FEATURE-BASED ESTIMATION
for applications in geosciences

FEATURE-BASED ESTIMATION
for applications in geosciences

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. ir. K.C.A.M. Luyben,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen
op maandag 10 december 2012 om 10.00 uur
door

Wiktoria ŁAWNICZAK

Master of Science in Applied Mathematics,
Delft University of Technology, The Netherlands

geboren te Zielona Góra, Poland

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. A.W. Heemink	Technische Universiteit Delft, promotor
Prof.dr.ir. J.D. Jansen	Technische Universiteit Delft, promotor
Prof. dr. W. R. Rossen	Technische Universiteit Delft
Prof.dr.ir. J. Biemond	Technische Universiteit Delft
Prof. dr. ir. F.C. van Geer	Utrecht University
Dr. P. J. van den Hoek	Shell International E&P
Dr. R.G. Hanea	Technische Universiteit Delft

Copyright © 2012 by Wiktoria Ławniczak
All rights reserved. No part of the material protected by this copyright notice may be reproduced or utilized 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.

Cover design: Proefschriftmaken.nl || Uitgeverij BOXPress
Printed & Lay Out by: Proefschriftmaken.nl || Uitgeverij BOXPress
Published by: Uitgeverij BOXPress, Oisterwijk

science of nonexistent things
Hugo Steinhaus about *mathematics*

Contents

1 Introduction **3**

2 Numerical modeling of reservoirs **9**
- 2.1 Modeling of reservoirs . . . 9
- 2.2 *simsim* . . . 12
- 2.3 State-space representation of the model . . . 16

3 Data assimilation and parameter estimation **17**
- 3.1 Problem setup . . . 17
- 3.2 The classical Kalman filter . . . 18
- 3.3 The Ensemble Kalman Filter - EnKF . . . 21
  - 3.3.1 The Ensemble Square-Root Filter - EnSRF . . . 22
- 3.4 Implementation issues of the Ensemble Kalman Filter . . . 24
- 3.5 Parameter estimation problem . . . 28

4 Ensemble Multiscale Filter - EnMSF **31**
- 4.1 Multiscale ensemble filtering for measurement update . . . 31
  - 4.1.1 Introduction . . . 31
  - 4.1.2 EnKF and EnMSF - theoretical background . . . 32
  - 4.1.3 Application . . . 40
  - 4.1.4 Conclusions . . . 43
- 4.2 Towards the use of the ensemble multiscale filter for history matching . . . 44
  - 4.2.1 Introduction . . . 44
  - 4.2.2 Pixel numbering scheme . . . 44
  - 4.2.3 Interpolation problem . . . 47
  - 4.2.4 Coarsening parameters in the EnMSF . . . 47
  - 4.2.5 The design of the tree and its impact . . . 49
  - 4.2.6 Covariance matrix approximation . . . 53
  - 4.2.7 History matching using EnMSF . . . 54
- 4.3 Conclusions . . . 58

5 Feature-based methods **61**
- 5.1 Grid distortion for a 2D reservoir model . . . 61
  - 5.1.1 Introduction . . . 61
  - 5.1.2 Basic notions . . . 63
  - 5.1.3 Field Alignment . . . 65
  - 5.1.4 Ensemble sequential data assimilation and direct search . . . 67

5.1.5 Grid distortion 68
5.1.6 Twin Experiment 76
5.1.7 Discussion and Conclusions 91
5.2 Grid distortion for a 3D groundwater flow model 91
5.2.1 The 3D grid distortion method 91
5.2.2 3D groundwater flow model 92
5.2.3 Experiment setup 93
5.2.4 Discussion and results 98
5.3 Grid distortion for a 3D reservoir model 101
5.3.1 Discussion and conclusions 108

6 Conclusions and recommendations **109**

Summary **119**

Samenvatting **121**

Curriculum Vitae **125**

Acknowledgments

This research was carried out within the context of the ISAPP Knowledge Centre. ISAPP (Integrated Systems Approach to Petroleum Production) is a joint project of the Netherlands Organization for Applied Scientific Research TNO, Shell Global Solutions International, and Delft University of Technology.

Banach [41] is dead but there are other great mathematicians I met on my way through schools. I would like to thank them for teaching me rigorous thinking and I hope that by writing this thesis I proved that their time was not completely wasted. Especially, I want to name: Alicja Gandecka, Jan Szajkowski, Dorota Krassowska, Mieczyslaw Trad, Anna Kubicka and Jolanta Misiewicz. All of them really scared me at the time but now I am grateful for the solid math basis I obtained.

I consider myself lucky to have four (4!) supervisors during the construction of this doctorate thesis. I would like to thank Dr. Remus Hanea for giving me the opportunity to start this PhD program, and his friendliness; Prof. Arnold Heemink for good sense of humor, great ideas and contagious peacefulness; Prof. Jan Dirk Jansen for being fair, supportive and consistent; and Prof. Dennis McLaughlin for all the time he took to work with me, I greatly appreciate the trips to Boston for multiple reasons. I am very grateful to Prof. Heemink for translating the propositions into Dutch and for all the help and understanding during the last part of getting this degree.

The toughest job was to make all the administration, papers, computers work. It would have never been done if it was not for: Dorothee Engering, Evelyn Sharabi, Carl Schneider, Jim Long and Anke Dahlmann. Thank you for being so kind.

For me the secrets of good teaching [44] are still to be discovered but I am glad I had good course coordinators: Roelof Koekoek, Harry Kneppers and Henric Corstens.

I realize that there are many great books to learn from but I think there are even more great people to learn from. The literature meetings organized by Martin Verlaan and discussions within our little group are an irreplaceable source of knowledge. Thank you all!

Always the office mates make a day at the university fun, at least mine did. All sorts of problems could be solved or created with Svetlana, Mariya and Cristi; I greatly enjoyed my time with Elena, Luis, Gift and Raf. Thanks guys! I also want to thank Angéline for a very pleasant cooperation, and colleagues and friends at DIAM.

I have made great friends during my studies that substituted for my family, thank you: my lovely roomies who have been making every-day life interesting and entertaining, Karolina for all the conversations, trips, her great sense of humor and emergency translation of the summary, and Angela for being full of positive inspiring energy; my Mexican family who always makes me feel welcome: Sandra, Camila y OswalDOS; my American family who provided me with a real home across the ocean: Arlene, Mike, Suzanne and

MichaelBaby; the wonderfully down-to-earth friend for her easiness of expression: Asia (with Olek and Zosia); the always-there, like brother to me: Marcin (with Monika and Natalia); the chilled and warm Paisas: Alejandra, Manuela y Sebastian. At the same time, away from Poland, I could also keep my dear quarter-of-a-century friends. I want to thank: Gosia (with Tomek and Klaudia) for her confidence and view on life, and Ola (with Daniel and Łucja) for always being in a good mood. And Adrian for being so fun and easygoing. An additional thank you goes to: Eka, Israel, Christiaan, Manuelito, Maryam, Diego and Nora, Gosia, Eduardo, Isabel, Tatiana and Dirk, Ivan, Asia, Anca, Joaquin and Marisa, Negar.

I would also like to thank Richard and the small Commercial team, for the great new beginnings.

Most of all I would like to thank my family in Poland. My grandparents who always miss me a lot; my parents that are always worried but keep it all cool; and my aunt who likes calling me.

Marco&F., I am glad we met just at the right time. Thank you for the good times together, you are irreplaceable.

Thank you. Dziekuje. Gracias. Bedankt.

W

Chapter 1

Introduction

The energy requirements around the world are expected to grow. Even with blooming and expanding green energy R&D, the developing society will still need petroleum for another tens of years. A concern might be that the oil and gas reserves that are easily accessible have, to a large extent, been produced already. The sources that are left are either in geologically complicated areas or contain heavy oil that is difficult to extract and process. Therefore, new technologies and any possible improvements to all stages of the oil recovery can contribute to the world energy supply.

Mathematical modeling of reservoirs

A computer model can help forecast some better known phenomena: in meteorology it improves weather prediction, in cyclone tracking it can save lives when people are evacuated on time from the endangered areas, in reservoir engineering it helps optimize economics. Nevertheless, computer simulations can never precisely predict the processes in the nature whether it is reservoir engineering, weather prediction, species spread, market fluctuations; the real world is far too complex. Therefore, data assimilation methods need to be employed.

Reservoir modeling plays an important role in enhanced oil recovery planning. It is a basis for field development and involves a significant number of researchers around the world in the industry as well as in the academia. The basic physics of fluid moving through a porous medium has been known since Henry Darcy in 1856 published his research. Modern computers allow to efficiently simulate the behavior of the fluids in different reservoir conditions and with several chemical components present.

Even though the models are not perfect due to simplifications and approximations, a bigger issue in reservoir engineering is reservoir condition specification. Unlike in some other disciplines, like meteorology for instance, it is not possible to observe the domain of interest that is deep underground or, even more challenging, under the sea bottom. For that reason, the crucial reservoir parameters like permeability or porosity are unknown. Therefore, initially the reservoir structural information comes from the geologists and geophysicists.

The way the sediments form and what processes create the landscape through the ages determine which structures are to be expected in the underground formations. Based on that information (and possible observations described later) an expert derives a detailed geological model. Two equivalent experts will produce two different property models according to their respective experience and interpretations. These property models are sometimes very detailed and need to be upscaled to make them suitable for flow simulation.

Observations of the behavior of the reservoirs

When an oil field is discovered there can be exploration data collected. The most common large scale observations in this case would be seismic observations. To gather seismic data, a controlled explosion is often performed and the signal of the reflected waves is recorded. The seismic waves reflect from subsurface structures in a different way depending on the structure's density, and that creates a seismic image. This image can be interpreted and inverted into subsurface structures.

However, today there are not many new fields being discovered. The industry highly relies on oil fields that are at various stages of development. Then, one wants to collect data for production improvement but at the lowest possible cost, and most data types are expensive to obtain.

Additionally to seismic data, there are other large scale observation techniques available, for example, gravity (Figure 1.1) or magnetic observations. While the wells are drilled in the field, the rock that is being removed can be a source of valuable information about the underground structure. It is a type of hard data (data that does not need to be inverted) that gives direct values of permeability and porosity through lab tests on a very small scale. Additionally, there can be wireline log data collected from instruments lowered down an open borehole. These can provide very accurate measurements in a small vicinity of the well.

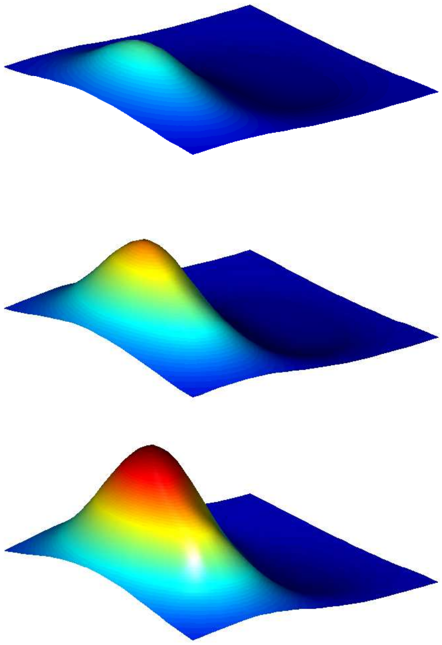


Figure 1.1: Simulated time-lapse gravity [micro Gal] measurements of a water-driven gas reservoir production process at (from the top) 4, 7 and 10 years. Courtesy of M. Glegola.

There are different types of wells, the more complicated the more expensive they get, reaching up to tens of millions of dollars per single well. The simplest wells are the vertical monitoring wells that only gather data and do not interact with the surroundings (common in hydrogeology). There are injection/production vertical wells that can also collect pressure and/or fluid flow data. The most technically advanced wells are the so-called smart horizontal wells that can reach areas that are difficult to access and can be kilometers away. These wells can sometimes be opened/closed to flow in separate segments along their lengths and, in that way, may provide additional data.

Data assimilation

In the case that a model is well calibrated and its parameters are known, it would have good predictive capacities, and it would be good enough to use the model in decision making concerning field development. When the data are rich and precise, they might be sufficient for successful operation of the field without using a model. Unfortunately, in most cases neither the model nor the data provide good-enough information due to simplifications and uncertainties, and the combination of the two can help to reduce the scarcity of information.

Data assimilation is a concept that combines the model with observed data in order to minimize uncertainties and improve predictions. The model as a theoretical representation of a natural process is not perfect and its uncertain parameters can be better estimated given acquired data. An improved model is expected to have better predictive capabilities, and in reservoir engineering can help planning the field development and designing production strategies.

Formally, data assimilation is a mathematical optimization problem. Typically, one is interested to minimize a mismatch between the data observed and the data modeled, and this mismatch is a basis for creating an objective (or cost) function. The objective function is a highly nonlinear function of its variables in large-scale applications. Then, the problems are difficult to solve and, therefore, a variety of data assimilation techniques exists, none of which is universally applicable. Additionally, the problem is non-unique, i.e. there exist its many solutions that minimize the given objective. In other words, real-life optimization problems are severely ill-posed.

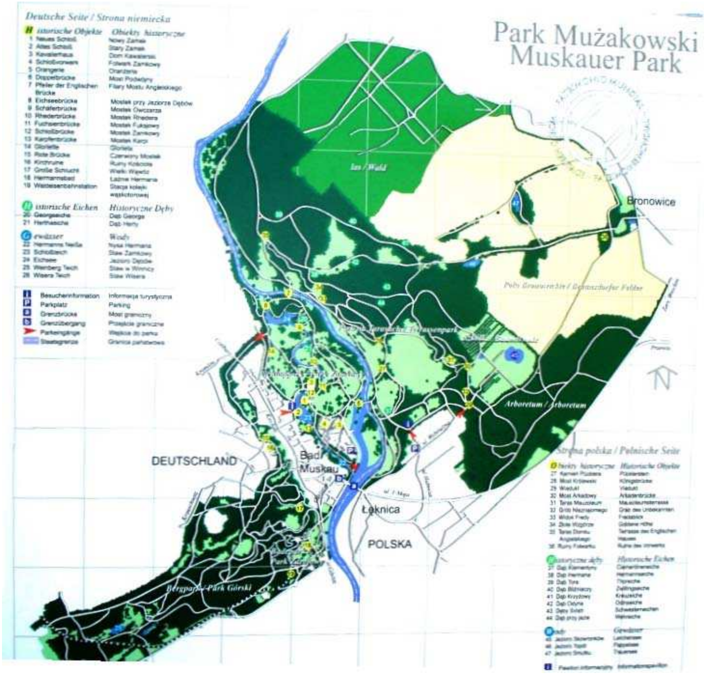


Figure 1.2: Map of Muskauer Park at the German-Polish border. A meandering river cuts through the park. The park's south-east border arc is the biggest terminal moraine in the world.

A variety of methods for data assimilation has been developed for the oil industry, [62]. Especially, within the last years emphasis has been put to produce geologically consistent permeability and porosity fields. Geological consistency requires the characteristics of the geometric structure of the parameters to be preserved during the data assimilation process, [10], [11]. These characteristics are properties of the fields that go beyond the mean and the covariance. Natural formations show a wide range of patterns like river meanders, salt domes, layer-cake formations, faults, wind or water eddies, or moraines (Figure 1.2). Dealing with patterns and handling images is clearly related to the image processing research area.

Image processing

The image processing discipline offers a variety of methods for dealing with images, patterns, features and other structures. Our goal is to combine data assimilation with pattern matching ideas. The two areas are not far apart, and a more detailed insight reveals many similarities. Image processing problems deal with phenomena without an underlying physical model, like a sequence of movie snapshots. The consecutive states (snapshots) are known and the goal is to define an automatic warping process from one image to another. The warping should be done in a natural way, i.e. in a way a human observer would do it, which reaches as far as cognitive science. An example could be face recognition when seeing one side of somebody's face would make us recognize a person in real life. Our brains interpolate that partial information in an instant but this every-day experience is a very complex task for computer face recognition software. Pattern recognition problems include also common finger-print matching, as present nowadays at airports, in laptops, or a police database. An ecologically oriented application is, for example, salamander matching, [95], where a digital picture of the animal needs to be compared to a collection of previously gathered images.

Applications like cyclone prediction, [2], moving fire fronts, [54], [6], precipitation and thunderstorms, [42], [100], [26], or epidemic spreading, [53], deal with structured, feature-driven fields. These approaches touch the area of image processing but at the same time contain dynamic models of the underlying phenomena. There, data assimilation is performed with a large amount of observations, namely, the parameter field of interest is observed but it is different from the model output. The two images present the same feature that might be displaced or aligned differently. The difficulty lies in making consistent corrections to the model predictions that have to be approached globally where underlying features are taken into account.

In this thesis we will adopt ideas from the image processing area within data assimilation methods for reservoir engineering applications. A typical problem in reservoir engineering is that the observations are usually very scarce and nonlinearly related to the variables that need to be estimated. We often deal with ill-posedness due to a large amount of unknown variables compared to the number of observations. The spatial uncertainty would usually be represented with a small ensemble of nodes, much smaller than the number of variables. This representation is too simple to picture the complex spatial uncertainty. This way spurious correlations arise between physically or ideologically distant states.

We will see that in reservoir engineering the unknown variables to be estimated are commonly permeability fields that are nothing else but images. Often, these images have a predefined structure that disappears during the data assimilation and, therefore, the whole process loses the geological background and realism.

Objective of the thesis

We propose two research directions to resolve the aforementioned problems:

- diminishing the spurious correlations between the sparse data and distantly located variables through upscaled treatment of the covariances,

- taking into account features in the domain and reducing the state size through an effective reparametrization.

