. \quad (3.60)$$

Alternatively to assimilating x_o once with uncertainty σ_o , it can also be attempted to assimilate x_o n times sequentially with higher uncertainty $\tilde{\sigma}_o$. In that case

$$\sigma_a^2|_0 = \sigma_f^2, \quad \sigma_a^2|_i = \frac{\sigma_a^2|_{i-1} \tilde{\sigma}_o^2}{\sigma_a^2|_{i-1} + \tilde{\sigma}_o^2}, \quad \sigma_a^2 = \sigma_a^2|_n, \quad (3.61)$$

or equivalently

$$\sigma_a^2 = \frac{\sigma_f^2 \tilde{\sigma}_o^2}{2^{n-1} \sigma_f^2 + \tilde{\sigma}_o^2}. \quad (3.62)$$

The number of duplicate measurements must be specified by the user. This thesis proposes to let σ_a be invariant under this approach; only the discontinuity in the first statistical moment is reduced while the second moment is left unchanged. From Eq. (3.60) and Eq. (3.62), it follows that

$$n = 1 + \log_2 \left(\frac{\tilde{\sigma}_o^2 \sigma_f^2 + \sigma_o^2}{\sigma_o^2 \sigma_f^2 + \tilde{\sigma}_o^2} \right) \quad \text{or} \quad \tilde{\sigma}_o^2 = 2^{n-1} \sigma_o^2. \quad (3.63)$$

Example: if $\tilde{\sigma}_a = \sigma_f$ (bring the uncertainty of "measured measurements" to the order of "predicted measurements"), then

$$n = 1 + \log_2 \left(\frac{\sigma_f^2 + \sigma_o^2}{2\sigma_o^2} \right), \quad (3.64)$$

so if $\sigma_f^2 = 10^{11}$ and $\sigma_o^2 = 10^3$, then $n \approx 26.58$. Setting $n = 27$ gives $\tilde{\sigma}_o^2 = 6.7 \cdot 10^{10}$.

3.4.4 Smoother

A long term production optimization algorithm typically runs over a time interval from the present to the end of the estimated lifetime of the reservoir. The input for such an algorithm is a reservoir simulator that has been history-matched using all available data up to the present. Both a reservoir simulator that has been history-matched using a variational method or a reservoir simulator that has been history-matched using a filter satisfy this condition. However, the history of the former simulator is consistent with all data, whereas only the present time of the latter simulator is consistent with all data. This is due to the sequential nature of a filter; data improves the present estimate of a reservoir simulator but does not have an impact back in time. Applications exist where it is necessary to estimate the past state of a reservoir using measurements taken at a later point in time. Variational methods can be used for such applications without modifications. A filter must be modified into a smoother; a filter estimates the present given the history, a smoother estimates the present given the history and the future (or it estimates the past given the history and the present). Mathematically, the results of a smoother can be seen as if obtained by a statistically optimal weighing of the outcome of two filters, one running forward and one running backward in time. Implementing a smoother is quite a lot harder than a filter.

In current implementations of closed-loop reservoir management, the production optimization and data assimilation are decoupled by an iterative scheme. Based on all available knowledge, an optimal production strategy is calculated. This strategy is then applied to the reservoir simulator and in the field. Measurements are assimilated whenever they become available. A new optimal strategy is obtained from the updated simulator. When the measurement errors are not correlated in time, a filter is quite suited in closed-loop reservoir management. Otherwise, a smoother must be used or other modifications must be made to a filter [section 3.4.5].

3.4.5 Adapting a filter to handle seismic data with time-correlated errors

3D seismic data can be assimilated into a reservoir simulator using a filter, provided that there is a forward model that predicts seismic data based on the outcome of the reservoir simulator. If seismic data are inverted to grid block pressures and saturations, they can be assimilated as if originating from production measurements. Different levels in between are also possible. In 4D seismic data inversion, the data typically consists of two (or more) sets of repeated data at different points in time. Usually 4D inversion is done by inverting the difference in data back to pressure changes and saturation changes rather than pressures and saturations themselves. Both sets of 3D data contain large uncertainties, but this is partially cancelled out by the differencing. 4D data seismic data can be assimilated using a smoother and has been done in [Skjervheim *et al.* (2005)].

This thesis proposes an alternative. Instead of differencing two sets of 3D data to reduce the uncertainty, both sets of 3D data can be assimilated by a method that explicitly takes uncertainties into account. A standard filter can not handle time-correlated data, so a modification needs to be made. An Ensemble Kalman filter can be used with a double ensemble size. Every ensemble member has a (non-identical) twin brother in the ensemble. Whenever production data are available, these are assimilated into both ensembles, but keeping the two ensembles separate. Whenever measurements become available that are correlated to previous measurements, half of the ensemble members are destroyed; the ones that contained the seismic data. New twin brothers are created as exact copies of the ensemble members that only contain production data. Now the seismic data are assimilated into the duplicate ensemble, keeping the original ensemble unharmed. In order to assimilate all 3D data sets simultaneously, the duplicate ensemble members also need to be augmented with historic versions of the original ensemble members.

3.4.6 Other low-order approximations

Other low order approximations exist, where the columns of \mathbf{L} , Eq. (3.50), are chosen differently. In the RRSQRT-filter (Reduced Rank Square Root) [Verlaan and Heemink (1997)], the ensemble members are updated in time without randomly sampling model errors. Instead a square root of the model error covariance is concatenated to \mathbf{L} , $\widehat{\mathbf{L}} = \left[\mathbf{L} \mid \left((n-1) \mathbf{P}_{\varepsilon_j} \right)^{1/2} \right]$, after which $\widehat{\mathbf{L}}$ is reduced back to its original size by selecting the n leading singular vectors. The POEnKF (Partial Orthogonal EnKF) [Verlaan and Heemink (1997)] and COFFEE (Complementary Orthogonal subspace Filter For Efficient Ensembles) [Heemink *et al.* (2001)] combine an ensemble of singular vectors with a randomly sampled ensemble. The spaces spanned by these two ensembles overlap partially. In POEnKF this overlap is ignored, in COFFEE the random ensemble members are projected onto the complement of the space that is spanned by the singular vector ensemble.

3.5 Hybrid method

3.5.1 Expectation maximization

The Expectation Maximization (EM) [Dellaert (2002), Bilmes (1998)] attempts to find the parameters θ that best explain the data \mathbf{y} by maximizing the likelihood function

$$L(\theta) := \ln p(\mathbf{y}|\theta). \quad (3.65)$$

The maximization problem is simplified by introducing auxiliary distribution functions $p(\mathbf{x})$ over the hidden variables \mathbf{x} . These variables are needed to forward predict the data given the parameters and contain the reservoir states. The likelihood can be written as

$$L(\theta) := \ln p(\mathbf{y}|\theta) = \ln \int p(\mathbf{x}, \mathbf{y}|\theta) d\mathbf{x} = \ln \int p(\mathbf{x}) \frac{p(\mathbf{x}, \mathbf{y}|\theta)}{p(\mathbf{x})} d\mathbf{x}, \quad (3.66)$$

and a lower bound can be found because of the concaveness of the \ln -function

$$L(\theta) \geq \int p(\mathbf{x}) \ln \frac{p(\mathbf{x}, \mathbf{y}|\theta)}{p(\mathbf{x})} d\mathbf{x} = \int (p(\mathbf{x}) \ln p(\mathbf{x}, \mathbf{y}|\theta) - p(\mathbf{x}) \ln p(\mathbf{x})) d\mathbf{x}. \quad (3.67)$$

The EM algorithm iterates over an expectation step and a maximization step.

- In the expectation step of the η -th iteration, the lower bound Eq. (3.67) is maximized with respect to the distribution functions over the hidden variables $p(\mathbf{x})$, while keeping the parameters fixed. It can be shown [Bilmes (1998)] that the maximum is obtained when

$$p_\eta(\mathbf{x}) = p(\mathbf{x}|\theta_{\eta-1}, \mathbf{y}), \quad (3.68)$$

for which the bound becomes an equality.

- In the M-step, the lower bound is maximized with respect to the parameters, while keeping $p(\mathbf{x})$ fixed, so

$$\theta_\eta = \arg \max_{\theta} \int p_\eta(\mathbf{x}) \ln p(\mathbf{x}, \mathbf{y}|\theta) d\mathbf{x}. \quad (3.69)$$

3.5.2 Integrating a filter in a variational method

The first (μ_x) and second (\mathbf{P}_x) order moments of $p_\eta(\mathbf{x})$ from the E-step, Eq. (3.68), can be calculated by an (Ensemble) Kalman smoother. Unlike what was discussed in section 3.4.4, the model parameters need not be augmented to the reservoir states. According to Eq. (3.68), or equivalently

$$\theta_\eta = \arg \min_{\theta} E_{p_\eta(\mathbf{x})} \{-\ln p(\mathbf{x}, \mathbf{y}|\theta)\} = \arg \min_{\theta} E_{p_\eta(\mathbf{x})} J, \quad (3.70)$$

the expectation of the objective function Eq. (3.5) with respect to $p_\eta(\mathbf{x})$, is minimized with respect to the parameters θ . Taking this expectation comes down to replacing the terms $(\mathbf{y} - \boldsymbol{\mu}_y)^T \mathbf{P}_y^{-1} (\mathbf{y} - \boldsymbol{\mu}_y)$ and $\boldsymbol{\varepsilon}^T \mathbf{P}_\varepsilon^{-1} \boldsymbol{\varepsilon}$ of Eq. (3.5) by $(\mathbf{x} - \boldsymbol{\mu}_x)^T \mathbf{P}_x^{-1} (\mathbf{x} - \boldsymbol{\mu}_x)$. Minimizing the objective

$$J = (\mathbf{x} - \boldsymbol{\mu}_x)^T \mathbf{P}_x^{-1} (\mathbf{x} - \boldsymbol{\mu}_x) + (\boldsymbol{\theta} - \boldsymbol{\mu}_\theta)^T \mathbf{P}_\theta^{-1} (\boldsymbol{\theta} - \boldsymbol{\mu}_\theta) \quad (3.71)$$

can be done with the existing adjoint reservoir simulator; the real measurements \mathbf{y} can be replaced by the analyzed reservoir states from the forward simulation \mathbf{x} .

Like a classical variational method, a gradient of the objective with respect to the parameters can be obtained by Eq. (3.19) and Eq. (3.21) applied to the ensemble mean of the forward simulations. Like a filter/smoothen with parameters augmented to the state, it is also possible to update the second order moment of the parameter probability distribution using EM. In order to do so, an ensemble of parameters must be sampled from the prior distribution, and Eq. (3.19) and Eq. (3.21) must be applied to every ensemble member separately.

For the full EM method, a gradient-based search algorithm in the M-step must converge before the E-step is again performed. Optionally, the search algorithm may be truncated after several (or just one) iteration. This will increase the number of iterations of the EM algorithm, but will reduce the computational costs of the M-step. It might also converge to a better parameter estimate.

The smoother in the E-step may be replaced by a filter [Swepe (1973), te Stroet (1995)]. Implementing a filter is significantly easier than implementing a smoother. The adjoint equations must be adapted to handle discontinuous trajectories, which is a small effort. The variational algorithm might need more iterations with a filter than a smoother but does not necessarily result in a better or a worse parameter estimate.

3.6 Relation between data assimilation methods

At first glance, Kalman filtering/smoothing and variational data assimilation might look like totally different methods. In a Kalman smoother, the prior probability distribution of the model parameters is updated by sequential time integration steps and measurement steps. The time update takes into account model errors that are not caused by errors in the parameters. The results are the first (mean) and second (covariance) order statistical moments of the posterior distribution of the reservoir states and model parameters conditioned on the data. In a variational method, the squared difference of the predicted and observed measurements is minimized, regularized by the squared deviations of the parameters from the prior. Optionally, model errors can be taken into account, meaning that the reservoir simulator is used as a weak constraint. If model errors are not taken into account, the reservoir simulator acts as a strong constraint. The different terms in the squared objective function can be given weights, in the form of scaling matrices.

Although smoothers and variational methods can be formulated independently, they both can be derived from a higher probabilistic principle, Bayes rule. The scaling matrices in the data-misfit objective function must then be interpreted as covariance matrices. When the prior is Gaussian and the system and sensor models are linear, then the posterior is also Gaussian. The first and second order statistical moments that are calculated by the smoother, are therefore sufficient information for the full posterior distribution. A weak constraint variational method like the Representer Method, finds the minimum of the objective function at the point where the posterior density is maximal, called the mode. For a multivariate Gaussian, the mode and mean coincide, meaning that a Kalman Smoother and the Representer Method result in identical parameter estimates (ignoring roundoff error). Both methods give some extra information. A smoother provides, besides the parameter estimates, parameter uncertainties in the form of a covariance matrix. The Representer Methods also specifies the separate effect of every measurement in space and time on the final parameter estimate.

When the prior is not Gaussian, or the system or sensor model is non-linear, a smoother and a weak constraint variational method are not equivalent in the parameter estimates. A smoother still gives the first and second order statistical moments, but these are no longer sufficient to fully represent a probability function. A weak constraint variational method calculates the mode of something that hopefully resembles the posterior probability density.

In a smoother, the reservoir states and model parameters are always consistent with all data. Whenever measurements become available, the history of the reservoir states is updated to be consistent with the new model parameters. To make future predictions, it is often sufficient if the reservoir states and model parameters are consistent with all data for the current time, rather than the full history. A Kalman Filter is a simplification of a Kalman Smoother, where every measurement is used only to make the current reservoir states and model parameters consistent, without modifying the whole reservoir history. In fact, a Kalman smoother can be implemented as weighted average of two Kalman filters; one running forward in time and one running backward in time.

Chapter 4

An efficient weak-constraint gradient-based parameter estimation algorithm using representer expansions⁴

Abstract

The discrepancy between observed measurements and their model-predicted antitheses can be used to improve either the model output alone or both the model output and the parameters that underlie the model. In case of parameter estimation, methods exist that can efficiently calculate the gradient of the discrepancy to changes in the parameters, assuming that there are no uncertainties in addition to the unknown parameters. Usually many different parameter sets exist that minimize the discrepancy locally, so the gradient must be regularized before it can be used by gradient-based minimization algorithms. This chapter proposes a method for calculating a gradient in the presence of additional model errors, through the use of representer expansions. The representer are data-driven basis functions that perform the regularization. All available data can be used during every iteration of the minimization scheme, as is the case in the classical Representer Method (RM). However, the method proposed here also allows adaptive selection of different portions of the data during different iterations to reduce computation time. The user now has the freedom to choose the number of basis functions and revise this choice at every iteration. The method also differs from the classic RM by introducing measurement representer in addition to state, adjoint and parameter representer and by the fact that no correction terms are calculated. Unlike the classic RM, where the minimization scheme is prescribed, the RM proposed here provides a gradient that can be used in any minimization algorithm.

The applicability of the modified method is illustrated with a synthetic example to estimate permeability values in an inverted 5-spot waterflooding problem.

⁴ This chapter is based on [Rommelse *et al.* (2007)], which was published as TUD-DIAM report 07-05 and submitted to SPE Journal

4.1 Introduction

4.1.1 Gradient-based parameter estimation

Data assimilation methods aim to improve numerical models by comparing actual measurements of a physical system with the numerical model predictions of these measurements [Bennett (2002), Lewis *et al.* (2006), Evensen (2007), Oliver *et al.* (2008)]. As the parameters of the numerical model are changed, consequently the predicted state variables and the predicted measurements change. The discrepancy between the "measured measurements" and the "predicted measurements" can be used to update only the state variables (state estimation) or also the parameters (parameter estimation) in order to decrease this discrepancy. When only the state of the model is predicted, the model itself is not corrected. Alternatively, the model parameters and hence the model itself, may be changed until the predicted output is satisfactorily close to the measurements. Parameter estimation aims at improving the predictive ability of the model, whereas state estimation attempts to find an initial estimate for a model that is assumed to already have a good predictive ability for the prediction interval of interest. Estimating initial states falls in the category parameter estimation, estimating all other states is state estimation. When the output of the model is used to make decisions, state estimation is appropriate for time scales on which the error in the model's predictive ability can be neglected. When the model is used for making long-term decisions, parameter estimation algorithms must be used.

This chapter focusses on gradient-based parameter estimation algorithms. More precisely, it proposes a method for calculating the gradient of the discrepancy with respect to changes in the parameters, in the presence of model errors.

4.1.2 Model errors; strong and weak constraints

Often, the discrepancy between the actual measurements and the predicted measurements is formulated using the Euclidean norm. The objective of the data assimilation is then to minimize the square of this norm with respect to the model parameters while the numerical model is used as a (strong) constraint.

However, there is an additional phenomenon that may cause the discrepancy. The model is an approximation, so even if the parameters were known, the model might still produce incorrect output. These errors can be modelled as extra parameters, which are also added to the objective function. The model is then used as a weak constraint in the minimization problem.

4.1.3 Notation

The state variables at time t_i are denoted by $\mathbf{x}_i, i \in \{0, \dots, n\}$. The models of interest in this chapter are written in function format

$$\mathbf{g}(\mathbf{x}_i, \mathbf{x}_{i-1}, \boldsymbol{\theta}, \boldsymbol{\varepsilon}_i) = \mathbf{0}, \quad (4.1)$$

where the Jacobian $\frac{\partial(\mathbf{x}_i, \mathbf{x}_{i-1}, \boldsymbol{\theta}, \boldsymbol{\varepsilon}_i)}{\partial \mathbf{x}_i}$ is assumed to be never singular. The model parameters are collected in the vector $\boldsymbol{\theta}$ and the model errors on interval $[t_{i-1}, t_i]$ are contained in the vector $\boldsymbol{\varepsilon}_i$. The measurement operator operates on all state variables at all time steps and is denoted by

$$\mathbf{y} = \mathbf{h}(\mathbf{x}_{\{0, \dots, n\}}). \quad (4.2)$$

The initial states may be part of the parameter estimation process, so $\mathbf{x}_0 = \mathbf{x}_0(\boldsymbol{\theta})$. The minimization relies on the availability of the first (mean) and second (covariance) order statistics of the model parameters and model errors. These are denoted by $\boldsymbol{\theta}^{prior}, \mathbf{P}_\theta, \boldsymbol{\varepsilon}_i^{prior} = \mathbf{0}$ and $\mathbf{P}_{\boldsymbol{\varepsilon}_i}$.

In the so-called strong constraint case, the model errors are explicitly set to zero. The objective function that has to be minimized is

$$\begin{aligned} J = & \frac{1}{2} (\mathbf{h}(\mathbf{x}_{\{0, \dots, n\}}) - \mathbf{m})^T \mathbf{P}_y^{-1} (\mathbf{h}(\mathbf{x}_{\{0, \dots, n\}}) - \mathbf{m}) + \\ & + \frac{1}{2} (\boldsymbol{\theta} - \boldsymbol{\theta}^{prior})^T \mathbf{P}_\theta^{-1} (\boldsymbol{\theta} - \boldsymbol{\theta}^{prior}) + \sum_{i=1}^n \lambda_i^T \mathbf{g}(\mathbf{x}_i, \mathbf{x}_{i-1}, \boldsymbol{\theta}, \mathbf{0}), \end{aligned} \quad (4.3)$$

where \mathbf{m} contains the actual physical measurements, possibly taken at different times. \mathbf{P}_y represents the uncertainty in the measurements in the form of an error covariance matrix. The last term of Eq. (4.3) represents the system equations \mathbf{g} that have been adjoined to the objective function with the aid of Lagrange multipliers λ_i ; see e.g. [Bennett (2006), Lewis *et al.* (2006), Oliver *et al.* (2008)]. If the prior parameters and the measurement errors have a Gaussian probability distribution and the reservoir simulator \mathbf{g} and sensor model \mathbf{h} are linear, Eq. (4.3) can be interpreted in a probabilistic setting as Bayes rule for updating a prior. The estimate that is found by minimizing Eq. (4.3) is then equivalent to a posterior that represents the mean of the probability function of the model parameters $\boldsymbol{\theta}$ conditional to the measurements; see e.g. [Gavalas *et al.* (1976)], [Zhang *et al.* (2005)] and [Oliver *et al.* (2008)] for further petroleum-related references. If the assumptions of Gaussian prior parameters and measurement errors and linear functions \mathbf{g} and \mathbf{h} are not fulfilled, Eq. (4.3) does not have a physical or probabilistic interpretation.

When model errors ε_i are taken into account they become additional parameters in the minimization process. They are assumed to be zero-mean, so the objective becomes

$$\begin{aligned}
 J = & \frac{1}{2} (\mathbf{h}(\mathbf{x}_{\{0, \dots, n\}}) - \mathbf{m})^T \mathbf{P}_y^{-1} (\mathbf{h}(\mathbf{x}_{\{0, \dots, n\}}) - \mathbf{m}) + \\
 & + \frac{1}{2} (\boldsymbol{\theta} - \boldsymbol{\theta}^{prior})^T \mathbf{P}_{\boldsymbol{\theta}}^{-1} (\boldsymbol{\theta} - \boldsymbol{\theta}^{prior}) + \frac{1}{2} \sum_{i=1}^n \varepsilon_i^T \mathbf{P}_{\varepsilon_i}^{-1} \varepsilon_i + \\
 & + \sum_{i=1}^n \lambda_i^T \mathbf{g}(\mathbf{x}_i, \mathbf{x}_{i-1}, \boldsymbol{\theta}, \varepsilon_i).
 \end{aligned} \tag{4.4}$$

In the data assimilation literature equation Eq. (4.4) is referred to as the weak constraint problem. For realistically-sized problems in reservoir engineering, minimization problems Eq. (4.3) and Eq. (4.4) are most efficiently solved using gradient-based methods where the gradient is obtained with an adjoint or co-state formulation. For petroleum-related applications of strong-constraint least-squares minimization problems, see e.g. [Chavent *et al.* (1975)], [Ruijian *et al.* (2003)], [Rodrigues (2006)] and [Oliver *et al.* (2008)] for further references.

4.1.4 Representer Method

The Representer Method (RM) was introduced in oceanography as an efficient gradient-based technique to solve the weak constraint least-squares minimization problem ([Bennett and McIntosh (1982)], [Eknes and Evensen (1997)]). However, although the representer method was introduced as a method to solve the weak constraint problem, it may equally well be applied to the strong constraint problem. Subsequent work by [Bennett (2002)] also addressed nonlinear applications for state estimation. [Baird and Dawson (2005)] applied the method to linear state estimation in single-phase reservoir flow, [Valstar *et al.* (2004)] extended it to nonlinear parameter estimation in ground water flow, and [Rommelse *et al.* (2006)], [Przybysz *et al.* (2007)] and [Baird and Dawson (2007)] further developed the method for use in two-phase reservoir flow. The RM simultaneously decomposes the deviation of the estimated parameters from the prior parameters into the isolated effects of every measurement. This regularizes the minimization problem and it also gives information that can be used to quantify the usefulness of every single measurement.

In this chapter, a modification of the classic RM is derived as a postprocessor that evaluates the effect of the measurements on the solution of the weak constraint minimization problem. It is then reformulated to produce a regularized gradient that can be used by any gradient-based minimization algorithm in order to find the solution of the weak constraint minimization problem. The method allows decomposition of the parameter vector into a general number of basis functions rather than a number that necessarily needs to be equal to the number of measurements as in the classical method. This makes the method computationally more attractive for applications where many measurements are available. Due to

a different linearization, no correction terms have to be calculated, as was the case in the earlier versions of the RM when applied to non-linear problems.

4.2 Gradient of the strong constraint minimization problem

4.2.1 Obtaining a gradient

The derivatives of Eq. (4.4) with respect to λ_i , ε_i , \mathbf{x}_i and θ are

$$\left(\frac{\partial J}{\partial \lambda_i}\right)^T = \mathbf{g}(\mathbf{x}_i, \mathbf{x}_{i-1}, \theta, \varepsilon_i), \quad (4.5)$$

$$\left(\frac{\partial J}{\partial \varepsilon_i}\right)^T = \mathbf{P}_{\varepsilon_i}^{-1} \varepsilon_i + \left(\frac{\partial \mathbf{g}(\mathbf{x}_i, \mathbf{x}_{i-1}, \theta, \varepsilon_i)}{\partial \varepsilon_i}\right)^T \lambda_i, \quad (4.6)$$

$$\begin{aligned} \left(\frac{\partial J}{\partial \mathbf{x}_i}\right)^T &= \left(\frac{\partial \mathbf{h}}{\partial \mathbf{x}_i}\right)^T \mathbf{P}_y^{-1} (\mathbf{h}(\mathbf{x}_{\{0, \dots, n\}}) - \mathbf{m}) + \\ &+ \left(\frac{\partial \mathbf{g}(\mathbf{x}_i, \mathbf{x}_{i-1}, \theta, \varepsilon_i)}{\partial \mathbf{x}_i}\right)^T \lambda_i + \left(\frac{\partial \mathbf{g}(\mathbf{x}_{i+1}, \mathbf{x}_i, \theta, \varepsilon_{i+1})}{\partial \mathbf{x}_i}\right)^T \lambda_{i+1}, \end{aligned} \quad (4.7)$$

$$\left(\frac{\partial J}{\partial \theta}\right)^T = \mathbf{P}_\theta^{-1} (\theta - \theta^{prior}) + \sum_{i=1}^n \left(\frac{\partial \mathbf{g}(\mathbf{x}_i, \mathbf{x}_{i-1}, \theta, \varepsilon_i)}{\partial \theta}\right)^T \lambda_i. \quad (4.8)$$

Eq. (4.5) and Eq. (4.6) are valid for $i \in \{1, \dots, n\}$, whereas Eq. (4.7) is valid for $i \in \{1, \dots, n-1\}$. For t_n , the term including λ_{i+1} is missing from Eq. (4.7). Alternatively, $\lambda_{n+1} = \mathbf{0}$ may be introduced. For t_0 , the term including λ_i is missing. In case the initial states \mathbf{x}_0 are part of the parameter estimation process, the term

$$\left(\frac{\partial \mathbf{x}_0(\theta)}{\partial \theta}\right)^T \left(\frac{\partial \mathbf{g}(\mathbf{x}_1, \mathbf{x}_0, \theta, \varepsilon_1)}{\partial \mathbf{x}_0}\right)^T \lambda_1 \quad (4.9)$$

should be added to Eq. (4.8). For the strong constraint case, where ε_i is explicitly set to $\mathbf{0}$, the gradient of the objective function with respect to the model parameters, $\left(\frac{\partial J}{\partial \theta}\right)^T$, can be calculated using Eq. (4.8), where the model states \mathbf{x}_i and adjoint states λ_i follow from sequentially solving Eq. (4.5) and Eq. (4.7) with the left-hand sides, $\left(\frac{\partial J}{\partial \lambda_i}\right)^T$ and $\left(\frac{\partial J}{\partial \mathbf{x}_i}\right)^T$, set to zero. Eq. (4.6) does not need to be used; instead $\varepsilon_i = \mathbf{0}$ is used.

4.2.2 Using the gradient

To solve the strong constraint minimization problem, a numerical routine must be implemented that evaluates J as given in Eq. (4.3), and $\left(\frac{\partial J}{\partial \theta}\right)^T$, as given in section 4.2.1. This routine can then be passed to any gradient-based minimization software package, together with a set of initial parameters (usually $\theta^{init} = \theta^{prior}$) and some appropriate minimization options that are algorithm-dependent.

Often the objective function has multiple local minima and the minimization process needs to be regularized. If low-order parameters \mathbf{b} are introduced such that $\theta = \theta^{prior} + \mathbf{Q}\mathbf{b}$, with $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}$, then a regularized gradient can be found as $\mathbf{Q}\mathbf{Q}^T \left(\frac{\partial J}{\partial \theta}\right)^T$. The orthogonal matrix \mathbf{Q} can for example be obtained by selecting several left-singular vectors (section 4.3.7) of a square root \mathbf{L} of the covariance matrix $\mathbf{P}_\theta = \mathbf{L}\mathbf{L}^T$.

4.3 Gradient of the weak constraint minimization problem

4.3.1 Local minimizer

In a stationary point (denoted by superscript s) of Eq. (4.4), all gradients are equal to zero, so

$$\mathbf{g}(\mathbf{x}_i^s, \mathbf{x}_{i-1}^s, \boldsymbol{\theta}^s, \boldsymbol{\varepsilon}_i^s) = \mathbf{0}, \quad (4.10)$$

$$\boldsymbol{\varepsilon}_i^s = -\mathbf{P}_{\varepsilon_i} \left(\frac{\partial \mathbf{g}(\mathbf{x}_i, \mathbf{x}_{i-1}, \boldsymbol{\theta}, \boldsymbol{\varepsilon}_i)}{\partial \boldsymbol{\varepsilon}_i} \right)^T \boldsymbol{\lambda}_i^s, \quad (4.11)$$

$$\begin{aligned} & \left(\frac{\partial \mathbf{h}(\mathbf{x}_{\{0, \dots, n\}}^s)}{\partial \mathbf{x}_i^s} \right)^T \mathbf{P}_y^{-1} \left(\mathbf{m} - \mathbf{h}(\mathbf{x}_{\{0, \dots, n\}}^s) \right) \\ &= \left(\frac{\partial \mathbf{g}(\mathbf{x}_i^s, \mathbf{x}_{i-1}^s, \boldsymbol{\theta}^s, \boldsymbol{\varepsilon}_i^s)}{\partial \mathbf{x}_i^s} \right)^T \boldsymbol{\lambda}_i^s + \left(\frac{\partial \mathbf{g}(\mathbf{x}_{i+1}^s, \mathbf{x}_i^s, \boldsymbol{\theta}^s, \boldsymbol{\varepsilon}_{i+1}^s)}{\partial \mathbf{x}_i^s} \right)^T \boldsymbol{\lambda}_{i+1}^s, \end{aligned} \quad (4.12)$$

$$\mathbf{P}_\theta^{-1} (\boldsymbol{\theta}^{prior} - \boldsymbol{\theta}^s) = \sum_{i=1}^n \left(\frac{\partial \mathbf{g}(\mathbf{x}_i^s, \mathbf{x}_{i-1}^s, \boldsymbol{\theta}^s, \boldsymbol{\varepsilon}_i^s)}{\partial \boldsymbol{\theta}^s} \right)^T \boldsymbol{\lambda}_i^s. \quad (4.13)$$

Unlike in section 4.2.1, the forward equations Eq. (4.10) and the adjoint equations Eq. (4.12) are now coupled because the model errors $\boldsymbol{\varepsilon}_i^s$ are no longer equal to zero; they are related to the adjoint states $\boldsymbol{\lambda}_i^s$ by Eq. (4.11).

4.3.2 Basis functions

The minimization algorithm is started with $\boldsymbol{\theta}^{start} = \boldsymbol{\theta}^{prior}$ and $\boldsymbol{\varepsilon}_i^{start} = \boldsymbol{\lambda}_i^{start} = \mathbf{0}$. Applying these prior conditions to Eq. (4.10) gives the prior system states \mathbf{x}_i^{prior} . Given the prior states, also prior measurements can be predicted, $\mathbf{h}(\mathbf{x}_{\{0, \dots, n\}}^{prior})$. The causes for the variables to move away from their prior are parameterized by \mathbf{b} . In the classic RM there is a 1-1-relationship between one such cause and an isolated measurement in space and time. In this chapter, this assumption is abandoned. Moreover, for computational purposes, it is interesting to assume that the number of parameters in the vector \mathbf{b} is (much) smaller than the number of measurements.

The deviations from the priors are now decomposed as

$$\mathbf{x}_i^s - \mathbf{x}_i^{prior} = \mathbf{R}_{\mathbf{x}_i} \mathbf{b}, \quad (4.14)$$

$$\boldsymbol{\lambda}_i^s = \mathbf{R}_{\boldsymbol{\lambda}_i} \mathbf{b}, \quad (4.15)$$

$$\boldsymbol{\theta}^s - \boldsymbol{\theta}^{prior} = \mathbf{R}_{\boldsymbol{\theta}} \mathbf{b}, \quad (4.16)$$

$$\mathbf{h}(\mathbf{x}_{\{0, \dots, n\}}^s) - \mathbf{h}(\mathbf{x}_{\{0, \dots, n\}}^{prior}) = \mathbf{R}_{\mathbf{y}} \mathbf{b}. \quad (4.17)$$

The columns of $\mathbf{R}_{\mathbf{x}_i}$, $\mathbf{R}_{\boldsymbol{\lambda}_i}$, $\mathbf{R}_{\boldsymbol{\theta}}$ and $\mathbf{R}_{\mathbf{y}}$ contain the state representers, the adjoint representers, the parameter representers and the measurement representers respectively. When the measurement operator \mathbf{h} is linear, the measurement representers $\mathbf{R}_{\mathbf{y}}$ can be constructed by applying \mathbf{h} to the matrix that is obtained by concatenating the state representers $\mathbf{R}_{\mathbf{x}_i}$ as row blocks. Alternatively, the RM can be formulated in terms of state representers without defining measurement representers, as is done in the classic RM. The introduction of measurement representers only has added value when the measurement operator is non-linear. Theoretically, it is also possible to introduce error representers

$$\boldsymbol{\varepsilon}_i^s = \mathbf{R}_{\boldsymbol{\varepsilon}_i} \mathbf{b}. \quad (4.18)$$

However, looking at Eq. (4.11), these error representers are nothing more than modified adjoint representers

$$\mathbf{R}_{\boldsymbol{\varepsilon}_i} = -P_{\boldsymbol{\varepsilon}_i} \left(\frac{\partial \mathbf{g}(\mathbf{x}_i, \mathbf{x}_{i-1}, \boldsymbol{\theta}, \boldsymbol{\varepsilon}_i^s)}{\partial \boldsymbol{\varepsilon}_i} \right)^T \mathbf{R}_{\boldsymbol{\lambda}_i}, \quad (4.19)$$

and have no practical application.

4.3.3 Representer equations

Substitution of Eq. (4.14), Eq. (4.15) and Eq. (4.16) into Eq. (4.12) and Eq. (4.13) results in

$$\begin{aligned} & \left(\frac{\partial \mathbf{h}(\mathbf{x}_{\{0, \dots, n\}}^s)}{\partial \mathbf{x}_i^s} \right)^T \mathbf{P}_y^{-1} \left(\mathbf{m} - \mathbf{h}(\mathbf{x}_{\{0, \dots, n\}}^s) \right) \\ &= \left(\frac{\partial \mathbf{g}(\mathbf{x}_i^s, \mathbf{x}_{i-1}^s, \boldsymbol{\theta}^s, \boldsymbol{\varepsilon}_i^s)}{\partial \mathbf{x}_i^s} \right)^T \mathbf{R}_{\lambda_i} \mathbf{b} + \left(\frac{\partial \mathbf{g}(\mathbf{x}_{i+1}^s, \mathbf{x}_i^s, \boldsymbol{\theta}^s, \boldsymbol{\varepsilon}_i^s)}{\partial \mathbf{x}_i^s} \right)^T \mathbf{R}_{\lambda_{i+1}} \mathbf{b}, \end{aligned} \quad (4.20)$$

$$\mathbf{R}_\theta \mathbf{b} = -\mathbf{P}_\theta \sum_{i=1}^n \left(\frac{\partial \mathbf{g}(\mathbf{x}_i^s, \mathbf{x}_{i-1}^s, \boldsymbol{\theta}^s, \boldsymbol{\varepsilon}_i^s)}{\partial \boldsymbol{\theta}^s} \right)^T \mathbf{R}_{\lambda_i} \mathbf{b}. \quad (4.21)$$

Eq. (4.20) can be simplified by requiring

$$\mathbf{P}_y^{-1} \left(\mathbf{m} - \mathbf{h}(\mathbf{x}_{\{0, \dots, n\}}^s) \right) = \mathbf{Q} \mathbf{b}. \quad (4.22)$$

How to choose the selection matrix \mathbf{Q} will be explained in section 4.3.7. In the classical formulation, where $\mathbf{Q} = \mathbf{I}$, \mathbf{b} contains the differences between the observed and predicted measurements, decorrelated by \mathbf{P}_y . Using requirement Eq. (4.22) provides a means to calculate the *adjoint representer* \mathbf{R}_{λ_i} :

$$\begin{aligned} & \left(\frac{\partial \mathbf{g}(\mathbf{x}_i^s, \mathbf{x}_{i-1}^s, \boldsymbol{\theta}^s, \boldsymbol{\varepsilon}_i^s)}{\partial \mathbf{x}_i^s} \right)^T \mathbf{R}_{\lambda_i} \\ &= \left(\frac{\partial \mathbf{h}(\mathbf{x}_{\{0, \dots, n\}}^s)}{\partial \mathbf{x}_i^s} \right)^T \mathbf{Q} - \left(\frac{\partial \mathbf{g}(\mathbf{x}_{i+1}^s, \mathbf{x}_i^s, \boldsymbol{\theta}^s, \boldsymbol{\varepsilon}_i^s)}{\partial \mathbf{x}_i^s} \right)^T \mathbf{R}_{\lambda_{i+1}}. \end{aligned} \quad (4.23)$$

The *parameter representer* \mathbf{R}_θ follow by removing \mathbf{b} from Eq. (4.21)

$$\mathbf{R}_\theta = -\mathbf{P}_\theta \sum_{i=1}^n \left(\frac{\partial \mathbf{g}(\mathbf{x}_i^s, \mathbf{x}_{i-1}^s, \boldsymbol{\theta}^s, \boldsymbol{\varepsilon}_i^s)}{\partial \boldsymbol{\theta}^s} \right)^T \mathbf{R}_{\lambda_i}. \quad (4.24)$$

Optionally, the term

$$\left(\frac{\partial \mathbf{x}_0(\boldsymbol{\theta}^s)}{\partial \boldsymbol{\theta}^s} \right)^T \left(\frac{\partial \mathbf{g}(\mathbf{x}_1^s, \mathbf{x}_0^s, \boldsymbol{\theta}^s, \boldsymbol{\varepsilon}_1^s)}{\partial \mathbf{x}_0^s} \right)^T \mathbf{R}_{\lambda_1} \quad (4.25)$$

has to be added to Eq. (4.24) for estimating the initial states. The *state representers* $\mathbf{R}_{\mathbf{x}_i}$ are obtained by differentiating Eq. (4.10) with respect to the representer coefficients \mathbf{b} ;

$$\frac{\partial \mathbf{g}(\mathbf{x}_i^s, \mathbf{x}_{i-1}^s, \boldsymbol{\theta}^s, \boldsymbol{\varepsilon}_i^s)}{\partial \mathbf{x}_i^s} \mathbf{R}_{\mathbf{x}_i} + \frac{\partial \mathbf{g}(\mathbf{x}_i^s, \mathbf{x}_{i-1}^s, \boldsymbol{\theta}^s, \boldsymbol{\varepsilon}_i^s)}{\partial \mathbf{x}_{i-1}^s} \mathbf{R}_{\mathbf{x}_{i-1}} + \quad (4.26)$$

$$+ \frac{\partial \mathbf{g}(\mathbf{x}_i^s, \mathbf{x}_{i-1}^s, \boldsymbol{\theta}^s, \boldsymbol{\varepsilon}_i^s)}{\partial \boldsymbol{\theta}^s} \mathbf{R}_{\boldsymbol{\theta}} \quad (4.27)$$

$$= \frac{\partial \mathbf{g}(\mathbf{x}_i^s, \mathbf{x}_{i-1}^s, \boldsymbol{\theta}^s, \boldsymbol{\varepsilon}_i^s)}{\partial \boldsymbol{\varepsilon}_i^s} \mathbf{P}_{\boldsymbol{\varepsilon}_i} \left(\frac{\partial \mathbf{g}(\mathbf{x}_i, \mathbf{x}_{i-1}, \boldsymbol{\theta}, \boldsymbol{\varepsilon}_i)}{\partial \boldsymbol{\varepsilon}_i^s} \right)^T \mathbf{R}_{\boldsymbol{\lambda}_i}$$

and the same is done with Eq. (4.17) to obtain the *measurement representers* $\mathbf{R}_{\mathbf{y}}$;

$$\mathbf{R}_{\mathbf{y}} = \sum_{i=0}^n \frac{\partial \mathbf{h}(\mathbf{x}_{\{0, \dots, n\}}^s)}{\partial \mathbf{x}_i^s} \mathbf{R}_{\mathbf{x}_i}. \quad (4.28)$$

Substitution of Eq. (4.17) into Eq. (4.22) indicates that the *representer coefficients* \mathbf{b} should be obtained as the least-squares solution of

$$(\mathbf{R}_{\mathbf{y}} + \mathbf{P}_{\mathbf{y}} \mathbf{Q}) \mathbf{b} = \mathbf{m} - \mathbf{h}(\mathbf{x}_{\{0, \dots, n\}}^{\text{prior}}). \quad (4.29)$$

4.3.4 Representer Method as iterative minimizer

The representer method can be used as post-processor after a local minimum of Eq. (4.4) has been found by another method, in which case equations Eq. (4.23) and Eq. (4.24) need to be calculated. If no such local minimum has yet been found, the representer method can also be used in an attempt to approach a minimum by Picard iterations. The idea of Picard iterations is to solve $\boldsymbol{\theta}$ from $\mathbf{f}(\boldsymbol{\theta}) = \mathbf{0}$ by $\boldsymbol{\theta}_{i+1} = \boldsymbol{\theta}_i \pm \mathbf{f}(\boldsymbol{\theta}_i)$. Here $\mathbf{f}(\boldsymbol{\theta}) = \frac{dJ(\boldsymbol{\theta})}{d\boldsymbol{\theta}}$. The steps that need to be taken then are (superscript s now stands for estimate, rather than stationary point):

1. Initialize the parameter estimate $\boldsymbol{\theta}^s$ equal to the parameter prior $\boldsymbol{\theta}^{\text{prior}}$.
2. Initialize the adjoint states $\boldsymbol{\lambda}_i^s$ and model errors $\boldsymbol{\varepsilon}_i^s$ equal to zero.
3. Run the non-linear model Eq. (4.10).
4. Choose \mathbf{Q} , as will be discussed in section 4.3.7.
5. Calculate the adjoint representers Eq. (4.23).
6. Calculate the parameter representers Eq. (4.24).
7. Calculate the state representers Eq. (4.26).
8. Calculate the measurement representers Eq. (4.28).

9. Calculate new representer coefficients Eq. (4.29).
10. Calculate new adjoint states Eq. (4.15).
11. Calculate the model errors Eq. (4.11).
12. Calculate new parameters. Eq. (4.16) can be used, or a line search can be included;

$$\boldsymbol{\theta}_{new}^s = (1 - \alpha) \boldsymbol{\theta}_{old}^s + \alpha (\boldsymbol{\theta}^{prior} + \mathbf{R}_\theta \mathbf{b}). \quad (4.30)$$

13. Go to 3 if stopping criterion has not been fulfilled.

It could be argued that with these definitions of the representer and representer coefficients, Eq. (4.14) through Eq. (4.17), and with the Picard iterations used in this formulation of the representer method, the results from steps 7 and 9 cannot be substituted in Eq. (4.14) to reproduce the results of step 3, in contrast to the original formulation of the representer method [Valstar *et al.* (2004), Baird and Dawson (2005), Przybysz *et al.* (2007)]. This argument is true for the modified RM, but a similar argument can be given for the original RM. In the original formulation of the RM, a correction term is used and the states are decomposed around the states of the previous iteration, rather than around the prior states. The criterion for calculating this correction term, \mathbf{x}_i^{corr} , is in fact that the original version of Eq. (4.14),

$$\mathbf{x}_i^s - \left(\mathbf{x}_i^f - \mathbf{x}_i^{corr} \right) = \mathbf{R}_{\mathbf{x}_i} \mathbf{b}, \quad (4.31)$$

remains valid. However, a Picard-type of approximation is made to come up with a workable equation for the correction term \mathbf{x}_i^{corr} , so Eq. (4.31) only holds if a minimum of the objective function has been found and not during the iteration process.

4.3.5 Obtaining a gradient

The measurements can be predicted after step 3 of section 4.3.4 has finished. Together with the input parameters (step 1) and the model errors (step 11), the objective Eq. (4.4) can be evaluated. Instead of step 12, a direction that decreases the objective can be calculated as

$$\mathbf{d}_{decrease} = \boldsymbol{\theta}^{prior} + \mathbf{R}_\theta \mathbf{b} - \boldsymbol{\theta}_{old}^s, \quad (4.32)$$

so an (approximate) gradient is given by

$$\left(\frac{\partial J}{\partial \boldsymbol{\theta}^s} \right)^T = \boldsymbol{\theta}^s - \boldsymbol{\theta}^{prior} - \mathbf{R}_\theta \mathbf{b}. \quad (4.33)$$

4.3.6 Computational issues in using the gradient

The objective and its gradient cannot directly be used in standard gradient-based minimization algorithms because the interface of a standard minimization algorithm is usually not equipped to handle the adjoint states and model errors. A standard minimization algorithm provides a set of parameters to a routine that evaluates the objective and its gradient and then modifies the parameters. Such a routine for the representer method would contain steps 3 to 12 of section 4.3.4 with the modification of section 4.3.5. The routine also modifies the values of the adjoint states and model errors, which the minimization algorithm does not know about. Moreover, during a line search of the minimization algorithm, the gradient routine modifies the model errors. When the line search method rejects a step, the model errors must be reset to the last accepted values. With some modifications, the interface of a minimization algorithm can be enriched to handle the routine of the representer method that evaluates the objective function and its gradient. Because of these modification to the minimization algorithm, the method is not suited to interface with third party software without the availability of the source code. However, if the source code is available, making the modifications is almost trivial.

Unlike the gradient of the strong constraint problem that needs regularization, as was discussed in section 4.2.2, the gradient of the weak constraint problem is already regularized by the representer expansions.

4.3.7 Choosing \mathbf{Q}

Eq. (4.23) states that \mathbf{Q} must be chosen such that several rows of $\frac{\partial \mathbf{h}(\mathbf{x}_{\{0, \dots, n\}}^s)}{\partial \mathbf{x}_i^s}$ are removed by the multiplication

$$\mathbf{Q}^T \frac{\partial \mathbf{h}(\mathbf{x}_{\{0, \dots, n\}}^s)}{\partial \mathbf{x}_i^s}. \quad (4.34)$$

Here, the terms $\frac{\partial \mathbf{h}}{\partial \mathbf{x}_i^s}$ are column blocks of the full measurement sensitivity matrix

$$\mathbf{J}_h = \frac{\partial \mathbf{h}(\mathbf{x}_{\{0, \dots, n\}}^s)}{\partial \mathbf{x}_{\{0, \dots, n\}}^s} = \left[\frac{\partial \mathbf{h}(\mathbf{x}_{\{0, \dots, n\}}^s)}{\partial \mathbf{x}_0^s} \quad \dots \quad \frac{\partial \mathbf{h}(\mathbf{x}_{\{0, \dots, n\}}^s)}{\partial \mathbf{x}_n^s} \right]. \quad (4.35)$$

It is needed to remove rows of the full \mathbf{J}_h instead of the individual blocks. In other words, the same linear transformation is used to remove rows of all individual blocks, or \mathbf{Q} is the same for all time steps. Eq. (4.23) allows \mathbf{Q} to be different for different time steps, but then the operation in Eq. (4.29) would no longer be well-defined.

For example, a singular value decomposition of \mathbf{J}_h can be used, so $\mathbf{J}_h = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, where $\mathbf{\Sigma}$ has the form

$$\mathbf{\Sigma} = \begin{bmatrix} \sigma_1 & & & 0 & 0 \\ & \ddots & & 0 & 0 \\ & & \sigma_m & 0 & 0 \end{bmatrix}, \quad (4.36)$$

and the bottom part of \mathbf{V}^T is filled with zeros. Extra zero-rows can be created in the product $\Sigma \mathbf{V}^T = \mathbf{U}^T \mathbf{J}_h$ by setting the smallest singular values to zero, which is equivalent to removing the bottom rows of \mathbf{U}^T . Therefore the \mathbf{Q} matrix that is proposed in this chapter is

$$\mathbf{Q} = \mathbf{U}_{[:,1:k]}, \quad (4.37)$$

which means that \mathbf{Q} is formed by calculating the left-singular vectors of \mathbf{J}_h in a matrix \mathbf{U} and then keeping only k columns. The singular values in Σ can even help to make a decision on the number of representer functions k , based on a preservation of energy principle.

4.4 Numerical experiments: twin experiment

4.4.1 Inverted 5-spot

Experiments were done on a 2D 2-phase waterflooding application with wells in an inverted 5-spot configuration. Water is injected at a rate of one pore volume per year and the production wells are constrained to flow at 0.25 pore volumes per year. The state of the reservoir is described by pressure and water saturation in all $21 \times 21 \times 1$ grid blocks of $10 \times 10 \times 20$ m. Capillary pressure is ignored, as well as gravity effects. All other fluid and reservoir parameters have been listed in Tbl. (4.1). Synthetic data are generated by picking one realization out of a database of 1000 realizations as the "true" permeability, Fig. (4.1), and running a reservoir simulation with model errors that are sampled as white noise, Fig. (4.2). Note that although in reality production data consist of phase rates, bottom hole pressures or tubing head pressures, we will use the gridblock pressures as "measurements" to assess the performance of the algorithms. The pressure and saturation responses in the well gridblocks after simulating with these model errors are shown in Fig. (4.3). This figure also shows 10 synthetic pressure "measurements" in all well gridblocks at 100 and 200 days of simulation. The other realizations from the database are used to construct a covariance matrix that is used in the objective function that has to be minimized.

4.4.2 Reservoir simulator in weak or stochastic mode

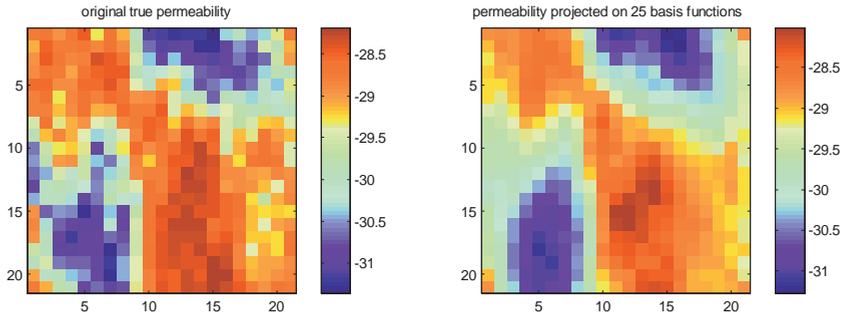
The 2-phase reservoir simulator can be written as

$$\frac{d}{dt} (\mathbf{f}_1(\mathbf{x})) = \mathbf{f}_2(\mathbf{x}, \boldsymbol{\theta}), \quad (4.38)$$

where \mathbf{x} contains the water saturation and water pressure (equal to oil pressure) for every grid block and $\boldsymbol{\theta}$ contains the permeabilities of all grid blocks. \mathbf{f}_1 describes the presence of

Symbol	Variable	Value	SI units	Value	Field units
h	Gridblock height	20	m	65.62	ft
$\Delta x, \Delta y$	Gridblock length/width	10	m	32.81	ft
μ_o	Oil dynamic viscosity	1.0×10^{-3}	$Pa \cdot s$	1.0	cP
μ_w	Water dynamic viscosity	1.0×10^{-3}	$Pa \cdot s$	1.0	cP
c_t	Total compressibility	1.0×10^{-8}	Pa^{-1}	7.0×10^{-5}	psi^{-1}
p_R	Initial reservoir pressure	10×10^6	Pa	1450.4	psi
$k_{r,o}^0$	Endpoint relative permeability, oil			1.0	—
$k_{r,w}^0$	Endpoint relative permeability, water			0.5	—
n_o	Corey exponent, oil			2.0	—
n_w	Corey exponent, water			2.0	—
S_{or}	Residual oil saturation			0.2	—
S_{wc}	Connate water saturation			0.2	—
ϕ	porosity			0.3	—

Table 4.1: Reservoir and fluid properties for the example

Figure 4.1: True permeability data $[\ln(m^2)]$ used to synthesize pressure data in the wells and best possible reconstruction using 25 basis functions.

water and oil mass in the grid blocks and \mathbf{f}_2 models the flow through the grid block interfaces. Injection/production is modelled as sources/sinks, which are included in \mathbf{f}_2 .

A fully implicit time discretization is used;

$$\tilde{\mathbf{g}}(\mathbf{x}_i, \mathbf{x}_{i-1}, \boldsymbol{\theta}) = \mathbf{f}_1(\mathbf{x}_i) - (t_i - t_{i-1}) \mathbf{f}_2(\mathbf{x}_i, \boldsymbol{\theta}) - \mathbf{f}_1(\mathbf{x}_{i-1}) = \mathbf{0}. \quad (4.39)$$

The model errors are introduced as additional sources/sinks in all grid blocks. In other words, after \mathbf{x}_i has been solved from Eq. (4.39), the water and oil masses in the grid blocks have not correctly been predicted and must still be modified. The prediction deteriorates as the time step $(t_i - t_{i-1})$ gets larger. Therefore the correction is modelled proportionally to $(t_i - t_{i-1})$. If the additional sources become too strong, then unrealistically high pressures will be observed. If the additional sinks become too strong, then saturations outside $[0, 1]$ will occur. In this chapter, the additional sinks are non-linearly constrained by \mathbf{f}_1 . The stochastic reservoir simulator has the form

$$\begin{aligned} & \mathbf{g}(\mathbf{x}_i, \mathbf{x}_{i-1}, \boldsymbol{\theta}, \boldsymbol{\varepsilon}_i) \\ = & \mathbf{f}_1(\mathbf{x}_i) - (t_i - t_{i-1}) \mathbf{f}_2(\mathbf{x}_i, \boldsymbol{\theta}) - \mathbf{f}_1(\mathbf{x}_{i-1}) + \min\{\mathbf{f}_1(\mathbf{x}_i), (t_i - t_{i-1}) \boldsymbol{\varepsilon}_i\} = \mathbf{0}. \end{aligned} \quad (4.40)$$

For the synthetic truth, $\boldsymbol{\varepsilon}_i$ is generated as white noise, Fig. (4.2). Applying this stochastic forcing to the reservoir simulator results in wiggly pressure and saturation responses in the wells, Fig (4.3). Although the pressure is not smooth in time, it is still smooth in space, Fig (4.4). This is not the case for the water saturation.

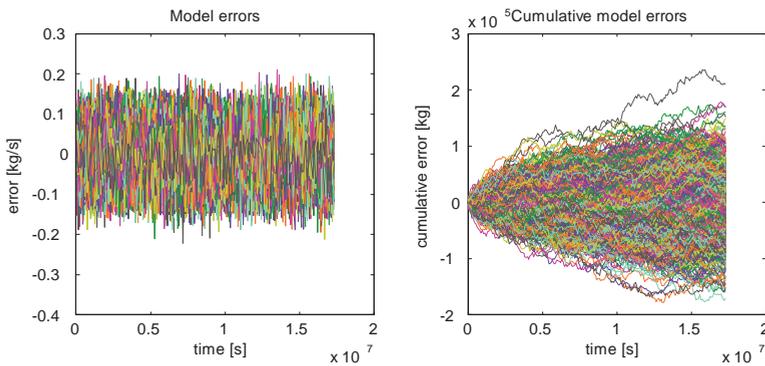


Figure 4.2: High model errors as functions of time; the $\boldsymbol{\varepsilon}_i$'s are sampled for all grid blocks and defined on daily intervals. They are plotted at the end of the intervals.

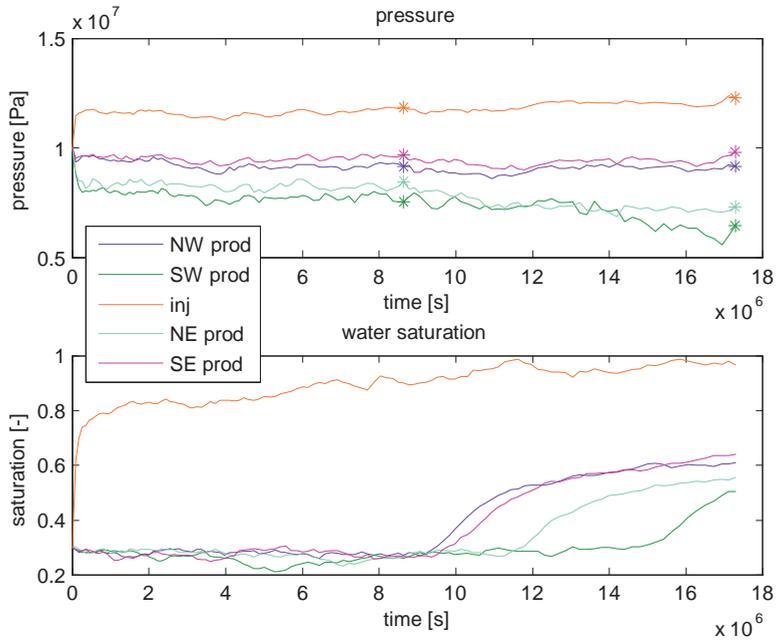


Figure 4.3: Pressure and saturation response in the well gridblocks when simulating with large model errors. The top plot also shows the 10 synthetic measurements, indicated by asterisks

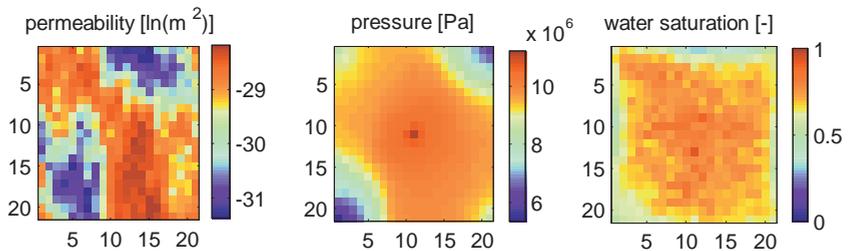


Figure 4.4: Spatial plots of permeability, pressure and water saturation at 200 days

4.4.3 Permeability reconstructed

Four cases were examined with zero, low, middle and high model errors with standard deviations of 0 , $5 \cdot 10^{-4}$, $5 \cdot 10^{-3}$ and $5 \cdot 10^{-2}$ [$kg\ s^{-1}$] respectively. For the first case, the gradient of the strong constraint problem was used and regularized by the leading 25 left-singular vectors of \mathbf{P}_θ . The best possible permeability that can be reconstructed using these basis functions is shown in Fig. (4.1). The gradient of the weak constraint problem was obtained and regularized by the Representer Expansions. For the strong constraint case, experiments were done with fixed basis functions and with Representers. In the former case, the classic adjoint method provides the gradient, in the latter case the gradient is obtained from the Representer Method. Steps 2, 10 and 11 of section 4.3.4 can be ignored when the RM is used for solving a strong constraint problem. The gradients were used in a steepest descent scheme and in the LBFGS [Gao and Reynolds (2006), Ulbrich (2002)] algorithm. Three different line search strategies were tried; a fixed step size, a step size that decreases exponentially while the objective function does not decrease and a quadratic line search. In the quadratic line search, the objective function along the line is approximated by a parabola, given the current value of the objective, its slope along the line and the value at a potential step size. A new potential step size is found at the minimum of the parabola. It is similar to Wolfe conditions [Nocedal and Wright (1999)] in the sense that it uses the gradient to determine the direction of the line search as well as using it to estimate the optimal step size. However, the Wolfe conditions contain some extra options to apply a relaxation factor or to be less greedy, whereas the quadratic line search method does not. Fig (4.5), Fig (4.6) and Fig (4.7) show the decrease of the objective function as function of iteration number, step size and the product of step size and the norm of the gradient, for both minimization algorithms, all four cases, and three different line search strategies. In general, the objective functions of the different cases contain different weight factors and are therefore not comparable. However, the synthetic true model errors are created in such a way, that they contribute identically to the objective functions of the high, middle and low cases.

The objective of the strong constraint problem gradually decreases when the adjoint method and steepest descent are used for the minimization. (L)BFGS and RM both require additional computation time per iteration. This can be compared to making an effort to calculate some approximation of the 2nd order derivatives of the objective function, the Hessian. Applying BFGS does not contribute extra when the RM is also applied. However, the BFGS does have a clear effect on the strong constraint problem that is solved with the adjoint method; with a relatively simple but effective line search algorithm (exponential), BFGS converges significantly faster than steepest descent. Although BFGS provides a better search direction than SD, this does not help the convergence if a fixed step size is used. For the strong constraint case with the adjoint method, the quadratic line search provides reduced computation time per iteration as well as faster convergence. In this case the advantage of BFGS over SD can no longer be observed. The quadratic line search does not work properly in combination with RM, because the slope of the objective along the search direction cannot be calculated accurately. In order to calculate this slope correctly, the dot product of the gradient and the search direction needs to be calculated. In case of SD with the adjoint method, the search direction is equal to the gradient. In case of BFGS, the gradient is modified by an approxi-

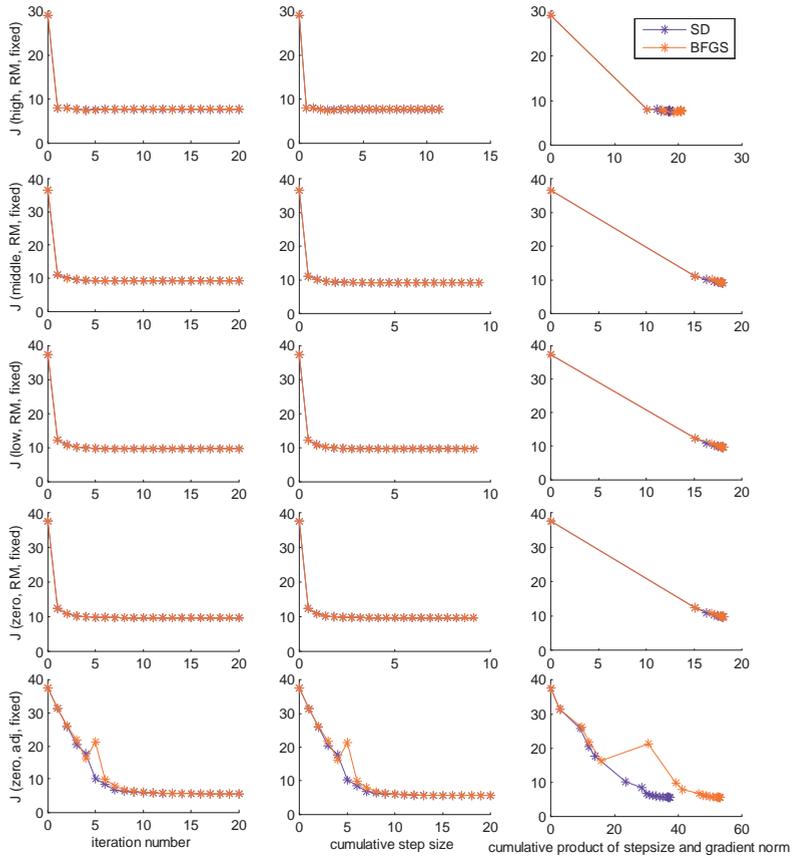


Figure 4.5: Three views on the objective functions; as function of iteration number, step size, and step size multiplied by the norm of the gradient. From top to bottom: high level model errors to zero model errors. RM is used for the weak constraint cases. Both RM and the adjoint with 25 svd basis functions are used for the strong constraint case. Steepest descent and LBFGS were applied. The step size is fixed.

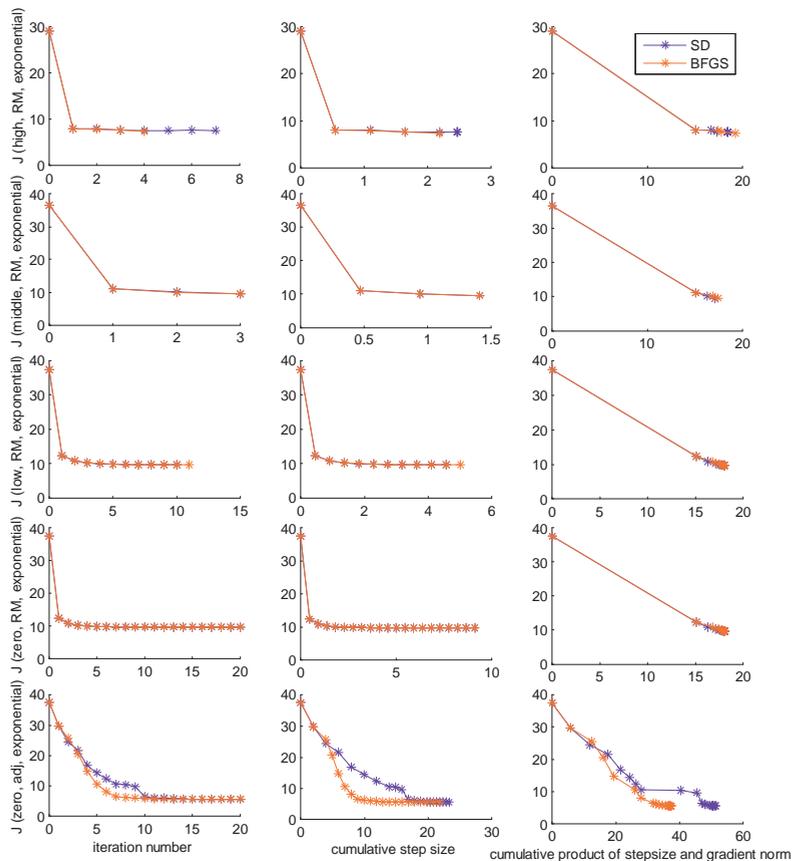


Figure 4.6: Three views on the objective functions; as function of iteration number, step size, and step size multiplied by the norm of the gradient. From top to bottom: high level model errors to zero model errors. RM is used for the weak constraint cases. Both RM and the adjoint with 25 svd basis functions are used for the strong constraint case. Steepest descent and LBFGS were applied. The step size was exponentially decreased until the objective improved.

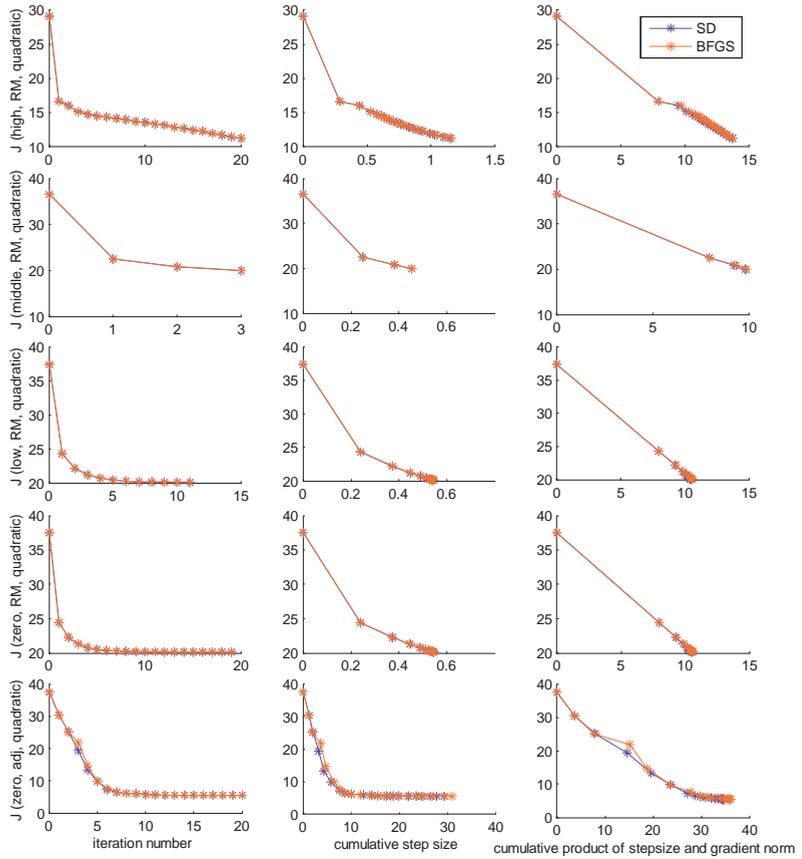


Figure 4.7: Three views on the objective functions; as function of iteration number, step size, and step size multiplied by the norm of the gradient. From top to bottom: high level model errors to zero model errors. RM is used for the weak constraint cases. Both RM and the adjoint with 25 svd basis functions are used for the strong constraint case. Steepest descent and L-BFGS were applied. The step size was determined by a quadratic line search.

mation of the Hessian to obtain the search direction. In case of RM, a regularized gradient or search direction is obtained directly, while the true gradient is unknown. As a result, the quadratic line search underestimates the step size and the algorithm stops before even finding a local minimum of the objective function.

The RM only needs one iteration for a linear problem (linear simulator (one fluid phase), estimating the initial state while the model parameters are assumed to be known, additive model errors). For more complicated problems more iterations are needed, but still most of the decrease of the objective function is achieved in the first iteration. RM converges even faster in case of higher model errors. This is due to the fact that the non-linear relationship between the model parameters and the state variables get overshadowed by the model errors, making the problem more linear.

The RM is greedier than the adjoint method; much more work is done per iteration, but less iterations are needed. It seems that the RM is punished slightly for its greed, because for the strong constraint case the objective function value of the adjoint method is a little smaller than the RM after convergence.

The prior permeability and the true permeability as well as the final reconstructions for the cases with zero, low, middle and high model errors using SD are shown in Fig. (4.8). In the strong constraint case, the gradient was regularized using 25 basis functions that were obtained as left-singular vectors of the permeability covariance matrix.