First, a type of ensemble Kalman filter is applied, namely, an ensemble multiscale filter. In reservoir engineering, ensemble sequential filtering started with [59] in 2002 and since then various modifications were implemented, [20]. The ensemble multiscale filter has originated from multiscale trees in image processing ([22], [92], pyramidal matching [2]), and it was first implemented for an incompressible flow model, [99]. It upscales the covariance matrix on a tree structure, extracting stronger dependencies and introducing localized improvements. We develop it for the estimation of the permeability in reservoir models and test it thoroughly for several cases in reservoir engineering.

Second, a feature modeling technique called grid distortion is proposed. In reservoir engineering a number of feature-based methods have been implemented already: level sets ([58], [14]), Karhunen-Loève expansion ([72]), discrete cosine transformation ([34], [35], [36]), channel parametrization ([97], [84]), training-image based sampling ([49] and references therein) and elastic gridding ([76], [75]). The grid distortion method is based on smooth grid generation techniques, [80], but it can equivalently be viewed as an image warping technique, [67]. It provides smooth distortions of images adjusting the features in the domain, and can be expressed using very few parameters, which makes the problem better-posed. We pair grid distortion with the ensemble Kalman filter algorithm as well as with deterministic search methods.

Overview of the thesis

Before presenting the research results, an introduction to the modeling of reservoirs is provided. In Chapter 2 we describe the basic reservoir equations, show a small example of a forward run of an in-house reservoir simulator, and outline a concept of a forward model. Data assimilation with (ensemble) Kalman filters and optimization methods used in the thesis comprise Chapter 3. There, we derive basic (ensemble) Kalman filter equations, point out some improvements, and present other parameter estimation methods. The ensemble multiscale filter and its application are presented in Chapter 4, first for a simple case and later as a full data assimilation scheme in the context of reservoir engineering application. The content of this chapter has been published in [46] and [45]. Feature-based methods are discussed in Chapter 5; here, the focus is mainly on the grid distortion method developed in this thesis. The leading equations for grid distortion are derived and multiple implementations in 2D and 3D cases are shown. There is an intention to publish this research soon. The conclusions are finally presented in Chapter 6.

Chapter 2

Numerical modeling of reservoirs

In this chapter, a brief description of reservoir modeling is presented, followed by specifications of the main model used in this thesis. A detailed overview of the subject can be found in [37], [4], [5].

2.1 Modeling of reservoirs

Let us present equations for a three dimensional problem in Cartesian space, where x and y are horizontal coordinates, z is the vertical coordinate, and t is time.

Several phenomena play a role in creating a flow model of a reservoir with oil and water, namely, the mass balance concept, Darcy's law, equations of state, and the capillary pressure concept. We explain them further here.

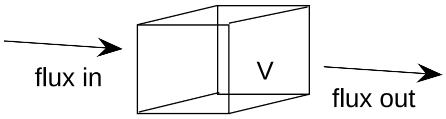


Figure 2.1: Representation of a volume with a flux going through it.

The mass conservation law gives rise to one mass balance equation per chemical component c. Let C_c be the mass concentration of component c in a unit volume V. Then, there exists a flux (a rate of change of mass), $\underline{F}$, transferring the mass through the volume (Figure 2.1). The amount of component c in the volume can be expressed as:

$$\iiint\limits_V C_c \, dV.$$

The mass concentration C_c can be expressed as $C_c = \phi \sum\limits_l \rho_l S_l y_{c,l}$ with ϕ - porosity, ρ_l - the density of phase l, S_l - the saturation of phase l, $y_{c,l}$ - the mass fraction of component c in phase l. Saturation of phase l is a fraction of the volume of the pores that is occupied by phase l; naturally, it is a value in the interval $[0,1]$ and the sum over all phases is

$$\sum_l S_l = 1.$$

The change in time of mass of component c is equal to

$$\frac{\partial}{\partial t}\left(\iiint\limits_V C_c \, dV\right) = \text{flux in} - \text{flux out}.$$

The difference in fluxes is a surface integral of the flux over a boundary of V, ∂V:

$$\text{flux in} - \text{flux out} = -\iint\limits_{\partial V} \underline{F} \cdot \underline{n} \, dA,$$

where $\underline{n}$ denotes a normal vector (in the direction out of the volume), and the symbol $\cdot$ indicates a dot product. This expression, according to the Divergence Theorem, is equal to:

$$-\iint\limits_{\partial V} \underline{F} \cdot \underline{n} \, dA = -\iiint\limits_V \nabla \cdot \underline{F} \, dV,$$

where $\nabla \cdot \underline{F}$ is the divergence of $\underline{F}$. This leads to

$$\frac{\partial}{\partial t}\left(\iiint\limits_V C_c \, dV\right) = -\iiint\limits_V \nabla \cdot \underline{F} \, dV,$$

and hence

$$\frac{\partial}{\partial t} C_c = -\nabla \cdot \underline{F}.$$

Finally, allowing additional input/output sources q leads to:

$$\frac{\partial}{\partial t} C_c = -\nabla \cdot \underline{F} + q. \tag{2.1}$$

Assuming dead components, i.e., components that do not travel between phases, and assuming that the various chemical components in the oil can be lumped into one 'pseudo component', two basic equations arise: one for oil and one for water (subscript w stands for water, o for oil):

$$C_w = \phi \rho_w S_w, \; C_o = \phi \rho_o S_o. \tag{2.2}$$

We know that for density ρ and velocity $\underline{v}$, flux is equal to $\underline{F} = \rho \underline{v}$, and we want to work out the velocity. Darcy's law is based on a proportionality between the pressure gradient, ∇p , and the negative velocity: $\nabla p \sim -\underline{v}$ or, more precisely,

$$-\frac{k}{\mu} \nabla p = \underline{v},$$

where k and μ denote the permeability tensor and viscosity, respectively. This is an equation for the case of one phase present. The two-phase Darcy's law differentiates between the water and oil equations:

$$-\frac{k}{\mu_w} k_{rw} \nabla p_w = \underline{v}_w,$$

$$-\frac{k}{\mu_o}k_{ro}\nabla p_o = \underline{v}_o.$$

Relative permeabilities, $k_{rw} \in [0,1]$ for water and $k_{ro} \in [0,1]$ for oil, are functions of water saturation. These relations are found experimentally. Additionally, in 3D domains the impact of gravity has to be taken into account and the two-phase Darcy's law becomes:

$$-\frac{k}{\mu_w}k_{rw}(\nabla p_w - \rho_w g \nabla d) = \underline{v}_w, \tag{2.3}$$

$$-\frac{k}{\mu_o}k_{ro}(\nabla p_o - \rho_o g \nabla d) = \underline{v}_o, \tag{2.4}$$

where g is the gravitational acceleration, ρ_w and ρ_o are water and oil densities, respectively, and d is depth pointing vertically downwards.

The compressibilities for water c_w, rock c_r and oil c_o are defined as:

$$c_w = \frac{1}{\rho_w}\frac{\partial \rho_w}{\partial p}, \; c_r = \frac{1}{\phi}\frac{\partial \phi}{\partial p}, \; c_o = \frac{1}{\rho_o}\frac{\partial \rho_o}{\partial p}.$$

The capillary pressure (p_c) constraint is related to the interfacial tension between the phases and the wetting properties of the rock which need to be determined experimentally:

$$p_c(S_w) = p_w - p_o,$$

where p_w and p_o are water and oil pressures, respectively.

For each phase we insert in Darcy's law (2.3) and (2.4), and concentration definitions (2.2), into Equation (2.1), given that $\underline{F}_l = \rho_l \underline{v}_l$ for $l \in \{w, o\}$:

$$\frac{\partial}{\partial t}(\phi \rho_l S_l) = -\nabla \cdot \left(\rho_l \left(-\frac{k}{\mu_l}k_{rl}(\nabla p_l - \rho_l g \nabla d)\right)\right) + q_l.$$

Using the chain rule gives:

$$\phi \underbrace{\frac{1}{\phi}\frac{\partial \phi}{\partial p}}_{c_r}\frac{\partial p}{\partial t}\rho_l S_l + \phi \rho_l \underbrace{\frac{1}{\rho_l}\frac{\partial \rho_l}{\partial p}}_{c_l}\frac{\partial p}{\partial t}S_l + \phi \rho_l \frac{\partial S_l}{\partial t} = \rho_l \nabla \cdot \left(\frac{k}{\mu_l}k_{rl}(\nabla p_l - \rho_l g \nabla d)\right) + q_l.$$

It leads to two coupled equations that need to be solved for S_w and p, [37], [5]:

$$\begin{aligned}\phi\left[S_w(c_w + c_r)\frac{\partial p}{\partial t} + \frac{\partial S_w}{\partial t}\right] \; = \; & \frac{\partial}{\partial x}\left[\frac{k}{\mu_w}k_{rw}\left(\frac{\partial p}{\partial x} - \rho_w g \frac{\partial d}{\partial x}\right)\right] + \\ & \frac{\partial}{\partial y}\left[\frac{k}{\mu_w}k_{rw}\left(\frac{\partial p}{\partial y} - \rho_w g \frac{\partial d}{\partial y}\right)\right] + \\ & \frac{\partial}{\partial z}\left[\frac{k}{\mu_w}k_{rw}\left(\frac{\partial p}{\partial z} - \rho_w g \frac{\partial d}{\partial z}\right)\right] + q_w,\end{aligned} \tag{2.5}$$

$$\begin{aligned}\phi\left[(1 - S_w)(c_o + c_r)\frac{\partial p}{\partial t} - \frac{\partial S_w}{\partial t}\right] \; = \; & \frac{\partial}{\partial x}\left[\frac{k}{\mu_o}k_{ro}\left(\frac{\partial p}{\partial x} - \rho_o g \frac{\partial d}{\partial x}\right)\right] + \\ & \frac{\partial}{\partial y}\left[\frac{k}{\mu_o}k_{ro}\left(\frac{\partial p}{\partial y} - \rho_o g \frac{\partial d}{\partial y}\right)\right] + \\ & \frac{\partial}{\partial z}\left[\frac{k}{\mu_o}k_{ro}\left(\frac{\partial p}{\partial z} - \rho_o g \frac{\partial d}{\partial z}\right)\right] + q_o,\end{aligned} \tag{2.6}$$

where $p = p_w = p_o$ when capillary forces are neglected. The initial conditions for $p(x, y, z)$ and $S_w(x, y, z)$ need to be specified and typically there is a no-flow condition assumed at the boundaries.

These equations can be spatially discretized and solved numerically with a finite difference scheme. Typically, a regular computational grid is used but advanced simulators might allow non-standard unstructured grids. The equations shown here are 3D, two-phase equations; gas can be added as a third phase. The presence of aquifers, faults, tracers can also be accounted for.

2.2 *simsim*

In this thesis a horizontal two-dimensional, two-phase (oil-water) reservoir model is used where the gravity effect can be neglected. Therefore, Equations (2.5) and (2.6) become:

$$\frac{\partial}{\partial x}\left(\frac{k}{\mu_w}k_{rw}\frac{\partial p}{\partial x}\right) + \frac{\partial}{\partial y}\left(\frac{k}{\mu_w}k_{rw}\frac{\partial p}{\partial y}\right) + q_w = \phi\left[S_w(c_w + c_r)\frac{\partial p}{\partial t} + \frac{\partial S_w}{\partial t}\right],$$

$$\frac{\partial}{\partial x}\left(\frac{k}{\mu_o}k_{ro}\frac{\partial p}{\partial x}\right) + \frac{\partial}{\partial y}\left(\frac{k}{\mu_o}k_{ro}\frac{\partial p}{\partial y}\right) + q_o = \phi\left[(1 - S_w)(c_o + c_r)\frac{\partial p}{\partial t} - \frac{\partial S_w}{\partial t}\right].$$

Reservoir simulator *simsim* (*sim*ple *sim*ulator, [37]) has been implemented and developed at Delft University of Technology.

After the equations are discretized in space for the two-phase flow, four methods for the time integration are available, [37], [5], [4]: explicit Euler, implicit Euler with Picard iteration, implicit Euler with Newton iteration, and IMPES (IMplicit Pressure Explicit Saturation).

The reservoir domain is 2D with no-flow through the boundaries. The wells (injectors or producers) can be implemented with prescribed initial pressures and rate constraints, or with prescribed initial rates and pressure constraints.

As an example of how *simsim* operates, a 49 × 49 reservoir is simulated for a year. The channelized permeability field and homogeneous porosity field are shown in the top row of Figure 2.2. There, the injection (I) and production (P) wells are indicated as white dots. The initial condition for reservoir pressure $\mathbf{p}_0$ (Pa) and water saturation $\mathbf{s}_0$ is $\left[\mathbf{p}_0^T\ \mathbf{s}_0^T\right]^T = \left[3 \cdot 10^7 \cdot \mathbb{1}_{49^2\times 1}^T\ 0.2 \cdot \mathbb{1}_{49^2\times 1}^T\right]^T$, where $\mathbb{1}$ is a column vector of ones. The streamlines, [37], between the three injectors at the left boundary and the three producers at the right boundary are shown in Figure 2.3. The wells operate under prescribed rates $(\pm 0.001\ m^3/s)$ and no pressure constraints.

After one year, the pressure and saturation fields look like in the bottom row of Figure 2.2. The gradual saturation change through time is shown in Figure 2.4.

The simulated data from each well can be collected and plotted against time. Figure 2.5 shows the bottom hole and the well grid block pressures together with the flow rates and water saturation in wells. Cumulative production data over the whole field are shown in Figure 2.6.

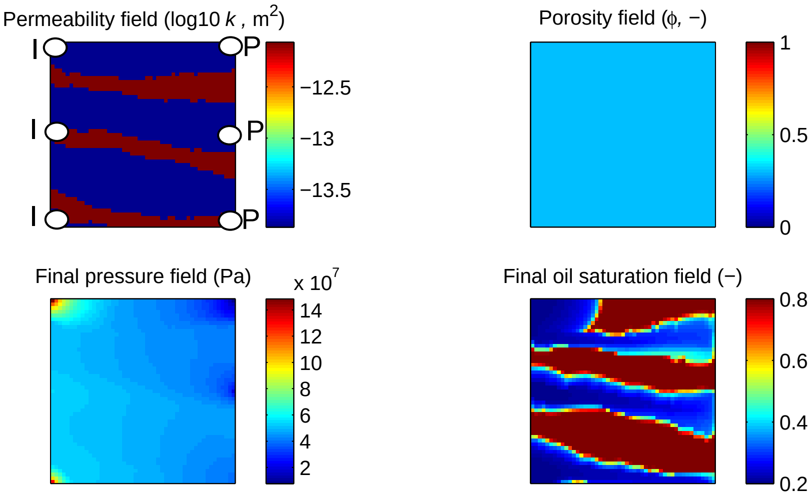


Figure 2.2: Permeability and porosity fields - top row, pressure and saturation after one year - bottom row.

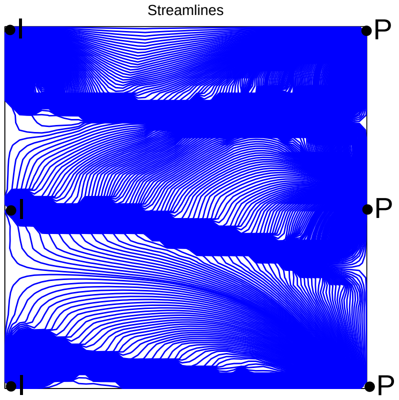


Figure 2.3: Streamlines with indicated wells.

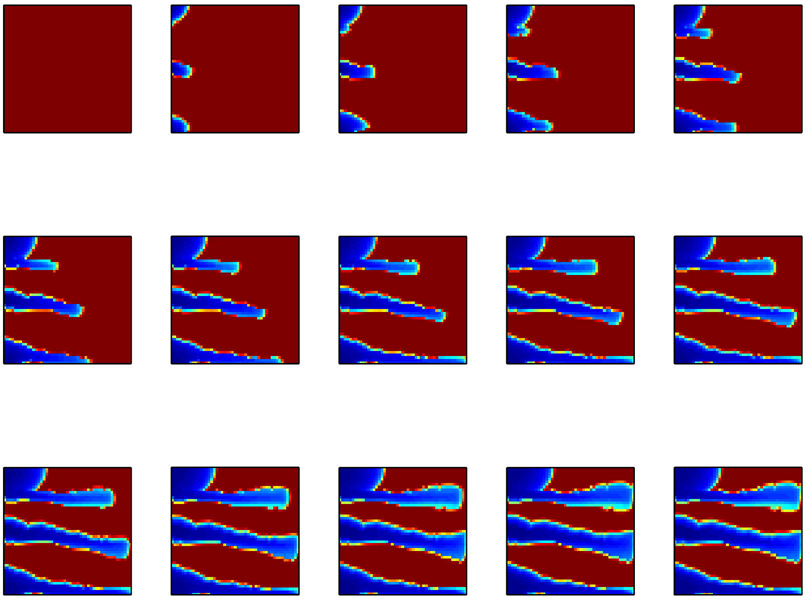


Figure 2.4: Saturation change during one year at equal time intervals.