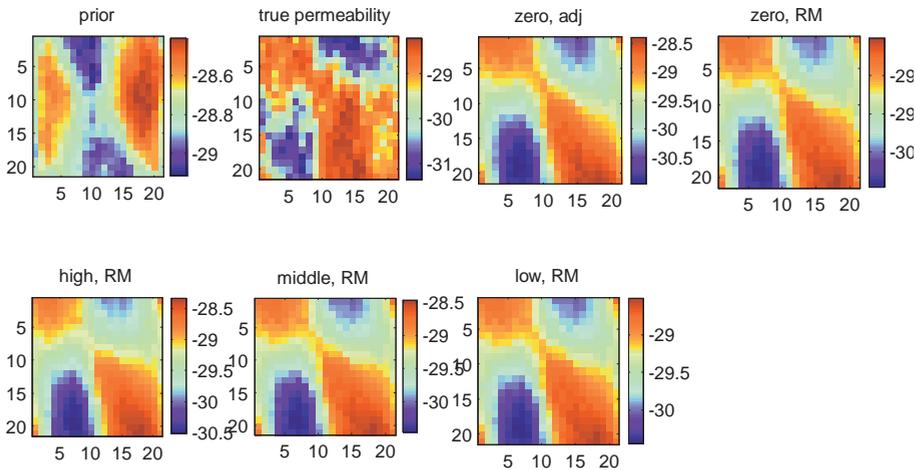


Figure 4.8: Prior permeability and final estimates in the cases with zero/low/middle/high model errors. The classical adjoint with 25 basis functions was used for the case with no model errors. RM was used for the other cases.

4.4.4 Additional output from minimization process

Besides reconstructing parameters, the (modified) RM gives additional information. Fig. (4.9) shows the model errors that were reconstructed by SD with exponential line search for the case where the truth was synthesized using high model errors. Compared to the original, Fig. (4.2), they are underestimated and smoothed. All the high resolution model errors that were not time-correlated cannot be reconstructed by the measurements that have a much lower time resolution. Consequently, artificial time correlation of the model errors is introduced by the RM. Fig. (4.10) shows the reconstructed pressure and saturation responses in the well gridblocks. These are smoothed as well. The parameter representers multiplied by their representer coefficients are plotted in Fig. (4.11). The different scales show that some measurements have a larger impact on the final permeability estimate than others, both in space (different columns) and time (different rows). Measurements at the same location at different moments in time affect the estimated permeability similarly and therefore result in similar representer functions. Fig. (4.12) shows all representers on the same scale.

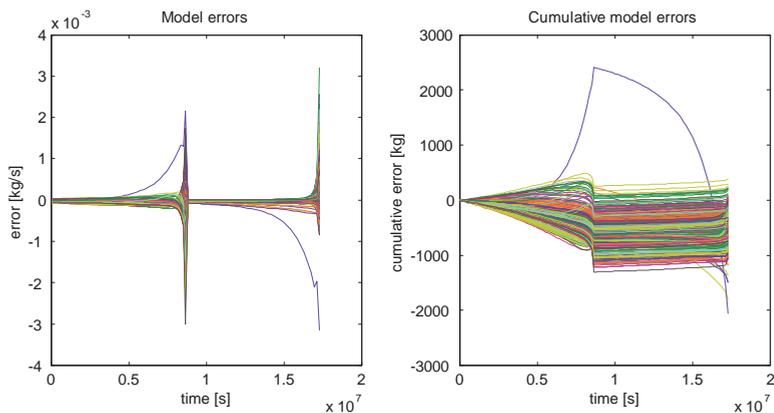


Figure 4.9: High model errors reconstructed by SD with exponential line search.

4.4.5 Order reduction

Fig. (4.8) was obtained without any order reduction (\mathbf{Q} was chosen equal to the identity matrix). Fig. (4.13) was created using an order reduction by a factor two; \mathbf{Q} is obtained by a permutation of the columns of the identity matrix and then adding the right most columns to the left most columns. The resulting columns are then normalized. The permutation is different and random for every iteration. In every iteration, the objective decreases by steepest

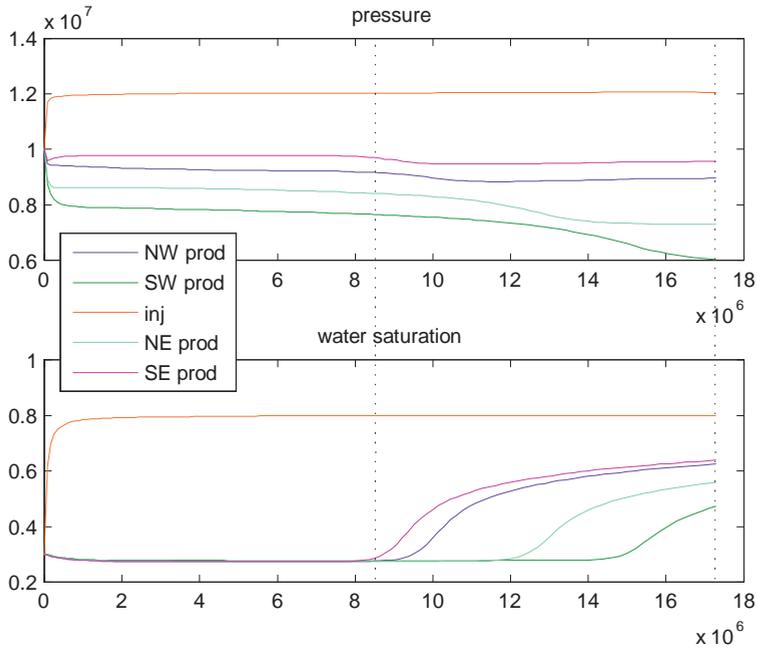


Figure 4.10: Pressure and saturation response in the well gridblocks reconstructed by LBFGS using the 10 pressure measurements in the 5 wells at the 2 measurement times indicated by the dotted lines.

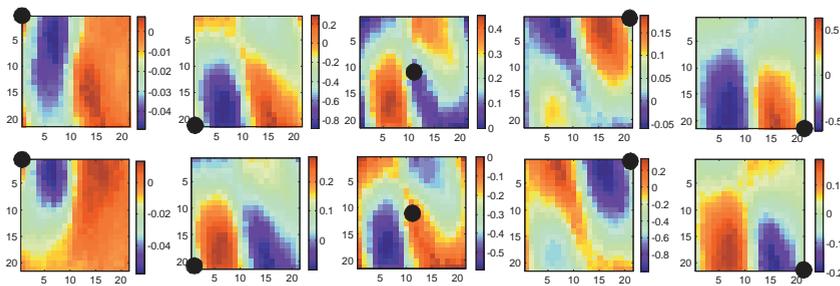


Figure 4.11: Parameter representers. Top row: pressure measurements after 100 days; bottom row: after 200 days; middle column: measurements obtained from injection well; other columns: north-west (NW), SW, NE and SE producers, indicated by the black bullets.

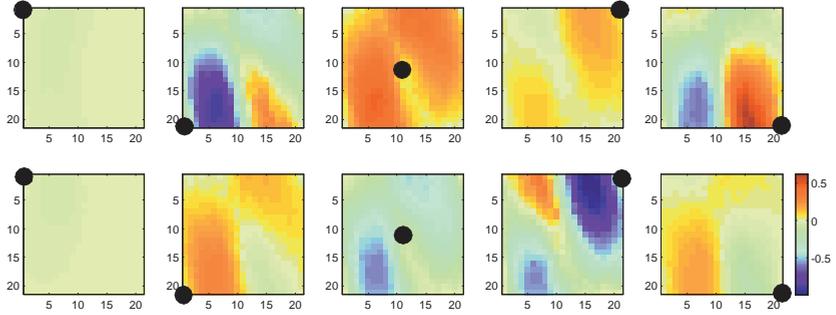


Figure 4.12: Scaled parameter representers. Top row: pressure measurements after 100 days; bottom row: after 200 days; middle column: measurements obtained from injection well; other columns: north-west (NW), SW, NE and SE producers, indicated by the black bullets.

descent and exponential line search. Since in every iteration \mathbf{Q} is chosen differently, also the objective function has a different interpretation in every iteration. Therefore it appears in Fig. (4.13) that the objective function sometimes increases, whereas in fact it is a slightly different objective function. If the columns of \mathbf{Q} are lumped too much, at some stage LBFGS reaches the stopping criterion after zero iterations.

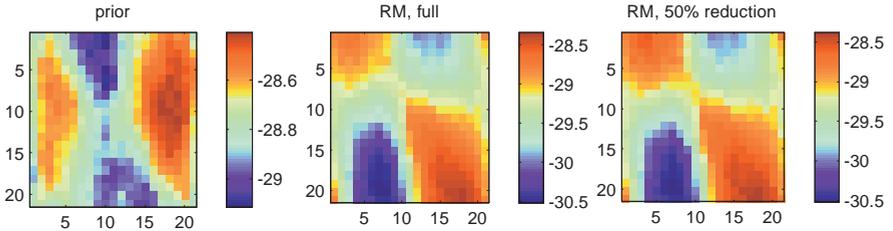


Figure 4.13: Prior permeability and final estimate for the case with high model errors using full RM and RM with 50% reduction. Steepest descend with exponential line search was used.

4.5 Discussion

4.5.1 Strong constraint solver and the RM as post-processor

In theory, the weak constraint minimization problem can easily be turned into a strong constraint minimization problem by treating the model errors as additional model parameters;

$$\tilde{\theta} = [\theta^T \quad \varepsilon_1^T \quad \dots \quad \varepsilon_n^T]^T. \quad (4.41)$$

Now any strong constraint solver can be used to solve the weak constraint problem. However, strong constraint solvers depend to a high degree on regularization techniques or methods to reduce the order of the parameter space. Usually basis functions are chosen and used during the entire minimization process. The result is then accepted as the solution of the minimization problem. The RM discussed in this chapter can be used as a post-processor to evaluate the outcome of the strong constraint solver and to update or overwrite the user-defined basis function to initialize a new strong constraint estimation procedure.

4.5.2 Variable time steps

Most modern simulators are equipped with a time-stepping mechanism that detects instabilities or unphysical values for the state variables and decreases the time step accordingly. Whenever possible the time step is increased again to reduce computation time. In an iterative method the length and the number of time steps therefore vary.

Building a strong constraint minimization problem out of a weak constraint one, as described in section 4.5.1, is not possible when successive iterations use different time steps, because the parameter vector Eq. (4.41) is only defined for one iteration. However, the modified RM can still be used. The model errors ε_i^s that were calculated in the old iteration must be interpolated to run the model in the new iteration Eq. (4.10). Here an integral average,

$$\varepsilon_i^j = \frac{1}{t_i^j - t_{i-1}^j} \int_{t_{i-1}^j}^{t_i^j} \varepsilon^{j-1} dt, \quad (4.42)$$

is used, where ε^{j-1} is the step function that is defined by

$$\{t_0^{j-1}, \dots, t_n^{j-1}\} \times \{\varepsilon_1^{j-1}, \dots, \varepsilon_n^{j-1}\} \quad (4.43)$$

from the old iteration.

An improvement in the parameters may cause the simulator to use more time steps, which could make the term $\frac{1}{2}\varepsilon^T P_\varepsilon^{-1} \varepsilon$ with $\varepsilon^T = [\varepsilon_1 \quad \dots \quad \varepsilon_n]^T$ in Eq. (4.4) increase dispropor-

tionately. Normalization factors can be added, so

$$\begin{aligned}
 J = & \frac{1}{2|\mathbf{m}|} (\mathbf{h}(\mathbf{x}_{\{0,\dots,n\}}) - \mathbf{m})^T \mathbf{P}_y^{-1} (\mathbf{h}(\mathbf{x}_{\{0,\dots,n\}}) - \mathbf{m}) + \\
 & + \frac{1}{2|\boldsymbol{\theta}|} (\boldsymbol{\theta} - \boldsymbol{\theta}^{prior})^T \mathbf{P}_\theta^{-1} (\boldsymbol{\theta} - \boldsymbol{\theta}^{prior}) + \frac{1}{2|\boldsymbol{\varepsilon}|} \boldsymbol{\varepsilon}^T \mathbf{P}_\varepsilon^{-1} \boldsymbol{\varepsilon} + \\
 & + \sum_{i=1}^n \lambda_i^T \mathbf{g}(\mathbf{x}_i, \mathbf{x}_{i-1}, \boldsymbol{\theta}, \boldsymbol{\varepsilon}_i),
 \end{aligned} \tag{4.44}$$

where $|\cdot|$ stands for counting the number of elements in a vector.

4.5.3 Measure of success

Variational data assimilation methods are designed to minimize some data-misfit objective, Eq. (4.4). Their success can be measured by which (local) minimum they can find and how fast they can find it. However, different performance measures can be explored as well. Figures Fig. (4.3) and Fig. (4.10) can be compared for example. When water breaks through in production wells, these become financially less profitable and eventually have to be shut in. The goal is to predict water breakthrough long before the water actually arrives at the production wells, so different control strategies can be applied to postpone the water breakthrough. How well the saturation profiles in the well gridblocks are reconstructed can be used as an alternative measure of success for a data assimilation algorithm. The difference between Fig. (4.3) and Fig. (4.10) must therefore be quantified somehow. [Cheng *et al.* (2005)] proposes to shift the curves in time to find a best fit; the shift quantifies how well the water breakthrough is estimated in time, the fit quantifies how well the behavior of the water during the breakthrough is estimated.

4.5.4 Use of parameter representers to quantify the impact of measurements

Fig. (4.11) shows the effect of every measurement on the final parameter estimate. Even when the modified RM is used in order-reduced mode, one extra iteration can be made after convergence of the method to produce all the parameter representers by running in full mode. Using the parameter representers, the usefulness of measurements must somehow be quantified, preferably expressed in terms of money. Care must be taken when interpreting these quantities. For example, a measurement can give a better understanding of the sub-surface, but it might also indicate that oil production will be lower than prognosed. This does not mean that the impact of the measurement should be quantified with a negative number. Research in this area is ongoing. Once the effect of measurements can be quantified, representers may be of help in designing measurement strategies.

4.5.5 Data selection

Attempts have been made to preprocess the data and discard the data with the most uncertainty to reduce the number of representer functions and reduce the computation time. [Schwaighofer and Tresp (2003)] mentions the Random and the Sparse Greedy Matrix Approximation (SGMA) versions of the Subset of Representers Method (SRM). Such preprocessing is based on the measurement uncertainty matrix \mathbf{P}_y and remains unchanged during the minimization process. This chapter proposes to choose a different preprocessing of the data at every iteration based on the measurement sensitivity matrix $\mathbf{J}_h = \frac{\partial \mathbf{h}(\mathbf{x}_{\{0, \dots, n\}}^s)}{\partial \mathbf{x}_{\{0, \dots, n\}}^s}$. Since the preprocessing itself costs computation time, the selection from the full dataset can also be used for more than one iteration. These two criteria can also be combined. In that case \mathbf{P}_y moves from Eq. (4.29) to Eq. (4.23) and the choice of \mathbf{Q} is based on $\mathbf{P}_y^{-1} \mathbf{J}_h$, which looks like a good compromise between how much the user trusts the measured data (\mathbf{P}_y) and how sensitive the forecasted measurements are to changes in the state variables. These criteria are also used in the Greedy Posterior Approximation version of SRM [Schwaighofer and Tresp (2003)], although there the number of basis functions is fixed.

4.5.6 Regularization

One might think that solving the representer coefficients \mathbf{b} from Eq. (4.22) instead of Eq. (4.29) makes the state representers, Eq. (4.26), and measurement representers, Eq. (4.28), obsolete. This may be true if extra, user-defined, regularization is applied to the resulting gradient, since the state- and measurement representer functions are part of the regularization.

4.5.7 Computational efficiency

Calculating one gradient comes down to one non-linear simulation to compute the reservoir states, and two sets of linear simulations to compute the adjoint/parameter representers and the state/measurement representers. No reduction is necessary for measurements that are sparse in time and space. For a large number of measurements, the full RM is not feasible. At every iteration different portions of the data must be selected to keep the number of representers limited. However this selection, based on svd, requires additional computation time. A good balance is application-dependent.

4.6 Conclusion

4.6.1 Recapitulation

This chapter introduces a modified formulation of the representer method. Since it can handle non-linear dynamics, non-linear measurement operators and non-linear model errors, it can deal with situations that are more realistic than previous implementations [Valstar *et al.* (2004), Baird and Dawson (2005), Rommelse *et al.* (2006), Przybysz *et al.* (2007), Baird and Dawson (2007)]. The derivation was explained and the method was illustrated by estimating the permeability of a reservoir in a 2-phase 5-spot waterflood setting. Experiments were done comparing a strong constraint with weak constraints of different magnitude. The use of gradients of the strong and weak constraint problems in steepest descent and LBFGS minimization schemes was illustrated. An example was shown where the number of representer was reduced by a factor two, without degrading the quality of the final permeability estimate.

4.6.2 Conclusions

The RM that was used in this chapter was modified from the original RM on four accounts: it can interact with different gradient-based minimization algorithms, the number of representer may be (much) smaller than the number of measurements, the representer functions are defined differently so no correction terms need to be calculated, and the "measurement representer" is introduced.

The modified RM method does not solve the weak constraint minimization problem directly; it produces a regularized gradient that can be used by any gradient-based minimization algorithm (after minor modifications). Solving the minimization problem is then left to this algorithm.

In the classic RM, the number of representer coefficients is equal to the number of measurements. In applications where there are many measurements, computational feasibility needs to be created by selecting a subset of the data, or, as was done in this chapter, compressing the measurement into a smaller amount of pseudo measurements. This can for example be done by analyzing the data covariance matrix or the Jacobian of the measurement operator with a singular value decomposition. This choice can be made before starting the iterative process and it can be chosen differently for various iterations. Even the number of basis functions can be changed during minimization.

Both in previous non-linear versions of the RM [Valstar *et al.* (2004), Baird and Dawson (2007)] and in the RM proposed in this chapter, the state variables are essentially decomposed around the prior state variables. In the modified RM, the prior state variables are explicitly calculated by a forward simulation with the prior parameters, whereas the classic RM splits the prior state variables into the results of the last simulation and a correction term. The

criterion that is used to calculate the correction term, is that at every iteration the linearized equations are solved. By using this criterion, the state variables of the last simulation and the correction term do no longer add up to the prior state variables. One could argue that the modified RM converges slower than the classic RM, since, besides the presence of a penalty term in the objective function that keeps the parameters close to the prior parameters, an additional penalty is introduced that keeps the state variables close to the prior state variables [Przybysz (2007)]. It can also be argued that the modified RM tries to solve the same non-linear equations as the classic RM without requiring that the linearized equations need to be solved at every iteration. Without this limitation, the modified RM may also converge faster than the classic RM. Convergence of the RM has not been proven for non-linear problems, and the convergence behavior is also not yet well understood. Convergence behavior of the classic RM and the modified RM should be further examined for larger, field-scale problems.

The classic RM is derived without "measurement representers", probably because in the case of a linear measurement operator, the measurement representers are constructed by concatenating the state representers of different time steps as row blocks into one large matrix. In this chapter the measurement operator is more complex, and measurement representers were introduced to handle this.

The RM only needs one single iteration for a linear problem. In general the RM uses more than one iteration for a non-linear problem, but still converges considerably faster than steepest descent. It also needs a lot more computation time per iteration. (L)BFGS can speed up steepest descent at little extra computation time. A small effort is made to calculate some approximation of the second order derivatives of the objective function, the Hessian. RM costs about as much computation time as explicitly calculating the Hessian. BFGS can also be applied on top of RM, but this does not contribute to faster convergence.

It is recommended to use a good line search algorithm. Although BFGS provides a better search direction than SD, this does not help the convergence if a fixed step size is used. For the strong-constraint case with the adjoint method, the quadratic line search performed so well for SD, that BFGS could not outperform it. BFGS did perform better with an exponential line search. The quadratic line search does not work properly in combination with RM, because the slope of the objective along the search direction cannot be calculated accurately. As a result, the quadratic line search underestimates the step size and the algorithm stops before even finding a local minimum of the objective function.

The RM achieves most of the decrease of the objective function in the first iteration. How much work is left for successive iterations depends on the magnitude of the model errors. If the model errors are large, then the non-linear relation between model parameters and state variables is overshadowed and the minimization problem becomes more linear. Hence, convergence is faster.

In the example considered in this chapter, order reduction can be used to reduce computation time without loss of quality of the estimated parameters. When the order is reduced to the extreme, the minimization algorithm reaches the stopping criterion before performing any iterations. Applications may exist where this becomes a problem for smaller reductions. With or without reduction, no proof of convergence of the RM exists for non-linear problems.

The resolution of the model errors that were reconstructed by RM was never higher than the resolution of the measurements. Compared to the high resolution model errors that were used to create the synthetic measurements, the RM introduced artificial correlation between the model errors on time intervals in between separate measurements.

Chapter 5

Comparison of the Ensemble Kalman Filter and a modified Representer Method for sensitivity to prior data⁵

Abstract

Data assimilation algorithms or computer-assisted history-matching methods are meant to improve the predictive capability of reservoir simulation models. They rely on two sources of uncertain information: measured data, typically production data such as well bore pressures and phase rates, and prior information, for example a statistical description of the reservoir properties. In a synthetic numerical experiment the uncertainties in the data and the prior are known; in a field application they are not. One could say that there are not only uncertainties in the data and the reservoir properties, but there is also "uncertainty in the uncertainty".

In this chapter the robustness of the Ensemble (Square Root) Kalman Filter and a gradient-based algorithm using Representer Expansions are compared with respect to prior input data. Some algorithm-dependent settings are explored to try to make the filter reproduce the results of the Representer Method: the ensemble size, the initialization method and the Kalman update. The concept of assimilating data more than once with dampened weighting factors (added uncertainty) is introduced.

First the equations that underlie the Ensemble Kalman Filter and the Representer Method are given. Then numerical experiments are presented and two measures of quantifying the success of the methods are introduced. According to one such measure, the Representer Method performed better for all numerical examples considered. The parameters of the filter can be chosen such that the filter with the correct input data is just as successful as the Representer Method, using the second measure. When the methods are fed with the wrong prior input, the second measure also favours the Representer Method, so for the examples considered in this chapter the Representer Method is less sensitive to "wrong" prior data than the Kalman Filter.

5.1 Introduction

Section 5.1.1 introduces the concept of running a reservoir simulator in strong (deterministic) or weak (stochastic) mode. The relationship between the Ensemble Filter and the Represen-

⁵ This chapter is based on [Rommelse *et al.* (2008a)], which was published as TUD-DIAM report 08-16 and submitted to Computational Geosciences

ter Method is discussed in section 5.1.2 and the two methods are interpreted as different approximations of Bayes rule.

5.1.1 Strong and weak constraint reservoir simulation

Running a reservoir simulator with incorrect model parameters, like permeability or porosity, causes a discrepancy between observations and measurement forecasts. There is an additional phenomenon that causes the discrepancy; the model is an approximation, so even if the parameters were known, the model would still produce incorrect output. These model errors can be explicitly set to zero for simple numerical experiments. The reservoir simulator is then used deterministically. For more realistic experiments or field applications, the model errors can be modelled as additional parameters, which are sampled by a Monte Carlo method or estimated along with the other model parameters. In this case the reservoir simulator is used in stochastic mode. In variational or gradient-based data assimilation, the terms deterministic and stochastic mode correspond to using the reservoir simulator as a strong or as a weak constraint respectively.

Running the reservoir simulator and predicting measurements can be written in general state-space notation as

$$\mathbf{x}_i = M(\mathbf{x}_{i-1}) \quad , \quad \mathbf{y} = H(\mathbf{x}) \quad , \quad (5.1)$$

where $\mathbf{x}_i, i \in \{0, \dots, n\}$ are vectors of state variables (typically pressures and saturations) at time t_i , $\mathbf{x} = [\mathbf{x}_1^T \ \dots \ \mathbf{x}_n^T]^T$, \mathbf{y} is a vector of measurements (outputs), and M and H are model and measurement operators respectively. In this chapter running the reservoir simulator and predicting measurements is denoted by

$$\mathbf{g}(\mathbf{x}_i, \mathbf{x}_{i-1}, \boldsymbol{\theta}, \boldsymbol{\varepsilon}_i) = \mathbf{0} \quad , \quad \mathbf{y} = \mathbf{h}(\mathbf{x}) \quad , \quad (5.2)$$

where \mathbf{g} and \mathbf{h} are vector-valued functions, where the model parameters (for example permeabilities and porosities) are collected in the vector $\boldsymbol{\theta}$ and the model errors on interval $[t_{i-1}, t_i]$ are contained in the vector $\boldsymbol{\varepsilon}_i$. In fact, the mass balance and Darcy equations [Aziz and Settari (1979)]

$$\frac{d}{dt}(\mathbf{f}_1(\mathbf{x})) = \mathbf{f}_2(\mathbf{x}, \boldsymbol{\theta}) \quad , \quad (5.3)$$

are discretized in time using a the implicit Euler scheme, resulting in a deterministic reservoir simulator:

$$\mathbf{g}(\mathbf{x}_i, \mathbf{x}_{i-1}, \boldsymbol{\theta}, \mathbf{0}) = \mathbf{f}_1(\mathbf{x}_i) - (t_i - t_{i-1}) \mathbf{f}_2(\mathbf{x}_i, \boldsymbol{\theta}) - \mathbf{f}_1(\mathbf{x}_{i-1}) = \mathbf{0} \quad . \quad (5.4)$$

The presence of water and oil mass in the grid blocks is described by \mathbf{f}_1 , while \mathbf{f}_2 models the flow through the grid block interfaces and the injection/production of fluids in the wells. The model errors are introduced as additional sources/sinks in all grid blocks. In other words, after \mathbf{x}_i has been solved from Eq. (5.4), the water and oil masses in the grid blocks have not correctly been predicted and must still be modified. The prediction gets worse as the time step $(t_i - t_{i-1})$ grows larger. Therefore the correction is modelled proportional to $(t_i - t_{i-1})$. If

the additional sources become too strong, unrealistically high pressures will be observed. If the additional sinks become too strong, saturations outside $[0, 1]$ will occur. In this chapter, the additional sinks are non-linearly constrained by \mathbf{f}_1 . The stochastic reservoir simulator has the form

$$\begin{aligned} & \mathbf{g}(\mathbf{x}_i, \mathbf{x}_{i-1}, \boldsymbol{\theta}, \boldsymbol{\varepsilon}_i) \\ = & \mathbf{f}_1(\mathbf{x}_i) - (t_i - t_{i-1}) \mathbf{f}_2(\mathbf{x}_i, \boldsymbol{\theta}) - \mathbf{f}_1(\mathbf{x}_{i-1}) + \min\{\mathbf{f}_1(\mathbf{x}_i), (t_i - t_{i-1}) \boldsymbol{\varepsilon}_i\} = \mathbf{0}. \end{aligned} \quad (5.5)$$

5.1.2 Bayesian data-assimilation

Reservoir simulation can be embedded in a stochastic or probabilistic framework; see e.g. [Gavalas *et al.* (1976)] for an early reference, [Zhang *et al.* (2005)] for applications and [Oliver *et al.* (2008)] which also contains further petroleum-related references. In this case the reservoir state variables (pressures and saturations in all grid blocks) do not have deterministic values, but are described by a multivariate probability distribution (or density) function (PDF). The stochastic nature of the state variables is caused by the uncertainty in the initial states, the uncertainty in the model parameters (permeability, porosity, etc.) and the fact that the reservoir simulator is imperfect (e.g. gravity or capillary effects were not modelled, or three components were modelled where five would have been more appropriate). The uncertainty in the measurements is caused by two effects; 1) sensors monitor a stochastic quantity and 2) they are subject to influences that might damage them or otherwise corrupt the data.

Bayes theorem for continuous probability distributions [Bayes (1763)]

$$f(\boldsymbol{\theta}|\mathbf{y}) = \frac{f(\mathbf{y}|\boldsymbol{\theta}) f(\boldsymbol{\theta})}{f(\mathbf{y})}, \quad f(\boldsymbol{\theta}|\mathbf{y}) \propto f(\mathbf{y}|\boldsymbol{\theta}) f(\boldsymbol{\theta}) = f(\boldsymbol{\theta}, \mathbf{y}), \quad (5.6)$$

states that the *posterior* density $f(\boldsymbol{\theta}|\mathbf{y})$ (the probability of the model parameters given the data) is proportional to the *prior* density $f(\boldsymbol{\theta})$ (the probability of the model parameters) multiplied by the *likelihood* $f(\mathbf{y}|\boldsymbol{\theta})$ of the data given the model parameters. This basically means that the reservoir simulator and a measurement model must be used to construct the joint density of the parameters and the measurement forecasts $f(\boldsymbol{\theta}, \mathbf{y})$. The domain is then restricted by substituting the observations, which causes the integral over the new function to no longer be equal to 1. Renormalization then gives the posterior density. The hidden dependence of the measurement forecasts on the reservoir states \mathbf{x} can also be explicitly written in Bayes theorem

$$f(\boldsymbol{\theta}, \mathbf{x}|\mathbf{y}) = \frac{f(\mathbf{x}, \mathbf{y}|\boldsymbol{\theta}) f(\boldsymbol{\theta})}{f(\mathbf{y})}, \quad f(\boldsymbol{\theta}, \mathbf{x}|\mathbf{y}) \propto f(\mathbf{x}, \mathbf{y}|\boldsymbol{\theta}) f(\boldsymbol{\theta}) = f(\boldsymbol{\theta}, \mathbf{x}, \mathbf{y}). \quad (5.7)$$

For practical applications, it is hardly ever possible to compute Eq. (5.6) analytically, so approximations have to be made. For the special case where the prior $f(\boldsymbol{\theta})$ is Gaussian and the measurement forecasts linearly depend on the parameters (which requires that both the

reservoir simulator and the sensor model are linear), the posterior $f(\boldsymbol{\theta}|\mathbf{y})$ is also Gaussian [Tarantola (2005)], which means that it suffices to only calculate the mean and covariance.

The Kalman Filter, which will be described in section 5.2.1, exploits this property and sequentially updates the two statistical moments by performing time updates and measurement updates. In the non-linear case, the Ensemble Kalman Filter [Evensen (2003), Evensen (2007)] samples the prior and updates the samples until a sampled representation of the posterior is found. However, at every measurement update the density is again assumed to be Gaussian.

When the posterior is Gaussian, the logarithm is proportional to a weighted average of the model errors, the measurement errors and the parameter errors [Tarantola (2005)], (see also section 5.3.1). This can be formulated as an objective that has to be minimized in order to find the mode of the posterior, which is equal to the mean for a Gaussian. This is done in variational methods. In the non-linear case, a mode of a probability distribution is found, but it is usually not clear how this distribution is related to the posterior. Only one mode is found if the posterior is multi-modal.

5.2 Ensemble Filter

5.2.1 EnKF

Due to increasing computer capacity, the Kalman filter [Kalman (1960), Gelb (1974)] is growing more popular for computer-assisted history-matching. Implementing an Ensemble Kalman Filter (EnKF) [Evensen (2003), Heemink *et al.* (2001)] is easy and does not require modifications to existing reservoir simulators; see. e.g. [Naevdal *et al.* (2005)] for an early application, and [Evensen (2007)] for further references.

In the Ensemble Kalman Filter, the prior is represented by an ensemble of samples $\{\boldsymbol{\theta}^{(1)}, \dots, \boldsymbol{\theta}^{(n)}\}$. This sampling can be done randomly or deterministically. In the absence of measurements, the samples are updated in the time update by the non-linear reservoir simulator

$$\mathbf{g}\left(\mathbf{x}_i^{(j)}, \mathbf{x}_{i-1}^{(j)}, \boldsymbol{\theta}^{(j)}, \boldsymbol{\varepsilon}_i^{(j)}\right) = \mathbf{0}. \quad (5.8)$$

Whenever measurements are available, the measurement update is performed. Augmented ensemble members are created by concatenating the reservoir states, the measurement forecasts and (in case of parameter estimation) the parameters

$$\mathbf{I}_f^{(j)} = \begin{bmatrix} \mathbf{x}_i^{(j)} \\ \mathbf{y}_f^{(j)} \\ \boldsymbol{\theta}^{(j)} \end{bmatrix}, \quad (5.9)$$

where the measurement forecasts ensemble is obtained by running an ensemble of (non-linear) measurement models

$$\mathbf{y}_f^{(j)} = \mathbf{h} \left(\mathbf{x}_i^{(j)} \right). \quad (5.10)$$

The forecasted ensemble members are then split into a mean and a set of deviations

$$\mathbf{l}_f^{(j)} = \bar{\mathbf{l}}_f + \tilde{\mathbf{l}}_f^{(j)}, \quad (5.11)$$

which are updated by

$$\mathbf{L}_f = \begin{bmatrix} \tilde{\mathbf{l}}_f^{(1)} & \dots & \tilde{\mathbf{l}}_f^{(n)} \end{bmatrix}, \quad \tilde{\mathbf{L}} = [\mathbf{0} \quad \mathbf{I} \quad \mathbf{0}] \mathbf{L}_f, \quad (5.12)$$

$$\mathbf{K} = \mathbf{L}_f \tilde{\mathbf{L}}^T \left(\tilde{\mathbf{L}} \tilde{\mathbf{L}}^T + (n-1) \mathbf{P}_y \right)^{-1}, \quad (5.13)$$

$$\bar{\mathbf{l}}_a = \bar{\mathbf{l}}_f + \mathbf{K} (\mathbf{m} - \bar{\mathbf{y}}_f), \quad \mathbf{L}_a = \mathbf{L}_f + \mathbf{K} \left([\tilde{\mathbf{m}}^{(1)} \quad \dots \quad \tilde{\mathbf{m}}^{(n)}] - \tilde{\mathbf{L}} \right), \quad (5.14)$$

to form the analyzed mean and deviations that can be used to form the analyzed ensemble members. Here \mathbf{m} is the vector of observed measurements. The Kalman gain matrix, Eq. (5.13), is built from the measurement forecast's autocovariance $\frac{\tilde{\mathbf{L}} \tilde{\mathbf{L}}^T}{n-1}$, the observation's autocovariance \mathbf{P}_y (which must be specified by the user) and the cross covariance of the measurement forecasts and the augmented forecasted ensemble members. The measurement errors $\tilde{\mathbf{m}}^{(j)}$ are sampled using \mathbf{P}_y . The analyzed measurements are a weighted average of the measurement forecasts and the observations. The jump from the forecasted values to the analyzed values is linearly extrapolated to the reservoir states and the parameters using the cross covariances between them.

5.2.2 ESRKF

In the EnKF, the errors in the observations $\tilde{\mathbf{y}}_o^{(j)}$ are sampled using \mathbf{P}_y . Without these errors, the uncertainty in the analyzed ensemble would be underestimated by

$$\mathbf{L}_a = (\mathbf{I} - \mathbf{K} [\mathbf{0} \quad \mathbf{I} \quad \mathbf{0}]) \mathbf{L}_f = \mathbf{L}_f - \mathbf{K} \tilde{\mathbf{L}}. \quad (5.15)$$

However, also deterministic solutions to this problem are available. In [Verlaan and Heemink (1997)], extra columns are added to $\tilde{\mathbf{L}}$ in Eq. (5.15). These columns are obtained as a square root \mathbf{L}_y of $\mathbf{P}_y = \mathbf{L}_y \mathbf{L}_y^T$. After the augmentation, the number of columns is reduced to the original number by selecting the leading left-singular vectors. In the Ensemble Square Root Kalman Filter (ESRKF), Eq. (5.15) is used, but with a modified Kalman gain matrix which, in the one-dimensional case, can be written as

$$\tilde{\mathbf{K}} = \frac{\mathbf{K}}{1 + \sqrt{\frac{(n-1)\mathbf{P}_y}{\tilde{\mathbf{L}} \tilde{\mathbf{L}}^T + (n-1)\mathbf{P}_y}}}. \quad (5.16)$$

For more dimensions, the modified Kalman gain matrix involves calculating the non-unique square roots of two matrices. Details can be found in [Whitaker and Hamill (2002)].