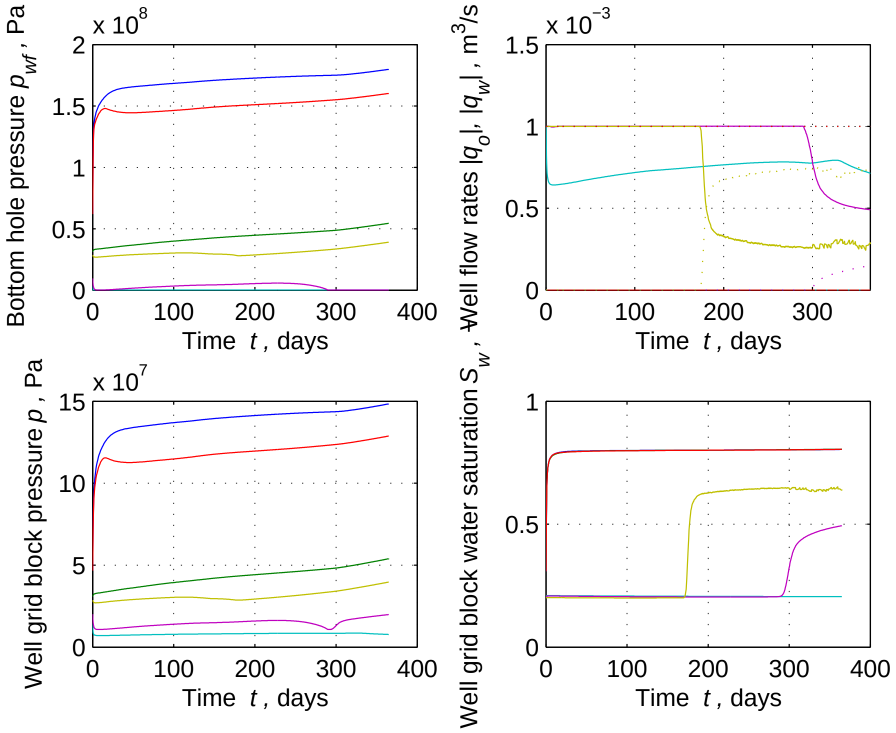


Figure 2.5: Simulated well data where colors indicate different wells.

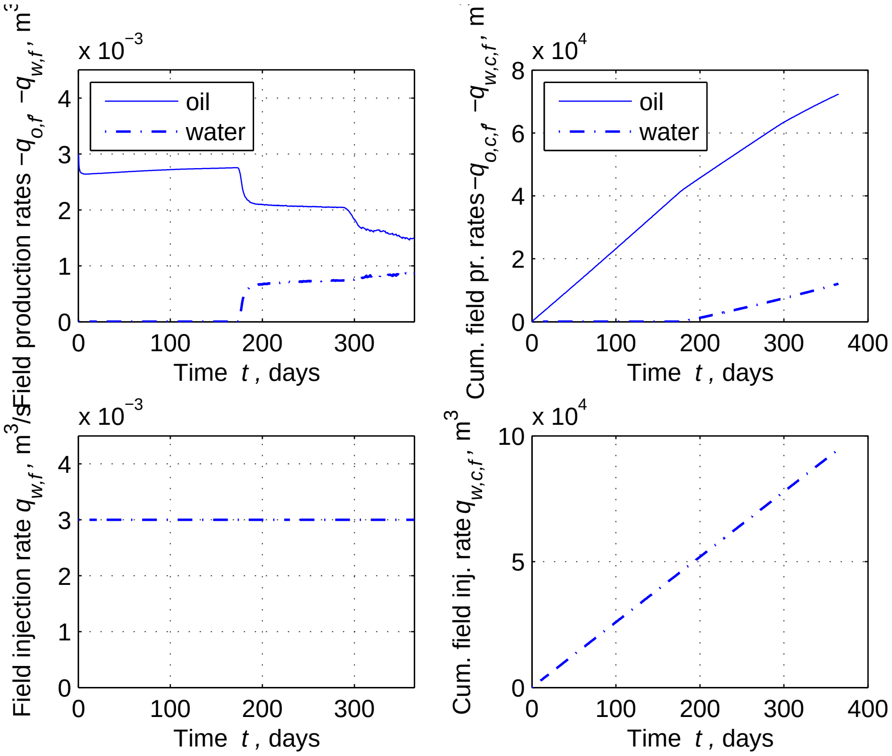


Figure 2.6: Simulated field data.

2.3 State-space representation of the model

Let $\mathbf{x}_k$ be a vector containing all the dynamic model variables at a discrete time k. In reservoir engineering it would typically be discretized pressure and saturation per grid block: $\mathbf{x}_k = \left[\mathbf{p}_k^T \ \mathbf{s}_k^T\right]^T$. Static variables, discretized permeability and porosity per grid block, are stored in vector $\mathbf{m} = \left[\mathbf{k}^T \ \boldsymbol{\phi}^T\right]^T$. The deterministic nonlinear model representation can be formulated as:

$$\mathbf{x}_{k+1} = \mathbf{f}_{k \to k+1}(\mathbf{m}, \mathbf{x}_k). \tag{2.7}$$

Here, $\mathbf{f}_{k \to k+1}(\mathbf{m}, \mathbf{x}_k)$ is a model operator that depends on specified static parameters $\mathbf{m}$ and propagates dynamic variables $\mathbf{x}_k$ from time k to $k+1$. An initial condition $\mathbf{x}_0 = \left[\mathbf{p}_0^T \ \mathbf{s}_0^T\right]^T$ needs to be specified. We assume no-flow boundary conditions.

Chapter 3

Data assimilation and parameter estimation

This chapter presents sequential ensemble data assimilation and parameter estimation methods that have been implemented for applications in this thesis. The descriptions of Kalman, ensemble Kalman and ensemble square root algorithms are presented. Implementation improvements and parameter estimation methods conclude the chapter.

3.1 Problem setup

The task of data assimilation is to improve the results of a numerical model of the phenomena one is interested in with the available observations. Even though the physics of many processes may be well-understood, the numerical models are often uncertain due to discretization errors and modeling approximations. Therefore, the model from Section 2.3, Equation (2.7), can in general contain a stochastic term expressing the model uncertainty. Moreover, a measurement equation is added for a full description of a state-space representation as given by:

$$\begin{cases} \mathbf{x}_{k+1} = \mathbf{f}_{k\to k+1}(\mathbf{m}, \mathbf{x}_k) + \boldsymbol{\varepsilon}_{k+1}, \\ \mathbf{y}_{k+1} = \mathbf{h}(\mathbf{x}_{k+1}) + \boldsymbol{\nu}_{k+1}. \end{cases} \tag{3.1}$$

The initial state $\mathbf{x}_0$ and the model parameters $\mathbf{m}$ need to be specified; $\mathbf{h}$ is a measurement operator expressing the relation between the state $\mathbf{x}_{k+1}$ and observations $\mathbf{y}_{k+1}$ at a given time; and $\boldsymbol{\varepsilon}$ and $\boldsymbol{\nu}$ are model noise and observation noise, respectively. The noise in the model is assumed to be normally distributed $\boldsymbol{\varepsilon} \sim N(\mathbf{0}, \mathbf{Q})$ with a model error covariance matrix $\mathbf{Q}$; the noise in the data is assumed to be normally distributed $\boldsymbol{\nu} \sim N(\mathbf{0}, \mathbf{R})$ with an observation error covariance matrix $\mathbf{R}$; and the model and observation errors are independent. The error covariance matrices might depend on time k.

The measurement operator $\mathbf{h}$ can depend on time when different data types are available at different times. Large scale measurements like seismic observations are expensive and not sensitive to small scale changes, therefore, they are collected relatively less frequently than well observations, for example. In real-life applications data come from instruments and can be contaminated with noise due to human or device error.

Upon observation arrival we can assess how well the model predicts just acquired observations, and on the basis of the mismatch deduct possible improvements to the model state. There exist many data assimilation techniques and they process the data in different ways. A common starting point is the specification of an objective function. Let us define a vector $\mathbf{x}_{[k]}$ that contains states $\mathbf{x}_k$ for all time steps k, and then the objective function to be minimized is

$$\begin{aligned} &J\left(\mathbf{x}_{[k]}\right) = \\ &\quad \frac{1}{2}\sum_k (\mathbf{y}_k - \mathbf{h}(\mathbf{x}_k))^T \mathbf{R}^{-1} (\mathbf{y}_k - \mathbf{h}(\mathbf{x}_k)) + \\ &\quad \frac{1}{2}\sum_k (\mathbf{x}^b - \mathbf{x}_k)^T \mathbf{B}^{-1} (\mathbf{x}^b - \mathbf{x}_k) + \\ &\quad \frac{1}{2}\sum_k (\mathbf{x}_{k+1} - \mathbf{f}(\mathbf{m}, \mathbf{x}_k))^T \mathbf{Q}^{-1} (\mathbf{x}_{k+1} - \mathbf{f}(\mathbf{m}, \mathbf{x}_k)). \end{aligned} \tag{3.2}$$

We are looking for $\mathbf{x}^*_{[k]} = \underset{\mathbf{x}_{[k]}}{\operatorname{argmin}}\ J\left(\mathbf{x}_{[k]}\right)$, where J is a nonnegative scalar function $J : \mathbb{R}^{n_{x_{[k]}}} \longrightarrow \mathbb{R}^+ \cup \{0\}$ with $n_{x_{[k]}}$ the size of vector $\mathbf{x}_{[k]}$. The function includes a quadratic data mismatch weighted by the noise covariance matrix $\mathbf{R}$, a quadratic background mismatch measuring how different the state $\mathbf{x}_k$ is from some background state $\mathbf{x}^b$, weighted by a given covariance matrix $\mathbf{B}$, and a quadratic model mismatch weighted by a given covariance matrix $\mathbf{Q}$.

3.2 The classical Kalman filter

Kalman filtering is a sequential optimization procedure that comprises of two steps: a forecast step and an update step, alternatingly applied until the last update time. First, the forecast step integrates the model to the first update time to collect the predicted observations, then the update is performed with the aid of the data, and the model continues with the new updated parameters until the next update time.

Let us consider a stochastic system like in Equation (3.1) but with linear model and observation operators:

$$\begin{cases} \mathbf{x}_{k+1} = \mathbf{F}_k \mathbf{x}_k + \boldsymbol{\varepsilon}_{k+1}, \\ \mathbf{y}_{k+1} = \mathbf{H}_k \mathbf{x}_{k+1} + \boldsymbol{\nu}_{k+1}, \end{cases} \tag{3.3}$$

where, as before, $\boldsymbol{\varepsilon} \sim N(\mathbf{0}, \mathbf{Q})$ and $\boldsymbol{\nu} \sim N(\mathbf{0}, \mathbf{R})$, and model parameters have been omitted. For simplicity of notation, we use $\mathbf{F}_k$ and $\mathbf{H}_k$ without the subscripts from here on. Then, the objective function in Equation (3.2) becomes:

$$\begin{aligned} &J\left(\mathbf{x}_{[k]}\right) = \\ &\quad \frac{1}{2}\sum_k (\mathbf{y}_k - \mathbf{H}\mathbf{x}_k)^T \mathbf{R}^{-1} (\mathbf{y}_k - \mathbf{H}\mathbf{x}_k) + \\ &\quad \frac{1}{2}\sum_k (\mathbf{x}^b - \mathbf{x}_k)^T \mathbf{B}^{-1} (\mathbf{x}^b - \mathbf{x}_k) + \\ &\quad \frac{1}{2}\sum_k (\mathbf{x}_{k+1} - \mathbf{F}\mathbf{x}_k)^T \mathbf{Q}^{-1} (\mathbf{x}_{k+1} - \mathbf{F}\mathbf{x}_k). \end{aligned} \tag{3.4}$$

At each update time we want to compute a distribution of state $\mathbf{x}_k$ given the data $\mathbf{y}_k$. This distribution can be shown to be Gaussian, [38]. Then, the estimate of the mean $\bar{\mathbf{x}}_k$, and the covariance $\mathbf{P}_k$ completely describe its shape. We want to derive the equations for $\bar{\mathbf{x}}_k$ and $\mathbf{P}_k$ that are time and measurement updates, respectively, where $\bar{\mathbf{x}}_k$ is an optimal state estimate and $\mathbf{P}_k$ is its corresponding minimized error at time k.

We will differentiate between forecasted and updated variables using superscripts f and a (for analyzed), respectively. In this way, $\mathbf{x}_k^f$ denotes forecasted dynamic variables at time k and $\mathbf{x}_k^a$ denotes analyzed dynamic variables at the same time after the update step; $\bar{\mathbf{x}}_k^f$ and $\bar{\mathbf{x}}_k^a$ denote forecasted and analyzed mean estimates at time k, respectively.

Since the model error mean is $\mathbf{0}$, the mean estimate $\bar{\mathbf{x}}_k$ propagates in time as given by:

$\bar{\mathbf{x}}_k$ time update

$$\bar{\mathbf{x}}_{k+1} = \mathbf{F}\bar{\mathbf{x}}_k. \tag{3.5}$$

Since the definition of a covariance matrix can be written as $\mathbf{P}_k = E[(\mathbf{x}_k - \bar{\mathbf{x}}_k)(\mathbf{x}_k - \bar{\mathbf{x}}_k)^T]$, let us define the forecasted and updated covariance matrices:

$$\mathbf{P}_k^f = E[(\mathbf{x}_k - \bar{\mathbf{x}}_k^f)(\mathbf{x}_k - \bar{\mathbf{x}}_k^f)^T],$$

$$\mathbf{P}_k^a = E[(\mathbf{x}_k - \bar{\mathbf{x}}_k^a)(\mathbf{x}_k - \bar{\mathbf{x}}_k^a)^T],$$

for the given mean estimates $\bar{\mathbf{x}}_k^f$ and $\bar{\mathbf{x}}_k^a$, where $E[\cdot]$ denotes the expected value.

To estimate the forward covariance propagation we compute:

$$\mathbf{x}_k - \bar{\mathbf{x}}_k \overset{(3.3),\,(3.5)}{=} \mathbf{F}\mathbf{x}_{k-1} + \boldsymbol{\varepsilon}_k - \mathbf{F}\bar{\mathbf{x}}_{k-1} = \mathbf{F}(\mathbf{x}_{k-1} - \bar{\mathbf{x}}_{k-1}) + \boldsymbol{\varepsilon}_k,$$

and this gives

$\mathbf{P}_k$ time update

$$\mathbf{P}_k = \mathbf{F}\mathbf{P}_{k-1}\mathbf{F}^T + \mathbf{Q}.$$

Let k be a fixed observation time. A Kalman filter aims at providing an optimal state estimate given the objective function at the observation time k. The objective function in Equation (3.4) for the $\bar{\mathbf{x}}_k^a$ estimate becomes:

$$J(\bar{\mathbf{x}}_k^a) = \frac{1}{2}(\mathbf{y}_k - \mathbf{H}\bar{\mathbf{x}}_k^a)^T\mathbf{R}^{-1}(\mathbf{y}_k - \mathbf{H}\bar{\mathbf{x}}_k^a) + \frac{1}{2}(\mathbf{x}_k^f - \bar{\mathbf{x}}_k^a)^T(\mathbf{P}_k^f)^{-1}(\mathbf{x}_k^f - \bar{\mathbf{x}}_k^a),$$

where the model mismatch term disappears due to Equation (3.5). Here the background mismatch term from Equation (3.4) is a distance to the forecast $\mathbf{x}_k^f$, and $\mathbf{P}_k^f$ denotes the error in this forecast mismatch.

A necessary condition for finding the objective function's minimum is that the function's gradient vanishes:

$$\nabla_{\bar{\mathbf{x}}_k^a} J = \mathbf{0}.$$

We compute the estimate $\bar{\mathbf{x}}_k^a$. Following [38], we get:

$$-\mathbf{H}^T\mathbf{R}^{-1}(\mathbf{y}_k - \mathbf{H}\bar{\mathbf{x}}_k^a) + (\mathbf{P}_k^f)^{-1}(\bar{\mathbf{x}}_k^a - \bar{\mathbf{x}}_k^f) = \mathbf{0},$$

and hence

$$\begin{aligned}
&\bar{\mathbf{x}}_k^a = \\
&\quad \left(\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} + (\mathbf{P}_k^f)^{-1}\right)^{-1}\left(\mathbf{H}^T\mathbf{R}^{-1}\mathbf{y}_k + (\mathbf{P}_k^f)^{-1}\bar{\mathbf{x}}_k^f\right) = \\
&\quad \left(\mathbf{R}\mathbf{H}^{-T}\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} + \mathbf{R}\mathbf{H}^{-T}(\mathbf{P}_k^f)^{-1}\right)^{-1}\mathbf{y}_k + \left(\mathbf{P}_k^f\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} + \mathbf{P}_k^f(\mathbf{P}_k^f)^{-1}\right)^{-1}\bar{\mathbf{x}}_k^f = \\
&\quad \left(\mathbf{H} + \mathbf{R}\mathbf{H}^{-T}(\mathbf{P}_k^f)^{-1}\right)^{-1}\mathbf{y}_k + \left(\mathbf{P}_k^f\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} + \mathbf{I}\right)^{-1}\bar{\mathbf{x}}_k^f.
\end{aligned} \tag{3.6}$$

We want to work out the terms in front of $\mathbf{y}_k$ and $\bar{\mathbf{x}}_k^f$ in Equation (3.6). We have

$$\left(\mathbf{H} + \mathbf{R}\mathbf{H}^{-T}(\mathbf{P}_k^f)^{-1}\right)^{-1} = \left[\left(\mathbf{H}(\mathbf{H}^{-T}(\mathbf{P}_k^f)^{-1})^{-1} + \mathbf{R}\right)\left(\mathbf{H}^{-T}(\mathbf{P}_k^f)^{-1}\right)\right]^{-1} = \mathbf{P}_k^f\mathbf{H}^T(\mathbf{H}\mathbf{P}_k^f\mathbf{H}^T + \mathbf{R})^{-1}, \tag{3.7}$$

and we want to show that

$$(\mathbf{P}_k^f\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} + \mathbf{I})^{-1} = \mathbf{I} - \mathbf{P}_k^f\mathbf{H}^T(\mathbf{H}\mathbf{P}_k^f\mathbf{H}^T + \mathbf{R})^{-1}\mathbf{H} \tag{3.8}$$

by simple multiplication:

$$\begin{aligned}
&\left(\mathbf{P}_k^f\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} + \mathbf{I}\right)\left(\mathbf{I} - \mathbf{P}_k^f\mathbf{H}^T(\mathbf{H}\mathbf{P}_k^f\mathbf{H}^T + \mathbf{R})^{-1}\mathbf{H}\right) = \\
&\quad \mathbf{I} + \mathbf{P}_k^f\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} - \mathbf{P}_k^f\mathbf{H}^T(\mathbf{H}\mathbf{P}_k^f\mathbf{H}^T + \mathbf{R})^{-1}\mathbf{H} - \mathbf{P}_k^f\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\mathbf{P}_k^f\mathbf{H}^T(\mathbf{H}\mathbf{P}_k^f\mathbf{H}^T + \mathbf{R})^{-1}\mathbf{H} = \\
&\quad \mathbf{I} + \mathbf{P}_k^f\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} - \mathbf{P}_k^f\mathbf{H}^T\mathbf{R}^{-1}\mathbf{R}(\mathbf{H}\mathbf{P}_k^f\mathbf{H}^T + \mathbf{R})^{-1}\mathbf{H} - \mathbf{P}_k^f\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\mathbf{P}_k^f\mathbf{H}^T(\mathbf{H}\mathbf{P}_k^f\mathbf{H}^T + \mathbf{R})^{-1}\mathbf{H} = \\
&\quad \mathbf{I} + \mathbf{P}_k^f\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} - \mathbf{P}_k^f\mathbf{H}^T\mathbf{R}^{-1}(\mathbf{R} + \mathbf{H}\mathbf{P}_k^f\mathbf{H}^T)(\mathbf{H}\mathbf{P}_k^f\mathbf{H}^T + \mathbf{R})^{-1}\mathbf{H} = \mathbf{I}.
\end{aligned}$$