5.2.3 Duplicated measurements

As mentioned in section 5.2.1, the analyzed measurements are a weighted average of the measurement forecasts and the observations. The difference between the forecasted values and the analyzed values is linearly extrapolated to the reservoir states and the model parameters using the cross covariances between them. These are both linear operations which can cause problems when the measurement operator is non-linear. Furthermore, the weighting factors that are used for the averaging are the user-defined uncertainty in the observations and the covariance of the measurement forecasts. This second order statistical moment is not appropriate when the state variables are predicted by a non-linear simulator, even when it is obtained by a EnKF rather than an Extended Kalman Filter (EKF) [Evensen (2003)]. In fact, using a EnKF without extra modifications often causes saturation values outside $[0, 1]$ and unrealistic pressure and permeability estimates. A quick engineering trick is to apply a dampening factor to the jump from the forecasted values to the analyzed values, which is equivalent to adding extra uncertainty to the observations.

Since in practice it costs money to obtain data, it is a waste to discard data or to add extra uncertainty to them. Results in this chapter are obtained by duplicating measurements to assimilate them more than once, but with extra uncertainty. The extra uncertainty ensures that the jump from the forecasted values to the analyzed values, which is overestimated by the linear measurement update, is dampened. The duplication of measurements ensures that the jump is not dampened too much, so no valuable data are discarded. When a measurement is predicted with variance σ_f^2 and the uncertainty of the sensor is σ_o^2 , then the measurement is assimilated n times with artificial uncertainty $\tilde{\sigma}_o^2$ using two criteria:

1. The artificial sensor uncertainty $\tilde{\sigma}_o^2$ is of the same magnitude as the measurement forecasts variance σ_f^2 , so $\tilde{\sigma}_o^2 \approx \sigma_f^2$.
2. The variance of the analyzed measurements is invariant under this method, so

$$\sigma_a^2 = \frac{\sigma_f^2 \sigma_o^2}{\sigma_f^2 + \sigma_o^2} = \frac{\sigma_f^2 \tilde{\sigma}_o^2}{2^{n-1} \sigma_f^2 + \tilde{\sigma}_o^2}. \quad (5.17)$$

Therefore, every measurement is used

$$n = 1 + \left\lceil \log_2 \left(\frac{\sigma_f^2 + \sigma_o^2}{2\sigma_o^2} \right) \right\rceil \quad (5.18)$$

times, with variance $\tilde{\sigma}_o^2 = 2^{n-1} \sigma_o^2$. By construction, the variance of the analyzed measurements is identical to what it would be without duplicated measurements. However, the jump from the forecasted values to the analyzed values is not; the sum of a set of small jumps is less than one big jump.

5.3 Representer Method

The discrepancy between observed measurements and their model predicted antitheses can be formulated as an objective function that has to be minimized. This minimization process is regularized by penalizing the deviation of the estimated parameters from the prior parameters, while the model errors may also be added to the objective function, as will be discussed in Section 5.3.1. If the posterior distribution is Gaussian, the mode (or mean) of the posterior minimizes the objective. If not, the objective has a different meaning and the weighting factors might have to be changed accordingly; covariance matrices that are sufficient to represent Gaussians are no longer appropriate and the user has to supply an alternative. When the posterior is multi-modal, the objective function has multiple local minima and additional regularization is required. Calculating a gradient of the objective function with respect to the model parameters can be done efficiently using an adjoint reservoir simulator in case the model errors are explicitly set to zero; see e.g. [Chavent *et al.* (1975)] for an early application and [Oliver *et al.* (2008)] for further references. Regularization is usually done by selecting a limited set of basis functions that are chosen as singular vectors of a root of the covariance matrix of the parameters. When model errors are taken into account, the reservoir simulator and its adjoint can no longer be used sequentially to produce a gradient. The Representer Method (RM) [Bennett and McIntosh (1982), Eknes and Evensen (1997), Bennett (2002), Valstar *et al.* (2004), Baird and Dawson (2005), Janssen *et al.* (2006), Przybysz *et al.* (2007), Baird and Dawson (2007), Rommelse *et al.* (2007)] can then be used to solve the weak constraint minimization problem. Regularization is also included in the method and the basis functions are data-driven rather than specified by the user. Unfortunately the computational cost is proportional to the number of measurements times the computational cost of solving the strong constraint problem using the adjoint system.

The RM used in this chapter was introduced in [Rommelse *et al.* (2007)] and differs from previous implementations. It produces a gradient that can be used in any minimization algorithm (with minor modifications); here the Limited-memory Broyden-Fletcher-Goldfarb-Shanno (LBFSGS) method [Gao and Reynolds (2006), Ulbrich (2002)] is used. Another feature of the method introduced in [Rommelse *et al.* (2007)] is the abandoning of the one-to-one relationship between the representer functions and isolated measurements in time and space, which forms an essential element of the classic RM, making the method computationally more attractive. Moreover, the concept of measurement representers is introduced to deal with non-linear measurement models and a new linearization is used that does no longer require the calculation of special correction terms.

5.3.1 Objective function and derivatives

The objective that has to be minimized is formulated as

$$\begin{aligned}
J &= \frac{1}{2} (\mathbf{h}(\mathbf{x}_{\{0, \dots, n\}}) - \mathbf{m})^T \mathbf{P}_y^{-1} (\mathbf{h}(\mathbf{x}_{\{0, \dots, n\}}) - \mathbf{m}) + & (5.19) \\
&+ \frac{1}{2} (\boldsymbol{\theta} - \boldsymbol{\theta}^{prior})^T \mathbf{P}_\theta^{-1} (\boldsymbol{\theta} - \boldsymbol{\theta}^{prior}) + \frac{1}{2} \sum_{i=1}^n \boldsymbol{\varepsilon}_i^T \mathbf{P}_{\varepsilon_i}^{-1} \boldsymbol{\varepsilon}_i + \\
&+ \sum_{i=1}^n \boldsymbol{\lambda}_i^T \mathbf{g}(\mathbf{x}_i, \mathbf{x}_{i-1}, \boldsymbol{\theta}, \boldsymbol{\varepsilon}_i).
\end{aligned}$$

If the reservoir simulator \mathbf{g} and the sensor model \mathbf{h} were linear and the prior were Gaussian, the weighting factors \mathbf{P}_θ , $\mathbf{P}_{\varepsilon_i}$ and \mathbf{P}_y could be chosen equal to the model parameter uncertainty, the model error uncertainty and the sensor uncertainty respectively, in order to find the same solution that an EnKF would find. Otherwise different factors may have to be chosen. In general, the initial reservoir states need to be estimated along with the permeability, porosity and other parameters, $\mathbf{x}_0 = \mathbf{f}(\boldsymbol{\theta})$, but in this chapter they are assumed known.

The derivatives of Eq. (5.19) with respect to $\boldsymbol{\lambda}_i$, $\boldsymbol{\varepsilon}_i$, \mathbf{x}_i and $\boldsymbol{\theta}$ are

$$\left(\frac{\partial J}{\partial \boldsymbol{\lambda}_i} \right)^T = \mathbf{g}(\mathbf{x}_i, \mathbf{x}_{i-1}, \boldsymbol{\theta}, \boldsymbol{\varepsilon}_i), \quad (5.20)$$

$$\left(\frac{\partial J}{\partial \boldsymbol{\varepsilon}_i} \right)^T = \mathbf{P}_{\varepsilon_i}^{-1} \boldsymbol{\varepsilon}_i + \left(\frac{\partial \mathbf{g}(\mathbf{x}_i, \mathbf{x}_{i-1}, \boldsymbol{\theta}, \boldsymbol{\varepsilon}_i)}{\partial \boldsymbol{\varepsilon}_i} \right)^T \boldsymbol{\lambda}_i, \quad (5.21)$$

$$\begin{aligned}
\left(\frac{\partial J}{\partial \mathbf{x}_i} \right)^T &= \left(\frac{\partial \mathbf{h}(\mathbf{x}_{\{0, \dots, n\}})}{\partial \mathbf{x}_i} \right)^T \mathbf{P}_y^{-1} (\mathbf{h}(\mathbf{x}_{\{0, \dots, n\}}) - \mathbf{m}) + & (5.22) \\
&+ \left(\frac{\partial \mathbf{g}(\mathbf{x}_i, \mathbf{x}_{i-1}, \boldsymbol{\theta}, \boldsymbol{\varepsilon}_i)}{\partial \mathbf{x}_i} \right)^T \boldsymbol{\lambda}_i + \left(\frac{\partial \mathbf{g}(\mathbf{x}_{i+1}, \mathbf{x}_i, \boldsymbol{\theta}, \boldsymbol{\varepsilon}_{i+1})}{\partial \mathbf{x}_i} \right)^T \boldsymbol{\lambda}_{i+1},
\end{aligned}$$

$$\left(\frac{\partial J}{\partial \boldsymbol{\theta}} \right)^T = \mathbf{P}_\theta^{-1} (\boldsymbol{\theta} - \boldsymbol{\theta}^{prior}) + \sum_{i=1}^n \left(\frac{\partial \mathbf{g}(\mathbf{x}_i, \mathbf{x}_{i-1}, \boldsymbol{\theta}, \boldsymbol{\varepsilon}_i)}{\partial \boldsymbol{\theta}} \right)^T \boldsymbol{\lambda}_i. \quad (5.23)$$

For the strong constraint case, where $\boldsymbol{\varepsilon}_i$ is explicitly set to $\mathbf{0}$, the gradient of the objective function with respect to the model parameters, $\left(\frac{\partial J}{\partial \boldsymbol{\theta}} \right)^T$, can be calculated using Eq. (5.23), where the model states \mathbf{x}_i and adjoint states $\boldsymbol{\lambda}_i$ follow from sequentially solving Eq. (5.20) and Eq. (5.22) with the left-hand sides, $\left(\frac{\partial J}{\partial \boldsymbol{\lambda}_i} \right)^T$ and $\left(\frac{\partial J}{\partial \mathbf{x}_i} \right)^T$, set to zero. Eq. (5.21) does not need to be used; instead $\boldsymbol{\varepsilon}_i = \mathbf{0}$ is used.

To interface with gradient-based minimization algorithms, a numerical routine must be implemented that evaluates J , Eq. (5.19), and $\left(\frac{\partial J}{\partial \boldsymbol{\theta}} \right)^T$, Eq. (5.20), Eq. (5.22) and Eq. (5.23). $\mathbf{Q}\mathbf{Q}^T \left(\frac{\partial J}{\partial \boldsymbol{\theta}} \right)^T$ is the regularized gradient if orthogonal basis functions are chosen for the regularization and placed as columns in the matrix \mathbf{Q} .

5.3.2 Local minimizer

In a stationary point of Eq. (5.19), all gradients are equal to zero, so

$$\mathbf{g}(\mathbf{x}_i, \mathbf{x}_{i-1}, \boldsymbol{\theta}, \boldsymbol{\varepsilon}_i) = \mathbf{0}, \quad (5.24)$$

$$\boldsymbol{\varepsilon}_i = -\mathbf{P}_{\boldsymbol{\varepsilon}_i} \left(\frac{\partial \mathbf{g}(\mathbf{x}_i, \mathbf{x}_{i-1}, \boldsymbol{\theta}, \boldsymbol{\varepsilon}_i)}{\partial \boldsymbol{\varepsilon}_i} \right)^T \boldsymbol{\lambda}_i, \quad (5.25)$$

$$\begin{aligned} & \left(\frac{\partial \mathbf{h}(\mathbf{x}_{\{0, \dots, n\}})}{\partial \mathbf{x}_i} \right)^T \mathbf{P}_y^{-1} (\mathbf{m} - \mathbf{h}(\mathbf{x}_{\{0, \dots, n\}})) \\ &= \left(\frac{\partial \mathbf{g}(\mathbf{x}_i, \mathbf{x}_{i-1}, \boldsymbol{\theta}, \boldsymbol{\varepsilon}_i)}{\partial \mathbf{x}_i} \right)^T \boldsymbol{\lambda}_i + \left(\frac{\partial \mathbf{g}(\mathbf{x}_{i+1}, \mathbf{x}_i, \boldsymbol{\theta}, \boldsymbol{\varepsilon}_{i+1})}{\partial \mathbf{x}_i} \right)^T \boldsymbol{\lambda}_{i+1}, \end{aligned} \quad (5.26)$$

$$\boldsymbol{\theta} = \boldsymbol{\theta}^{prior} - \mathbf{P}_{\boldsymbol{\theta}} \sum_{i=1}^n \left(\frac{\partial \mathbf{g}(\mathbf{x}_i, \mathbf{x}_{i-1}, \boldsymbol{\theta}, \boldsymbol{\varepsilon}_i)}{\partial \boldsymbol{\theta}} \right)^T \boldsymbol{\lambda}_i. \quad (5.27)$$

Calculating and regularizing a gradient as was done in section 5.3.1, is no longer possible since the forward model Eq. (5.24) and the adjoint model Eq. (5.26) are now coupled because the model errors $\boldsymbol{\varepsilon}_i$ are no longer equal to zero; they are related to the adjoint states $\boldsymbol{\lambda}_i$ by Eq. (5.25).

5.3.3 Representer expansions

Since it is not possible to obtain a gradient of the weak constraint minimization problem directly and regularize it, the Representer Method was introduced to regularize Eq. (5.24) through Eq. (5.27) and then find a minimization scheme. Unlike in section 5.3.1, where the basis functions are chosen by the user, in the RM the regularization is done symbolically and then substituted into Eq. (5.24) through Eq. (5.27) to obtain the basis functions. In [Valstar *et al.* (2004), Przybysz *et al.* (2007)] the number of basis functions is chosen equal to the number of measurements, such that

$$\mathbf{x}_i = \mathbf{x}_i^{forecast} + \mathbf{x}_i^{correction} + \mathbf{R}_{\mathbf{x}_i} \mathbf{b}, \quad (5.28)$$

$$\boldsymbol{\lambda}_i = \mathbf{R}_{\boldsymbol{\lambda}_i} \mathbf{b}, \quad (5.29)$$

$$\boldsymbol{\theta} = \boldsymbol{\theta}^{prior} + \mathbf{R}_{\boldsymbol{\theta}} \mathbf{b}, \quad (5.30)$$

where the columns of $\mathbf{R}_{\mathbf{x}_i}$, $\mathbf{R}_{\boldsymbol{\lambda}_i}$ and $\mathbf{R}_{\boldsymbol{\theta}}$ are the basis functions, called the state representers, adjoint representers and parameter representers. The representer coefficients are contained in the vector \mathbf{b} .

In [Rommelse *et al.* (2007)] a different regularization is introduced. The minimization algorithm is started with $\boldsymbol{\theta}^{init} = \boldsymbol{\theta}^{prior}$ and $\boldsymbol{\varepsilon}_i^{init} = \boldsymbol{\lambda}_i^{init} = \mathbf{0}$. Applying these initial

conditions to Eq. (5.24) gives the initial system states \mathbf{x}_i^{init} . These are not initial in the sense that they are defined at t_0 , but they are initial in the minimization algorithm. Given the initial states, also initial measurements can be predicted, $\mathbf{h}(\mathbf{x}_{\{0,\dots,n\}}^{init})$. The causes for the variables to move away from their prior are parameterized by \mathbf{b} . The 1-1-relationship between one such cause and an isolated measurement in space and time is abandoned. For computational purposes, it is interesting to assume that the number of parameters in the vector \mathbf{b} is (much) smaller than the number of measurements. The deviations from the priors are decomposed as

$$\mathbf{x}_i - \mathbf{x}_i^{prior} = \mathbf{R}_{\mathbf{x}_i} \mathbf{b}, \quad (5.31)$$

$$\lambda_i = \mathbf{R}_{\lambda_i} \mathbf{b}, \quad (5.32)$$

$$\theta - \theta^{prior} = \mathbf{R}_\theta \mathbf{b}, \quad (5.33)$$

$$\mathbf{h}(\mathbf{x}_{\{0,\dots,n\}}) - \mathbf{h}(\mathbf{x}_{\{0,\dots,n\}}^{prior}) = \mathbf{R}_y \mathbf{b}. \quad (5.34)$$

The columns of \mathbf{R}_y contain the measurement representers.

5.3.4 Representer equations

After substitution of Eq. (5.31), Eq. (5.32) and Eq. (5.33) into Eq. (5.24), Eq. (5.26) and Eq. (5.27) and some manipulation [Rommelse *et al.* (2007)], the representer equations are obtained. These can be used to obtain a regularized gradient for the weak constraint minimization problem as follows:

1. If the model errors ε_i have not yet been approximated by a previous iteration of the minimization algorithm, initialize them to zero. Given the model parameters θ and the model errors ε_i , run the non-linear model Eq. (5.24). After the model has been run for the first time, save the prior states \mathbf{x}_i^{prior} and forecast the prior measurements $\mathbf{h}(\mathbf{x}_{\{0,\dots,n\}}^{prior})$.
2. Choose the selection matrix \mathbf{Q} , as discussed in detail in [Rommelse *et al.* (2007)]. A convenient choice is $\mathbf{Q} = \mathbf{U}_{[:,1:k]}$, where \mathbf{U} is a matrix of left singular vectors of the measurement sensitivity matrix, and where k is typically much smaller than the number of measurements.
3. Calculate the adjoint representers \mathbf{R}_{λ_i} from

$$\begin{aligned} & \left(\frac{\partial \mathbf{g}(\mathbf{x}_i, \mathbf{x}_{i-1}, \theta, \varepsilon_i)}{\partial \mathbf{x}_i} \right)^T \mathbf{R}_{\lambda_i} \\ &= \left(\frac{\partial \mathbf{h}(\mathbf{x}_{\{0,\dots,n\}})}{\partial \mathbf{x}_i} \right)^T \mathbf{Q} - \left(\frac{\partial \mathbf{g}(\mathbf{x}_{i+1}, \mathbf{x}_i, \theta, \varepsilon_{i+1})}{\partial \mathbf{x}_i} \right)^T \mathbf{R}_{\lambda_{i+1}}. \end{aligned} \quad (5.35)$$

4. Calculate the parameter representers \mathbf{R}_θ from

$$\mathbf{R}_\theta = -\mathbf{P}_\theta \sum_{i=1}^n \left(\frac{\partial \mathbf{g}(\mathbf{x}_i, \mathbf{x}_{i-1}, \boldsymbol{\theta}, \varepsilon_i)}{\partial \boldsymbol{\theta}} \right)^T \mathbf{R}_{\lambda_i}. \quad (5.36)$$

5. Calculate the state representers $\mathbf{R}_{\mathbf{x}_i}$ from

$$\begin{aligned} & \frac{\partial \mathbf{g}(\mathbf{x}_i, \mathbf{x}_{i-1}, \boldsymbol{\theta}, \varepsilon_i)}{\partial \mathbf{x}_i} \mathbf{R}_{\mathbf{x}_i} + \frac{\partial \mathbf{g}(\mathbf{x}_i, \mathbf{x}_{i-1}, \boldsymbol{\theta}, \varepsilon_i)}{\partial \mathbf{x}_{i-1}} \mathbf{R}_{\mathbf{x}_{i-1}} + \\ & + \frac{\partial \mathbf{g}(\mathbf{x}_i, \mathbf{x}_{i-1}, \boldsymbol{\theta}, \varepsilon_i)}{\partial \boldsymbol{\theta}} \mathbf{R}_\theta \\ = & \frac{\partial \mathbf{g}(\mathbf{x}_i, \mathbf{x}_{i-1}, \boldsymbol{\theta}, \varepsilon_i)}{\partial \varepsilon_i} \mathbf{P}_{\varepsilon_i} \left(\frac{\partial \mathbf{g}(\mathbf{x}_i, \mathbf{x}_{i-1}, \boldsymbol{\theta}, \varepsilon_i)}{\partial \varepsilon_i} \right)^T \mathbf{R}_{\lambda_i}. \end{aligned} \quad (5.37)$$

6. Calculate the measurement representers \mathbf{R}_y from

$$\mathbf{R}_y = \sum_{i=0}^n \frac{\partial \mathbf{h}(\mathbf{x}_{\{0, \dots, n\}})}{\partial \mathbf{x}_i} \mathbf{R}_{\mathbf{x}_i}. \quad (5.38)$$

7. Calculate new representer coefficients \mathbf{b} as the least-squares solution given by

$$(\mathbf{R}_y + \mathbf{P}_y \mathbf{Q}) \mathbf{b} = \mathbf{m} - \mathbf{h}(\mathbf{x}_{\{0, \dots, n\}}^{prior}). \quad (5.39)$$

8. Calculate new adjoint states Eq. (5.32).

9. Calculate the model errors Eq. (5.25).

10. Calculate the regularized gradient of the weak constraint minimization problem

$$\left(\frac{\partial J}{\partial \boldsymbol{\theta}} \right)^T = \boldsymbol{\theta} - \boldsymbol{\theta}^{prior} - \mathbf{R}_\theta \mathbf{b}. \quad (5.40)$$

Comments:

- In the routine that calculates the gradient, also the model errors are updated. For minimization algorithms that calculate more than one gradient per iteration, there must be a mechanism in the gradient routine that can store the model errors as well as the temporary changes. The changes must be made definite whenever that is signaled by the minimization algorithm, for example after a successful line search. This requires a minor update to the minimization routine, meaning that third-party software libraries can only be used if the source code is also available.
- By setting the model errors equal to zero and ignoring steps 8 and 9, the RM can also be used to obtain a regularized gradient for the strong constraint minimization problem.

Symbol	Variable	Value	SI units	Value	Field units
h	Gridblock height	20	m	65.62	ft
$\Delta x, \Delta y$	Gridblock length/width	10	m	32.81	ft
μ_o	Oil dynamic viscosity	1.0×10^{-3}	$Pa s$	1.0	cP
μ_w	Water dynamic viscosity	1.0×10^{-3}	$Pa s$	1.0	cP
c_t	Total compressibility	1.0×10^{-8}	Pa^{-1}	7.0×10^{-5}	psi^{-1}
p_R	Initial reservoir pressure	10×10^6	Pa	1450.4	psi
k_{ro}^0	Endpoint relative permeability, oil			1.0	—
k_{rw}^0	Endpoint relative permeability, water			0.5	—
n_o	Corey exponent, oil			2.0	—
n_w	Corey exponent, water			2.0	—
S_{or}	Residual oil saturation			0.2	—
S_{wc}	Connate water saturation			0.2	—
ϕ	porosity			0.3	—

Table 5.1: Reservoir and fluid properties for the example

5.4 Numerical experiments

Experiments were done on a 2D 2-phase waterflooding application with wells in an inverted 5-spot configuration that was also used in [Rommelse *et al.* (2007)]. Water is injected at a rate of one pore volume per year and the production wells are constrained to flow at 0.25 pore volumes per year. The state of the reservoir is described by pressure and water saturation in all $21 \times 21 \times 1$ grid blocks of $10 \times 10 \times 20$ m . Capillary pressure and gravity effects are ignored. All other fluid and reservoir parameters have been listed in Tbl. (5.1). Synthetic data are generated by picking one realization out of a database of 1000 realizations as the "true" permeability, Fig. (5.1), and running a reservoir simulation with model errors that are sampled as white noise, Fig. (5.2). Note: although in reality production data consist of phase rates, bottom hole pressures or tubing head pressures, we will use the gridblock pressures as "measurements" to assess the performance of the algorithms. The pressure and saturation responses in the well gridblocks after simulating with these model errors are shown in Fig. (5.3). This figure also shows 10 synthetic pressure "measurements" in all well gridblocks at 100 and 200 days of simulation. The other realizations from the database are used to construct a covariance matrix that is used in the objective function that has to be minimized.

5.4.1 Correct prior and prior with exponentially decreasing correlation length

The filter is initialized by choosing an ensemble of permeability fields. For a variational method, like the RM, an initial permeability estimate must be specified as well as the uncertainty. These are extracted from the database as the mean and the sample covariance matrix. For an honest comparison of the methods, the same mean and covariance matrix should also be used to initialize the filter. In order to do so, samples must be drawn from the covariance matrix and added to the mean to obtain the ensemble members. This sampling can be done randomly, which is equivalent to picking ensemble members from the database, or determin-

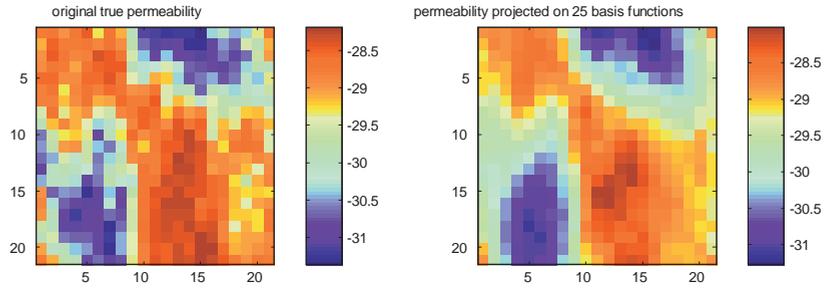


Figure 5.1: True permeability [$\ln(m^2)$]

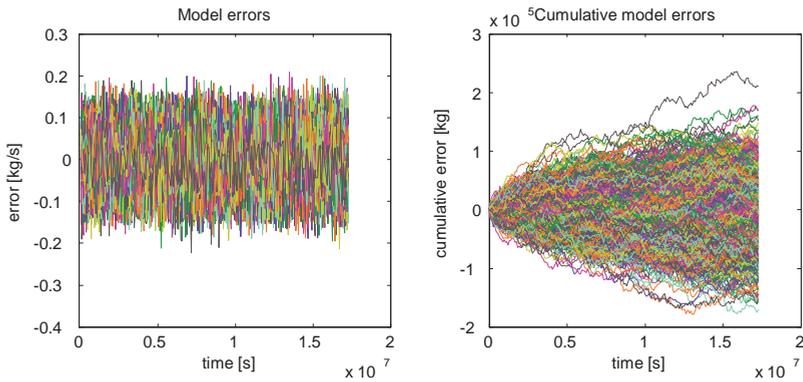


Figure 5.2: Synthetic true model errors

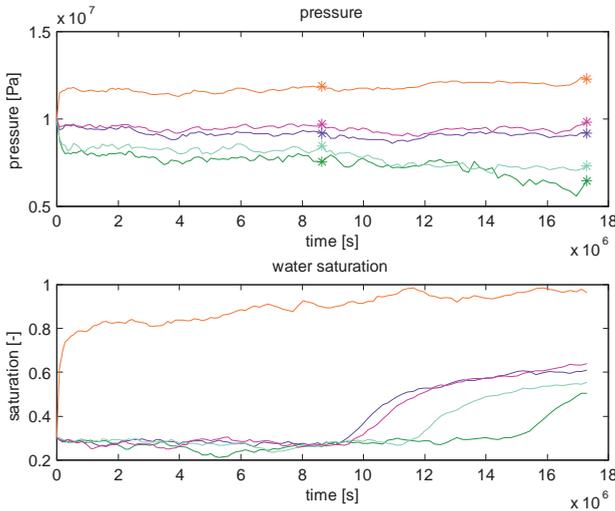


Figure 5.3: True pressure and saturation in the well gridblocks as functions of time. The top plot contains the 10 synthetic measurements at 100 and 200 days.

istically, for example by selecting the principal components of a square root of the covariance matrix.

In field applications, specifying such a covariance matrix for the permeability is a rather arbitrary process. It could be said that there is "uncertainty in the uncertainty", meaning that it is not clear how the covariance matrix should be chosen. In this chapter, the effect of specifying the wrong covariance matrix is investigated. The "wrong" covariance matrix is obtained from the "correct" covariance matrix by preserving the magnitude of the variance, but imposing a different spatial correlation pattern. The covariance between the permeability values in two grid blocks i and j decreases exponentially as function of the distance between grid blocks i and j ; the covariance is specified as

$$\tilde{P}_{ij} = \sqrt{P_{ii}P_{jj}}e^{-\frac{dist(i,j)}{l}}, \quad (5.41)$$

where l is the correlation length; it is obtained by minimizing the square-sum-difference of \tilde{P}_{ij} and P_{ij} , resulting in a value of 72.43 [m], Fig. (5.4)

5.4.2 Filter results

Some initial ensemble members are shown in Fig. (5.5). Fig. (5.6) presents the initial mean and covariance. The final estimates obtained with the correct prior covariance and 50

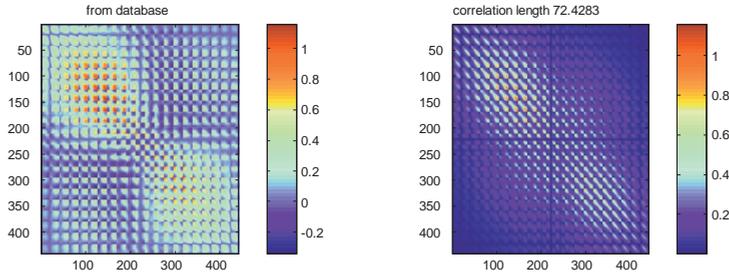


Figure 5.4: Correct prior covariance matrix and prior with exponentially decreasing covariance

ensemble members of ESRKF are shown in Fig. (5.7). The estimated mean, the left plot of Fig. (5.8), resembles the true permeability, Fig. (5.1). A more quantitative comparison is given in section 5.4.4. The results using the prior with exponentially decreasing covariance are presented in Fig. (5.9) and Fig. (5.10).

Fig. (5.11) shows the first four statistical moments of the ensemble after picking 100 realizations from the database. The covariance is a matrix, but only the diagonal elements are plotted. The third- and fourth-order moments are higher order tensors, but also only the "diagonals" are plotted. Fig. (5.13) shows what happens when a singular value decomposition is used to initialize the ensemble. The probability density that is represented by the ensemble is skewed and therefore does not at all represent a Gaussian probability density. This negatively impacts the performance of the EnKF. A random orthogonal matrix can be used to de-skew the ensemble, preserving the first and second order statistical moments. This is shown in figures Fig. (5.12) and Fig. (5.14).

The randomness that is introduced by picking realizations from the database is largely suppressed by picking a large enough set of realizations, 100. If (5.11) were created again using 100 different realizations from the database, it looks quite similar, Fig. (5.15). This is not the case if the ensemble size is smaller.

5.4.3 Representer results

Permeability estimates of the RM using the correct prior and the prior with exponentially decreasing covariance are shown in Fig.(5.16) and Fig. (5.17). Fig. (5.18) and Fig. (5.20) show the parameter representer. The model errors that are reconstructed by the RM are presented in Fig. (5.22).

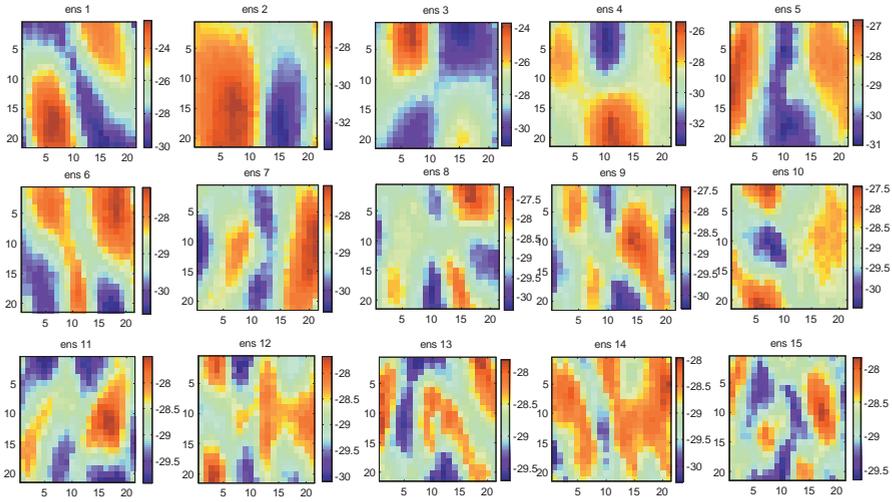


Figure 5.5: Some initial ensemble members

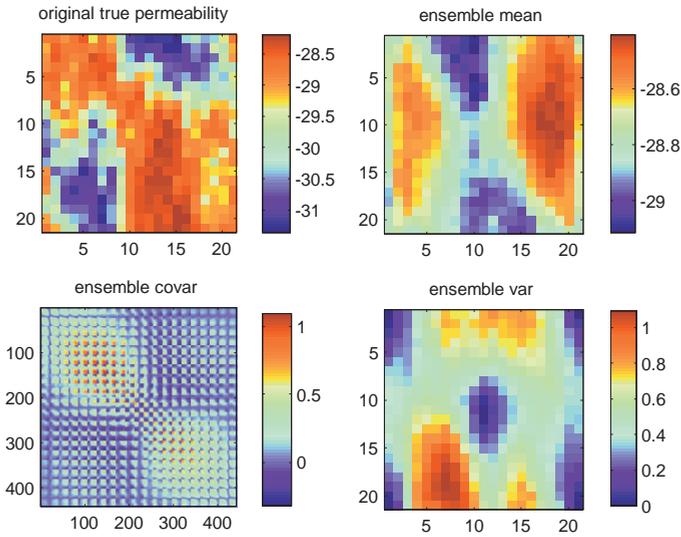


Figure 5.6: Synthetic true permeability, initial ensemble mean, covariance and variance.

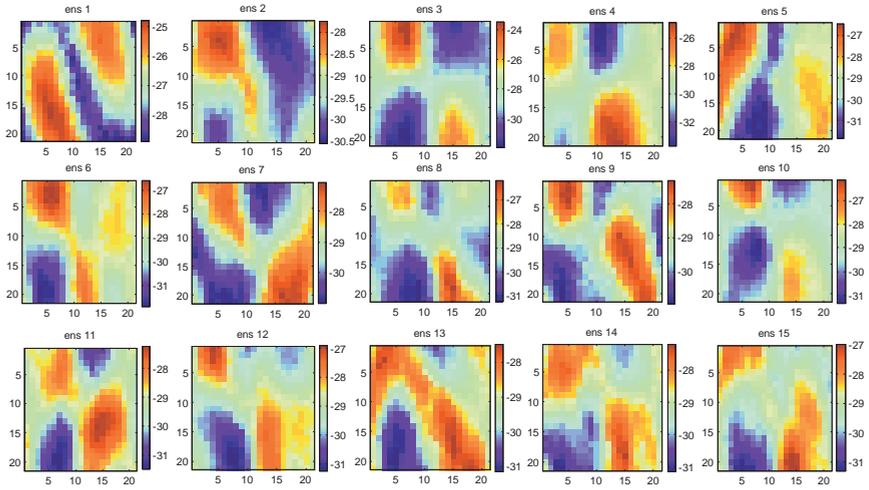


Figure 5.7: Some ensemble members, estimated with ESRKF using the correct prior.

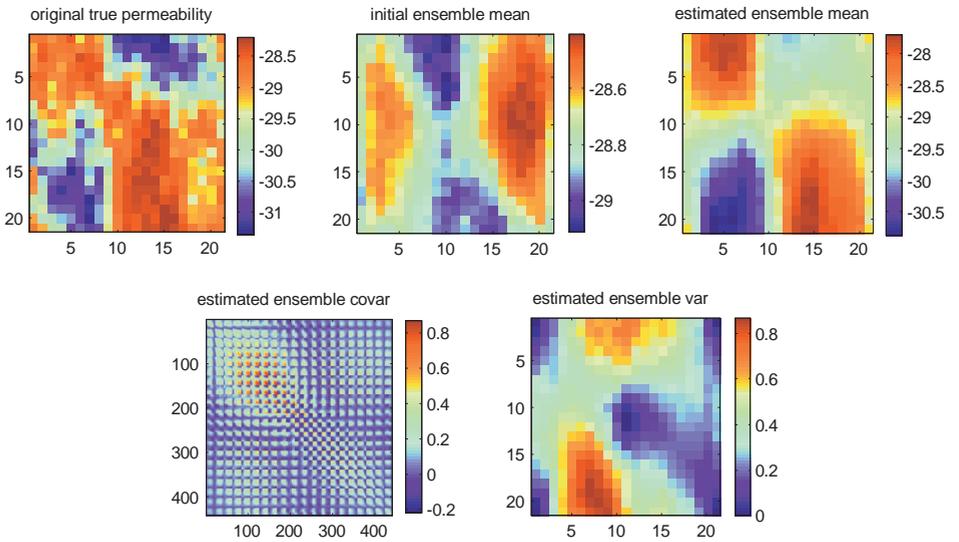


Figure 5.8: Synthetic true permeability, initial ensemble mean, and estimated ensemble mean, covariance and variance, estimated with ESRKF using the correct prior.

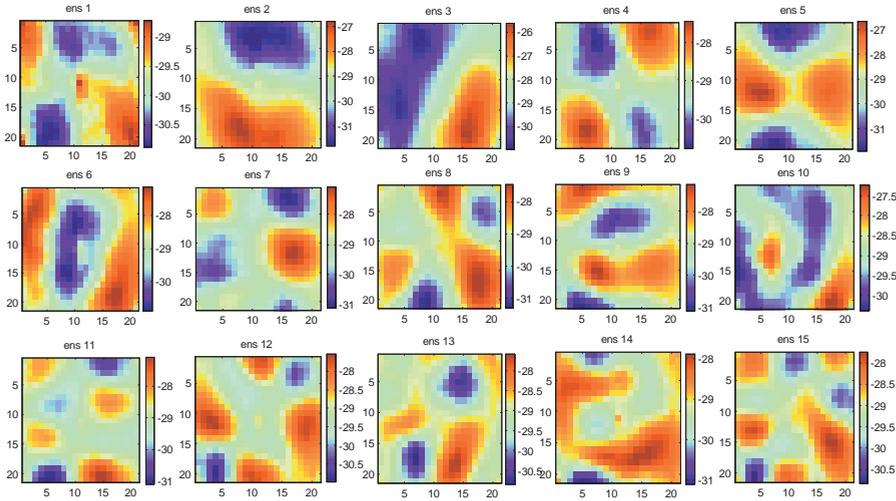


Figure 5.9: Some ensemble members, estimated with ESRKF using a prior with exponentially decreasing covariance.

5.4.4 Alternative quantification of success

Unlike in a field application, in a synthetic experiment the reconstructed parameters can be compared to the ones that were used to synthesize the data. Consequently, the RMSE between all ensemble members and the truth can be calculated. A histogram is plotted in Fig. (5.23) for the case where the correct prior was used.

Another important quantification of the success of a data assimilation algorithm in reservoir management comes from looking at the water breakthrough curves in the wells. It is too late to react to water breakthrough in the production wells after it has been observed. A data assimilation method may therefore add considerable value if a reasonable prediction of the water breakthrough can be made (long) before it actually occurs. Only then the control strategy of the (smart) wells can be changed to delay the water breakthrough. Fig. (5.24) shows the water saturation curves for the synthetic truth in all wells. It also shows the same curves for one ensemble member of an EnKF. The curves are shifted in time to optimally fit the truth. The shift is a measure of how well the water breakthrough could be predicted. It is calculated as the integral average of the difference of the true and the predicted curve in the saturation domain:

$$\frac{1}{S_{final} - S_{init}} \int_{S_{init}}^{S_{final}} (t_{true}(S) - (t_{predicted} - t_{shift})) dS = 0. \quad (5.42)$$

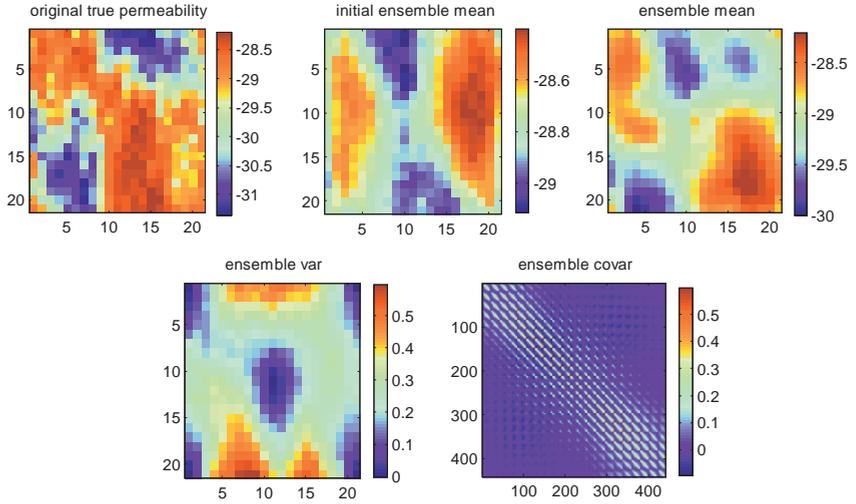


Figure 5.10: Synthetic true permeability, initial ensemble mean, and estimated ensemble mean, covariance and variance, estimated with ESRKF using an exponential prior.

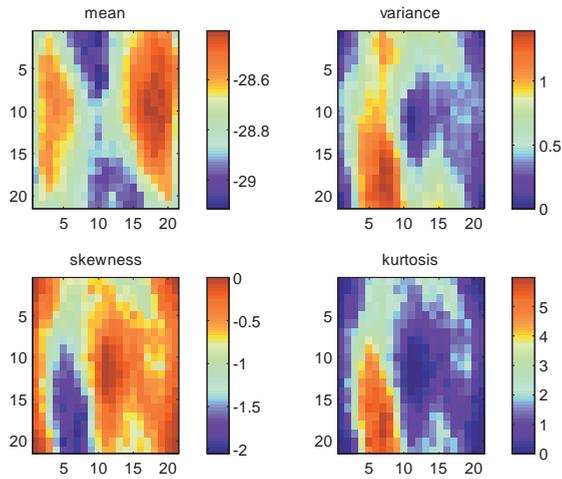


Figure 5.11: Four statistical moments of the ensemble. 100 members were picked randomly from the database.

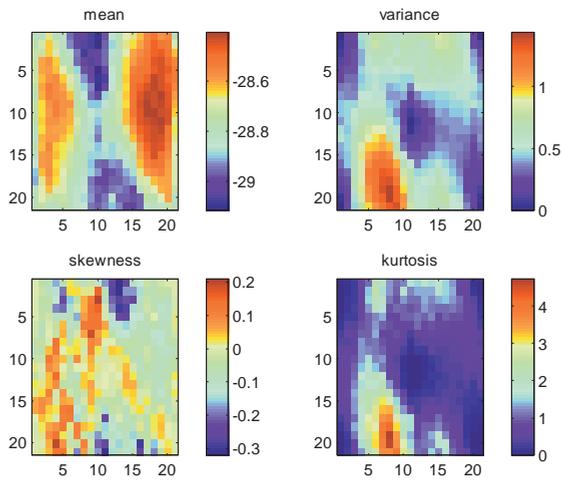


Figure 5.12: Four statistical moments of the ensemble. 100 members were picked randomly from the database and then 100 random linear combinations were created.

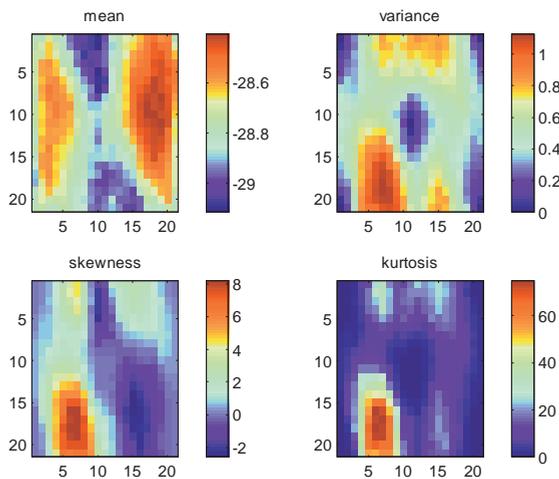


Figure 5.13: Four statistical moments of the ensemble. 100 leading singular vectors were calculated from the database.

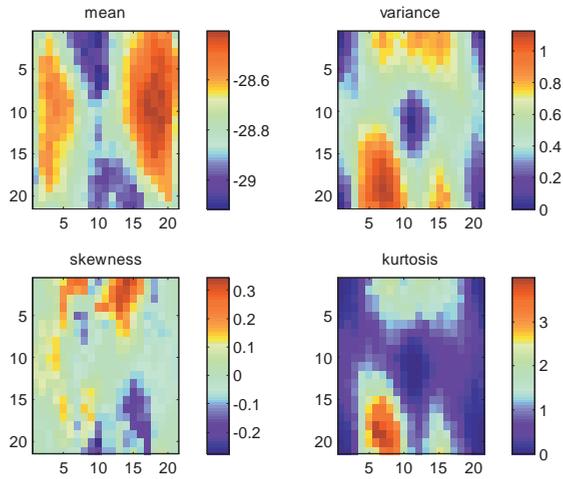


Figure 5.14: Four statistical moments of the ensemble. 100 leading singular vectors were calculated from the database and then 100 random linear combinations were created.

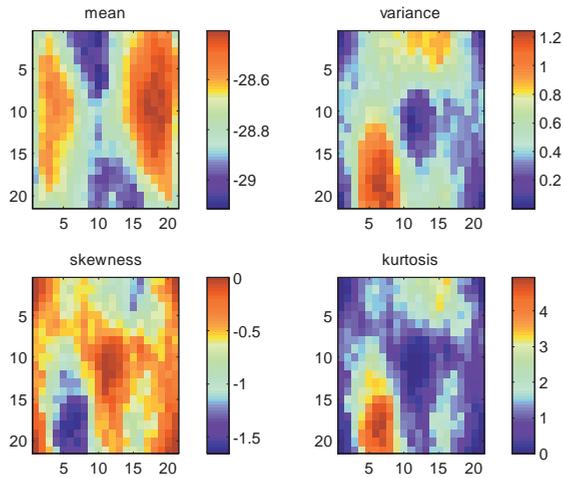


Figure 5.15: Four statistical moments of the ensemble. 100 members were picked again from the database, different from Fig. (5.11).

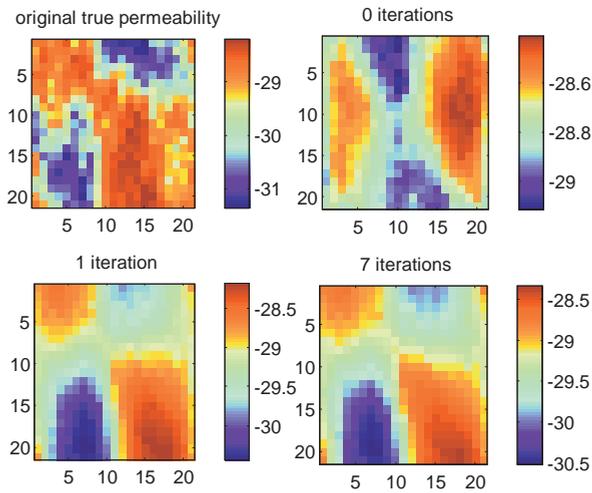


Figure 5.16: Permeability estimates of the RM using the correct prior

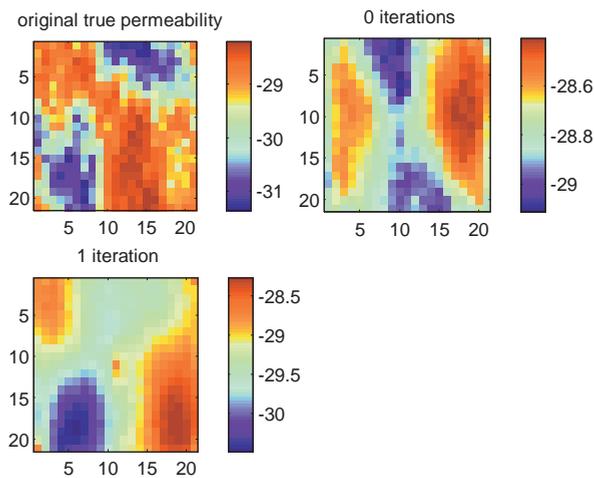


Figure 5.17: Permeability estimates of the RM using the prior with exponentially decreasing covariance

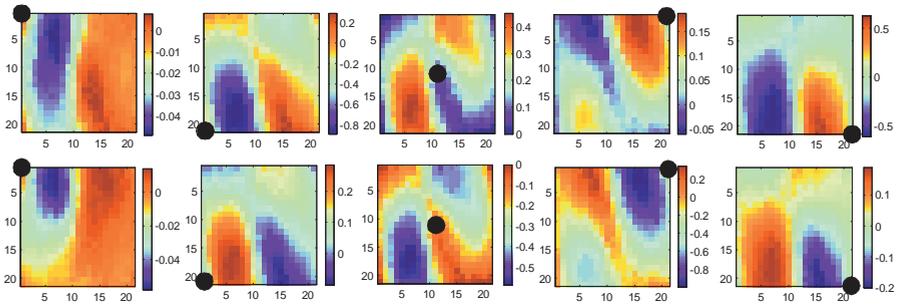


Figure 5.18: Parameter representers using correct prior. Different columns represent the deviation of the permeability estimates from the prior by different measurement locations (the wells). The locations are denoted by the black dots. Different rows represent different assimilation times.

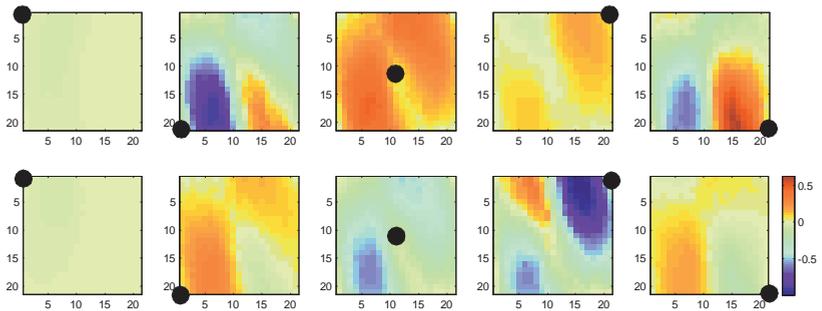


Figure 5.19: Parameter representers using correct prior, plotted on the same scale. Different columns represent the deviation of the permeability estimates from the prior by different measurement locations (the wells). The locations are denoted by the black dots. Different rows represent different assimilation times.

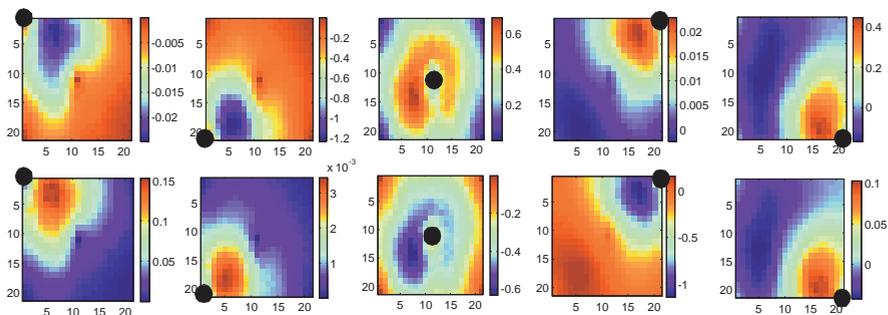


Figure 5.20: Parameter representers using exponential prior. Different columns represent the deviation of the permeability estimates from the prior by different measurement locations (the wells). The locations are denoted by the black dots. Different rows represent different assimilation times.

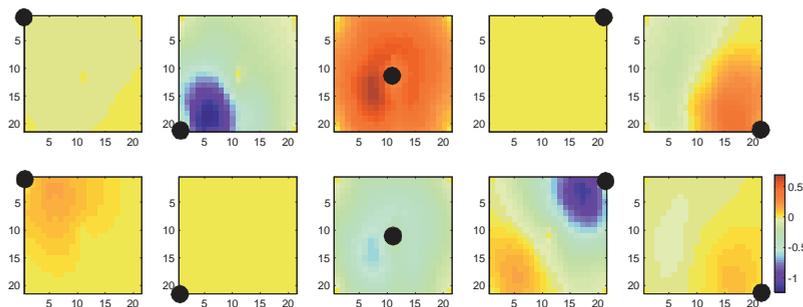


Figure 5.21: Parameter representers using exponential prior, plotted on the same scale. Different columns represent the deviation of the permeability estimates from the prior by different measurement locations (the wells). The locations are denoted by the black dots. Different rows represent different assimilation times.

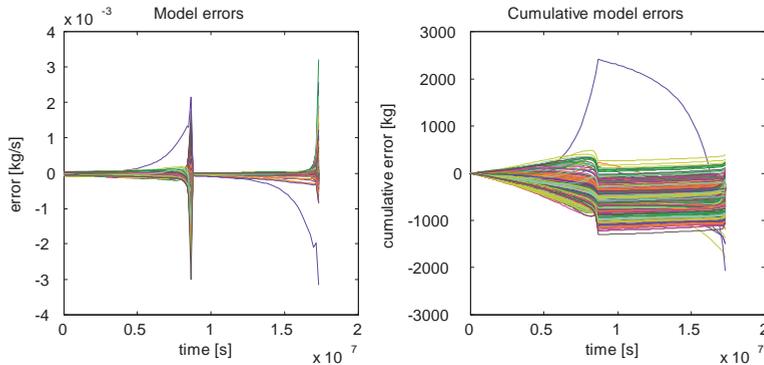


Figure 5.22: Model errors reconstructed by the RM using the correct prior

After t_{shift} has been calculated, the area between the true and the shifted predicted curve can be used to quantify the performance of the assimilation method to predict the shape of the water front as it breaks through:

$$A = \int_{S_{init}}^{S_{final}} |t_{true}(S) - (t_{predicted} - t_{shift})| dS. \quad (5.43)$$

A histogram can be plotted, Fig. (5.25), when t_{shift} is calculated for all ensemble members. For comparison with the RM, where the ensemble consists of only one ensemble member, the mean of the ensemble of t_{shift} values is used to quantify the performance of the filter.

Some numbers:

Forty experiments were done to gain insight in the effects of different prior, ensemble size, initialization method, de-skewing method and measurement update on the performance of the EnKF/ESRKF. In [Rommelse *et al.* (2007)] the performance of the RM was investigated for different magnitudes of model errors, different line search algorithms, different minimization algorithms and compared to the results of adjoint-based strong constraint parameter estimation algorithms, but only the correct prior was used in the data assimilation process. The RM options that led to the best results of that research were used in this chapter and applied to a twin experiment where a prior with exponentially decreasing covariance was used. Fig. (5.26) visualizes the results of the forty-two experiments; it shows the cross plots of two different quality measures for all experiments. $\|t_{shift}\|$ is calculated as the 2-norm of the 5-dimensional vector with the means of Fig. (5.25). RMSE is the mean from Fig. (5.23). The five different plots represent results obtained with different ensemble sizes, or with different measurement updates, but they all contain the same RM results for comparison. In the ESRKF, measurements are assimilated sequentially using the duplication as described in section 5.2.3. In the EnKF, measurements at different times are assimilated sequentially,

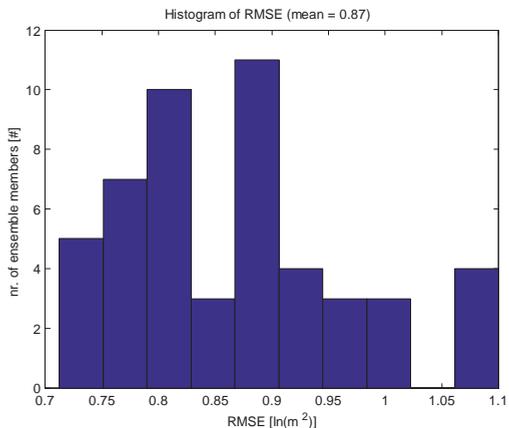


Figure 5.23: Root-mean-square difference between estimated permeability and the truth, for all ensemble members. 50 ensemble members were randomly picked from the database.

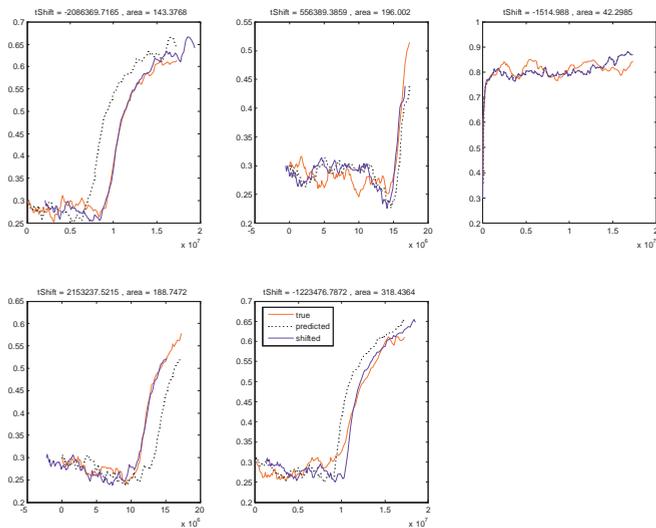


Figure 5.24: True, predicted and shifted [s] water breakthrough curves of one ensemble member for all wells

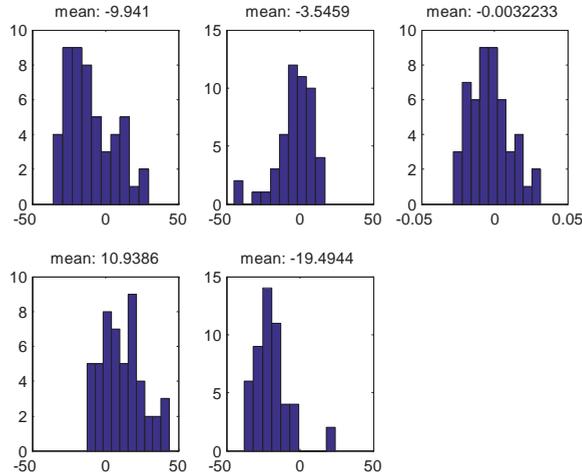


Figure 5.25: True, predicted and shifted [days] water breakthrough curves of all ensemble member for all wells

but measurements that were taken at the same time are also assimilated simultaneously and without duplication. Effects of randomness between the different experiments are reduced by storing and re-using the random numbers. This way "between-subject" variability is reduced and the effects of "within-subject" variation can be studied [van der Poel and Jansen (2004)]. The results of the ensemble methods are independent of the output of the random number generator if the ensemble size is chosen sufficiently large. For a smaller ensemble size (10, 25), results vary, but the effect of the ensemble size decreases between 50 through 100.

In terms of RMSE, the filter always performs worse than the RM. In terms of predicting the water breakthrough curves, the performance of the RM can be reached by a filter if the ensemble size is chosen sufficiently large and the correct prior is used. This is not true if a prior with exponentially decreasing covariance is used, so it can be said that the RM was less sensitive to using the "wrong" prior in this application than the filter. No matter what prior is used, the filter usually performs best if it is initialized by SVD and then de-skewed. If no de-skewing is applied, then randomly initializing the ensemble usually produces better results than using SVD for the correct prior, but SVD performs better for the exponential prior. This can be explained as follows; randomly picking ensemble members from the database gives a rather symmetric ensemble, whereas the SVD ensemble is very skewed. For the prior with exponential decreasing covariance, no such database is available. Picking ensemble members

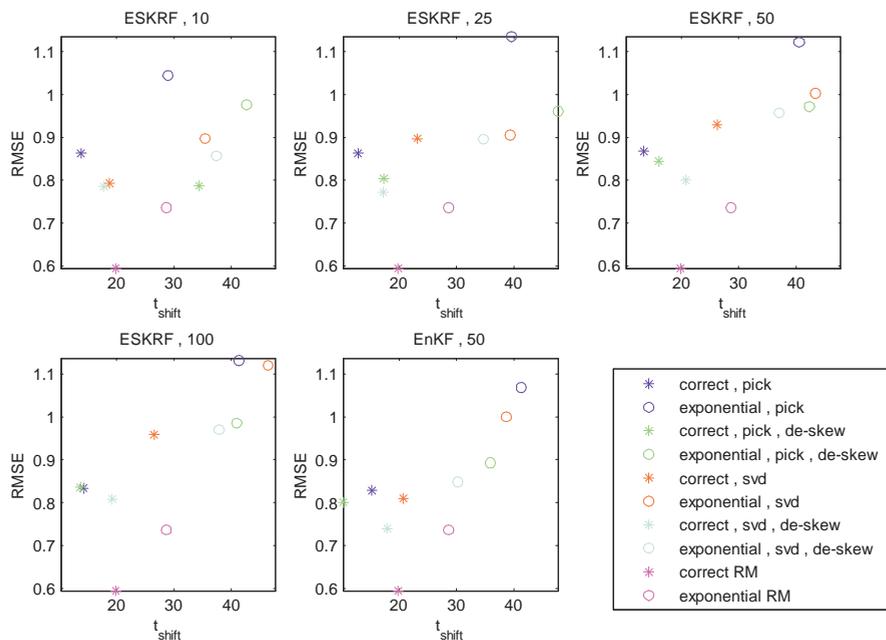


Figure 5.26: Crossplot of the two quality measures (RMSE and t_{shift}) for different ensemble sizes, EnKF or ESRKF, correct prior and prior with exponential decreasing covariance, ensemble randomly picked or obtained by svd and with or without de-skewing of the ensemble. All plots also contain the results of the Representer Method.

involves picking columns from a Cholesky vector of the covariance matrix and results in a skewed ensemble as well.

It is worth mentioning that ESRKF with duplicated measurements performs reasonably better than EnKF in terms of RMSE and much better in terms of $\|t_{Shift}\|$ in the presence of no or little measurement errors. In the presence of moderate or large measurement errors, the results of EnKF or ESRKF are similar. The measurement update of EnKF makes a crude Gaussian assumption, whereas the version with duplicated measurements makes a set of smaller assumptions. The EnKF is a major improvement over the Extended Kalman filter for dealing with nonlinearities in the reservoir model, but it is recommended not to use the EnKF for non-linear problems without a modified measurement update if the measurements are very accurate. Experiments to support this statement are not presented in this chapter, because the scope of this chapter is to compare the EnKF/ESRKF with the RM and not to show all details of the EnKF.

5.5 Conclusions

The RM always performed better than the filter, in terms of RMSE between estimated and true parameters. If the posterior probability of the parameters given the measurements is not symmetric or even multi-modal, it makes sense that the RM performs better. In that case the probability that the true parameters lie close to the mode of the posterior is larger than the probability that the true parameters lie near the mean of the posterior. Also, the recursive filter introduces errors during every measurement update. These errors accumulate. An iterative algorithm has the opportunity in every iteration to reduce the errors that were made in the previous iteration.

When using the correct prior, predicting the water breakthrough can be done equally well by the filter as by the RM if the ensemble size is chosen sufficiently large. This also shows that models that are not equally well history-matched might produce future predictions equally well. This implies that a good history-match may not guarantee good predictions.

When using the prior with exponentially decreasing covariance, the filter does not perform as well as the RM, both in terms of RMSE and predicted water breakthrough. It can be said that the RM is less sensitive to using a "wrong" prior than a filter in this application.

The best filter results can be obtained if the ensemble is initialized with SVD and then de-skewed. Without de-skewing, SVD creates ensembles that are very skewed, causing poor results in estimating model parameters. Randomly initializing the ensemble for the case with the exponential prior involves picking columns from a Cholesky factor of the covariance matrix, which also results in a skewed ensemble.

Chapter 6

Variational estimation of permeability and model errors from 3D and 4D seismic data using model-driven regularization⁶

Abstract

Automated history-matching methods, or data assimilation algorithms, can be used to support decision-making tools in closed-loop reservoir management. In variational data assimilation, the discrepancy between observed measurements and their model predicted antitheses is minimized with respect to parameters that underlie the reservoir model. Assuming that there are no uncertainties in addition to the unknown parameters, there are methods that can efficiently calculate the gradient of the discrepancy to changes in the parameters. Usually many different parameter sets exist that locally minimize the discrepancy, so the gradient must be regularized before it can be used by gradient-based minimization algorithms. In the presence of model errors, more advanced methods must be employed, like the Representer Method (RM).

This chapter proposes a variational method to estimate the permeability of a reservoir rock from static and time-lapse seismic data, that simultaneously estimates the model errors in the reservoir simulator. Unlike in the (classic or modified) RM, the regularization and the computation of the gradient are decoupled, which can save a lot of computation time. An analytical gradient is available, whereas the RM uses an approximated one.

First the Variational Parameter Estimation method with Model-driven Regularization (VPERM) is introduced and the relation to the RM is explained. Experiments are presented, using three different data sets. The first data set contains synthetic pressure measurements in the injection and production wells, the second and third data sets contain P-wave impedance data in addition to pressure data. The second data set contains the baseline survey and the monitor, the third one contains the baseline and the difference.

VPERM produces results that are similar to the results of the RM, but in less computation time. Both methods produce good results when only pressure data from the wells are available. Sometimes seismic data removes outliers from the "history-future" crossplot; better predictions are obtained from models that are not necessarily better history-matched. In other cases, seismic data gives better history-matched models and better predictions.

⁶ This chapter is based on [Rommelse *et al.* (2008b)], which was published as TUD-DIAM report 08-18 and submitted to Computational Geosciences

6.1 Introduction

This chapter investigates the applicability of a weak constraint variational method to estimate reservoir permeability from 3D or 4D seismic data with a reservoir simulator that contains model errors. Sections 6.1.1 and 6.1.2 introduce the variational parameter estimation and weak constraints approaches. [Skjervheim *et al.* (2005)] used the Ensemble Kalman Filter (EnKF) [Evensen (2003)] to estimate permeability from seismic data in the presence of model errors. In order to do so, a non-linear rock physics model was augmented to the non-linear reservoir model to predict changes in seismic response from changes in fluid pressure and mixture. A weak constraint variational method is promising, since [Rommelse *et al.* (2008a)] showed that for a linear measurement operator the Representer Method (RM) [Bennett and McIntosh (1982), Eknes and Evensen (1997), Bennett (2002), Valstar *et al.* (2004), Baird and Dawson (2005), Janssen *et al.* (2006), Przybysz *et al.* (2007)] outperformed the EnKF in terms of two quality measures; root mean square difference between estimated and synthetic true permeability, and the ability to predict water breakthrough in production wells.

6.1.1 Variational parameter estimation

Data assimilation methods aim to improve numerical models by comparing actual measurements of a physical system with the numerical model predictions of these measurements. The predicted state variables and the predicted measurements change as the parameters of the numerical model are perturbed. The discrepancy between the "measured measurements" and the "predicted measurements" can be used to update the parameters in order to decrease this discrepancy. In variational methods, the discrepancy is formulated as an objective function that has to be minimized, often using the Euclidean norm. The dynamic model that relates parameters to state variables, is usually adjoined to the objective function with the aid of Lagrange multipliers [Bennett (2006)]. In reservoir management, the dynamic model is implemented in the form of a reservoir simulator, the parameters of interest can for example be porosity or permeability, and the state of the system is typically described by pressure and saturation values in all grid blocks. The first or higher order variations are iteratively used to improve the parameter estimate. Gradients can be approximated or efficiently calculated by an adjoint reservoir simulator [Rommelse *et al.* (2007)]. Usually reservoir simulators use time-implicit numerical schemes and first order derivatives, Jacobians, are used to speed up the non-linear solvers. It is quite a task to construct the adjoint simulator, even when the Jacobians are available in a reservoir simulator. Higher order derivatives are usually not available. In the LBFSGS [Gao and Reynolds (2006), Ulbrich (2002)] minimization algorithm used in this chapter, second order derivatives are approximated by monitoring the first order derivatives over successive iterations and used in a Gauss-Newton scheme.

6.1.2 Model errors; strong and weak constraints

The discrepancy between measured measurements and predicted measurements is often formulated using the Euclidean norm. The objective of the data assimilation is to minimize the square of this norm with respect to the model parameters while the numerical model is used as a (strong) constraint.

However, there is an additional phenomenon that may cause the discrepancy. The model is an approximation, so even if the parameters were known, the model may still produce incorrect output. These errors can be modelled as extra parameters, which are also added to the objective function using the 2-norm. The model is then used as a weak constraint in the minimization problem.

6.1.3 Regularization

Variational methods often get trapped in local minima. Regularization methods must be employed to reduce the number of local minima. When model errors are taken into account, these can be modelled as additional model parameters to turn the weak constraint minimization problem into a strong constraint problem of much higher order. Regularization then becomes increasingly important.

The RM [Bennett and McIntosh (1982), Bennett (2002), Valstar *et al.* (2004)] turns the weak constraint problem into a strong constraint problem of the same order as the number of measurements. The regularization is part of the minimization scheme and the basis functions are recalculated at every iteration. [Rommelse *et al.* (2007)] proposes an approximated gradient and an additional reduction of the dimension of the parameter space.

This chapter proposes a Variational Parameter Estimation algorithm that uses Model-driven Regularization (VPERM, section 6.2), similar to the RM. The regularization is decoupled from the minimization scheme. Basis functions can be updated at any iteration, but they do not necessarily need to be. The weak constraint problem is reduced to a strong constraint problem of an order that is much lower than the number of measurements. Assimilating seismic data would not have been feasible otherwise. The gradient is analytical if the basis functions are kept fixed, unlike the approximated gradient of the RM.

6.2 The VPERM method

This section introduces the VPERM method; Variational Parameter Estimation Regularized by Model-driven basis functions.