Substituting Equations (3.7) and (3.8) in Equation (3.6), we obtain an expression for the update equation in the Kalman filter, [38]:

$\bar{\mathbf{x}}_k$ measurement update

$$\begin{aligned}
&\bar{\mathbf{x}}_k^a = \\
&\quad \left(\mathbf{H} + \mathbf{R}\mathbf{H}^{-T}(\mathbf{P}_k^f)^{-1}\right)^{-1}\mathbf{y}_k + \left(\mathbf{P}_k^f\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} + \mathbf{I}\right)^{-1}\bar{\mathbf{x}}_k^f = \\
&\quad \mathbf{P}_k^f\mathbf{H}^T\left(\mathbf{H}\mathbf{P}_k^f\mathbf{H}^T + \mathbf{R}\right)^{-1}\mathbf{y}_k + \left(\mathbf{I} - \mathbf{P}_k^f\mathbf{H}^T(\mathbf{H}\mathbf{P}_k^f\mathbf{H}^T + \mathbf{R})^{-1}\mathbf{H}\right)\bar{\mathbf{x}}_k^f = \\
&\quad \bar{\mathbf{x}}_k^f + \mathbf{P}_k^f\mathbf{H}^T(\mathbf{H}\mathbf{P}_k^f\mathbf{H}^T + \mathbf{R})^{-1}(\mathbf{y}_k - \mathbf{H}\bar{\mathbf{x}}_k^f).
\end{aligned} \tag{3.9}$$

The term

$$\mathbf{K} = \mathbf{P}_k^f\mathbf{H}^T(\mathbf{H}\mathbf{P}_k^f\mathbf{H}^T + \mathbf{R})^{-1} \tag{3.10}$$

is called the Kalman gain.

The updated covariance matrix $\mathbf{P}_k^a = E[(\mathbf{x}_k - \bar{\mathbf{x}}_k^a)(\mathbf{x}_k - \bar{\mathbf{x}}_k^a)^T]$ for the given mean estimate $\bar{\mathbf{x}}_k^a$ can now be computed from

$$\begin{aligned}
&\mathbf{x}_k - \bar{\mathbf{x}}_k^a \overset{(3.9)}{=} \\
&\quad \mathbf{x}_k - \bar{\mathbf{x}}_k^f - \mathbf{K}(\mathbf{y}_k - \mathbf{H}\bar{\mathbf{x}}_k^f) \overset{(3.3)}{=} \\
&\quad \mathbf{x}_k - \bar{\mathbf{x}}_k^f - \mathbf{K}(\mathbf{H}\mathbf{x}_k + \boldsymbol{\nu}_k - \mathbf{H}\bar{\mathbf{x}}_k^f) = \\
&\quad (\mathbf{I} - \mathbf{K}\mathbf{H})(\mathbf{x}_k - \bar{\mathbf{x}}_k^f) - \mathbf{K}\boldsymbol{\nu}_k,
\end{aligned}$$

which gives

$\mathbf{P}_k$ measurement update

$$
\begin{aligned}
&\mathbf{P}_k^a = \\
&\quad (\mathbf{I}-\mathbf{KH})\mathbf{P}_k^f(\mathbf{I}-\mathbf{KH})^T - \mathbf{KRK}^T = \\
&\quad (\mathbf{I}-\mathbf{KH})\mathbf{P}_k^f - (\mathbf{I}-\mathbf{KH})\mathbf{P}_k^f\mathbf{H}^T\mathbf{K}^T - \mathbf{KRK}^T = \\
&\quad (\mathbf{I}-\mathbf{KH})\mathbf{P}_k^f - \mathbf{K}(\mathbf{K}^{-1}\mathbf{P}_k^f - \mathbf{HP}_k^f + \mathbf{RH}^{-T})\mathbf{H}^T\mathbf{K}^T \overset{(3.10)}{=} \\
&\quad (\mathbf{I}-\mathbf{KH})\mathbf{P}_k^f - \mathbf{K}\left((\mathbf{HP}_k^f\mathbf{H}^T + \mathbf{R})\mathbf{H}^{-T}(\mathbf{P}_k^f)^{-1}\mathbf{P}_k^f - \mathbf{HP}_k^f + \mathbf{RH}^{-T}\right)\mathbf{H}^T\mathbf{K}^T = \\
&\quad (\mathbf{I}-\mathbf{KH})\mathbf{P}_k^f.
\end{aligned}
$$

Kalman filtering, [40], was originally developed for linear, Gaussian problems. It provides the optimal estimate of the state of the system and the covariance of the estimation error. It is also able to propagate these statistics in time. Models are, however, rarely linear and variables are Gaussian only in some specific cases. Lately, for computational reasons and to allow for nonlinear models, the ensemble Kalman filter (EnKF) was introduced, [16], [20], and became a new standard for sequential data assimilation.

3.3 The Ensemble Kalman Filter - EnKF

We present a sequential ensemble-based algorithm for nonlinear Gaussian processes, recall Equations (3.1):

$$
\begin{cases}
\mathbf{x}_{k+1} = \mathbf{f}_{k\to k+1}(\mathbf{m}, \mathbf{x}_k) + \boldsymbol{\varepsilon}_{k+1}, \\
\mathbf{y}_{k+1} = \mathbf{h}(\mathbf{x}_{k+1}) + \boldsymbol{\nu}_{k+1}.
\end{cases}
$$

The ensemble Kalman filter, [16], represents the distribution of the state vector $\mathbf{x}_k \in \mathbb{R}^{n_s\times 1}$ by a sample (a sample from the distribution of interest), i.e., a collection of possible realizations, also known as an ensemble, with n_e members:

$$\mathbf{X} = [\mathbf{x}^1\ \mathbf{x}^2\ \dots\ \mathbf{x}^{n_e}] \in \mathbb{R}^{n_s\times n_e}.$$

The time index has been omitted for clarity since all the variables considered here are taken at the same time step, namely, a discrete update step.

Let $\mathbf{Y}$ be a matrix holding n_e copies of the observation vector:

$$\mathbf{Y} = [\mathbf{y}\ \mathbf{y}\ \dots\ \mathbf{y}] \in \mathbb{R}^{n_o\times n_e},$$

where n_o is the number of observation points. The vector of observations $\mathbf{y}$ is an input to the data assimilation algorithm.

Let $\hat{\mathbf{Y}}$ be a matrix holding an ensemble of predicted measurements from each replicate:

$$\hat{\mathbf{Y}} = [\mathbf{h}(\mathbf{x}^1)\ \mathbf{h}(\mathbf{x}^2)\ \dots\ \mathbf{h}(\mathbf{x}^{n_e})] \in \mathbb{R}^{n_o\times n_e}.$$

This collection of vectors is a result of integrating the given model to the current update time for each ensemble member, n_e-times, and computing the forecasted observation values. For complicated models this forecast step can be very time consuming for large ensemble sizes.

The ensemble Kalman update is expressed as:

$$\mathbf{X}^a = \mathbf{X}^f + \mathbf{K}(\mathbf{Y} - \hat{\mathbf{Y}}),$$

and the Kalman gain $\mathbf{K} \in \mathbb{R}^{n_s \times n_o}$ is equal to:

$$\mathbf{K} = Cov(\mathbf{X}^f, \hat{\mathbf{Y}})[Cov(\hat{\mathbf{Y}}, \hat{\mathbf{Y}}) + \mathbf{R}]^{-1},$$

where $Cov(\cdot, \cdot)$ denotes a sample (cross-)covariance matrix.

This algorithm was initially described in [16] in 1994. The original formulation was wrong and its correction was published in [8] where it is explained that to preserve correct statistics the observations need to be perturbed. Since then a number of improvements have been introduced of which many were applied to reservoir engineering, [1]. Let $\mathbf{Y}$ be an ensemble of perturbed measurements:

$$\mathbf{Y} = [\mathbf{y} + \boldsymbol{\nu}^1 \ \mathbf{y} + \boldsymbol{\nu}^2 \ ... \ \mathbf{y} + \boldsymbol{\nu}^{n_e}] \in \mathbb{R}^{n_o \times n_e},$$

where $\boldsymbol{\nu}$ is a measurement error sampled from a normal zero mean distribution with a given covariance matrix $\mathbf{R}$ that expresses our belief in the uncertainty of the measurements.

The setup of the ensemble Kalman filter, where the distribution of a variable is represented by a sample, is easy to implement. Even though it has been used successfully in many applications, it is known to cause complications and some of them are discussed here further.

The square root implementation of the ensemble Kalman filter is presented since it is the version used throughout this work. The basic idea of the ensemble Kalman filter is an introduction to almost every major section for the sake of completeness. It might also be presented from different (but equivalent) angles.

3.3.1 The Ensemble Square-Root Filter - EnSRF

In the classical ensemble Kalman filter as formulated in the previous section, the construction of the ensemble of perturbed measurements $\mathbf{Y}$ contains noise that can be an additional source of sampling error for small ensemble sizes, [19]. Therefore, square root algorithms were developed to avoid the use of perturbed measurements. Note that the version presented here includes a matrix of perturbations as a sample representation of a covariance matrix, [18], which lowers the computational burden.

Square root algorithms use an ensemble representation like the EnKF, but update the mean and deviations from the mean separately as in the traditional Kalman filter. In fact, both equations, the update of the mean and the update of the perturbations, could be directly derived from the classical Kalman filter equations but with the modifications regarding the ensemble representations. A square root version of the filter is used throughout this thesis. The basis of the algorithm was taken from [18], Section 7.4.2[1] therein, and improved as described in [70] and [51]. Other versions of square root filters exist, see for example [83].

Let us again define:

- $\mathbf{X} = [\mathbf{x}^1 \ \mathbf{x}^2 \ ... \ \mathbf{x}^{n_e}] \in \mathbb{R}^{n_s \times n_e}$ - an ensemble of state vectors,

---

[1] Implementation thanks to Dr. B. Jafarpour.

- $\mathbf{Y} = [\mathbf{y} + \boldsymbol{\nu}^1 \ \mathbf{y} + \boldsymbol{\nu}^2 \ ... \ \mathbf{y} + \boldsymbol{\nu}^{n_e}] \in \mathbb{R}^{n_o \times n_e}$ - an ensemble of perturbed measurements,
- $\hat{\mathbf{Y}} = [\mathbf{h}(\mathbf{x}^1) \ \mathbf{h}(\mathbf{x}^2) \ ... \ \mathbf{h}(\mathbf{x}^{n_e})] \in \mathbb{R}^{n_o \times n_e}$ - an ensemble of predicted measurements.

Additionally, we need:

- $\mathbb{1}_n \in \mathbb{R}^{n \times n}$ - a matrix where each element is equal to $\frac{1}{n}$,
- $\mathbf{I}$ - identity matrix of a proper size,
- $\bar{\mathbf{X}} = \mathbf{X}\mathbb{1}_{n_e} \in \mathbb{R}^{n_s \times n_e}$ - a matrix storing the ensemble mean n_e-times,
- $\mathbf{X}' = \mathbf{X} - \mathbf{X}\mathbb{1}_{n_e} = \mathbf{X} - \bar{\mathbf{X}}$ - an ensemble-perturbation matrix,
- $\mathbf{E} = [\boldsymbol{\nu}^1 \ \boldsymbol{\nu}^2 \ ... \ \boldsymbol{\nu}^{n_e}]$ - an ensemble of measurement perturbations,
- $\mathbf{S} = \hat{\mathbf{Y}} - \hat{\mathbf{Y}}\mathbb{1}_{n_e}$ - a matrix holding perturbations of the predicted measurements.

We want the above variables to be understood as (derived from) forecasted variables, for the ease of notation omitting the superscript f. The EnSRF defines the updated ensemble as a sum of updated ensemble mean and updated ensemble perturbations, that is:

$$\mathbf{X}^a = (\bar{\mathbf{X}})^a + (\mathbf{X}')^a$$

and

$$(\bar{\mathbf{X}})^a = \mathbf{X}\mathbb{1}_{n_e} + \mathbf{X}'\mathbf{S}^T\mathbf{X}_1(\mathbf{I} + \boldsymbol{\Sigma}_1^2)^{-1}\mathbf{X}_1^T(\mathbf{Y}\mathbb{1}_{n_e} - \hat{\mathbf{Y}}\mathbb{1}_{n_e}),$$

$$(\mathbf{X}')^a = \mathbf{X}'\mathbf{V}_2\sqrt{\mathbf{I} - \boldsymbol{\Sigma}_2^T\boldsymbol{\Sigma}_2}\mathbf{V}_2^T.$$

The last multiplication by $\mathbf{V}_2^T$ provides the unbiasedness of the filter, [70], [51], [19], since it was shown that the original algorithm without the last multiplication does not conserve the mean. To compute these equations we need to know matrices: $\boldsymbol{\Sigma}_1$, $\mathbf{X}_1$, $\boldsymbol{\Sigma}_2$, and $\mathbf{V}_2$. Let the operator $\overset{\text{tSVD}}{=}$ denote the thin singular value decomposition, [27], then the following sequence of equations completes the EnSRF update step:

1. $\mathbf{S} \overset{\text{tSVD}}{=} \mathbf{U}_0\boldsymbol{\Sigma}_0\mathbf{V}_0^T$,
2. $\mathbf{X}_0 = \boldsymbol{\Sigma}_0^{-1}\mathbf{U}_0^T\mathbf{E}$,
3. $\mathbf{X}_0 \overset{\text{tSVD}}{=} \mathbf{U}_1\boldsymbol{\Sigma}_1\mathbf{V}_1^T$,
4. $\mathbf{X}_1 = \mathbf{U}_0(\boldsymbol{\Sigma}_0^{-1})^T\mathbf{U}_1$,
5. $\mathbf{X}_2 = (\mathbf{I} + \boldsymbol{\Sigma}_1^2)^{-\frac{1}{2}}\mathbf{X}_1^T\mathbf{S}$,
6. $\mathbf{X}_2 \overset{\text{tSVD}}{=} \mathbf{U}_2\boldsymbol{\Sigma}_2\mathbf{V}_2^T$.

Note that the equation for the perturbation update $(\mathbf{X}')^a$ comes from the covariance measurement update in the classical Kalman filter $\mathbf{P}^a = (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{P}^f$ since $\mathbf{P}^a = (\mathbf{X}')^a[(\mathbf{X}')^a]^T$, $\mathbf{P}^f = \mathbf{X}'(\mathbf{X}')^T$ and we can write, [19]:

$$\begin{aligned}
\mathbf{P}^a &= (\mathbf{I}-\mathbf{KH})\mathbf{P}^f \\
\mathbf{P}^a &= \mathbf{P}^f - \mathbf{P}^f\mathbf{H}^T(\mathbf{HP}^f\mathbf{H}^T+\mathbf{R})^{-1}\mathbf{HP}^f \\
(\mathbf{X}')^a[(\mathbf{X}')^a]^T &= \mathbf{X}'(\mathbf{X}')^T - \mathbf{X}'(\mathbf{X}')^T\mathbf{H}^T(\mathbf{HX}'(\mathbf{X}')^T\mathbf{H}^T+\mathbf{R})^{-1}\mathbf{HX}'(\mathbf{X}')^T \\
(\mathbf{X}')^a[(\mathbf{X}')^a]^T &= \mathbf{X}'\left[\mathbf{I}-(\mathbf{X}')^T\mathbf{H}^T(\mathbf{HX}'(\mathbf{X}')^T\mathbf{H}^T+\mathbf{R})^{-1}\mathbf{HX}'\right](\mathbf{X}')^T
\end{aligned}$$

where we used Equation (3.10) for the Kalman gain. Then, if we write

$$\mathbf{I}-(\mathbf{X}')^T\mathbf{H}^T(\mathbf{HX}'(\mathbf{X}')^T\mathbf{H}^T+\mathbf{R})^{-1}\mathbf{HX}' = \mathbf{TT}^T$$

we obtain

$$(\mathbf{X}')^a[(\mathbf{X}')^a]^T = \mathbf{X}'\mathbf{TT}^T(\mathbf{X}')^T = \mathbf{X}'\mathbf{T}(\mathbf{X}'\mathbf{T})^T.$$

Finally, the square root perturbation update in a general form is:

$$(\mathbf{X}')^a = \mathbf{X}'\mathbf{T}.$$

Matrix $\mathbf{T}$ is called a square root transformation matrix and it is not unique, see [51] and [70] for a detailed discussion.

3.4 Implementation issues of the Ensemble Kalman Filter

The EnKF or EnSRF even in the most efficient form will almost never give satisfactory results when implemented for the first time in a new application. To be able to apply it successfully, different modifications, [1], [3], [19], can be used that depend on the problems encountered in a given setup. A number of improvements used in our implementations is presented in this section.

In today's research we want not only the dynamic model variables $\mathbf{x}_k$ to be estimated but also additional uncertain parameters $\mathbf{m}$. When $\mathbf{x}_k$ follows the model equation $\mathbf{x}_{k+1} = \mathbf{f}_{k\to k+1}(\mathbf{m}, \mathbf{x}_k)$, $\mathbf{m}$ does not change in time, i.e., $\mathbf{m}_{k+1} = \mathbf{I}\mathbf{m}_k$ where $\mathbf{I}$ is an identity matrix.

Updating all variables

Typically, all variables, dynamic and static, are updated and they comprise the state vector. Then, the state-space representation is of the form:

$$\begin{cases}
\begin{bmatrix} \mathbf{x}_{k+1} \\ \mathbf{m}_{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{k\to k+1}(\mathbf{m}_k, \mathbf{x}_k) \\ \mathbf{I}\mathbf{m}_k \end{bmatrix} + \begin{bmatrix} \boldsymbol{\varepsilon}_{k+1} \\ \mathbf{0} \end{bmatrix}, \\
\mathbf{y}_{k+1} = \mathbf{h}(\mathbf{x}_{k+1}) + \boldsymbol{\nu}_{k+1}.
\end{cases}$$

The measurement operator $\mathbf{h}$ works on the model outputs, where the model runs from time k to $k+1$.

In reservoir engineering the reservoir simulator is considered to contain no randomness. It is assumed that the model represents the physics of the fluid dynamics perfectly, and that all uncertainty is in the model parameters. Therefore, the state-space representation can be written as:

$$\begin{cases} \begin{bmatrix} \mathbf{x}_{k+1} \\ \mathbf{m}_{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{k\to k+1}(\mathbf{m}_k, \mathbf{x}_k) \\ \mathbf{I}\mathbf{m}_k \end{bmatrix}, \\ \mathbf{y}_{k+1} = \mathbf{h}(\mathbf{x}_{k+1}) + \boldsymbol{\nu}_{k+1}. \end{cases}$$

A common problem in data assimilation for this case is the un-physical updates of the variables. The laws of physics that determine the shape of pressure or saturation fields are not taken into account when a linear update is performed. The fields represent a dynamic state at a given point in time that is used for simulation initialization and must be physically meaningful.

Parameter update

One way to avoid un-physical updates of the dynamic variables is updating the constant parameters only, and leaving the dynamic variables unchanged at the update time step. Then, the state-space representation is rewritten as:

$$\begin{cases} \mathbf{m}_{k+1} = \mathbf{I}\mathbf{m}_k, \\ \mathbf{y}_{k+1} = \mathbf{h}\left(\mathbf{f}_{k\to k+1}(\mathbf{m}_k, \mathbf{x}_k)\right) + \boldsymbol{\nu}_{k+1}. \end{cases}$$

In this case, the model process continues with dynamic variables that were the result of a forward model $\mathbf{x}_k = \mathbf{f}_{k-1\to k}(\mathbf{m}_{k-1}, \mathbf{x}_{k-1})$, therefore, they are physically meaningful. Nevertheless, the dynamic variables $\mathbf{x}_k$ are now inconsistent with the constant variables $\mathbf{m}_k$ since the latter are a result of a linear update. The inconsistency might not allow the forward model to continue. The solution to this problem could be using reruns.

Reruns

Reruns, [62], [96], are used if one can afford running the model from time zero after each analysis time using specified initial conditions, and updated constant variables from the last update step. The advantage is that the model starts with the updated constant variables from an initial state such that no inconsistency appears, and it is unlikely to introduce physically-impossible relations. Then, the state-space representation is of the form:

$$\begin{cases} \mathbf{m}_{k+1} = \mathbf{I}\mathbf{m}_k, \\ \mathbf{y}_{k+1} = \mathbf{h}\left(\mathbf{f}_{0\to k+1}(\mathbf{m}_k, \mathbf{x}_0)\right) + \boldsymbol{\nu}_{k+1}. \end{cases}$$

Alternatively, one can rerun the model only after selected update steps. These can be the update steps with more observations available or update steps fulfilling a given criterium, [15].

Asynchronous scheme

In the case when observations are obtained very frequently, it becomes inefficient to update the state at each measurement time step. The model would need to be run and stopped very often whether we proceed in between time steps or use reruns. The asynchronous EnKF [71] was proposed to improve data assimilation for this kind of case studies.

Let us divide the time domain into assimilation windows indexed by K. Window K contains n_K measurement times. Index k_K runs through the measurement times in window K. The predicted data for time window $K+1$ becomes a collection of n_{K+1} predicted data sets and the state-space representation becomes:

$$\begin{cases} \mathbf{m}_{K+1} = \mathbf{I}\mathbf{m}_K, \\ \mathbf{y}_{K+1} = \begin{bmatrix} \mathbf{h}\left(\mathbf{f}_{\cdot\to k_{K+1}=1}(\mathbf{m}_K, \mathbf{x}_\cdot)\right) \\ \mathbf{h}\left(\mathbf{f}_{\cdot\to k_{K+1}=2}(\mathbf{m}_K, \mathbf{x}_\cdot)\right) \\ \vdots \\ \mathbf{h}\left(\mathbf{f}_{\cdot\to k_{K+1}=n_{K+1}}(\mathbf{m}_K, \mathbf{x}_\cdot)\right) \end{bmatrix} + \boldsymbol{\nu}_{K+1}, \end{cases}$$

where $\mathbf{x}_\cdot$ is a dynamic variable from a chosen earlier point in time.

Variable transformation

At the update step, a magnitude of the updated variable might exceed its feasible boundaries, [63], and the variable becomes physically meaningless. One solution is to 'crop' the outliers to stay within the boundaries of the interval of feasible values. It can, however, create non-smooth changes in the variable's values which is not always desirable. Another solution is to project the variable values to a space with infinite bounds, update it, and project back to its bounded space. Let $[min_u,\ max_u]$ be an interval of possible values for variable u, and let T be a bijection such that:

$$T : [min_u,\ max_u] \longrightarrow \mathbb{R},$$

and then its inverse function is $T^{-1} : \mathbb{R} \longrightarrow [min_u,\ max_u]$.

At any update time (time subscript omitted for clarity) the following steps are taken:

- forward-transformation T on the forecasted variable:

$$T(u^f) = \widetilde{u}^f,$$

- the update

$$update(\widetilde{u}^f) = \widetilde{u}^a,$$

- backward-transformation T^{-1} on the update:

$$T^{-1}(\widetilde{u}^a) = u^a.$$

Note that variable u is understood as a single physical notion. If one wishes to update variables with different magnitudes, values or physical interpretation, several T functions are needed.

If the transformation is applied in this work, it is the one introduced in [24]. Let $m_m = \frac{max_u + min_u}{2}$ and $m_r = \frac{max_u - min_u}{2}$. Then the forward transformation is:

$$T(u^f) = \widetilde{u}^f = \ln \frac{u^f - min_u}{max_u - u^f}.$$

The backward transformation after the update is:

$$T_\alpha^{-1}(\widetilde{u}^a) = u^a = m_m + m_r \frac{\exp(\alpha \widetilde{u}^a) - 1}{\exp(\alpha \widetilde{u}^a) + 1},$$

and it is illustrated in Figure 3.1 for several α values. For the implementations in this thesis we simply pick $\alpha = 1$. Different values of α reflect a tradeoff between the smoothness and the accuracy of the backward-transformation.

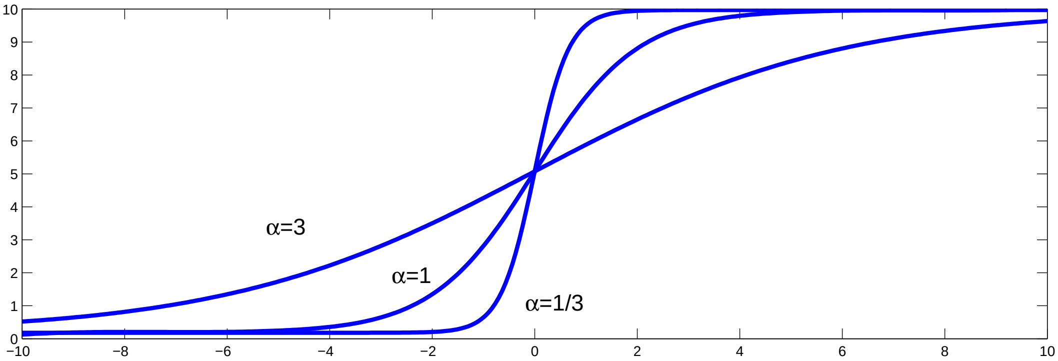


Figure 3.1: The backward transformation with different values of α.

Localization

Often, small ensembles can introduce false (much too large or much too small) spatial correlations between variables. These correlations can trigger an ensemble Kalman filter to misjudge the magnitude of a measurement update. It in turn might lead to ensemble collapse in the sense that it would converge to a state different from the true state.

To correct for a misrepresented uncertainty, a covariance localization has been proposed and a review of possible methods is presented in [1]. In general, the localization aims at correcting an ensemble covariance matrix $\mathbf{P}_e$ with an element-by-element multiplication by a covariance-localization matrix $\boldsymbol{\rho}$:

$$\boldsymbol{\rho} \circ \mathbf{P}_e : \ (\boldsymbol{\rho} \circ \mathbf{P}_e)_{i,j} = (\boldsymbol{\rho})_{i,j} \cdot (\mathbf{P}_e)_{i,j},$$

and using the latter in place of $\mathbf{P}_e$ in the ensemble Kalman measurement update equations. Provided that the covariance-localization matrix $\boldsymbol{\rho}$ is positive-definite, and since the ensemble covariance $\mathbf{P}_e$ is always positive-semidefinite with positive main diagonal entries, the Schur product $\boldsymbol{\rho} \circ \mathbf{P}_e$ is positive-definite, according to the Schur product theorem, [1]. That is, the Schur product is again a covariance.

Different localization techniques are used depending on the problem at hand. Most certainly, a localization is necessary if only a small ensemble can be used or there is a vast amount of spatially distributed data available.

3.5 Parameter estimation problem

In reservoir engineering the model parameters $\mathbf{m}$, containing for example permeability or porosity fields, are uncertain. Therefore, often one wants to estimate their value given observations, and the objective function in (3.2) turns into:

$$\begin{aligned}
&J\left(\mathbf{x}_{[k]}(\mathbf{m})\right) = \\
&\quad \frac{1}{2}\sum_k (\mathbf{y}_k - \mathbf{h}(\mathbf{x}_k(\mathbf{m})))^T \mathbf{R}^{-1} (\mathbf{y}_k - \mathbf{h}(\mathbf{x}_k(\mathbf{m}))) + \\
&\quad \frac{1}{2}\sum_k (\mathbf{x}^b - \mathbf{x}_k(\mathbf{m}))^T \mathbf{B}^{-1} (\mathbf{x}^b - \mathbf{x}_k(\mathbf{m})) + \\
&\quad \frac{1}{2}(\mathbf{m}_0 - \mathbf{m})^T \mathbf{P}_0^{-1} (\mathbf{m}_0 - \mathbf{m}),
\end{aligned}$$

where we do not consider the model noise, and add a prior term where $\mathbf{m}_0$ is an initial guess and $\mathbf{P}_0$ its error. Since $\mathbf{m}$ has no underlying dynamics, the problem turns into a parameter estimation problem and one wants to estimate $\mathbf{m}^*$ such that

$$\mathbf{m}^* = \underset{\mathbf{m}}{\operatorname{argmin}}\ J(\mathbf{x}_{[k]}(\mathbf{m})).$$

The objective is a scalar function, $J : \mathbb{R}^{n_m} \longrightarrow \mathbb{R}^+ \cup \{0\}$, whose domain is a set of possible values of parameters $\mathbf{m}$ for which $J(\mathbf{x}_{[k]}(\mathbf{m}))$ makes sense, and n_m is the size of vector $\mathbf{m}$. Minimizing a function of several variables might be challenging especially if n_m is very large. In reservoir engineering applications n_m can reach 10^6 and even more for real life large fields.

To find the objective function's minimum $\mathbf{m}^*$ we start from investigating its gradient with respect to the sought parameters, $\nabla_{\mathbf{m}} J$. Using the chain rule, the formula for the gradient can be computed and it is equal to:

$$\begin{aligned}
&\nabla_{\mathbf{m}} J = \\
&\quad -\sum_k \left(\frac{\partial \mathbf{x}_k}{\partial \mathbf{m}}\right)^T \left(\frac{\partial \mathbf{h}}{\partial \mathbf{x}_k}\right)^T \mathbf{R}^{-1} (\mathbf{y}_k - \mathbf{h}(\mathbf{x}_k(\mathbf{m}))) \\
&\quad -\sum_k \left(\frac{\partial \mathbf{x}_k}{\partial \mathbf{m}}\right)^T \mathbf{B}^{-1} (\mathbf{x}^b - \mathbf{x}_k(\mathbf{m})) \\
&\quad -\mathbf{P}_0^{-1} (\mathbf{m}_0 - \mathbf{m}).
\end{aligned}$$

This gradient is not always available or easy to compute, therefore, gradient-based minimization methods (e.g. BFGS [24], variational methods [12], [47], [52], steepest descent) might be prohibitive. It comes from the fact that it is difficult to compute the derivatives in the gradient formula. If the observation operator is simple, $\frac{\partial \mathbf{h}}{\partial \mathbf{x}_k}$ could be easy to compute. The main difficulty are usually the derivatives $\frac{\partial \mathbf{x}_k}{\partial \mathbf{m}}$ which are the derivatives of the forward model with respect to the model parameters. Variational techniques, [47], implement adjoints that are used to efficiently compute the gradient; for complex nonlinear models they are difficult to derive, too. Additionally, due to a high nonlinearity of a cost function they might be ineffective in finding a global minimum. The gradients could alternatively be computed using a simple finite difference scheme but for the large sizes of $\mathbf{m}$ it is not feasible.

Another approach to solving the minimization problem is the use of gradient-free methods (e.g. Kalman filtering [40], [1], particle filters [85], [86], pattern search methods [48], or genetic algorithms [66], [73]). In this thesis only gradient-free methods were implemented and they are presented in detail in the following sections.

In the case when the parameter space is very large and the observations are scarce, there are many degrees of freedom while fitting the data, and the solution is most certainly non-unique. Then it might be advantageous to constrain the parameter space to only some type of solutions, and express it using a smaller set of variables via a given transformation. Let us reduce the parameter space $\mathbf{m} = \mathbf{m}(\boldsymbol{\alpha})$ and rewrite the objective as a function of $\boldsymbol{\alpha}$:

$$\begin{aligned}
&J(\mathbf{x}_{[k]}(\mathbf{m}(\boldsymbol{\alpha}))) = \\
&\quad \frac{1}{2}\sum_k (\mathbf{y}_k - \mathbf{h}(\mathbf{x}_k(\mathbf{m}(\boldsymbol{\alpha}))))^T \mathbf{R}^{-1} (\mathbf{y}_k - \mathbf{h}(\mathbf{x}_k(\mathbf{m}(\boldsymbol{\alpha})))) + \\
&\quad \frac{1}{2}\sum_k (\mathbf{x}^b - \mathbf{x}_k(\mathbf{m}(\boldsymbol{\alpha})))^T \mathbf{B}^{-1} (\mathbf{x}^b - \mathbf{x}_k(\mathbf{m}(\boldsymbol{\alpha}))) + \\
&\quad \frac{1}{2}(\boldsymbol{\alpha}_0 - \boldsymbol{\alpha})^T \mathbf{P}_0^{-1} (\boldsymbol{\alpha}_0 - \boldsymbol{\alpha}).
\end{aligned}$$

Analogically, the prior term contains an initial guess $\boldsymbol{\alpha}_0$ and its error $\mathbf{P}_0$. Here we are looking for the minimum

$$\boldsymbol{\alpha}^* = \underset{\boldsymbol{\alpha}}{\operatorname{argmin}}\ J(\mathbf{x}_{[k]}(\mathbf{m}(\boldsymbol{\alpha}))),$$

where $J : \mathbb{R}^{n_\alpha} \longrightarrow \mathbb{R}^+ \cup \{0\}$ and $n_\alpha \ll n_m$ for n_α being the size of vector $\boldsymbol{\alpha}$. The problem posed this way might still be too complicated to solve with gradient-based methods even though the size of the parameter space has been reduced. It is often due to a high degree of nonlinearity associated with the reduced space.

The problem setup in this thesis always concerns estimation of permeability parameter (implicitly $\boldsymbol{\alpha}$ or explicitly $\mathbf{m}$), that is, only constant uncertain parameters are sought. Not only Kalman filtering can be applied to find the desired parameters but also other parameter estimation techniques could be efficient. We choose to use direct search methods, [48], for a reduced-parameter search space since they are computationally efficient only if the number of variables is small and this is the case in our studies. Direct search methods are slow but less prone to stuck in a local minimum. They are global optimization schemes that are appropriate for functions with complicated dependencies like in the examples constructed in this thesis. Not for all the direct search methods there exists a convergence proof but many successful implementations encourage their use. They search through the parameter space along specified paths (called meshes or lattices) and are not based on gradients. The algorithm computes the objective function value for each point and compares it to the best solution so far. It redefines the mesh when a better solution is found or all the points in the current mesh have been visited. The direct search algorithms differ with respect to the way the mesh is being defined. The advantage of search methods is that they are relatively easy to implement compared to gradient-based techniques. They can also form a hybrid with gradient-based techniques, for example, as an initial-guess estimator.