6.2.1 High dimensional gradient of the weak constraint problem

The state variables, pressures and saturations in all grid blocks, at time t_i are denoted by \mathbf{x}_i , $i \in \{0, \dots, n\}$. Running the reservoir simulator and predicting measurements is denoted by

$$\mathbf{g}(\mathbf{x}_i, \mathbf{x}_{i-1}, \boldsymbol{\theta}, \boldsymbol{\varepsilon}_i) = \mathbf{0} \quad , \quad \mathbf{y} = \mathbf{h}(\mathbf{x}_{\{0, \dots, n\}}), \quad (6.1)$$

where the model parameters are collected in the vector $\boldsymbol{\theta}$ and the model errors on interval $[t_{i-1}, t_i]$ are contained in the vector $\boldsymbol{\varepsilon}_i$. The initial states may be part of the parameter estimation process, so $\mathbf{x}_0 = \mathbf{x}_0(\boldsymbol{\theta})$. The minimization relies on the availability of the first (mean) and second (covariance) order statistics of the model parameters and model errors. These are denoted by $\boldsymbol{\theta}^{prior}$, \mathbf{P}_θ , $\boldsymbol{\varepsilon}_i^{prior} = \mathbf{0}$ and $\mathbf{P}_{\varepsilon_i}$.

The objective function that has to be minimized is

$$\begin{aligned} J &= \frac{1}{2} (\mathbf{h}(\mathbf{x}_{\{0, \dots, n\}}) - \mathbf{m})^T \mathbf{P}_y^{-1} (\mathbf{h}(\mathbf{x}_{\{0, \dots, n\}}) - \mathbf{m}) + \\ &+ \frac{1}{2} (\boldsymbol{\theta} - \boldsymbol{\theta}^{prior})^T \mathbf{P}_\theta^{-1} (\boldsymbol{\theta} - \boldsymbol{\theta}^{prior}) + \frac{1}{2} \sum_{i=1}^n \boldsymbol{\varepsilon}_i^T \mathbf{P}_{\varepsilon_i}^{-1} \boldsymbol{\varepsilon}_i + \\ &+ \sum_{i=1}^n \lambda_i^T \mathbf{g}(\mathbf{x}_i, \mathbf{x}_{i-1}, \boldsymbol{\theta}, \boldsymbol{\varepsilon}_i), \end{aligned} \quad (6.2)$$

where \mathbf{m} contains the actual physical measurements, possibly taken at different times, and \mathbf{h} is the measurement operator that operates on all state variables at all time steps. \mathbf{P}_y represents the uncertainty in the measurements in the form of an error covariance matrix. The last term of Eq. (6.2) represents the system equations \mathbf{g} that have been adjoined to the objective function with the aid of Lagrange multipliers λ_i in the usual fashion [Bennett (2006)].

The derivatives of Eq. (6.2) with respect to λ_i , \mathbf{x}_i , $\boldsymbol{\theta}$ and $\boldsymbol{\varepsilon}_i$ are

$$\left(\frac{\partial J}{\partial \lambda_i} \right)^T = \mathbf{g}(\mathbf{x}_i, \mathbf{x}_{i-1}, \boldsymbol{\theta}, \boldsymbol{\varepsilon}_i), \quad (6.3)$$

$$\begin{aligned} \left(\frac{\partial J}{\partial \mathbf{x}_i} \right)^T &= \left(\frac{\partial \mathbf{h}(\mathbf{x}_{\{0, \dots, n\}})}{\partial \mathbf{x}_i} \right)^T \mathbf{P}_y^{-1} (\mathbf{h}(\mathbf{x}_{\{0, \dots, n\}}) - \mathbf{m}) + \\ &+ \left(\frac{\partial \mathbf{g}(\mathbf{x}_i, \mathbf{x}_{i-1}, \boldsymbol{\theta}, \boldsymbol{\varepsilon}_i)}{\partial \mathbf{x}_i} \right)^T \lambda_i + \left(\frac{\partial \mathbf{g}(\mathbf{x}_{i+1}, \mathbf{x}_i, \boldsymbol{\theta}, \boldsymbol{\varepsilon}_{i+1})}{\partial \mathbf{x}_i} \right)^T \lambda_{i+1}, \end{aligned} \quad (6.4)$$

$$\left(\frac{\partial J}{\partial \boldsymbol{\theta}} \right)^T = \mathbf{P}_\theta^{-1} (\boldsymbol{\theta} - \boldsymbol{\theta}^{prior}) + \sum_{i=1}^n \left(\frac{\partial \mathbf{g}(\mathbf{x}_i, \mathbf{x}_{i-1}, \boldsymbol{\theta}, \boldsymbol{\varepsilon}_i)}{\partial \boldsymbol{\theta}} \right)^T \lambda_i, \quad (6.5)$$

$$\left(\frac{\partial J}{\partial \boldsymbol{\varepsilon}_i} \right)^T = \mathbf{P}_{\varepsilon_i}^{-1} \boldsymbol{\varepsilon}_i + \left(\frac{\partial \mathbf{g}(\mathbf{x}_i, \mathbf{x}_{i-1}, \boldsymbol{\theta}, \boldsymbol{\varepsilon}_i)}{\partial \boldsymbol{\varepsilon}_i} \right)^T \lambda_i. \quad (6.6)$$

Setting the first two derivatives equal to zero gives the reservoir simulator, Eq. (6.1) and the adjoint reservoir simulator

$$\begin{aligned} & \left(\frac{\partial \mathbf{h}(\mathbf{x}_{\{0, \dots, n\}})}{\partial \mathbf{x}_i} \right)^T \mathbf{P}_y^{-1} (\mathbf{m} - \mathbf{h}(\mathbf{x}_{\{0, \dots, n\}})) \\ &= \left(\frac{\partial \mathbf{g}(\mathbf{x}_i, \mathbf{x}_{i-1}, \boldsymbol{\theta}, \mathbf{0})}{\partial \mathbf{x}_i} \right)^T \boldsymbol{\lambda}_i + \left(\frac{\partial \mathbf{g}(\mathbf{x}_{i+1}, \mathbf{x}_i, \boldsymbol{\theta}, \mathbf{0})}{\partial \mathbf{x}_i} \right)^T \boldsymbol{\lambda}_{i+1}. \end{aligned} \quad (6.7)$$

Thus, for the weak constraint case, the gradient $\left[\frac{\partial J}{\partial \boldsymbol{\theta}} \quad \frac{\partial J}{\partial \varepsilon_1} \quad \dots \quad \frac{\partial J}{\partial \varepsilon_n} \right]^T$ is calculated by running the simulator and the adjoint simulator, using the current estimates of the parameters and model errors as input, and substituting the results in Eq. (6.5) and Eq. (6.6). The gradient can be used in any gradient-based search algorithm to find a local minimum of the objective function.

6.2.2 Low dimensional gradient of a strong constraint problem

The low dimensional problem is parameterized by a vector \mathbf{b} . In this low dimensional space there is no model and hence no model errors. The model is only defined in the high order space. From the low order parameters \mathbf{b} , the high order parameters $\boldsymbol{\theta}$ are constructed by

$$\boldsymbol{\theta} = \tilde{\boldsymbol{\theta}} + \mathbf{R}_\theta \mathbf{b}, \quad (6.8)$$

and the high order model errors are constructed by

$$\boldsymbol{\varepsilon}_i = \tilde{\boldsymbol{\varepsilon}}_i + \mathbf{R}_{\varepsilon_i} \mathbf{b}. \quad (6.9)$$

The columns of the matrices \mathbf{R}_θ and \mathbf{E}_i contain the parameter and model error basis functions. Whenever these basis functions are calculated, they are based on the results of the last iteration $\tilde{\boldsymbol{\theta}}$ and $\tilde{\boldsymbol{\varepsilon}}_i$ and the low order parameters are reset to $\mathbf{b} = \mathbf{0}$. Before the first iteration, $\tilde{\boldsymbol{\theta}}$ is initialized to $\tilde{\boldsymbol{\theta}} = \boldsymbol{\theta}^{prior}$ and $\tilde{\boldsymbol{\varepsilon}}_i$ is initialized to $\tilde{\boldsymbol{\varepsilon}}_i = \mathbf{0}$. The RM also introduces state representers $\mathbf{R}_{\mathbf{x}_i}$ as

$$\mathbf{x}_i = \tilde{\mathbf{x}}_i + \mathbf{R}_{\mathbf{x}_i} \mathbf{b}, \quad (6.10)$$

and adjoint representers $\mathbf{R}_{\boldsymbol{\lambda}_i}$ as

$$\boldsymbol{\lambda}_i = \tilde{\boldsymbol{\lambda}}_i + \mathbf{R}_{\boldsymbol{\lambda}_i} \mathbf{b}. \quad (6.11)$$

The RM does not update $\tilde{\boldsymbol{\theta}}$, $\tilde{\boldsymbol{\varepsilon}}_i$ and $\tilde{\boldsymbol{\lambda}}_i = \mathbf{0}$, meaning that the quality of the parameterization decreases as parameters move further away from the prior. $\tilde{\mathbf{x}}_i$ is formed from the results of the last iteration plus a correction term that is updated at every iteration. Effectively $\tilde{\mathbf{x}}_i$ is never updated and it is obtained by running the reservoir simulator on the prior parameters $\boldsymbol{\theta}^{prior}$ without model errors. [Rommelse *et al.* (2007)] also introduced measurement representers

$$\mathbf{h}(\mathbf{x}_{\{0, \dots, n\}}) = \tilde{\mathbf{h}} + \mathbf{R}_y \mathbf{b}. \quad (6.12)$$

Using the basis functions, Eq. (6.8) and Eq. (6.9), the low order gradient $\left(\frac{\partial J}{\partial \mathbf{b}}\right)^T$ is calculated as

$$\begin{aligned} \left(\frac{\partial J}{\partial \mathbf{b}}\right)^T &= \left(\frac{\partial \theta}{\partial \mathbf{b}}\right)^T \left(\frac{\partial J}{\partial \theta}\right)^T + \sum_{i=1}^n \left(\frac{\partial \varepsilon_i}{\partial \mathbf{b}}\right)^T \left(\frac{\partial J}{\partial \varepsilon_i}\right)^T = \\ &= \mathbf{R}_\theta^T \left(\frac{\partial J}{\partial \theta}\right)^T + \sum_{i=1}^n \mathbf{R}_{\varepsilon_i}^T \left(\frac{\partial J}{\partial \varepsilon_i}\right)^T, \end{aligned} \quad (6.13)$$

with $\left(\frac{\partial J}{\partial \theta}\right)^T$ and $\left(\frac{\partial J}{\partial \varepsilon_i}\right)^T$ from Eq. (6.5) and Eq. (6.6).

6.2.3 Choosing the basis functions

To find new basis functions \mathbf{R}_θ and $\mathbf{R}_{\varepsilon_i}$, equations Eq. (6.5) and Eq. (6.6) are set to zero and linearized around $(\tilde{\theta}, \tilde{\mathbf{x}}_i, \tilde{\lambda}_i, \tilde{\varepsilon}_i)$. The results are:

$$\mathbf{R}_\theta = -\mathbf{P}_\theta \sum_{i=1}^n \left(\frac{\partial \mathbf{g}(\mathbf{x}_i, \mathbf{x}_{i-1}, \theta, \varepsilon_i)}{\partial \theta}\right)^T \mathbf{R}_{\lambda_i} \quad (6.14)$$

and

$$\mathbf{R}_{\varepsilon_i} = -\mathbf{P}_{\varepsilon_i} \left(\frac{\partial \mathbf{g}(\mathbf{x}_i, \mathbf{x}_{i-1}, \theta, \varepsilon_i)}{\partial \varepsilon_i}\right)^T \mathbf{R}_{\lambda_i}. \quad (6.15)$$

Obviously, first basis functions \mathbf{R}_{λ_i} for the adjoint variables must be chosen. Similar to Eq. (6.14) and Eq. (6.15), equations for the adjoint basis functions can be found

$$\begin{aligned} -\left(\frac{\partial \mathbf{h}(\mathbf{x}_{\{0, \dots, n\}})}{\partial \mathbf{x}_i}\right)^T \mathbf{P}_y^{-1} \frac{\partial \mathbf{h}(\mathbf{x}_{\{0, \dots, n\}})}{\partial \mathbf{x}_i} \mathbf{R}_{\mathbf{x}_i} &= \left(\frac{\partial \mathbf{g}(\mathbf{x}_i, \mathbf{x}_{i-1}, \theta, \varepsilon_i)}{\partial \mathbf{x}_i}\right)^T \mathbf{R}_{\lambda_i} \\ &+ \left(\frac{\partial \mathbf{g}(\mathbf{x}_{i+1}, \mathbf{x}_i, \theta, \varepsilon_{i+1})}{\partial \mathbf{x}_i}\right)^T \mathbf{R}_{\lambda_{i+1}}, \end{aligned} \quad (6.16)$$

which depend on the state variable basis functions $\mathbf{R}_{\mathbf{x}_i}$, that can be calculated from

$$\frac{\partial \mathbf{g}}{\partial \mathbf{x}_i} \mathbf{R}_{\mathbf{x}_i} + \frac{\partial \mathbf{g}}{\partial \mathbf{x}_{i-1}} \mathbf{R}_{\mathbf{x}_{i-1}} + \frac{\partial \mathbf{g}}{\partial \theta} \mathbf{R}_\theta = \mathbf{O}. \quad (6.17)$$

In the RM, Eq. (6.16) and Eq. (6.17) can be solved sequentially, whereas here they are coupled through Eq. (6.14). If an estimate of the basis functions \mathbf{R}_θ for the parameters θ is available, for example from a previous iteration, new basis functions can be calculated by sequentially solving Eq. (6.17), Eq. (6.16) and Eq. (6.14). This can be used in a Picard iteration scheme. Here a different approach is taken; $\mathbf{R}_{\mathbf{x}_i}$ in Eq. (6.16) is obtained by a

singular value decomposition:

$$\begin{aligned} & \left(\frac{\partial \mathbf{h}(\mathbf{x}_{\{0, \dots, n\}})}{\partial \mathbf{x}_i} \right)^T \mathbf{P}_y^{-1} \frac{\partial \mathbf{h}(\mathbf{x}_{\{0, \dots, n\}})}{\partial \mathbf{x}_i} \mathbf{R}_{\mathbf{x}_i} \\ &= \mathbf{U} \Sigma \mathbf{V}^T \mathbf{X}_i = \mathbf{U} \Sigma \begin{bmatrix} \mathbf{V}_1^T \\ \mathbf{V}_2^T \end{bmatrix} \mathbf{V}_1 = \mathbf{U} \Sigma \begin{bmatrix} \mathbf{I} \\ \mathbf{O} \end{bmatrix}. \end{aligned} \quad (6.18)$$

When the left-most columns of $\mathbf{U} \Sigma$ are substituted in Eq. (6.16) to replace the left-hand-side, the resulting equation can be used to find approximations for \mathbf{R}_{λ_i} . Thereafter, these can be used in Eq. (6.14) and Eq. (6.15).

6.2.4 Quadratic line search

After the objective $J_0 = J(\boldsymbol{\theta})$ and the gradient $\mathbf{g}_0 = \frac{dJ}{d\boldsymbol{\theta}}$ have been evaluated, a step must be taken in the direction of the gradient or a modified direction \mathbf{d} . If the step size is parameterized by s , then $\tilde{J}(s) = J(\boldsymbol{\theta} - s\mathbf{d})$ must be minimized. The gradient is not only used to find the direction of the line search, but it also helps in finding the step size, as is the case with Wolfe conditions [Nocedal and Wright (1999)]. However, here a different approach is taken:

1. Choose some potential step size s_p .
2. Evaluate the objective $\tilde{J}(s_p) = J(\boldsymbol{\theta} - s_p \mathbf{d})$.
3. If $\tilde{J}(s_p) \leq \tilde{J}(0) - \mathbf{g}_0^T \mathbf{d}$, then accept s_p and end line search.
4. Calculate a new potential step size

$$\tilde{s}_p := \frac{1}{2} s_p \frac{\mathbf{g}_0^T \mathbf{d} s_p}{\mathbf{g}_0^T \mathbf{d} s_p + J_{s_p} - J_0}. \quad (6.19)$$

5. Goto 2 if $\tilde{J}(\tilde{s}_p) > \tilde{J}(0)$.

Comments:

- Eq. (6.19) minimizes the parabola that is defined by $\tilde{J}(0) = J_0$, $\tilde{J}(s_p) = J_{s_p}$, and $\frac{d\tilde{J}}{ds}(0) = -\mathbf{g}_0^T \mathbf{d}$.
- If $J_{s_p} = J_0$, then $\tilde{s}_p := \frac{s_p}{2}$, unless $\|\mathbf{g}_0\| = 0$ (but if that were the case, then $J(\boldsymbol{\theta})$ is a local minimum of J).
- In step 2, the objective function is evaluated. In step 5 it is evaluated again. However, in numerical experiments [section 6.4], the step from 5 to 2 is never made.

6.3 The models

6.3.1 Weak constraint reservoir simulator

A 2-phase (water/oil) reservoir simulator can be represented symbolically as

$$\frac{d}{dt} (\mathbf{f}_1(\mathbf{x})) = \mathbf{f}_2(\mathbf{x}, \boldsymbol{\theta}), \quad (6.20)$$

where \mathbf{x} contains the water saturation and water pressure (equal to oil pressure) for every grid block and $\boldsymbol{\theta}$ contains the permeabilities of all grid blocks. \mathbf{f}_1 describes the presence of water and oil mass in the grid blocks and \mathbf{f}_2 models the flow through the grid block interfaces. Injection/production is modelled as sources/sinks, which are included in \mathbf{f}_2 . Eq. (6.20) is formed by mass balance and Darcy equations [Aziz and Settari (1979)].

A fully implicit time discretization results in

$$\tilde{\mathbf{g}}(\mathbf{x}_i, \mathbf{x}_{i-1}, \boldsymbol{\theta}) = \mathbf{f}_1(\mathbf{x}_i) - (t_i - t_{i-1}) \mathbf{f}_2(\mathbf{x}_i, \boldsymbol{\theta}) - \mathbf{f}_1(\mathbf{x}_{i-1}) = \mathbf{0}. \quad (6.21)$$

The model errors are introduced as additional sources/sinks in all grid blocks. In other words, after \mathbf{x}_i has been solved from Eq. (6.21), the water and oil masses in the grid blocks have not correctly been predicted and must still be modified. The prediction grows worse as the time step $(t_i - t_{i-1})$ increases. The correction is therefore modelled proportional to $(t_i - t_{i-1})$. If the additional sources become too strong, unrealistically high pressures will be observed. If the additional sinks become too strong, saturations outside $[0, 1]$ will occur. In this chapter, the additional sinks are non-linearly constrained by \mathbf{f}_1 . The stochastic reservoir simulator has the form

$$\begin{aligned} \mathbf{g}(\mathbf{x}_i, \mathbf{x}_{i-1}, \boldsymbol{\theta}, \boldsymbol{\varepsilon}_i) &= \mathbf{0} = \\ &= \mathbf{f}_1(\mathbf{x}_i) - (t_i - t_{i-1}) \mathbf{f}_2(\mathbf{x}_i, \boldsymbol{\theta}) - \mathbf{f}_1(\mathbf{x}_{i-1}) + \min\{\mathbf{f}_1(\mathbf{x}_i), (t_i - t_{i-1}) \boldsymbol{\varepsilon}_i\}. \end{aligned} \quad (6.22)$$

6.3.2 Rock-physics model

In addition to production data, seismic data can be used to estimate parameters in a reservoir simulation model, even though acoustic wave propagation is a totally different physical phenomenon than multiphase fluid flow. Seismic data can consist of travel time data and amplitude data, recorded by geophones or hydrophones. Travel time data are available as the difference between the time at which human-made sources initiate acoustic waves that travel into the subsurface and the time at which the reflected waves are recorded, usually at the surface, but sometimes inside wellbores. In "passive seismic" the travel time data are the difference between recordings of the same acoustic wave by different sets of geophones in different well bores. In this case "mother nature" is used instead of a human-made source.

Like reservoir engineers use reservoir simulators to predict the flow of fluids through the subsurface, geophysicists make use of wave propagation simulators to predict how acoustic waves travel through the subsurface. And like history-matching in reservoir engineering, inverse methods are applied in geophysics that bring the predictions of wave simulations closer to the actual recorded waves by changing the parameters of the wave simulator. The parameter sets of both simulators have some overlap. Porosity for example, is an important parameter in both simulators. Theoretically, seismic data and production data could be merged into one big dataset and the parameter sets of both simulators could be merged to create one big inversion or parameter estimation problem. This has never been done. Usually one inversion problem is solved partially and the results are used as pseudo-data in the other inversion problem.

This chapter assumes that the seismic data has been inverted partially, until the level of seismic impedance. The P-wave impedance is augmented to the production data to help the history-matching of the reservoir simulation model. Besides a reservoir simulator, also an impedance simulator is needed to predict seismic impedance from fluid pressures and saturations. This dependence is shown in Fig (6.1) and described in the rest of this section.

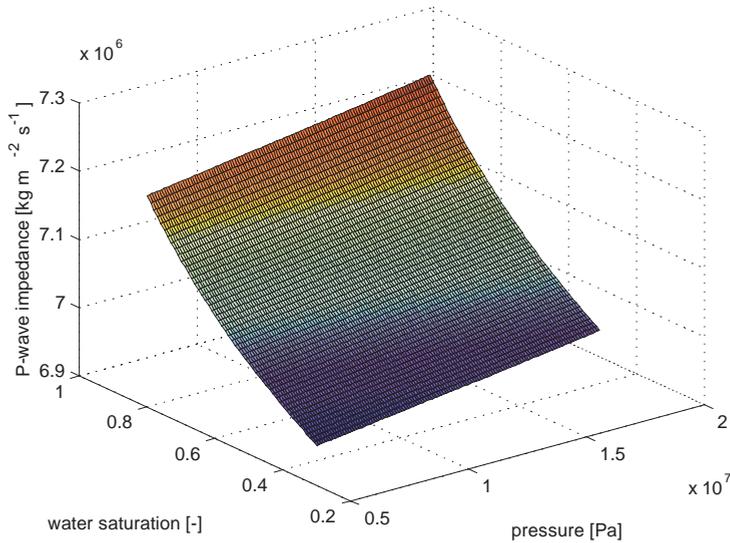


Figure 6.1: The dependence of the P-wave impedance on the fluid pressure and water saturation.

The P-wave impedance $Z_p [kg\ m^{-2}\ s^{-1}]$ and S-wave impedance $Z_s [kg\ m^{-2}\ s^{-1}]$ are defined as the velocities of the P and S-waves through the subsurface, $V_p [m\ s^{-1}]$ and $V_s [m\ s^{-1}]$, multiplied by the density of the (fluid-filled reservoir) rock $\rho_b [kg\ m^{-3}]$, which is a weighted average of the densities of the sandstone, $\rho_s [kg\ m^{-3}]$, and the fluids

$$\rho_b = \phi (S_w \rho_w + S_o \rho_o) + (1 - \phi) \rho_s. \quad (6.23)$$

Including the dependence of the velocities on the fluid filled bulk modulus $K_{ff} [Pa]$ and shear modulus $\mu_m [Pa]$, the impedances can be written as

$$Z_p = \sqrt{\left(K_{ff} + \frac{4}{3}\mu_m\right) \rho_b} \quad , \quad Z_s = \sqrt{\mu_m \rho_b}. \quad (6.24)$$

The fluid filled bulk modulus K_{ff} is calculated according to Gassmann [Mavko *et al.* (1998)]

$$K_{ff} = K_s \left(\frac{1 - \frac{K_m}{K_s} + K_m \phi \left(\frac{1}{K_f} - \frac{1}{K_s} \right)}{1 - \frac{K_m}{K_s} + K_s \phi \left(\frac{1}{K_f} - \frac{1}{K_s} \right)} \right), \quad (6.25)$$

or with $\alpha = \phi \left(\frac{1}{K_f} - \frac{1}{K_s} \right)$ and $\beta = 1 - \frac{K_m}{K_s}$:

$$K_{ff} = K_s \left(\frac{K_m \alpha + \beta}{K_s \alpha + \beta} \right), \quad (6.26)$$

which combines the bulk moduli of the sandstone, $K_s [Pa]$, the dry-frame $K_m [Pa]$ and the fluid $K_f [Pa]$. This last modulus is calculated from the bulk modulus of oil, $K_o [Pa]$ and bulk modulus of water, $K_w [Pa]$, by

$$K_f = \frac{1}{\frac{S_w}{K_w} + \frac{S_o}{K_o}} = \frac{K_w K_o}{S_w K_o + S_o K_w}. \quad (6.27)$$

The elastic moduli, the porosity and the densities are assumed uncertain. However, deterministic values are used and noise is added to the synthetic P-wave impedance, rather than to the underlying parameters separately, section [6.3.3].

6.3.3 Measuring impedance with synthetic noise

Noise is added after the true impedance has been synthesized. However, this is done in such a way that the difference of impedance measured at different times contains less noise than the separate measurements. So

$$\tilde{Z}_{t_1} = Z_{t_1} + e_d + e_{t_1}, \quad (6.28)$$

$$\tilde{Z}_{t_2} = Z_{t_2} + e_d + e_{t_2}, \quad (6.29)$$

$$\tilde{Z}_{t_1} - \tilde{Z}_{t_2} = Z_{t_1} - Z_{t_2} + e_{t_1} - e_{t_2}. \quad (6.30)$$

If the error in the impedance measurements is modelled by a Gaussian distribution

$$e_d + e_{t_1} \sim N\left(0, \sigma_Z^2\right), \quad (6.31)$$

and so is the error in the impedance difference

$$e_{t_1} - e_{t_2} \sim N\left(0, \sigma_{dZ}^2\right), \quad (6.32)$$

then e_{t_1} , e_{t_2} and e_d are also Gaussian

$$e_{t_1}, e_{t_2} \sim N\left(0, \frac{\sigma_{dZ}^2}{2}\right), \quad (6.33)$$

$$e_d \sim N\left(0, \sigma_Z^2 - \frac{\sigma_{dZ}^2}{2}\right). \quad (6.34)$$

The uncertainty in the baseline survey σ_Z^2 , the monitor σ_Z^2 and the difference σ_{dZ}^2 are parameters of interest in the numerical experiments. The uncertainties in the synthetic noise, $\frac{\sigma_{dZ}^2}{2}$ and $\sigma_Z^2 - \frac{\sigma_{dZ}^2}{2}$, are derived from these.

6.4 Numerical experiments

Twin experiments were done on a horizontal 2D 2phase waterflooding application with two different permeability and well configurations; an inverted 5-spot with vertical wells (5SPOT) and two horizontal wells perpendicular to two high-permeable streaks (2STREAKS). Synthetic pressure and impedance data were generated by running a reservoir simulator on a "synthetic truth" permeability field. The synthetic truths for 5SPOT and 2STREAKS and the well configurations for both experiments are shown in Fig. (6.2).

Water is injected at a rate of one pore volume per year and the production wells are constrained to a quarter (5SPOT) or one (2STREAKS) pore volumes per year. The state of the reservoir is described by pressure and water saturation in all 21x21x1 grid blocks of 10x10x20 m each. Capillary pressure and gravity effects are ignored. For 5SPOT, the true permeability is picked as one realization out of a database of 1000 realizations. The others realizations are used to construct a covariance matrix which is used in the objective function. For 2STREAKS, the truth is an academic caricature and realizations were sampled from a covariogram [Vossepoel and Douma (2008)]. The synthetic true model errors were sampled as white noise, similar to chapter 4. The state variables of the synthetic truth, located at the well positions, are shown in Fig. (6.3) for 5SPOT and Fig. (6.4) for 2STREAKS as functions of time. The synthetic pressure measurements are the reservoir pressures in the grid blocks that are penetrated by wells at 100 and 200 days of the simulation, resulting in 10 measurements for 5SPOT and 84 for 2STREAKS. Measurement errors are sampled and added to the

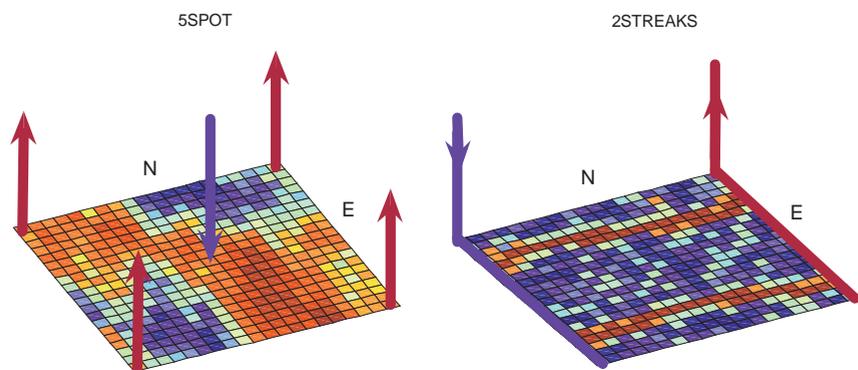


Figure 6.2: Well configurations. One vertical injector and four vertical producers for 5SPOT and two horizontal wells for 2STREAKS

synthetic measurements. Fig. (6.5) and Fig. (6.6) show the state variables after 200 days of simulation. The 882 impedance measurements, including measurement noise, are shown in Fig. (6.7) and Fig. (6.8).

Fig. (6.9) and Fig. (6.10) show some permeability estimates and the objective function during 38 iterations of VPERM on 5SPOT. Fig. (6.11) presents the estimated well responses. Fig. (6.12), Fig. (6.13) and Fig. (6.14) show the same for 2STREAKS. All iterations use the pressure data from the grid blocks that are penetrated by wells. Baseline and time-lapse data are used starting from iteration 20. If impedance data are used from the first iteration, a different local minimum of the objective function is found, usually one that gives a worse history-match. Fig. (6.15) presents the objective function of RM for the cases where seismic data are used in all iterations or only after the twentieth iteration. The first case converges much slower than the second. Moreover, the root-mean-square-error (RMSE) between the true and the estimated parameters is better for the second case; 0.5660 compared to 0.7536. If a history-matching workflow without seismic data gives reasonable results, it is not yet a trivial exercise to add seismic data. Additional data do potentially give more information, but also increases the dimensionality of the non-linear estimation problem. Using impedance data from the twentieth iteration instead of the first one is a practical trick to get a variational history-matching workflow to work with impedance data. The extra seismic data introduces new local minima in the neighborhood of the initial/prior estimate. Modifications to get an Ensemble Kalman Filter to work with saturation data for 2STREAKS were investigated by [Vossepoel and Douma (2008)].

Once a history-matching workflow is set up to work with or without seismic data, the influence of the seismic data can be assessed. In this research seismic data never had a negative effect and usually had a positive effect. In order to assess the effect of seismic data,

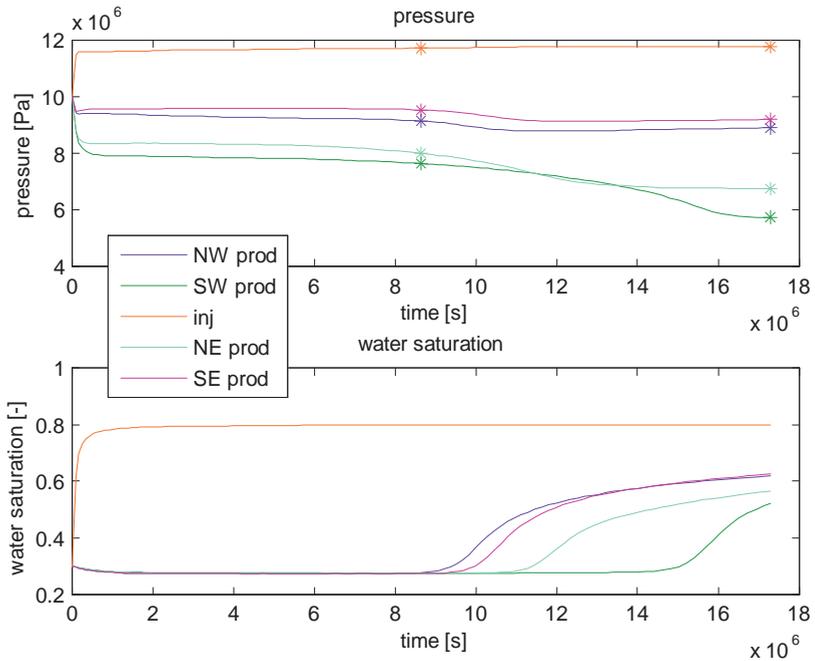


Figure 6.3: State variables of the synthetic truth for 5SPOT. Only the 5 grid blocks that are penetrated by the injection well and the north west (NW), SW, NE and SE production wells are shown. The top plot also shows synthetic pressure measurements prior to adding measurement noise, indicated by the asterisks.

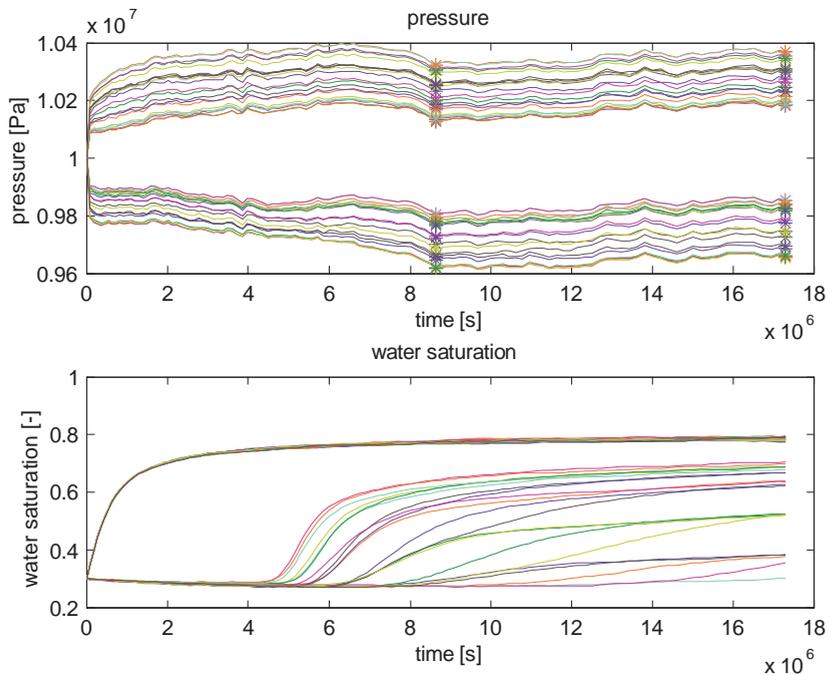


Figure 6.4: State variables of the synthetic truth for 2STREAKS. Only the 42 grid blocks that are penetrated by the horizontal injector and producer are shown. The top plot also shows synthetic pressure measurements prior to adding measurement noise, indicated by the asterisks.

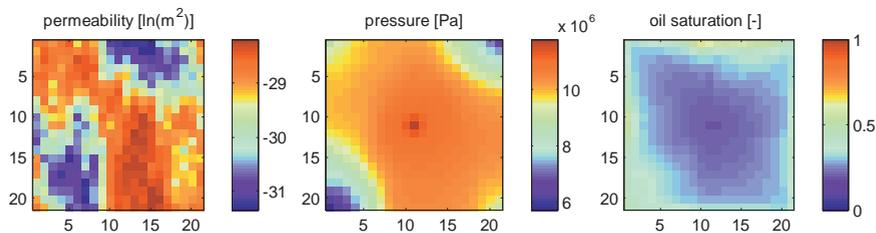


Figure 6.5: State variables after 200 days of simulation for 5SPOT.