In Section 5, a Latin hypercube search, [56], is applied. The algorithm samples points from permutations of a finite set of n consecutive natural numbers where n is the number of variables in the search space. The permutation values are perturbed slightly and scaled to fit the domain of interest. This way with every draw we obtain points that are nearly uniformly distributed over the feasibility interval. One iteration comprises of $15n$ draws, therefore, the more variables to be estimated, the slower the algorithm. In general, consecutive draws are not related to each other and hence the algorithm does not get stuck in local extrema. The disadvantage is that since the points have a prescribed form (almost uniformly distributed), the extremum might be impossible to find.

Chapter 4

Ensemble Multiscale Filter - EnMSF

This chapter describes the study that has been made on the ensemble multiscale filter for reservoir models. Section 4.1 presents one update time step on the EnMSF compared to the classical ensemble Kalman filter, without accounting for model dynamics. The EnMSF algorithm itself is described in Section 4.1.2. Section 4.2 includes the reservoir dynamics and details the study of the covariance matrix in EnMSF.

4.1 Multiscale ensemble filtering for measurement update[1]

4.1.1 Introduction

History matching (HM) is a process of adjusting the variables in a reservoir simulation model until it closely reproduces the past behavior of the reservoir. The accuracy of the history matching depends on the quality of the reservoir model and the quality and quantity of the data available. There are gradient based HM methods which require a minimization of a cost function over the entire time domain. In a real and large scale application it is an expensive procedure and it can be stuck in local minima. Due to the presence of uncertainties in both, the data and the model, it is hard and expensive.

One way to solve these problems is to use sequential data assimilation schemes (Kalman filtering). In the past years successful applications of Kalman filter theory were reported in many areas of research: the meteorological applications, [25], [12], nonlinear shallow-water storm-surge models, [88], atmospheric chemistry and transport modeling, [74], [89], [30].

The ensemble Kalman filter (EnKF) has also entered the world of reservoir engineering. Several publications have discussed the use of EnKF with oil reservoir models: [59], [60], [61], [28], [23], [50], [90], [77], showing promising results and at the same time indicating possible drawbacks.

The EnKF is based on the representation of the probability density of the state estimate by a finite number N (N being much smaller than the number of elements in the state vector) of randomly generated system states (ensemble members). This method falls

---

[1] This section is based on the article in Computational Geosciences, 13:245-254, 2009 where the coauthors are: R.G. Hanea, A.W. Heemink, D. McLaughlin.

into the Bayesian inversion approach and may provide a solution to the combined parameter and state estimation problem. The result is an ensemble of analyzed solutions (the combination between the measurements and the reservoir model) which approximates the posterior probability density function for the model parameters in the best way.

The ensemble size limits the number of degrees of freedom used to represent forecast and analysis errors. It makes the calculation of the error covariances practical for modest-sized ensembles. One important consequence of the use of small-sized ensembles is the sampling error problem. After a certain number of assimilation steps the ensemble loses its variance and leads to filter divergence. In [32] they conclude that the use of a small number of members in an ensemble often produces spuriously large magnitude background error covariances between greatly separated grid points (unphysical correlations). They noted that the EnKF analysis scheme could be improved by excluding observations at great distances from the grid point being analyzed by performing a covariance localization. Examples of this approach include methods based on covariance filtering with Schur products, [29], [33], and methods that perform updates in small blocks of grid cells ,[55], [65]. These methods improve computational efficiency and suppress the negative effect of sampling errors. The covariances that are used for localization will have an impact on the description of the physical correlation carried by the forecast covariance. Therefore, there is a risk of introducing correlations that are physically not possible. A number of researchers have observed and discussed the imbalances introduced by the localization schemes in the meteorological applications, [52], [57].

In this chapter we focus on data assimilation with the ensemble multiscale filter (EnMSF), [99], for estimation of oil reservoir permeability. This new approach solves some of the limitations of EnKF by allowing spatial localization.

Multiscale estimation is based on the concept of using a multiscale tree that describes the spatial correlations. The method is based on an algorithm, [92], inspired by image processing research. The degree of freedom to choose a certain tree and to set up the parameters for the update of the ensemble makes the method very appealing. At the same time, one should be aware of the strong dependence of the method performance on the choices mentioned above.

An interesting feature of the algorithm will be shown. An influence of its setup on the quality of the estimates in case of a reservoir engineering application will be investigated. Due to the complexity of the method we look only at a one time step update of the ensemble. In Section 4.1.2 the theoretical background for the EnKF and for the EnMSF is presented, and the assumptions that need to be made are described. In Section 4.1.3 a 2D, two-phase example is presented with seismic data. The numbering of the cells in the numerical grid is discussed. The conclusions follow in Section 4.1.4.

4.1.2 EnKF and EnMSF - theoretical background

The problem can be formulated as follows. There is noisy data at every point in the domain. Given the data we want to estimate what the true state is. In our case, the data and the state represent the same variable. There is no dynamic model involved. Nevertheless, we talk about forecasted and analyzed states to remain in the context of the sequential data assimilation. The forecasted state is simply the state without the measurements, and the analyzed state is the state where the measurements are already accounted for.

The state vector $\mathbf{x}$ needs to be defined and it is a collection of variables representing the model result. Superscripts for $\mathbf{x}$ are used in the equations: f representing the forecast state and a representing the analysis state.

All the available data are stored in vector $\mathbf{y}$. The way to compare the measured values with the state vector is to use a function from the state space to observation space called the observation operator $\mathbf{H}$:

$$\mathbf{y} = \mathbf{H}\mathbf{x}. \tag{4.1}$$

Through the observation operator $\mathbf{H}$, a forecast for the observed data can be made from the forecast of the state. Uncertainties in the measurements need to be specified as well. Therefore, Equation (4.1) modifies to:

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \boldsymbol{\nu}.$$

The observation operator $\mathbf{H}$ is often a collection of interpolation operators from the state to the observation points (conversions from state variables to the observed parameters); $\boldsymbol{\nu}$ is the observation noise, $\boldsymbol{\nu} \sim N(\mathbf{0}, \mathbf{R})$. The covariance matrix $\mathbf{R}$ needs to be specified.

Measurement update of the ensemble Kalman filter

The Ensemble Kalman filter was introduced by [16] and has been successfully used in many applications, [21], [32]. This Monte Carlo approach is based on the representation of the probability density of the state estimate by an ensemble of possible states, $\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^N$. Each ensemble member is assumed to be a single sample from a distribution of the true state. Whenever necessary, statistical moments are approximated with sample statistics.

- Forecasted state ensemble: $\mathbf{x}^{1,f},\ \mathbf{x}^{2,f},\ \dots,\ \mathbf{x}^{N,f}$.
- Analysis step:

$$\begin{aligned}
\bar{\mathbf{x}}^f &= \tfrac{1}{N}\textstyle\sum_{i=1}^{N} \mathbf{x}^{i,f},\\
\mathbf{E}^f &= [\mathbf{x}^{1,f} - \bar{\mathbf{x}}^f,\ \mathbf{x}^{2,f} - \bar{\mathbf{x}}^f,\ \dots,\ \mathbf{x}^{N,f} - \bar{\mathbf{x}}^f],\\
\mathbf{P}^f &= \tfrac{1}{N-1}\mathbf{E}^f(\mathbf{E}^f)^T,\\
\mathbf{K} &= \mathbf{P}^f\mathbf{H}^\top[\mathbf{H}\mathbf{P}^f\mathbf{H}^\top + \mathbf{R}]^{-1},\\
\mathbf{x}^{i,a} &= \mathbf{x}^{i,f} + \mathbf{K}[\mathbf{y} - \mathbf{H}(\mathbf{x}^{i,f}) + \boldsymbol{\nu}^i],
\end{aligned}$$

where $i = 1,\ \dots,\ N$. Here, $\boldsymbol{\nu}^i$ are realizations of the noise process $\boldsymbol{\nu}$.

Measurement update of the ensemble multiscale filter

The ensemble multiscale filter, [99], provides an alternative way to perform the update step. The original ensemble covariance is represented by a tree structure and physically long distance dependencies are kept through the relations between the tree nodes.

It consists of three basic steps:

1. assigning grid cells (pixels) to the finest scale nodes and computing the tree parameters from sample propagation through the tree (tree construction);
2. upward sweep (moving information upwards in the tree);
3. downward sweep (spreading information downwards in the tree to the finest scale).

The ensemble members are partitioned with respect to grid geometry and settings (like the pixel numbering and the tree specification). The multiscale algorithm places the partition at the finest scale nodes (leaf nodes) and computes the parameters at the upper tree nodes. Now, the upward and downward steps can be performed and the output is a set of updated replicates.

Example

Since the ensemble multiscale algorithm is more complex than EnKF, first, a little example is shown. The example greatly simplifies the method but allows to grasp the general idea.

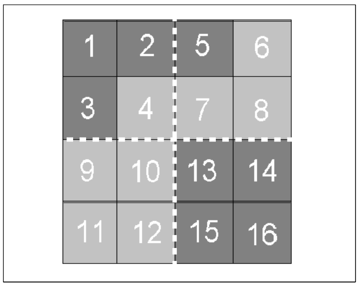


Figure 4.1: The initial grid division.

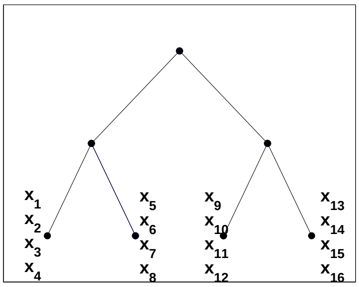

x_1 x_2 x_3 x_4
x_5 x_6 x_7 x_8
x_9 x_{10} x_{11} x_{12}
x_{13} x_{14} x_{15} x_{16}

Figure 4.2: Tree with pixel values assigned to the leaf nodes (fine scale nodes).

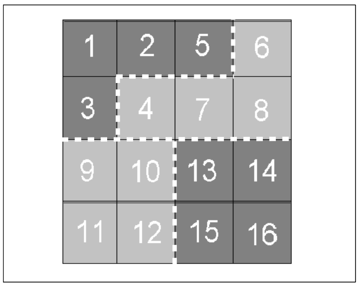


Figure 4.3: The middle scale representation on the grid.

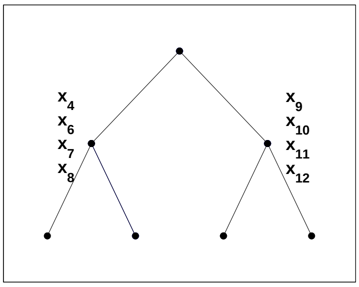


Figure 4.4: Pixels selected for the middle scale of the tree.

Each pair of Figures: 4.1-4.2, 4.3-4.4, 4.5-4.6, shows a grid and corresponding tree states. This is the first stage of the EnMSF - tree construction. The 4 × 4 grid is a representation of permeability where grey is high permeability and black is low.

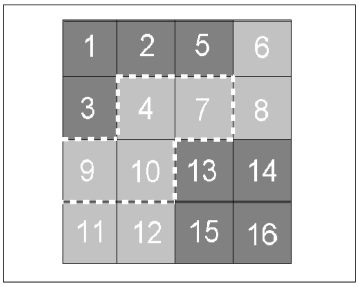


Figure 4.5: Root node representation on the grid.

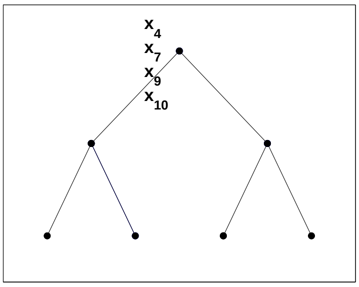


Figure 4.6: Pixels selected for the root (top) node of the tree.

The grid cells (pixels) are numbered and each group of four is assigned to a leaf node of a binary tree (Figures 4.1 and 4.2). Here it should be noted that having an ensemble of size N:

$$\begin{matrix} x_1^1 \\ x_2^1 \\ \vdots \\ x_{16}^1 \end{matrix}, \quad \begin{matrix} x_1^2 \\ x_2^2 \\ \vdots \\ x_{16}^2 \end{matrix}, \quad \ldots, \quad \begin{matrix} x_1^N \\ x_2^N \\ \vdots \\ x_{16}^N \end{matrix} \quad ,$$

the first leaf node, for example, contains a matrix with the first four states of each ensemble member:

$$\begin{matrix} x_1^1 \\ x_2^1 \\ x_3^1 \\ x_4^1 \end{matrix}, \quad \begin{matrix} x_1^2 \\ x_2^2 \\ x_3^2 \\ x_4^2 \end{matrix}, \quad \ldots, \quad \begin{matrix} x_1^N \\ x_2^N \\ x_3^N \\ x_4^N \end{matrix} \quad .$$

The state at each higher scale node is a linear combination of states at its direct children. At the middle scale four most influential states are kept at each of the two nodes (Figures 4.3 and 4.4). They happen to be the high permeability channel. These eight values are used to compute the four states at the root node (Figures 4.5 and 4.6) which is the center of the high permeability channel. This is the end of the tree construction part when all the nodes contain sets of parameters needed to perform the upward and downward sweeps.

Assume that a measurement is available in pixel no. 1. It is placed at the node which had pixel no. 1 assigned to it, the first leaf node (a circle in Figure 4.7). Going up the tree a Kalman-based update is performed and at the end the root node contains the knowledge from the measurement. Downward sweep (Figure 4.8) spreads the knowledge from the root node to all the other nodes. In consequence, the finest scale contains the analyzed states x_{i^a} (x_i^a), $i = 1, 2, ..., 16$.

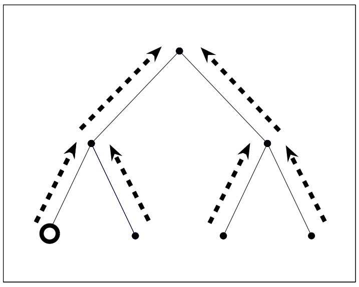


Figure 4.7: A scheme of the upward sweep.

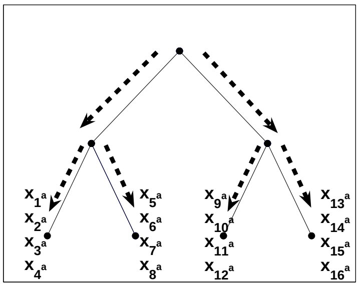


Figure 4.8: A scheme of the downward sweep and final updated values.

Clearly, the ensemble filter operates on an ensemble representing a distribution of the truth. For simplicity, the example shows one grid representation. It should be clear though that the states at the tree nodes come from the dependencies in the ensemble.

Some mathematics in the algorithm

The most complex is step 1 containing crucial assumptions and many flexible variables. Steps 2 and 3 are based on Kalman filter theory. Some mathematical details are presented here to enrich the simple example shown above. The full detailed description can be found in [99].

Some necessary notation is shown in Figure 4.9.

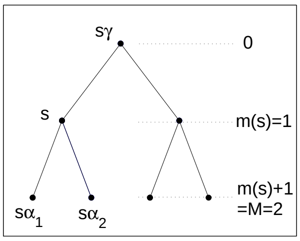


Figure 4.9: Notation: $s\alpha_i$ - the ith child of node s, $s\gamma$ - the parent of node s, $m(s)$ - the scale where s is placed, M - finest scale, 0 - the root node.

Additionally, some symbols used in the text are:

NOTATION

$\chi(s)$	State vector at node s.
$\chi_M(s)$	The vector of finest-scale states descended from s.
$\chi(s\|s)$	The state at node s after the upward sweep.
$\chi(s\|S)$	The state at node s after the downward sweep.
$\chi(s\gamma\|s)$	Projected state at node $s\gamma$.
^{j}	Superscript indicating an ensemble member.

Any other symbols are explained in the text.

The whole process starts with assigning the grid cells (pixels) to the leaf nodes of the tree. The cells can be numbered in various ways what determines the assignment. Two choices are shown in the next section. Assigned pixels provide states at the fine scale nodes of the tree.

A state at each non-fine-scale node s is a linear combination of the states at its children:

$$\chi(s) = V(s) \begin{bmatrix} \chi(s\alpha_1) \\ \vdots \\ \chi(s\alpha_q) \end{bmatrix}, \tag{4.2}$$

where matrices $V(s)$ are obtained as follows.

We search for a set of $V(s)$'s that provides *the scale-recursive Markov property* on the tree, i. e. decorrelates $q+1$ following sets of one scale: first q sets are all the children of the node s, and the set $q+1$ contains all the other nodes in this scale that are not children of s. The decorrelation is a minimization of conditional cross-covariances between the mentioned sets, given node s.

The tree that will approximate the forecast covariance matrix well should be based on *the scale-recursive Markov property.* The set of $V(s)$'s providing the scale-recursive Markov property perfectly would have a very high dimension since it would keep the total dependence between the finest states on the upper scale. Therefore, for practical purpose the state dimensions in coarser scales will be constrained. This is easier if $V(s)$'s are *block diagonal*; each block corresponds to a different and only one child of s.

The way $V(s)$'s are built

$V(s)$ has the form:

$$V(s) = diag[V_1(s), ..., V_q(s)],$$

where $V_i(s)$ is a matrix corresponding to the ith child of s, $s\alpha_i$, for $i = 1, ..., q$.

There are two constraints hidden here. The first one limits the number of rows in matrices $V_i(s)$ to $d_i(s)$. The second one, if necessary, coarsens the number of rows in $V(s)$.

Constructing matrices $V_i(s)$

To obtain $V_i(s)$'s, q conditional covariances need to be minimized, for each non-fine-scale node s. Those are the conditional cross-covariances between child i ($i = 1, ..., q$) and the rest of the nodes in the same scale, given the parent. Since direct minimization is inconvenient, the algorithm uses a *predictive efficiency method*.