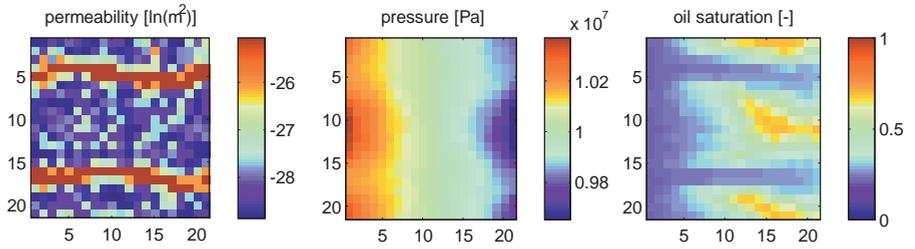


Figure 6.6: State variables after 200 days of simulation for 2STREAKS.

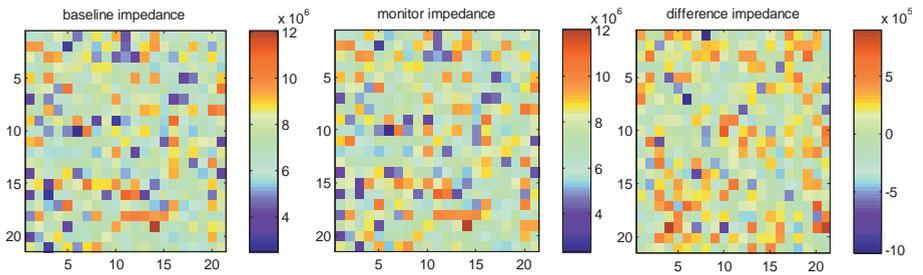


Figure 6.7: Impedance measurements for 5SPOT.

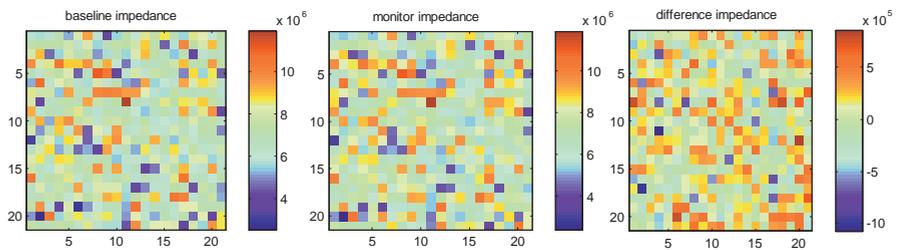


Figure 6.8: Impedance measurements for 2STREAKS.

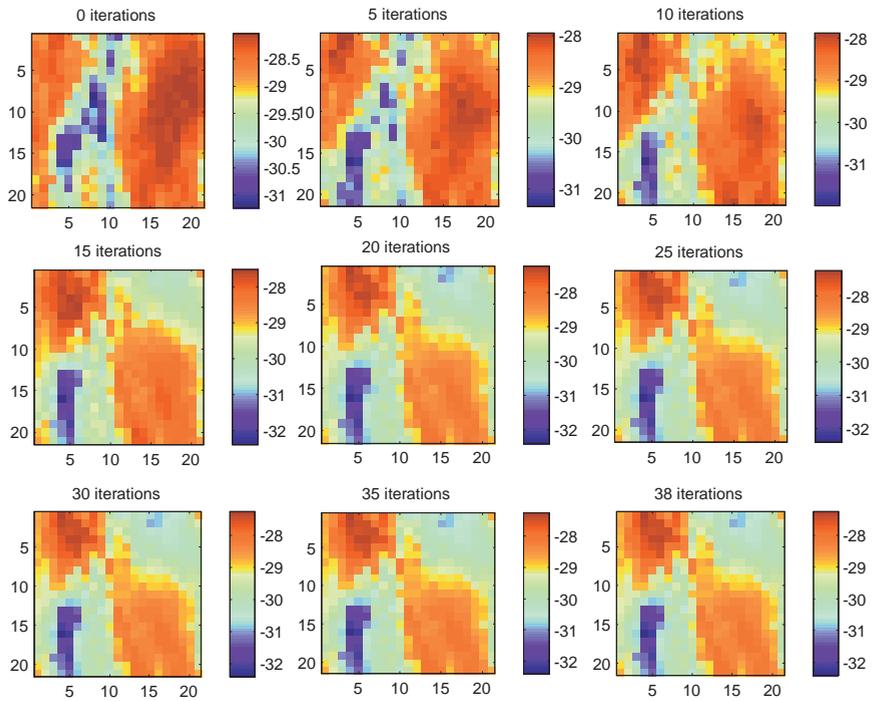


Figure 6.9: Permeability estimate during 38 iterations with VPERM for 5SPOT. Pressure data in the wells is used for all iterations, baseline and time-lapse impedance data is used in iterations 21 and up.

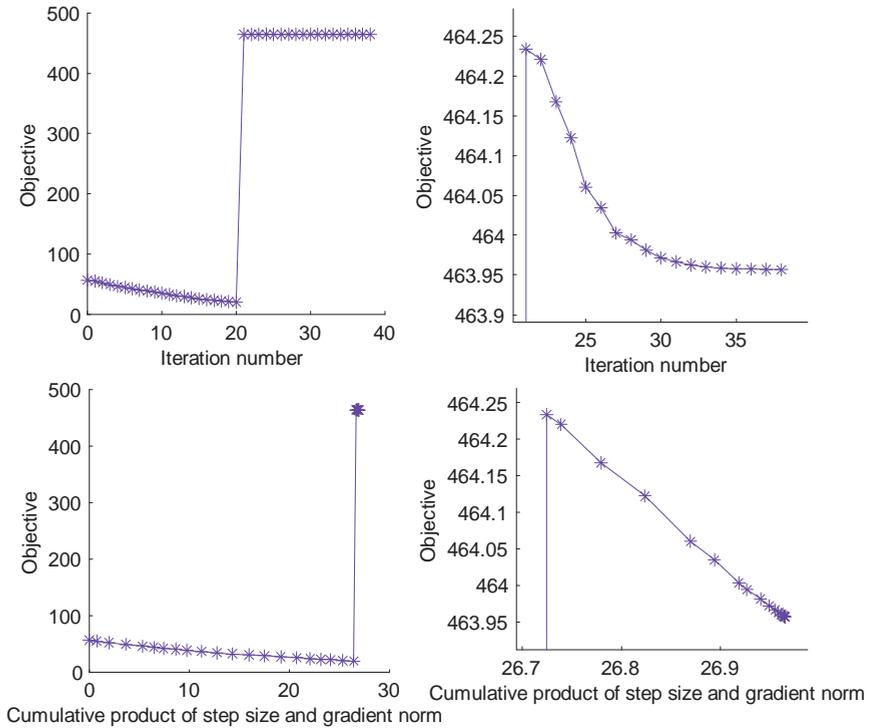


Figure 6.10: Objective function during 38 iterations with VPERM for 5SPOT. Pressure data in the wells is used for all iterations, baseline and time-lapse impedance data is used in iterations 21 and up. The plots on the right are close-ups of the plots on the left.

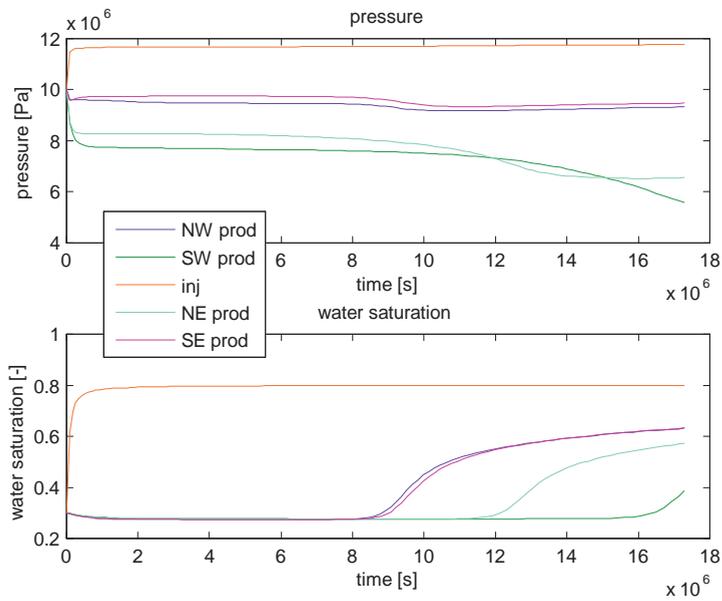


Figure 6.11: Estimated well response after 38 iterations with VPERM for 5SPOT. Pressure data in the wells is used for all iterations, baseline and time-lapse impedance data is used in iterations 21 and up.

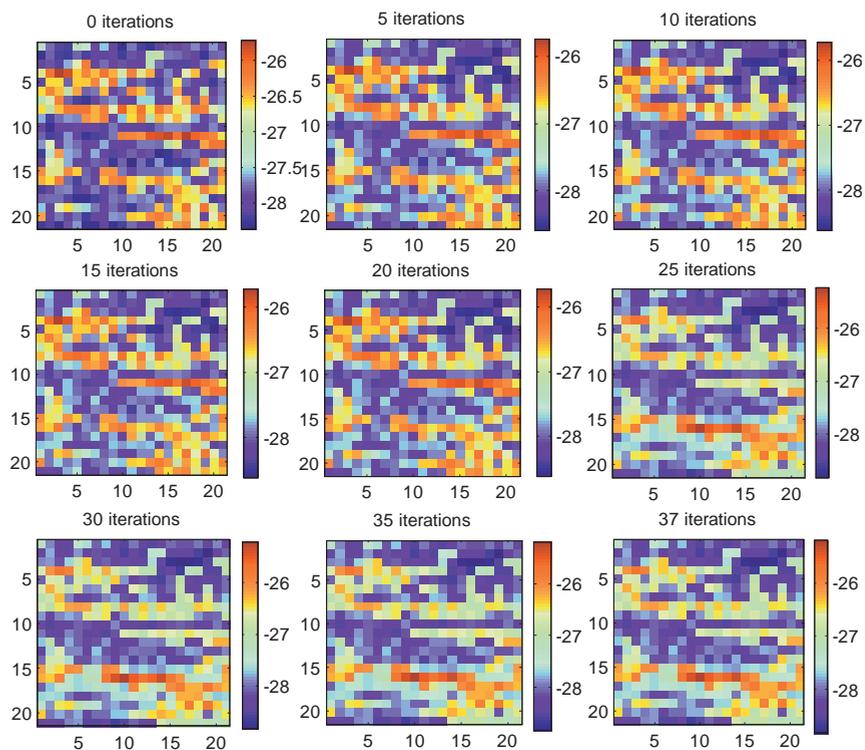


Figure 6.12: Permeability estimate during 37 iterations with VPERM for 2STREAKS. Pressure data in the wells is used for all iterations, baseline and time-lapse impedance data is used in iterations 21 and up.

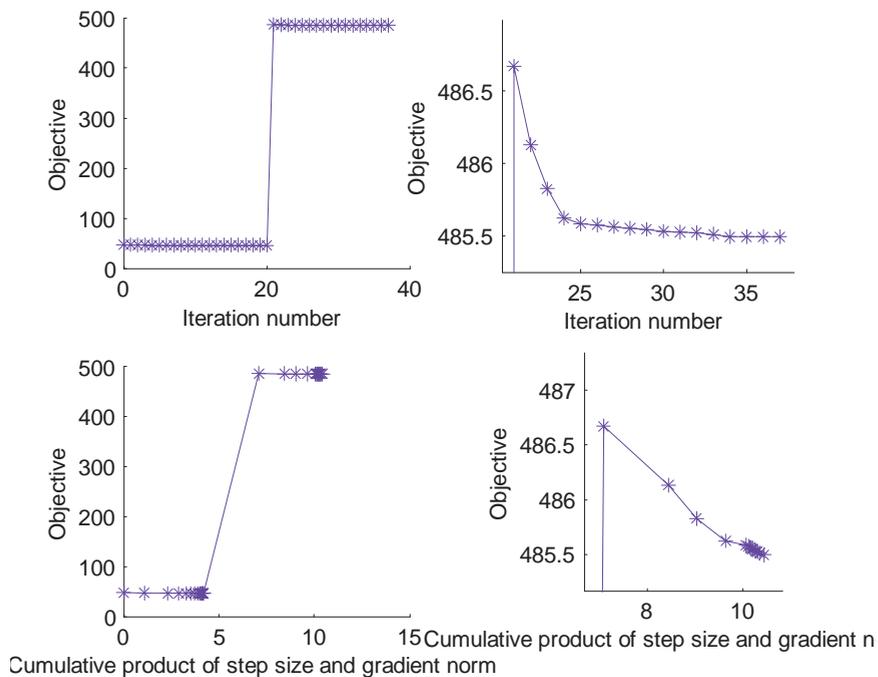


Figure 6.13: Objective function during 37 iterations with VPERM for 2STREAKS. Pressure data in the wells are used for all iterations, baseline and time-lapse impedance data are used in iterations 21 and up. The plots on the right are close-ups of the plots on the left.

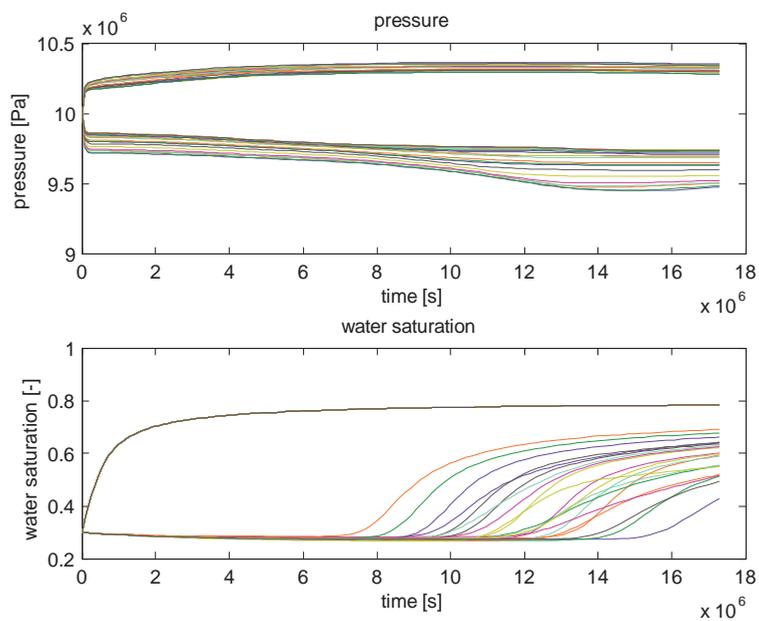


Figure 6.14: Estimated well response after 37 iterations with VPERM for 2STREAKS. Pressure data in the wells are used for all iterations, baseline and time-lapse impedance data are used in iterations 21 and up.

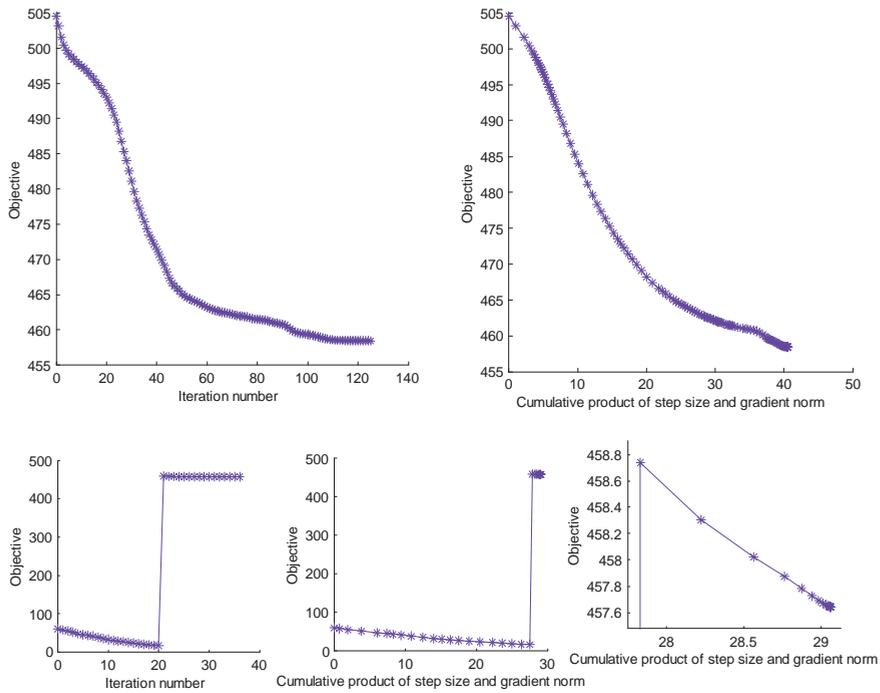


Figure 6.15: Objective function of 5SPOT using RM. Top: all iterations use pressure data and baseline and time-lapse impedance data. Bottom: impedance data are only used after iteration 20.

crossplots were made between the history-match and the "future-match", Fig. (6.16), Fig. (6.17) and Fig. (6.18), where the history-matching workflows were repeated with different priors/initial estimates and with three different datasets. All datasets contain the pressure data in the grid blocks that are penetrated by wells. The second and third dataset contain the baseline impedance data. The second dataset contains the monitor impedance data and the third dataset contains the time-lapse impedance data. The "history-match" is quantified by the Root Mean Square Error (RMSE) between the estimated permeability and the synthetic truth permeability. The "future-match" is quantified by looking at the accuracy of the water breakthrough curves. In fact, it is the square of the 2-norm of a vector that contains the time shifts of the estimated well responses to best match the synthetic well responses. For 5SPOT, Fig. (6.16), 3D seismic data does contribute to better history-matched models, although these models do not have better predictive capability than models that were history-matched without seismic data. 4D seismic data reduces the outliers in the crossplot, meaning that the models do have a better future predictive ability, without being better history-matched. This is also the message of [Walker and Lane (2007)]. It must be noted that 3D seismic data are actually 2D, and 4D are actually 3D, since the reservoir model is only 2D and not 3D. For 2STREAKS, Fig. (6.18), 3D seismic data have hardly any effect. 4D seismic data give better history-matched models as well as better "future-matched" models. Fig. (6.17) is a combination of 5SPOT and 2STREAKS; 2STREAKS is used as the synthetic truth and realizations were created by subtracting the mean from 5SPOT, adding the synthetic truth of 2STREAKS and performing some random shuffle operations on the permeability fields. Again, the availability of seismic data removes the outliers in the crossplot.

The VPERM method turns a high-dimensional weak constraint parameter estimation problem into a lower dimensional strong constraint parameter estimation problem by choosing basis functions for the parameters and the model errors inspired by the representer method. In the absence of model errors, it is very common to reduce the dimensionality of the strong constraint problem by choosing basis functions for the parameters, for example by calculating the principal components of the parameter covariance using SVD. Optionally, these basis functions may also be used for the parameters in a VPERM workflow. Sometimes this even results in faster convergence or convergence to a better local minimum, Fig. (6.19).

6.5 Conclusions

This chapter introduces a weak constraint Variational Parameter Estimation algorithm with Model-driven Regularization (VPERM). The regularization turns the high order weak constraint parameter estimation problem into a lower order strong constraint problem. The basis functions that perform the regularization are model-driven, like in the Representer Method (RM). However, if the basis functions are kept fixed in the VPERM, then an analytical gradient can be obtained, whereas the gradient in the RM is merely an approximation; it is the direction of a step in an iterative scheme with Picard iteration that is not proven to converge.

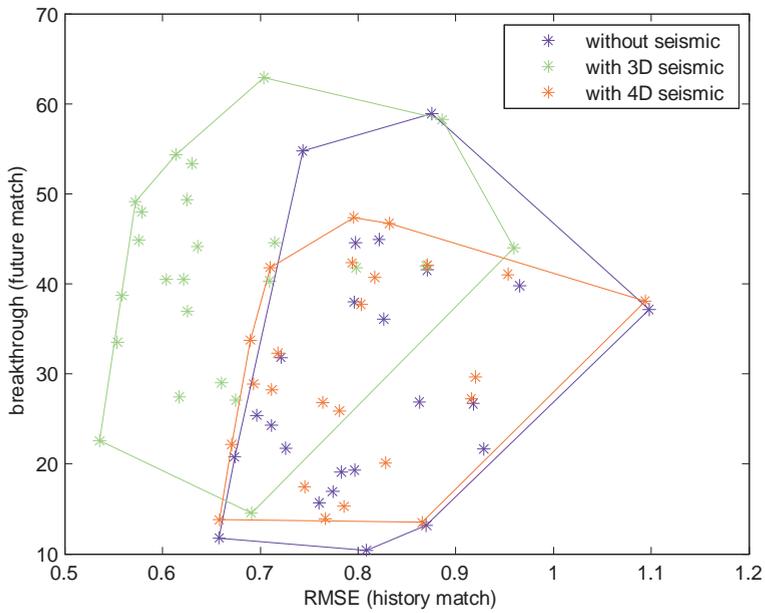


Figure 6.16: History-match future-match crossplot for 5SPOT using 25 different priors and 3 different datasets.

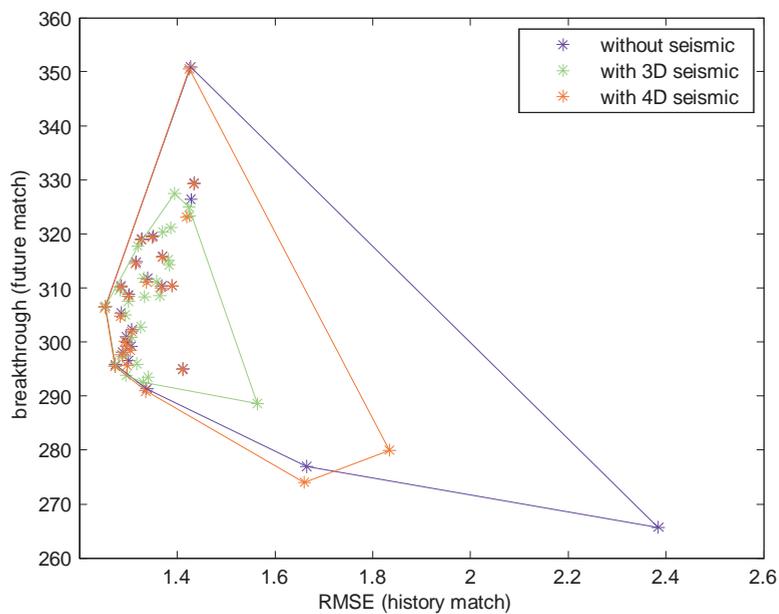


Figure 6.17: History-match future-match crossplot for 2STREAKS/5SPOT using 25 different priors and 3 different datasets.

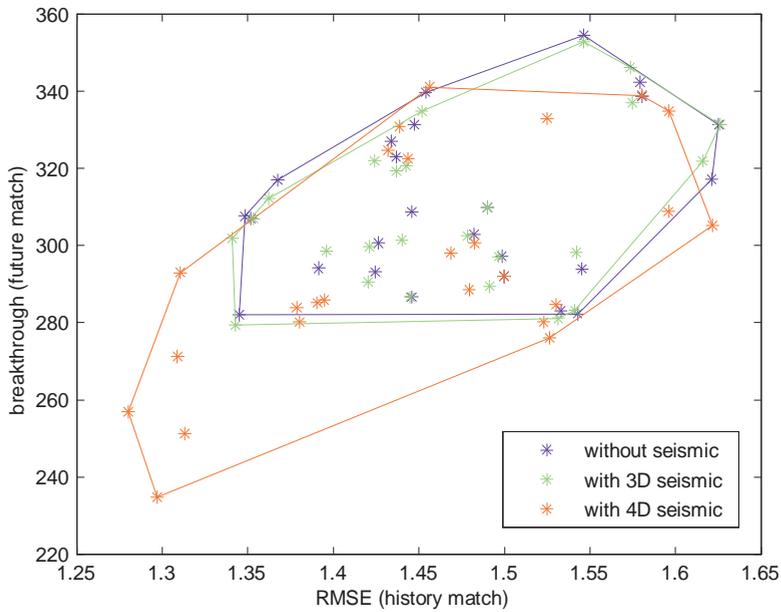


Figure 6.18: History-match future-match crossplot for 2STREAKS using 25 different priors and 3 different datasets.

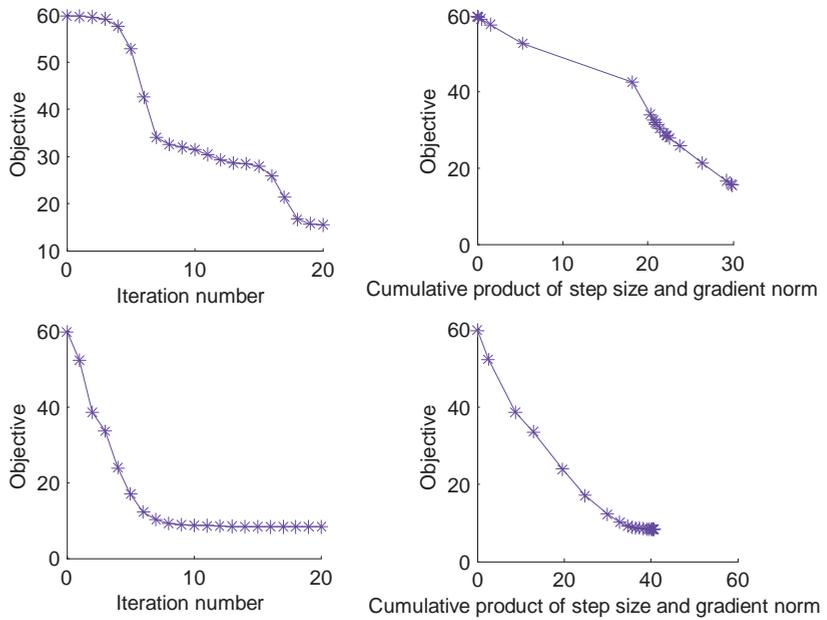


Figure 6.19: Top: VPERM for 5SPOT without seismic data. Bottom: parameter basis functions replaced using SVD.

In the RM, basis functions, or representers, are calculated in every iteration. In VPERM the basis functions are not updated at every iteration. This reduces computation time per iteration, but might also increase the total number of iterations.

Experiments were done with VPERM, attempting to assimilate production data, baseline and monitor P-wave impedance data and difference P-wave impedance data. From these experiments, the following can be concluded:

- VPERM produces results that are similar to the results of the RM, but in less computation time per iteration. Most figures in this chapter show VPERM results and only Fig. (6.15) shows RM results. However, many experiments were done with both methods. In fact VPERM was developed when RM provided disappointing results (slow convergence, too high history-match RMSE) with seismic data, but then performed similar to RM. Both methods produce good results when only pressure data obtained from the wells are available.
- If a data assimilation workflow gives reasonable results without seismic data, adding seismic data is not a trivial exercise. Although additional data does potentially give more information, it also increases the dimensionality of the parameter estimation problem. The seismic data introduced new local minima of the objective function in the neighborhood of the initial estimate. Perturbing the initial estimate in a random direction will improve the seismic impedance predictions in some gridblocks, but will also move predicted impedance values away from the true impedance in other grid blocks. In a mean-square-difference type of objective function these effects are cancelled out if there are as many improvements as deteriorations. In the direction of the objective function gradient there are only improvements, as long as the step size is small enough. In the neighborhood of the initial estimate the step size must be chosen so small, that improvement is only marginal and even below reasonable thresholds. In this research, using pressure data in all iterations and using seismic data from the 20th iteration gave good results. Similar modifications must be used for other data assimilation workflows. For example, [Vossepoel and Douma (2008)] investigated modifications for the Ensemble Kalman Filter to give good results with saturation data on the 2STREAKS model that was also used in this chapter.
- Using seismic data with VPERM never gives worse results than not using seismic data.
- In some cases, assimilating seismic data does not result in better history-matched models, whereas the models do give better future predictions; outliers in the history-future crossplot are removed. This confirms the results of [Walker and Lane (2007)]. In other cases, both the history-match and the "future-match" are improved by seismic data.
- Sometimes convergence can be improved, or a better local minimum of the objective function can be found, if the parameter basis functions are overwritten with the principal components of the prior parameter covariance.

Chapter 7

Summary and conclusions

7.1 Summary

In this thesis two data assimilation techniques, the Kalman filter and the Representer Method, were adopted from other areas of research and, with modifications, applied to petroleum reservoir applications. Also a new method was introduced, the VPERM method: Variational Parameter Estimation Regularized by Model-driven basis functions.

The specific kind of Kalman filter that was used in this thesis, was the Ensemble Square Root Kalman Filter (ESRKF). To improve the measurement update of the ESRKF, it was decomposed as a set of measurement updates with more uncertain data. The magnitude of the added uncertainty was chosen to increase the uncertainty of the data up to the level of the uncertainty in the parameters of interest. The number of times that every measurement was assimilated was chosen such that this multiple-updating-strategy only affected the estimated mean of the parameters but not the estimated covariance. In this way the added uncertainty is merely a numerical method and does not physically decrease the value of the data.

The Representer Method (RM) in this thesis was modified from previous implementations on four accounts; the number of representer may be (much) smaller than the number of measurements, the representer functions are defined differently so that no correction terms need to be calculated, it can interact with different gradient-based minimization algorithms and the "measurement representer" was introduced. The number of representer may be chosen by the user or can follow from a singular value decomposition and an energy preserving criterion. The RM has not been proven to converge for general non-linear problems, neither for previous implementations nor for the modified version. The number of iterations required depends very much on the specific problem. The convergence behavior of the classic RM and the modified RM should be further examined for larger, field-scale problems. The classic RM is derived without "measurement representer" because in the case of a linear measurement operator, the measurement representer are constructed by concatenating the state representer of different time steps as row blocks into one large matrix. Measurement representer were introduced to handle more complex measurement operators.

The VPERM method (Variational Parameter Estimation Regularized by Model-driven basis functions) was introduced in this thesis. The basis functions that perform the regularization are model-driven, like in the RM, and turn the high order weak constraint parameter estimation problem into a lower order strong constraint problem. Unlike RM, an analytical gradient can be obtained in VPERM by keeping the basis functions fixed. The "gradient" in RM is in fact the direction of a step in an iterative scheme with Picard iterations. The basis functions, or representer, are calculated in RM in every iteration. In VPERM, the basis

functions are not updated at every iteration. This reduces computation time per iteration, but might also increase the total number of iterations.

The data assimilation algorithms were tested using synthetic reservoirs. Data was generated with reservoir models and noise was added. The data assimilation algorithms started with the same reservoir models but with different parameters and are used to reconstruct the original parameters. This allows the use of two quality measures that are not available for field cases. The performance of the methods was measured by the root-mean-square-error (RMSE) between estimated and synthetic true parameters and the root-sum-square time shift between estimated water breakthrough curves and true water breakthrough curves in the well gridblocks.

First, experiments were done to improve the methods. For the Kalman filter, this led to the choice of ESRKF with duplicated measurements with added uncertainty. It showed the need to preprocess the sampled parameter ensemble before starting the actual data assimilation. For the RM some relevant lessons were learned about gradient based methods with or without Hessian estimating functionality and about line search methods. The RM was compared with the classic adjoint method. This method also underlies the RM, but uses user-defined regularization instead of the more advanced model-driven regularization of the RM.

Using the best settings and options that were found for the methods, the algorithms were compared to each other investigating their sensitivity with respect to uncertainty information that the user has to supply. This is a relevant topic since the user usually has to guess how uncertain the measurements are, whereas the guess itself introduces new uncertainty. The RM turned out to be less sensitive for the examples that were examined in this thesis.

Finally, the applicability of RM and VPERM was examined with respect to reservoir models. VPERM produces results that are similar to the results of the RM, but in less computation time per iteration and without using significantly more iterations. Both methods produce good results when only production data are assimilated. When seismic data was added, performance of VPERM never decreased. Depending on which performance measure was used, seismic data did increase performance. In some cases, assimilating seismic data does not result in better history-matched models (first performance measure), whereas the models do give better future predictions (second performance measure). Outliers in the history-future crossplot are removed. In other cases, both the history-match and the "future-match" are improved by seismic data.

7.2 Conclusions

It is recommended to use a good line search algorithm. Although (L)BFGS provides a better search direction than SD, this does not help the convergence if a fixed step size is used. For the strong-constraint case with the adjoint method, the quadratic line search performed so well for SD, that BFGS could not outperform it. BFGS did perform better with an expo-

mental line search. The quadratic line search does not work properly in combination with RM because the slope of the objective along the search direction cannot be calculated accurately. As a result, the quadratic line search underestimates the step size and the algorithm stops before even finding a local minimum of the objective function. (L)BFGS approximates a Hessian by monitoring the behavior of the gradient over successive iterations at very little cost. RM costs about as much computation time as explicitly calculating the Hessian. BFGS can also be applied on top of RM, but this does not contribute to faster convergence.

The RM achieves most of the decrease of the objective function in the first iteration. How much work is left for successive iterations depends on the magnitude of the model errors. If the model errors are large, then the non-linear relation between model parameters and state variables is overshadowed and the minimization problem becomes more linear. Hence, convergence is faster.

Model errors should not be modelled with a higher resolution than the measurements. Compared to the high resolution model errors that were used to create the synthetic measurements in this thesis, the variational methods introduced artificial correlation between the model errors on time intervals in between separate measurements.

The RM always performed better than the filter in terms of RMSE between estimated and true parameters. This makes sense for parameter estimation problems where the posterior probability of the parameters given the measurements is not symmetric or even multi-modal. The recursive filter introduces errors during every measurement update and the errors accumulate as the filter progresses. An iterative algorithm has the opportunity in every iteration to reduce the errors that were made in the previous iteration. Filter results can be improved by choosing the initial ensemble size sufficiently large and by making sure that the ensemble is de-skewed.

The RM was less sensitive to using a "wrong" prior than a filter for the examples considered in this thesis in terms of the two performance measures that were used.

VPERM turns the high order weak constraint parameter estimation problem into a lower order strong constraint problem. If the basis functions are kept fixed in the VPERM then, unlike the RM, an analytical gradient can be obtained. VPERM produces results that are similar to the results of the RM, but in less computation time per iteration. Sometimes convergence can be improved or a better local minimum of the objective function can be found, if the parameter basis functions of VPERM are overwritten with the principal components of the prior parameter covariance.

Adding seismic data to a history-matching application does not necessarily give better results. Although additional data do potentially give more information, they also increase the dimensionality of the parameter estimation problem. The seismic data introduced new local minima of the objective function in the neighborhood of the initial estimate. The seismic data become valuable if the data assimilation methods are adapted to handle the increased complexity that comes with the introduction of the extra data. Using seismic data with VPERM never produces worse results than not using seismic data. In some cases, assimilating seismic data does not result in better history-matched models, whereas the models do give better future predictions; outliers in the history-future crossplot are removed.

Nomenclature

\aleph	set of all grid block indices [-]
\aleph_ξ	set of indices of neighbors of grid block ξ [-]
η	index of neighbor of grid block ξ [#]
λ_t	total mobility [$m^{-2} s$]
λ_α	mobility of component/phase α [$m^{-2} s$]
$\mu_\alpha^{\xi\eta}$	viscosity of component/phase α on $\xi\eta$ -interface [$kg m^{-1} s^{-1}$]
μ_m	shear modulus of matrix [Pa]
ω^ξ	well factor in grid block ξ [m^3]
\otimes	element-wise multiplication operator
ϕ^ξ	porosity of grid block ξ [-]
λ_j	Lagrange multiplier/adjoint state vector for interval $t_{j-1} - t_j$
μ_θ	prior model parameters
θ	parameter vector
ε	model error vector
ε_n	model errors [$kg s^{-1}$]
\mathbf{b}	model parameters in low order space or representer coefficients
$\mathbf{J}_{\mathbf{f}_1^x}$	Jacobian of \mathbf{f}_1 w.r.t. \mathbf{x} [-]
\mathbf{K}	Kalman matrix
\mathbf{m}	vector with measurements
\mathbf{P}_θ	model parameter uncertainty covariance
\mathbf{P}_ε	model error covariance
\mathbf{P}_y	measurement uncertainty covariance
\mathbf{Q}	selection matrix
\mathbf{R}_{λ_i}	adjoint representers
\mathbf{R}_θ	parameter representers
$\mathbf{R}_{\varepsilon_i}$	model error representers
$\mathbf{R}_{\mathbf{x}_i}$	state representers
\mathbf{R}_y	measurement representers
\mathbf{x}	time-dependent state vector
\mathbf{x}_n	state vector at time t_n
$\mathbf{x}_{t_j}^{corr}$	correction term used in classic representer method
\mathbf{y}	vector with measurements
$\rho_\alpha^{\xi\eta}$	density of component/phase α on $\xi\eta$ -interface [$kg m^{-3}$]
ρ_α^ξ	density of component/phase α in grid block ξ [$kg m^{-3}$]
ρ_b	bulk density [$kg m^{-3}$]
ρ_s	density of sandstone [$kg m^{-3}$]
\hat{q}_α	volume (at STC) of injected/produced liquid per grid block volume [$s-1$]
$\tilde{\lambda}_\alpha$	mobility of component/phase α including permeability [s]
\tilde{q}_α	mass injection/production density of component/phase α [$kg m^{-3} s^{-1}$]
ξ	index of grid block ξ [#]

$A^{\xi\eta}$	area of $\xi\eta$ -interface [m^2]
b_α	formation volume factor of phase α [-]
$h^{\xi\eta}$	distance between centers of grid blocks ξ and η [m]
J	objective function
$k^{\xi\eta}$	permeability on $\xi\eta$ -interface in perpendicular direction [m^2]
$k_{r\alpha}^{\xi\eta}$	relative permeability of component/phase α on $\xi\eta$ -interface [-]
K_f	bulk modulus of fluid [Pa]
K_m	bulk modulus of matrix [Pa]
K_o	bulk modulus of oil [Pa]
K_s	bulk modulus of sandstone [Pa]
K_w	bulk modulus of water [Pa]
K_{ff}	fluid filled bulk modulus [Pa]
p_α^ξ	pressure of component/phase α in grid block ξ [$kg\ m^{-1}\ s^{-2}$]
p_{wb}	wellbore pressure [$kg\ m^{-1}\ s^{-2}$]
Q	wellbore flow rate at the surface [$m^3\ s^{-1}$]
$Q_\alpha^{\xi\eta}$	mass flux through $\xi\eta$ -interface [$m^3\ s^{-1}$]
q_α^ξ	injection/production of component/phase α in grid block ξ [$kg\ s^{-1}$]
$R_{\beta\alpha}$	solution β - α ratio [-]
S_α^ξ	saturation of component/phase α in grid block ξ [-]
t	time [s]
$t_\alpha^{\xi\eta}$	transmissibility of component/phase α on $\xi\eta$ -interface [$m\ s$]
V^ξ	volume of grid block ξ [m^3]
V_p	P-wave velocity [$m\ s^{-1}$]
V_s	S-wave velocity [$m\ s^{-1}$]
Z_p	P-wave impedance [$kg\ m^{-2}\ s^{-1}$]
Z_s	S-wave impedance [$kg\ m^{-2}\ s^{-1}$]

Bibliography

- Aavatsmark, I. and Eigestad, G. T. (2006). Numerical convergence of the MPFA O-method and U-method for general quadrilateral grids. *International Journal for Numerical Methods in Fluids*, **51**(9-10), 939–961.
- Aziz, K. and Settari, A. (1979). *Petroleum reservoir simulation*. Applied Science Publishers.
- Baird, J. and Dawson, C. (2005). The representer method for data assimilation in single-phase Darcy flow in porous media. *Computational Geosciences*, **9**(4), 247–271. DOI: 10.1007/s10596-005-9006-2.
- Baird, J. and Dawson, C. (2007). The representer method for two-phase flow in porous media. *Computational Geosciences*, **11**(3), 235 – 248. DOI: 10.1007/s10596-007-9048-8.
- Bayes, T. (1763). An essay towards solving a problem in the doctrine of chances. By the late Rev. Mr. Bayes, F. R. S. communicated by Mr. Price, in a letter to John Canton, A. M. F. R. S. *Philosophical Transactions*, (53), 370 – 418.
- Benner, P. and Mena, H. (2004). BDF methods for large-scale differential Riccati equations. In *Sixteenth International Symposium on: Mathematical Theory of Network and Systems*, Leuven, Belgium.
- Bennett, A. F. (2002). *Inverse modeling of the ocean and atmosphere*. Cambridge University Press.
- Bennett, A. F. (2006). *Lagrangian fluid dynamics*. Cambridge Monographs on Mechanics. Cambridge University Press, New York.
- Bennett, A. F. and McIntosh, P. C. (1982). Open ocean modeling as an inverse problem; tidal theory. *Journal of Physical Oceanography*, **12**(10), 1004 – 1018.
- Berkhout, A. J. (2004). *The data-driven Delphi value chain*, volume 2 of *Delphi, The Reservoir Characterization and Flow Simulation Project*. Delft University of Technology, Den Haag.
- Bilmes, J. A. (1998). A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models. Technical Report TR-97-021, U.C. Berkeley.

- Cao, Y., Li, S., Petzold, L., and Serban, R. (2002). Adjoint sensitivity analysis for differential-algebraic equations: the adjoint DAE system and its numerical solution. *SIAM Journal on Scientific Computing*, **24**(3), 1076–1089.
- Chavent, G., Dupuy, M., and Lemonnier, P. (1975). History matching by use of optimal theory. *SPE Journal*, **15**(1), 74 – 86. DOI: 10.2118/4627-PA.
- Cheng, H., Datta-Gupta, A., and Zhong, H. (2005). A comparison of travel time and amplitude inversion for production data integration into geologic models: sensitivity, non-linearity and practical implications. *SPE Journal*, **10**(1), 75 – 90.
- Cohn, S. E., Sivakumaran, N. S., and Todling, R. (1994). A fixed-lag Kalman smoother for retrospective data assimilation. *Monthly Weather Review*, **122**(12), 2838–2867.
- Datta-Gupta, A., Yoon, S. S., Nordaas, K., and Vasco, D. W. (2001). Streamlines, ray tracing and production tomography: generalization to compressible flow. *Petroleum Geoscience*, **7**, S75–S86.
- Dellaert, F. (2002). The expectation maximization algorithm. Technical Report GIT-GVU-02-20, Georgia Institute of Technology.
- Draper, N. R. and Smith, H. (1998). *Applied regression analysis*. Wiley-Interscience, 3 edition.
- EIA (2006). International energy outlook 2006. Technical Report DOE/EIA-0484(2006), Energy Information Administration, U.S. Department of Energy.
- EIA (2008). Annual energy outlook 2008. Technical Report DOE/EIA-0384(2008), Energy Information Administration, U.S. Department of Energy.
- Eknes, M. and Evensen, G. (1997). Parameter estimation solving a weak constraint formulation for an Ekman flow model. *Journal of Geophysical Research*, **102**(6C), 12479 – 12491.
- Evensen, G. (2003). The ensemble Kalman filter: theoretical formulation and practical implementation. *Ocean Dynamics*, **53**.
- Evensen, G. (2004). A note on the use of square root analysis schemes for the EnKF. Technical report, Hydro Research Centre.
- Evensen, G. (2007). *Data assimilation: the ensemble Kalman filter*. Springer, Berlin.

- Gao, G. and Reynolds, A. C. (2006). An improved implementation of the LBFGS algorithm for automatic history matching. *SPE Journal*, **11**(1), 5 – 17. DOI: 10.2118/90058-PA.
- Gavalas, G. R., Shah, P. C., and Seinfeld, J. H. (1976). Reservoir history matching by Bayesian estimation. *SPE Journal*, **16**(6), 337 – 350. DOI: 10.2118/5740-PA.
- Gelb, A. (1974). *Applied optimal estimation*. The MIT Press.
- Hairer, E. and Wanner, G. (1996). *Solving ordinary differential equations II: stiff and differential-algebraic problems*. Springer Verlag, Berlin, Germany, 2 edition.
- Heemink, A. W., Verlaan, M., and Segers, A. J. (2001). Variance reduced ensemble Kalman filtering. *Monthly Weather Review*, **129**(7), 1718 – 1728.
- Heijn, T., Markovinovic, R., and Jansen, J. D. (2004). Generation of low-order reservoir models using system-theoretical concepts. *SPE Journal*, **9**(2), 202–218.
- Jafarpour, B. and McLaughlin, D. B. (2007). Efficient permeability parameterization with the discrete cosine transform. Technical Report SPE 106453.
- Jansen, J. D., Brouwer, D. R., Naevdal, G., and van Kruijsdijk, C. P. J. W. (2005). Closed-loop reservoir management. *First Break*, **23**, 43–48.
- Janssen, G. M. C. M., Valstar, J. R., and van der Zee, S. E. A. T. M. (2006). Inverse modeling of multimodal conductivity distributions. *Water Resources Research*, **42**(3), Art. No. W03410.
- Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, pp 35–45.
- Leverett, M. C. (1941). Capillary behaviour in porous solids. *Trans, American Institution of Mining and Metallurgical Engineers*, **142**, 152–169.
- Lewis, J. M., Lakshmivarahan, S., and Dhall, S. K. (2006). *Dynamic data assimilation: a least squares approach*. Cambridge University Press, Cambridge.
- Li, B., Chen, Z., and Huan, G. (2004). Comparison of solution schemes for black oil reservoir simulations with unstructured grids. *Computer methods in applied mechanics and engineering*, **193**(3-5), 319–355.

- Lien, M., Brouwer, D. R., Manseth, T., and Jansen, J. D. (2006). Multiscale regularization of flooding optimization for smart field management. In *SPE Intelligent Energy Conference and Exhibition*, Amsterdam.
- Lioen, W. M. and de Swart, J. J. B. (1998). Test set for initial value problem solvers. Technical Report MAS-R9832, Centrum voor Wiskunde en Informatica.
- Mavko, G., Mukerji, T., and Dvorkin, J. (1998). *The rock physics handbook; tools for seismic analysis in porous media*. Cambridge University Press, Cambridge.
- Morrow, N. (1970). Physics and thermodynamics of capillary action in porous media. *Industrial and Engineering Chemistry*, **62**(6), 32–56.
- Naevdal, G., Johnsen, L. M., Aanonsen, S. I., and Vefring, E. H. (2005). Reservoir monitoring and continuous model updating using ensemble Kalman filter. *SPE Journal*, **10**(1), 66 – 74.
- Neuman, S. P. (1977). Theoretical derivation of Darcy's law. *Acta Mechanica*, **25**(3-4), 153–170.
- Nielsen, L. K. (2006). *Reservoir characterization by a binary level set method and adaptive multiscale estimation*. Ph.D. thesis, University of Bergen.
- Nocedal, J. and Wright, S. J. (1999). *Numerical optimization*. Springer Verlag, New York.
- Oliver, D. S., Reynolds, A. C., and Liu, N. (2008). *Inverse theory for petroleum reservoir characterization and history matching*. Cambridge University Press, Cambridge.
- Peaceman, D. W. (1977). *Fundamentals of numerical reservoir simulation*. Elsevier, Amsterdam.
- Petzold, L. R. (1983). A description of DASSL: a differential/algebraic system solver. *Scientific Computing*, pp 65–68.
- Plug, W. J., Bruining, J., Slob, E. C., and Gorriti, A. G. (2006). Numerical validation of various mixing rules used for up-scaled geo-physical properties. In *Computational Methods in Water Resources*, Copenhagen.
- Przybylski, J. (2007). Classic and modified representer method (personal communication).
- Przybylski, J. K. (2004). *Application of the representer method to identify uncertain parameters in large scale numerical reservoir model*. Msc thesis, Delft University of Technology.

Przybysz, J. K., Hanea, R. G., Jansen, J. D., and Heemink, A. W. (2007). Application of the representer method for parameter estimation in numerical reservoir models. *Computational Geosciences*, **11**(1), 73 – 85. DOI: 10.1007/s10596-006-9035-5.

Rodrigues, J. R. P. (2006). Calculating derivatives for automatic history matching. *Computational Geosciences*, **10**(1), 119–136. DOI: 10.1007/s10596-005-9013-3.

Rommelse, J. R., Lin, H. X., and Chan, T. F. (2004). Efficient active contour and K-means algorithms in image segmentation. *Scientific Programming*, **12**(2), 101–120.

Rommelse, J. R., Kleptsova, O., Jansen, J. D., and Heemink, A. W. (2006). Data assimilation in reservoir management using the representer method and the ensemble Kalman filter. In *Proceedings of the 10th European Conference on the Mathematics of Oil Recovery*, Amsterdam, Netherlands.

Rommelse, J. R., Jansen, J. D., and Heemink, A. W. (2007). An efficient gradient-based parameter estimation algorithm using representer expansions. Submitted July 2008 for publication in SPEJ. Preprint available as. Technical Report 07-05, Delft University of Technology.

Rommelse, J. R., Jansen, J. D., Heemink, A. W., Wilschut, F., and Kleptsova, O. (2008a). Comparison of the ensemble Kalman filter and a modified representer method for sensitivity to prior data. Submitted July 2008 for publication in COMG. Preprint available as. Technical Report 08-16, Delft University of Technology.

Rommelse, J. R., Jansen, J. D., and Heemink, A. W. (2008b). Variational estimation of permeability and model errors from 3D and 4D seismic data using model-driven regularization. Submitted November 2008 for publication in COMG. Preprint available as. Technical Report 08-18, Delft University of Technology.

Ruijian, L., Reynolds, A. C., and Oliver, D. S. (2003). History matching of three-phase flow production data. *SPE Journal*, **8**(40), 328 – 340. DOI: 10.2118/87336-PA.

Schwaighofer, A. and Tresp, V. (2003). Transductive and inductive methods for approximate Gaussian process regression. *Advances in Neural Information Processing Systems*, **15**, 953 – 960.

Skjervheim, J. A., Evensen, G., Aanonsen, S. I., Ruud, B. O., and Johansen, T. A. (2005). Incorporating 4D seismic data in reservoir simulation models using ensemble Kalman filter. In *SPE Annual Technical Conference and Exhibition*, Dallas.

- Smaoui, N. and Garrouch, A. A. (1997). A new approach combining Karhunen-Loeve decompositions and artificial neural networks for estimating tight gas sand permeability. *Journal of Petroleum Science and Engineering*, **18**(1-2), 101 – 112.
- Strebelle, S. (2002). Conditional simulation of complex geological structures using multiple-point statistics. *Mathematical Geology*, **34**(1), 1–21.
- Strikwerda, J. C. (2004). *Finite difference schemes and partial differential equations*. SIAM.
- Sweppe, F. C. (1973). *Uncertain dynamic systems*. Prentice Hall.
- Tarantola, A. (2005). *Inverse problem theory and methods for model parameter estimation*. SIAM, Philadelphia.
- te Stroet, C. B. M. (1995). *Calibration of stochastic groundwater flow models*. Phd thesis, Technische Universiteit Delft.
- Thiele, M. R. (2001). Streamline simulation. In *6th International Forum on Reservoir Simulation*, Schloss Fuschl, Austria.
- Ulbrich, S. (2002). Limited-Memory BFGS-Verfahren mit Powell-Wolfe Schrittweitenregel. <http://www.mathematik.tu-darmstadt.de:8080/ags/ag10/Team/ulbrich/opt3/codes.html>.
- Valstar, J. R. (2001). *Inverse modelling of groundwater flow and transport*. Ph.D. thesis, Delft University of Technology.
- Valstar, J. R., McLaughlin, D. B., te Stroet, C. B. M., and van Geer, F. C. (2004). A representer-based inverse method for groundwater flow and transport applications. *Water Resources Research*, **40** W05116. DOI:10.1029/2003WR002922.
- van der Poel, R. and Jansen, J. D. (2004). Probabilistic analysis of the value of a smart well for sequential production of a stacked reservoir. *Journal of Petroleum Science and Engineering*, **44**, 155 – 172.
- Verlaan, M. and Heemink, A. W. (1997). Tidal flow forecasting using reduced rank square root filters. *Stochastic Hydrology and Hydraulics*, **11**(5), 349–368.
- Versteeg, H. and Malalasekera, W. (1996). *An introduction to computational fluid dynamics: the finite volume method approach*. Prentice Hall.

- Vossepoel, F. and Douma, S. G. (2008). Assimilating saturation data with the ensemble Kalman filter (personal communication).
- Walker, G. J. and Lane, H. S. (2007). Assessing the accuracy of history-matching predictions and the impact of time-lapse seismic data; a case study for the Harding reservoir. In *SPE Reservoir simulation symposium*, Houston.
- Welch, G. and Bishop, G. (1995). An introduction to the Kalman filter. Technical report, University of North Carolina.
- Wheeler, M. F. (1988). *Numerical simulation in oil recovery*. Springer, New York.
- Whitaker, J. S. and Hamill, T. (2002). Ensemble data assimilation without perturbed observations. *Monthly Weather Review*, **130**.
- Woodbury, M. A. (1950). Inverting modified matrices. Technical report, Statistical research group, Princeton University.
- Zhang, F., Skjervheim, J. A., Reynolds, A. C., and Oliver, D. S. (2005). Automatic history matching in a Bayesian framework, example applications. *SPEREE*, **8**(3), 214 – 223. DOI: 10.2118/84461-PA.
- Zienkiewicz, O. C., Taylor, R. L., and Zhu, J. Z. (2005). *The finite element method: its basis and fundamentals*. Butterworth-Heinemann, 6 edition.

Appendix A

Rock-physics derivatives

Derivatives with respect to pressure:

$$\frac{\partial \rho_b}{\partial p} = \phi \left(S_w \frac{\partial \rho_w}{\partial p} + S_o \frac{\partial \rho_o}{\partial p} \right), \quad (\text{A.1})$$

$$\frac{\partial Z_p}{\partial p} = \frac{1}{2Z_p} \left(K_{ff} + \frac{4}{3}\mu_m \right) \frac{\partial \rho_b}{\partial p}. \quad (\text{A.2})$$

Derivatives with respect to saturation:

$$\frac{\partial K_f}{\partial S_w} = -\frac{K_f^2}{K_w}, \quad \frac{\partial K_f}{\partial S_o} = -\frac{K_f^2}{K_o}, \quad (\text{A.3})$$

$$\begin{aligned} \frac{\partial K_f}{\partial S} &= \frac{\partial K_f}{\partial S_w} \frac{\partial S_w}{\partial S} + \frac{\partial K_f}{\partial S_o} \frac{\partial S_o}{\partial S} = \\ &= -\frac{K_f^2}{K_w} \frac{\partial S}{\partial S} - \frac{K_f^2}{K_o} \frac{\partial (1-S)}{\partial S} = \\ &= K_f^2 \left(\frac{1}{K_o} - \frac{1}{K_w} \right), \end{aligned} \quad (\text{A.4})$$

$$\begin{aligned} \frac{\partial \alpha}{\partial S} &= -\frac{\phi}{K_f^2} \frac{\partial K_f}{\partial S} = \\ &= \phi \left(\frac{1}{K_w} - \frac{1}{K_o} \right), \end{aligned} \quad (\text{A.5})$$

$$\frac{\partial K_{ff}}{\partial \alpha} = K_s \frac{K_m - K_{ff}}{K_s \alpha + \beta}, \quad (\text{A.6})$$

$$\begin{aligned} \frac{\partial K_{ff}}{\partial S} &= \frac{\partial K_{ff}}{\partial \alpha} \frac{\partial \alpha}{\partial S} = \\ &= K_s \frac{K_m - K_{ff}}{K_s \alpha + \beta} \phi \left(\frac{1}{K_w} - \frac{1}{K_o} \right), \end{aligned} \quad (\text{A.7})$$

$$\frac{\partial \rho_b}{\partial S} = \phi (\rho_w - \rho_o), \quad (\text{A.8})$$

$$\frac{\partial Z_p}{\partial S} = \frac{1}{2Z_p} \left(\left(K_{ff} + \frac{4}{3}\mu_m \right) \frac{\partial \rho_b}{\partial S} + \frac{\partial K_{ff}}{\partial S} \rho_b \right). \quad (\text{A.9})$$

Summary

The research presented in this thesis aims at improving computer models that allow simulations of water, oil and gas flows in subsurface petroleum reservoirs. This is done by integrating, or assimilating, measurements into physics-based models. In recent years petroleum technology has developed rapidly. Nowadays wells can be drilled to a depth of up to 10 km, not just vertically, but also at an angle, horizontally or with branches. Moreover, downhole valves can be installed which can be opened or closed from the surface and advanced sensors can be placed in the subsurface. This technology has the potential to drain petroleum reservoirs much more efficiently. In order to do so, the technology needs to be used sensibly, which requires adequate knowledge of subsurface physical processes. Large amounts of measurements can contribute to this, but conventional methods are often ad hoc and not suited to handle the large amounts of data that are available nowadays. Good “data assimilation” methods are very important to ensure that the growing demand for energy in the near future can be met.

The objective of this thesis is to apply data assimilation techniques, invented and developed in other areas of research, to petroleum reservoir engineering, to modify them to be better suited for their new application, and to investigate how they can help to integrate both production data and seismic data to support decision-making in petroleum reservoir management. Two data assimilation techniques, the Kalman filter and the Representer Method, were adopted from other areas of research and, with modifications, applied to petroleum reservoir applications. Also the VPERM method, Variational Parameter Estimation Regularized by Model-driven basis functions, was introduced. The measurement update step of the Kalman filter was split into a set of update steps that are less sensitive to errors that are introduced because the reservoir model or the sensor model are nonlinear. Also the initialization of the filter with samples from the uncertainty in the reservoir permeability was investigated. The Representer Method (RM) in this thesis was modified from previous implementations on four accounts; the number of representers may be smaller than the number of measurements, the representer functions are defined differently so no correction terms need to be calculated, it can interact with different gradient-based minimization algorithms and the "measurement representer" was introduced. The VPERM method was inspired by RM and uses model-driven regularization with basis functions that are very similar to RM. VPERM turns the high order weak constraint parameter estimation problem into a lower order strong constraint problem. Unlike RM, an analytical gradient can be obtained in VPERM by keeping the basis functions fixed.

Synthetic reservoirs were used to test the data assimilation algorithms. Two quality measures were specified to quantify the performance of the methods, the root-mean-square-error (RMSE) between estimated and synthetic true parameters and the root-sum-square time shift between estimated water breakthrough curves and true water breakthrough curves in the well gridblocks. These quality measures are only applicable to synthetic cases and not to field cases.

First, experiments were done to improve the methods. For the Kalman filter this led to a specific variation of Kalman filter, the ESRKF, with an improved measurement update step and a preprocessing method for the initialized ensemble that was used before starting the actual data assimilation. For the RM some relevant lessons were learned about gradient based methods with or without Hessian estimating functionality and about line search methods. The RM

was compared with the classic adjoint method. This method also underlies the RM, but applies user-defined regularization instead of the more advanced model-driven regularization of the RM.

Using the best settings and options that were found for the methods, the algorithms were compared to each other investigating their sensitivity with respect to uncertainty information that the user has to supply. This is a relevant topic since the user usually has to guess how uncertain the measurements are, whereas the guess itself introduces new uncertainty. The RM turned out to be less sensitive for the examples that were examined in this thesis.

Finally, the applicability of RM and VPERM was examined with respect to seismic data and reservoir models. VPERM produces results that are similar to the results of the RM, but in less computation time per iteration and without using significantly more iterations. Both methods produce good results when only production data are assimilated. When seismic data were added, performance of VPERM never decreased. Depending on which performance measure was used, seismic data did increase performance. In some cases, assimilating seismic data does not result in better history-matched models, whereas the models do give better future predictions. Outliers in the history-future crossplot are removed. In other cases, both the history-match and the "future-match" are improved by seismic data. Seismic data do potentially provide additional information, but also increase the dimensionality of the parameter estimation problem. Data assimilation methods must be and have been adapted to handle this increased complexity to give value to the seismic data.

Samenvatting

Het onderzoek, waarop dit proefschrift is gebaseerd, richt zich op het verbeteren van computermodellen waarmee water-, olie- en gasstromingen in ondergrondse reservoirs kunnen worden doorgerekend. Dit gebeurt door het integreren, ofwel assimileren, van meetgegevens in modellen die gebaseerd zijn op fysische principes. De afgelopen jaren is de ontwikkeling van oliewinningstechnologie snel gegaan. Putten kunnen tegenwoordig tot wel 10 km diep geboord worden, en niet alleen vertikaal, maar ook om een hoek, horizontaal of met aftakkingen. Bovendien kunnen er tegenwoordig ondergrondse kranen geïnstalleerd worden die van boven de grond geopend of gesloten kunnen worden en kunnen zeer geavanceerde sensoren onder de grond geplaatst worden. Deze technologie heeft de potentie om oliereservoirs veel efficiënter leeg te halen. Hiertoe moet de technologie wel verstandig benut worden, waarvoor een goed begrip van de ondergrondse fysische processen nodig is. De vele meetgegeven kunnen hieraan bijdragen, maar conventionele methodes zijn vaak erg ad hoc en niet berekend op de grote hoeveelheden data die tegenwoordig beschikbaar zijn. Goede “data assimilatie”-methoden zijn dus zeer belangrijk om in de nabije toekomst aan de groeiende vraag naar energie te kunnen voldoen.

Het doel van dit proefschrift is het toepassen van data assimilatietechnieken, uitgevonden en ontwikkeld in andere takken van onderzoek, in petroleum reservoir engineering, ze te modificeren om ze beter geschikt te maken voor hun nieuwe toepassing, en uit te zoeken hoe ze van dienst kunnen zijn om zowel productie data als seismische data te benutten om het beslissingsproces in petroleum reservoir management te ondersteunen. Twee data assimilatie technieken, de Kalman filter en de Representer Methode, zijn gekopieerd uit andere wetenschappen en gemodificeerd voor toepassing in petroleum reservoir engineering. Ook is de VPERM methode, Variational Parameter Estimation Regularized by Model-driven basis functions, geïntroduceerd. De measurement-update stap van de Kalman filter is opgesplitst in een set van kleinere update stappen die minder gevoelig zijn voor fouten die veroorzaakt worden doordat het reservoir model of het sensor model niet-lineair zijn. Ook is de initialisatie van de filter met realisaties uit de onzekere ondergrondse gesteente doorstroombaarheid onderzocht. De Representer Methode (RM) in dit proefschrift is op vier punten afgeweken van eerdere implementaties; het aantal representers kan kleiner zijn dan het aantal metingen, de representer functies zijn op een andere manier gedefinieerd zodat er geen correctie termen berekend hoeven te worden, het kan ingebed worden in verschillende gradient-gebaseerde minimalisatie algoritmes en de "meet representer" is geïntroduceerd. De VPERM methode was geïnspireerd door de RM en gebruikt model-gedreven regularisatie met basis functies die erg lijken op die van de RM. VPERM verandert het hoogdimensionale weak-constraint parameterschattingsprobleem in een lager-dimensionaal strong-constraint probleem. In tegenstelling tot RM is er bij VPERM een analytische gradient beschikbaar door de basis functies constant te houden.

Synthetische reservoirs zijn gebruikt om de data assimilatie algoritmes te testen. Twee kwaliteitsmaten zijn gebruikt om de performance van de methodes te kwantificeren. De eerste

is de root-mean-square-error (RMSE) tussen de geschatte parameters en de werkelijke synthetische parameters. De tweede is de root-sum-square van de time-shift tussen de geschatte water doorbraak grafieken en de werkelijke water doorbraak grafieken van alle gridblocks met een olieput. Deze kwaliteitsmaten zijn alleen toepasbaar voor synthetische experimenten en niet voor field cases.

Ten eerste zijn er experimenten uitgevoerd om de methodes te verbeteren. Voor de Kalman filter heeft dit geleid tot een specifieke variatie van de Kalman filter, de ESRKF, met een verbeterde measurement-update stap en een preprocessing methode voor het initialiseren van het ensemble voordat de eigenlijke data assimilatie plaatsvindt. Voor de RM konden enkele relevante lessen geleerd worden over gradient-gebaseerde methodes met of zonder functionaliteit voor het schatten van de Hessian en over line-search methodes. De RM is vergeleken met de klassieke adjoint methode. Deze methode is tevens de basis voor de RM, maar past user-defined regularisatie toe in plaats van de meer geavanceerde model-driven regularisatie van de RM.

Met de beste settings en opties die voor de methodes gevonden werden, zijn de methodes met elkaar vergeleken en is uitgezocht hoe gevoelig de methodes zijn met betrekking tot informatie die de gebruiker aan de methodes dient te verschaffen. Dit is een relevant onderwerp, aangezien de gebruiker doorgaans moet schatten hoe onzeker de metingen zijn, terwijl de schatting zelf weer nieuwe onzekerheid introduceert. De RM bleek minder gevoelig in het geval van de voorbeelden die in dit proefschrift zijn beschouwd.

Tenslotte is de toepasbaarheid van de RM en VPERM voor seismische data en reservoir modellen onderzocht. VPERM produceert resultaten die vergelijkbaar zijn met de resultaten van de RM, maar in minder rekentijd per iteratie en zonder significant meer iteraties nodig te hebben. Beide methodes presteren goed als er alleen maar productie data gebruikt worden. Zodra seismische data wordt toegevoegd, wordt de performance van VPERM daar nooit slechter van. Afhankelijk van welke performancemaat gebruikt wordt, wordt de performance juist verhoogd door het toevoegen van seismische data. In sommige gevallen leidt het assimileren van seismische data niet noodzakelijk tot modellen die beter bij de geschiedenis van het reservoir aansluiten, terwijl de modellen wel beter geschikt worden om voorspellingen mee te maken. Uitschieters in de geschiedenis-toekomst crossplot worden verwijderd. In andere gevallen worden zowel de "history-match" als de "future-match" verbeterd door seismische data. Seismische data bevatten in potentie extra informatie naast productie data, maar ze verhogen ook de dimensionaliteit van het parameterschattingprobleem. Data assimilatie methodes moeten aangepast worden, en zijn in dit proefschrift aangepast, om te kunnen omgaan met de verhoogde complexiteit en zo de waarde van seismische data volledig te benutten.

Acknowledgments

Several people have actively contributed to this thesis and many others have supported me. I am very grateful to all of them. Many thanks to

- Tillie and Ton for supporting me and the rest of their children during the PhD and all other phases of our lives.
- Tillie, Ton, Jeannie and Leo for supporting my 9-month trip to the U.S. when I was an MSc student. Before my trip I hardly spoke English. I would not have dared to do PhD research and write a thesis in English without that great learning experience.
- Cor and Jan-Dirk for hiring me as a PhD student.
- Arnold for recommending me to Cor and Jan-Dirk.
- Roald, Willem-Jan, Cor, Jan-Dirk, Hans and Pacelli for teaching me the basics of the petroleum industry. When I came as an MSc from the Math department I did not have a clue about petroleum reservoirs or porous media. In my world, oil was just waiting in a big hole in the subsurface, waiting for us to drain empty by drilling and sticking straws in the ground.
- Arnold and Jan-Dirk for supervising my PhD research. In the initial stage I had the freedom to learn and set out my own paths. In the later stage they made sure that I kept my focus on a limited number of paths, and with their valuable comments they ensured that most paths led to nice results.
- Roald, Willem-Jan, Renato, Maarten, Jorn, Gijs, Gerben, Mario, Ali, Koen, Hein, Patrick, Saikat, Bouko, Nicolai, Ainoa, Cas, Rouhi, Gosia, Christiaan, Twan, Anke-Jannie and Mattia for a friendly atmosphere between PhD students.
- Renato for many intellectually stimulating discussions about research, life, serious issues and nonsense.
- Olga, Frank, Justyna, Sippe, Femke and Remus for valuable comments and cooperation.
- Dennis for welcoming me in his research group and Behnam for welcoming me in his home during one of my trips to MIT.
- Lydia, Margot, Marlijn, Hannie and Ralph for support at everyday logistical issues.
- Dietz-lab people for friendly coffee breaks.

- Dries, Cor and Jan-Dirk for criticizing my presentations before I had to appear before the Delphi sponsors.
- my Delphi colleagues and sponsors for the pleasant trips to Houston.
- Jan-Dirk, Arnold, Cor, Paul, Dennis, Al and Chris for being part of my promotion committee.
- Cor, Dennis and Al for flying over the ocean to be part of my promotion committee.
- Gijs and Nanda for standing by my side during my thesis defense. I am proud to have two such promising academics as my paranymphs.
- Han, for giving the outside of the thesis the professional appearance that it has. I am happy that I gave up my own amateuristic attempts.
- my basketball team mates and skydiving buddies. Without some nice hobbies and people to share them with, I would not have been able to release the stress that comes with doing PhD research.
- Ton, Tillie, Gijs, Barbara, Julia, Josephine, Nanda, Bart and Daan for all the moral support.
- Ton and Gijs for kicking my butt once or twice in the final phase of writing the thesis. Finishing a thesis alongside a full-time job at Shell has turned out to be more intense than I had anticipated.
- Wilma for your support in last stage and for your love. I love you too.
- everybody that I forgot to mention. Shame on me.

About the author

Joris Rommelse was born on March 2nd 1979 in Haarlem, The Netherlands. After skipping the last year of elementary school, he received his high school diploma (vwo) from the Solyvius College in Hoofddorp in 1996. He studied Applied Mathematics at Delft University of Technology, where he obtained a Masters degree in 2002. The research for his MSc thesis was carried out during a 9-month visit to the University of California, Los Angeles. At the TUDelft he was awarded best MSc graduate of the mathematics department, class of 2001-2002, for his study results and graduation research. From October 2002 until March 2007 he did the PhD research that led to this thesis, interrupted by a 3 month internship at Shell International Exploration & Production in Rijswijk. April 1st 2007 he started working as a scientific software consultant for Alten Nederland (currently Alten PTS) and as a contractor in Shell's reservoir simulation team.