Predictive efficiency method

The method is more efficient to compute than all the conditional cross-covariances. It picks $V_i(s)$'s which minimize the departure from optimality of the estimate:

$$\hat{z}_{ic}(s) = E[z_{ic}(s)|V_i(s)z_i(s)],$$

where $z_i(s)$ is a vector of states at node $s\alpha_i$ ($= \chi(s)$) and $z_{ic}(s)$ is a vector of states on all nodes at scale $m(s) + 1$ except node $s\alpha_i$. It was proved, [22], that they are given by the first $d_i(s)$ rows of:

$$V_i'(s) = U_i^T(s)Cov[z_i(s)]^{-1/2},$$

where $U_i(s)$ contains the column eigenvectors of:

$$Cov^{-1/2}[z_i(s)]Cov[z_i(s), z_{ic}(s)]Cov^T[z_i(s), z_{ic}(s)]Cov^{-T/2}[z_i(s)].$$

Here it should be noted that $d_i(s)$ are chosen by the user. The picked rows have the highest corresponding eigenvalues. The reason is that we assume that the column eigenvectors of $U_i(s)$ are lined in a decreasing (corresponding eigenvalue) order.

The size of $\chi(s)$ in Equation (4.2) is controlled by the setup of $V(s)$, that is, it was $V(s)$ that allowed keeping four states at the upper scale nodes in the example.

When all the states are computed and the measurements are placed at the tree nodes, the upward and downward sweeps can be carried out.

Going up the tree the algorithm updates the states at the nodes. Then each node s gets the value $\chi^j(s|s)$, where $\chi^j(s|s)$ is the state vector updated with all the measurements in the subtree rooted at s. At the top of the tree the value for the root node is obtained, $\chi^j(0|0)$. This is the basis to perform the downward sweep of the algorithm. $\chi^j(0|0)$ is the initial point, namely $\chi^j(0|S)$. Going down the tree the value $\chi^j(s|S)$ is assigned to each node s. That is the state value containing the knowledge from all given measurements. This way at the end of the sweep we get updated ensemble states at the finest scale which can be used to perform the next forecast step.

The equations leading the upward and downward sweeps are:

The upward sweep equation

$$\chi^j(s|s) = \chi^j(s) + K(s)[Y^j(s) - \hat{Y}^j(s)]$$

The states $\chi^j(s)$ at each node s are updated with perturbed measurements $Y^j(s)$ using weighting factor $K(s)$ and predicted measurements $\hat{Y}^j(s)$, and:

$$K(s) = \widehat{Cov}[\chi(s), \hat{Y}(s)][\widehat{Cov}[\hat{Y}(s)] + R(s)]^{-1},$$

$$\begin{cases} R(s) = r(s), & \text{m(s)=M;} \\ R(s) = diag[K(s\alpha_1)R(s\alpha_1)K^T(s\alpha_1), ..., K(s\alpha_q)R(s\alpha_q)K^T(s\alpha_q)], & \text{m(s)<M;} \end{cases}$$

$$\begin{cases} Y^j(s) = y(s) + e^j(s), & \text{m(s)=M;} \\ Y^j(s) = \begin{bmatrix} K(s\alpha_1)Y^j(s\alpha_1) \\ \vdots \\ K(s\alpha_q)Y^j(s\alpha_q) \\ y(s) + e^j(s) \end{bmatrix}, & \text{m(s)<M;} \end{cases}$$

$$\begin{cases} \hat{Y}^j(s) = h(s)\chi^j_M(s), & \text{m(s)=M;} \\ \hat{Y}^j(s) = \begin{bmatrix} K(s\alpha_1)\hat{Y}^j(s\alpha_1) \\ \vdots \\ K(s\alpha_q)\hat{Y}^j(s\alpha_q) \\ h(s)\chi^j_M(s) \end{bmatrix}, & \text{m(s)<M.} \end{cases}$$

The downward sweep equation

$$\chi^j(s|S) = \chi^j(s|s) + J(s)[\chi^j(s\gamma|S) - \chi^j(s\gamma|s)]$$

Previous states $\chi^j(s|s)$ at each node obtain the knowledge from all measurements through the weighting parameter $J(s)$:

$$\begin{aligned} J(s) &= \widehat{Cov}[\chi(s|s)]F^T(s)\widehat{Cov}^{-1}[\chi(s\gamma|s)], \\ F(s) &= \widehat{Cov}[\chi(s\gamma)]A(s)^T\widehat{Cov}^{-1}[\chi(s)], \\ A(s) &= \widehat{Cov}[\chi(s), \chi(s\gamma)]\widehat{Cov}^{-1}[\chi(s\gamma)]. \end{aligned}$$

The state $\chi^j(0|S) = \chi^j(0|0)$ is initially known from the upward sweep and projected replicates $\chi^j(s\gamma|s)$ can be computed based on matrices $V(s)$ and:

$$\chi^j(s\gamma|s) = F(s)\chi^j(s|s) + w'^j(s).$$

Matrix $F(s)$ is like above and $w'^j(s)$ is a zero-mean random perturbation with covariance $Q'(s)$:

$$Q'(s) = \widehat{Cov}[\chi(s\gamma)] - F(s)A(s)\widehat{Cov}[\chi(s\gamma)].$$

The whole procedure explained above, with the three steps, is able to approximate the forecast error covariance by constructing the tree and then to get the updated ensemble by moving up and down the tree assimilating the available measurements. At the end, the updated ensemble is obtained at the finest scale.

4.1.3 Application

In real-life applications the data are collected in the field; in theoretical applications (so-called twin experiments) the data is simulated with the aid of a setup that is called the truth or the reference case. A twin experiment is prepared here for the algorithm to check its performance. The results are compared to the ensemble Kalman filter's as it is described in [18]. All shown results are one update time results.

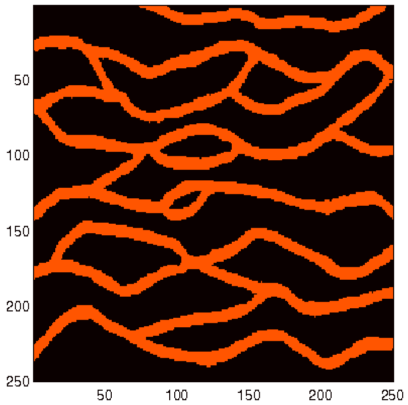


Figure 4.10: The training image 250×250.

Given the training image[2] (Figure 4.10) an ensemble was generated using SGeMS (The Stanford Geostatistical Modeling Software). Algorithm *snesim*, [10], generated 2D samples of permeability fields with grid size 64×64 from the training image with grid size 250×250. Each of 100 replicates is built of two values of permeability: high 10,000 mD (yellow) and low 500 mD (red). The first replicate was assumed to be the truth (Figure 4.11) and removed from the ensemble.

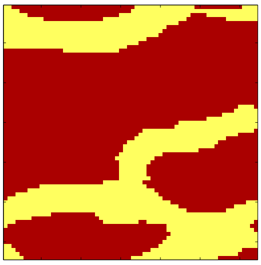


Figure 4.11: The true permeability 64×64.

The values of the observations are the perturbed values of the truth. It means that the permeability field is updated with permeability measurements. In practice these values

---

[2] A training image is an image representing the features and the distribution of ensemble members, [10].

cannot be measured anywhere except for at wells. Therefore, this example is not realistic but allows to test almost any possible setup.

Throughout the tests the tree is a quadtree (four children for every parent), there are 16 pixels assigned to each finest-scale node and 16 states preserved at coarser scale nodes.

The task is to assimilate large scale data. We assume it is possible to obtain the measurement in every pixel of the field, and that the data are very noisy. The number of data points in space is very large. It is known that EnKF is not an efficient tool to assimilate a very large amount of observations. The standard deviation of the measurement noise is, therefore, equal to a large value of 9. The data are shown in Figure 4.12.

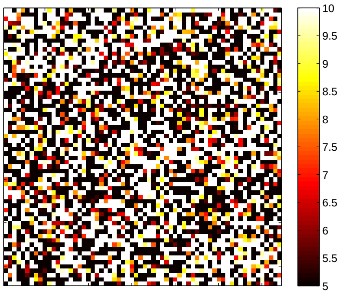


Figure 4.12: The permeability data.

The ensemble multiscale filter will be run twice. Each time with a different grid numbering. The numbering schemes are shown in Figures 4.13 and 4.14.

1	2	5	6	17	18	⋯
3	4	7	8	19	20	
9	10	13	14			
11	12	15	16			
⋮						

Figure 4.13: A square manner numbering of the pixels in the numerical grid.

1	2	3	4	5	6	⋯
65	66	67	68	69	70	⋯
129	130					
⋮						

Figure 4.14: A row wise numbering of the pixels in the numerical grid.

The numbering can express our belief in the dependencies in the actual field. The square manner numbering (Figure 4.13) keeps groups of pixels close in the grid close in the tree. It is not a perfect mapping though. For example, pixels 6 and 17 are direct neighbors but they are placed at different nodes.

The other approach (Figure 4.14) numbers the pixels row wise as if one believes that the channels are horizontal. It can be improved if there is some prior knowledge available, for example, about the channel placement.

It is visible in the results that interesting artifacts come from those two different approaches.

The plots of the prior, EnKF estimation and EnMSF with square and row wise numbering estimations are shown in Figures 4.15 - 4.18. The prior is relatively smooth and it is the best estimate if no data is given (the mean of the ensemble). Any proper assimilation should give an improvement to the prior which is the case in here.

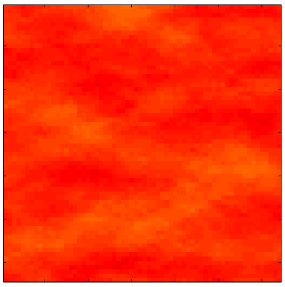


Figure 4.15: A mean of the ensemble members - the prior.

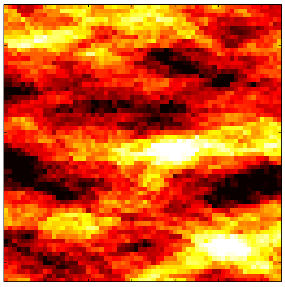


Figure 4.16: Assimilation with EnKF.

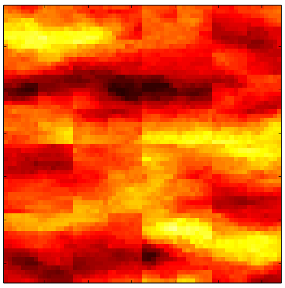


Figure 4.17: Assimilation with EnMSF and numbering scheme like in Figure 4.13.

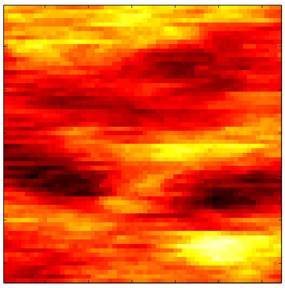


Figure 4.18: Assimilation with EnMSF and numbering scheme like in Figure 4.14.

The comparison of the performances is based on a root mean square error (RMSE) values and visual judgment. If two matrices of size $N \times M$ are $A_{N\times M} = \{a_{ij}\}$, $B_{N\times M} = \{b_{ij}\}$ then:

$$\mathrm{RMSE}(A, B) = \sqrt{\frac{1}{N \cdot M} \sum_{i=1}^{N} \sum_{j=1}^{M} (a_{ij} - b_{ij})^2}.$$

Table 4.1 contains RMSE between the truth and: the prior, EnKF, EnMSF + square numbering, EnMSF + row wise numbering.

The RMSE measures, roughly, the mean difference between respective pixels. It is a point not global measure, it cannot give information on large scale features. Additionally, one update step should not only rely on the RMSE. Hence, the visual comparison is also useful. It might suggest a need to search for a completely different measure of similarity.

Prior	EnKF	EnMSF+square	EnMSF+row wise
1.4002	1.3356	1.0795	1.0773

Table 4.1: RMSE between the truth and different results.

The plot of EnKF in Figure 4.16 is smooth and it seems like it sharpens a contrast in the prior. Its RMSE is not satisfactory either. The two versions of EnMSF (Figure 4.17 and Figure 4.18) show artifact, lines which come from the numbering schemes used. Nevertheless, the plots extract the high permeability channels quite well. The two approaches were going to show that EnMSF can be adjusted to a given problem, especially when some prior knowledge is available about the channel orientation or location.

4.1.4 Conclusions

The ensemble multiscale filter is a new technique for reservoir engineering. The method has been developed from image processing. The goal is to show an application of this filter to a simple reservoir engineering problem and to analyze its potential.

It is known that large data sets cause computational problems for Kalman filters. Also, with more data, the structure of the system noise is less important. Therefore, there is a need for efficient tools to handle this kind of applications.

Multiscale filtering is a way of representing the covariance matrix in the assimilation process by a tree structure. This simplification preserves the strongest correlations between the grid cells. The most complicated part of the method is the definition of the tree; it contains crucial assumptions and flexible parameters. There are features that influence filter's performance that can be adjusted to solve particular problems. Here, we focused on the numbering schemes which can represent our belief in the field dependencies. Certainly, it is very efficient to manipulate when some prior knowledge about the field is available.

The two numbering schemes shown represent different ideas. The first one, square like, might be universal to keep close pixels on the grid close in the tree. The second, row wise, can be suggested by horizontal flow information. Both schemes show good performance compared to EnKF in case of large data sets. The perfect mixture would be created when an approximate position of the channel was known. The shape or way of numbering could be adjusted to the feature.

Since the EnMSF is a complex and interesting algorithm it needs further experiments and investigation. Full runs with a reservoir simulator and more tests are required.

4.2 Towards the use of the ensemble multiscale filter for history matching[3]

4.2.1 Introduction

The EnMSF, [99], was introduced in Section 4.1.2 and here we present a sensitivity analysis of the EnMSF with respect to several algorithm parameters. We learnt that the numbering scheme has a strong influence on update performance with coarsened tree. We proceed with presenting the related covariance study and an interpolation problem with scarcer data set.

The numbering scheme is a feature that occurs to have the strongest impact on the filter's performance but additionally we decide to investigate the tree coarsening properties that have been kept default to that point. For several numbering schemes we manipulate algorithm coarsening parameters, namely, tree shape, upper-scale state coarsening (*cdim*), and the decorrelating neighborhood radius. First, the covariance study is shown that considers several parameter setups. Then, the full history matching experiments are presented where a reservoir simulator models the time change of the variables. Here, for the first time, the EnMSF is shown as a parameter estimator in a sequential updating scheme where the estimated and the observed variables are different physical notions. Now, a full state-space representation can be formulated following Section 3.4:

$$\begin{cases} \mathbf{m}_{k+1} = \mathbf{I}\mathbf{m}_k, \\ \\ \mathbf{y}_{k+1} = \mathbf{h}\left(\mathbf{f}_{k\to k+1}(\mathbf{m}_k, \mathbf{x}_k)\right) + \boldsymbol{\nu}_{k+1}. \end{cases}$$

A permeability parameter $\mathbf{m}$ is static, and the observations $\mathbf{y}$ are obtained through the observation function $\mathbf{h}$ extracting simulated data from the results of the reservoir simulator $\mathbf{f}$, where $\mathbf{x}$ represents the dynamic variables, grid-block pressure and saturation.

4.2.2 Pixel numbering scheme

We want to investigate the performance of the filter given several options for numbering the pixels in the grid and assigning them to the tree structure. Since the EnMSF extracts strong dependencies from pixels belonging to particular subtrees, it is expected that a different assignment pattern might modify the result.

The sensitivity analysis presents a comparison of covariances: the forecasted ensemble covariance and the covariance used in the EnMSF. The approximated covariance in the EnMSF will be called the tree covariance.

A tree applied in the EnMSF for this example is presented in Figure 4.19. There are four children per node and four pixels at each leaf node. Each middle scale node can keep up to 16 states, the root node can keep up to 64 states.

---

[3] This section is based on the proceedings of 11^{th} European Conference on the Mathematics of Oil Recovery - Bergen, Norway, 8-11 September 2008 where the coauthors are: R.G. Hanea, A.W. Heemink, D. McLaughlin, J.D. Jansen.

The true covariance between pixel coordinates (x_i, y_i) and (x_j, y_j) is given by:

$$C(i,j) = \exp\left(-\sqrt{\frac{(x_i - x_j)^2}{4^2} + \frac{(y_i - y_j)^2}{9^2}}\right),$$

for $i, j = 1, ..., 8$. The correlation is stronger along the x-axis. Figure 4.20 shows the sample cross-covariance between the pixel in the center (the lightest) and all the other pixels in the grid.

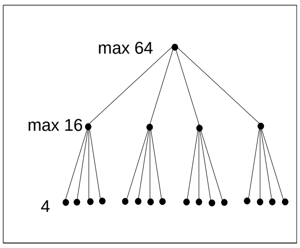


Figure 4.19: The tree for 8 × 8 example.

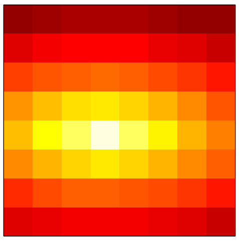


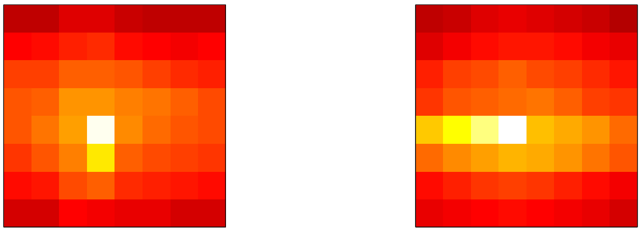
Figure 4.20: The prior sample cross-covariance.

Figure 4.21 shows four tree cross-covariances analogous to the one in Figure 4.20. All of them were computed using the same tree (Figure 4.19) and severe coarsening (there was only one state kept at each coarser scale node). Each covariance is coarsened in a different way depending on the numbering scheme. Vertical and horizontal numbering (4.21(a) and 4.21(b)) keep stronger correlations along the respective directions. Pixels numbered in the clusters like in Figure 4.1 preserve the correlation within the cluster, Figure 4.21(c). The last randomly numbered field (Figure 4.21(d)) is not able to show any structure.

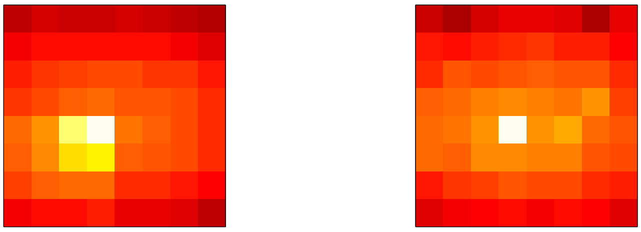
If all the states were kept at the higher scales of the tree, the covariance would be totally reconstructed independently of the numbering scheme. So, the numbering scheme gains importance only if a coarsening is applied. No prior information carried by the ensemble is cut off or prioritized by the filter if there is no truncation.

The coarsening can be advantageous if there is some additional knowledge about the grid. For example, that some areas are of a common nature which is not included in the prior ensemble.

To see the actual data assimilation results, applications are shown in the next section.



(a) Pixels numbered vertically. (b) Pixels numbered horizontally.



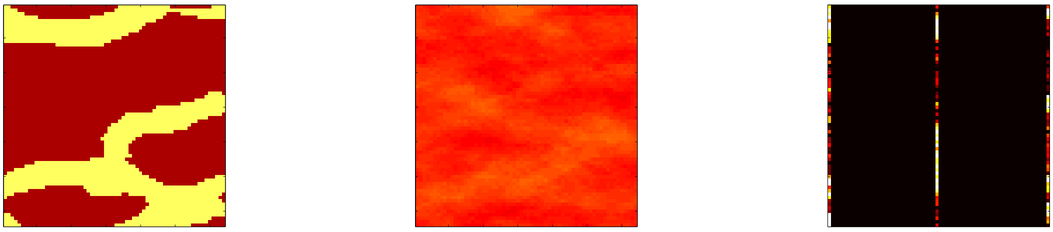
(c) Pixels numbered in groups. (d) Pixels numbered randomly.

Figure 4.21: The tree cross-covariance for severely coarsened tree and different numbering schemes.

4.2.3 Interpolation problem

The numbering schemes discussed in the previous section appeared to have a big impact on the covariance representation in the EnMSF. Here, a simple *one time* assimilation with a truncated tree is going to show how a coarsened tree covariance influences the result.

The true permeability field is given as in Figure 4.22(a), 64×64. The high permeability channels are light yellow and the low permeability background is dark red. A prior channelized ensemble with 100 members is given and its mean is shown in Figure 4.22(b). The correlation is stronger in horizontal direction since the channels are mostly horizontal. The permeability observations are taken along three vertical lines: one in the middle and two at the edges (Figure 4.22(c)). Since the permeability is a measured and estimated parameter, the problem is a simple interpolation problem.



(a) The true permeability.

(b) The prior of 100 permeability replicates.

(c) The measurements along the three vertical lines.

Figure 4.22: The setup of the interpolation problem.

The multiscale tree has four children per node, 16 pixels at the finest scale nodes and only 4 states kept at each coarser scale node. Four numbering schemes were applied and the EnMSF results are shown in Figure 4.23.

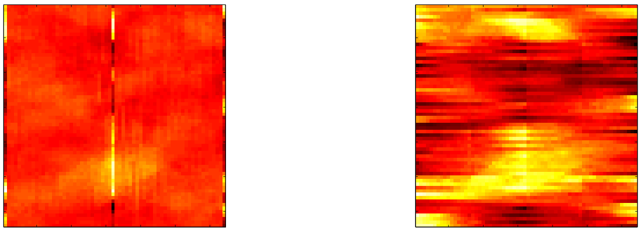
Since the state truncation was severe the results differ significantly. Vertical numbering (4.23(a)) assimilates the data only along the observation lines, it does not have a power to reach in the horizontal direction. When pixels are numbered along the rows (4.23(b)) the data is interpolated horizontally. If this numbering is additionally consistent with the channel orientation, then the assimilation is advantageous.

The clustered numbering (4.23(c)) spreads the data to a 'nearby' group of pixels. It is easy to notice that the pixels in line with the middle observations have been assigned to the tree together with their left neighbors; middle observations are not projected to the right plane of the grid.

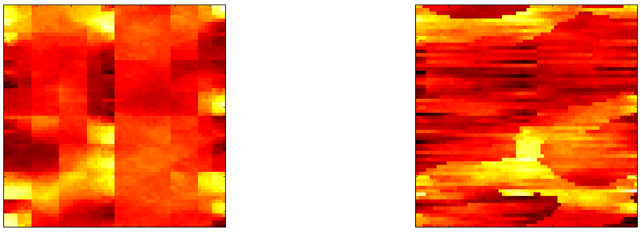
The last numbering scheme (4.23(d)) was based on the truth like in Figure 4.22(a). Low pixel numbers are assigned to the high permeability channels first and then the rest of the pixels is numbered row wise. Therefore, the numbering exactly mirrors the truth. The data should then be spread with respect to the true shapes. It can be handy when prior geological knowledge (not included in the initial ensemble) is available.

4.2.4 Coarsening parameters in the EnMSF

There are many parameters driving the EnMSF. These parameters need to be understood and studied to discover their influence, importance and sensitivity of the filter. Some



(a) Vertical numbering (b) Horizontal numbering.



(c) Pixels numbered in groups. (d) Numbering along the feature.

Figure 4.23: The EnMSF assimilation results with different numbering schemes.

conclusions will be drawn from studies done in this section. We want to investigate the tree shape, the state coarsening at the upper tree scales (*cdim*), and the conditioning neighborhood radius, and test the parameters versus different numbering schemes.

The research is based on the covariance matrix reconstruction. It is investigated how well the so-called tree covariance matrix represents the true and sample covariance matrices (the true covariance matrix might be given - as it is in this case - or computed from a very large sample; sample covariance matrix is a covariance of a given ensemble). The tree covariance matrix is a covariance matrix that is used by the EnMSF in an assimilation algorithm. It is computed on the basis of parameters assigned to the nodes of the tree after the tree construction step.

On the root (top) node of the tree an ensemble is sampled from a normal distribution with zero mean and the covariance matrix computed for that node in the tree construction step. The ensemble has as many members as there are replicates used initially for the tree construction. It is propagated to the finest scale nodes with the downward transition matrices attached to each node. The matrices come from the tree construction step as well. A covariance matrix of the ensemble from the finest scale is the tree covariance matrix.

In this experiment the true covariance matrix is again given by:

$$C(i,j) = \exp\left(-\sqrt{\frac{(x_i - x_j)^2}{4^2} + \frac{(y_i - y_j)^2}{9^2}}\right), \tag{4.3}$$

for $i, j = 1, ..., 8$. $C(i,j)$ means that the covariance is computed between the points (x_i, y_i) and (x_j, y_j) in the Cartesian coordinates. The grid size is 8×8. The tests are restricted to that size since it is computationally difficult to handle larger grids.

The covariance in Equation (4.3) has its denominators (4^2, 9^2) chosen such that the correlation of each grid block is longer in the horizontal than in the vertical direction. For illustration, the cross-covariance of one grid block with all the other grid blocks is presented. The true cross-covariance is shown in Figure 4.24 (consistently with further plots, it is the cross-covariance between a pixel in the 7th row, 3rd column with all the remaining pixels).

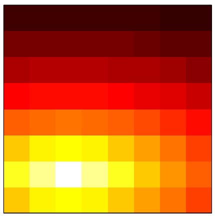


Figure 4.24: The true cross-covariance like in Equation (4.3).

The results were compared by computing root-mean-square errors (RMSE) between covariance matrices. If two matrices of size $N \times M$ are $A_{N\times M} = \{a_{ij}\}$, $B_{N\times M} = \{b_{ij}\}$ then again:

$$\mathrm{RMSE}(A,B) = \sqrt{\frac{1}{N\cdot M}\sum_{i=1}^{N}\sum_{j=1}^{M}(a_{ij}-b_{ij})^2}.$$

The RMSE values are multiplied by 10^3.

The tree covariance matrix has been compared to the true and sample covariance matrices. Since the difference between comparisons is negligible (not shown here), only the results against the true covariance are presented. Each simulation was run 1000 times. An average RMSE distance between a covariance matrix of a sample of 10, 100 replicates and the true covariance matrix is equal to 12.2478, 3.4772, respectively. It can be, therefore, assumed that a difference of around 10 should not be significant.

4.2.5 The design of the tree and its impact

The 64 pixels were distributed over the fine scale nodes of the tree in Figure 4.25 in groups of four. Each coarser scale node has four children (a quad tree). A number of states at a coarser scale node will be called *cdim*. The smaller its value the more coarsened the original sample covariance matrix. An additional coarsening factor is a neighborhood radius (in here equal to: 1, 2, 4, 8, 16, 64 as marked on the plots). It denotes a decorrelation radius length within one tree scale. The following results will show an impact of the two types of coarsening on a small (10 members) and on a large (100 members) ensemble.

Figures 4.27 and 4.28 picture the RMSEs for two different numbering schemes. Figure 4.27 had the numbering scheme adjusted to a feature and we will call it a 'channel' numbering scheme. If one expects there is a channel (like in Figure 4.26 in this example) running through the field, one could choose to first number the pixels horizontally inside the channel and then continue numbering outside it. That is how the numbering scheme for the runs in Figure 4.27 is constructed. A random numbering scheme was used for the results in Figure 4.28. Figures 4.27 and 4.28 contain six plots each, for six values

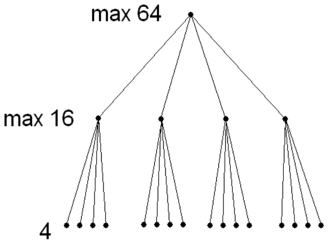


Figure 4.25: The quad tree used for the tests. There are 4 pixels in every finest scale node. It implies a maximum of 16 states at the nodes in the middle scale, and a maximum of 64 at the root node (depending on the middle scale).

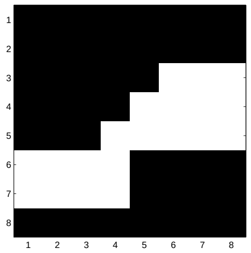


Figure 4.26: The assumed channel.

of *cdim*: 64, 32, 16, 8, 4, 1, placed in the reading order. Every plot is scaled to interval $[0, 65]$ in the x-axis (neighborhood radius) and to $[0, 220]$ in the y-axis (RMSE).

The algorithm while computing the states at coarser scale nodes takes the minimum between *cdim* and a total number of states at its children. The case $cdim = 64$ represents a full covariance matrix reconstruction and, therefore, is equivalent to an ensemble Kalman filter. The RMSE is very small, negligible, and the ensemble with 100 members (solid line) is more correct then the one with 10 (dashed line). The same observations apply to $cdim = 32$. Then, the states are coarsened only at the root node of the tree.

When $cdim = 16$, the RMSE is still low for both ensembles. For smaller neighborhoods (more severe coarsening) the larger ensemble performs worse. Still, the difference is not significant.

Interesting results are obtained for $cdim = 8$. For the channel numbering (Figure 4.27) the error increases mildly, especially for the large ensemble and the small neighborhoods. The random numbering scheme shows a significantly worse performance for the ensemble of 100. The remaining plots in Figure 4.28 show a similar trend of preferring the small ensemble results.

The channel numbering scheme shows a different performance. With $cdim = 4$ both curves, for 10 and 100 ensemble members, are close to each other even though for small neighborhoods the small ensemble performs better. The largest mismatch can be seen with $cdim = 1$. It can be expected since the coarsening is severe and a greater ensemble might introduce a larger misfit.

We will investigate what the impact of the coarsening is on the covariance matrix. Examples with the current and additional numbering schemes will be shown.

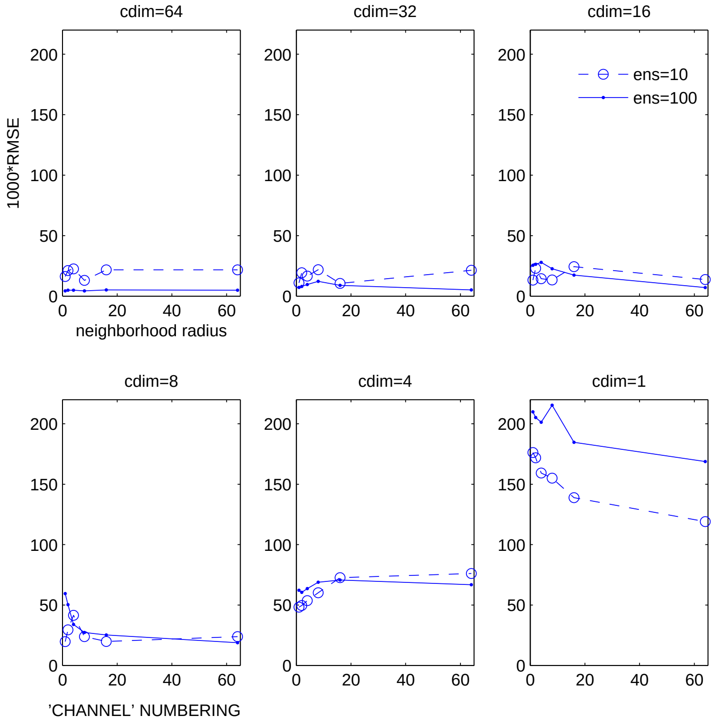


Figure 4.27: The plots of 10^3·RMSEs versus the neighborhood radius for different number of states kept at the coarser scales of the tree (*cdim*), '`channel`' numbering scheme.

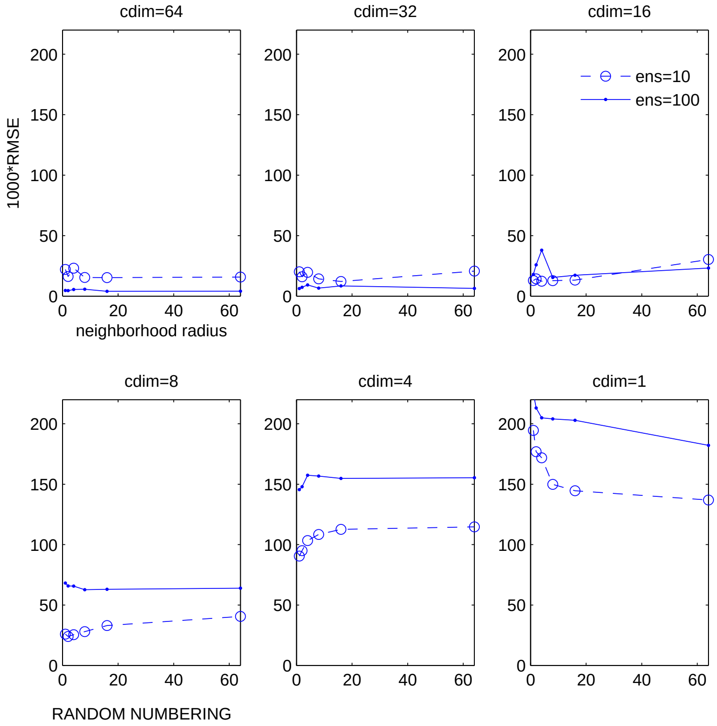
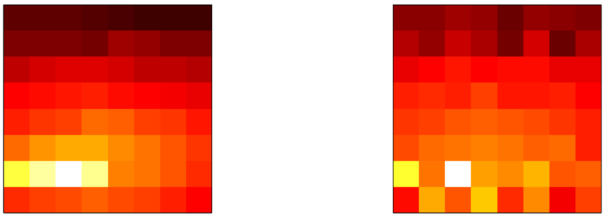


Figure 4.28: The plots of 10^3·RMSEs versus the neighborhood radius for different number of states kept at the coarser scales of the tree (*cdim*), `random` numbering scheme.

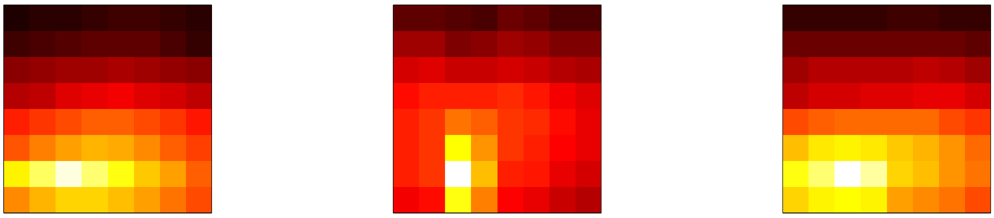
4.2.6 Covariance matrix approximation

For the EnMSF to be computationally feasible a truncation has to be implemented. It is due to the fact that a full tree structure requires storage of several large matrices at every node. Therefore, it needs to be reviewed how the truncation influences the covariance matrix and what the result of this coarsening is.

Having in mind the true cross-covariance from Figure 4.24, five truncated tree cross-covariances are shown in Figure 4.29. All the plots are kept in a convention used in Figure 4.24. Plots in Figure 4.29 were generated with 10 replicates, *cdim* and neighborhood radius equal to 4.



(a) 'Channel' numbering. (b) Random numbering.



(c) Row wise numbering. (d) Column wise numbering. (e) Square numbering (see Figure 4.1).

Figure 4.29: Cross-covariances for different numbering schemes.

Each cross-covariance shows a different pattern oriented with respect to the underlying numbering scheme. The numbering proves important and may dominate the initial dependence structure with significant truncation.

The ensemble multiscale filter occurs to be an interesting approach to sequential updating with build-in localization possibilities. Since reservoir engineering applications are not rich in measurements, the full potential of the method cannot be investigated. We conclude that the filter could be beneficial when used in a data-rich field.

Coming section presents the ensemble multiscale filter applied to a reservoir engineering problem versus the ensemble square root filter for comparison. It is presented in the light of this section's results.

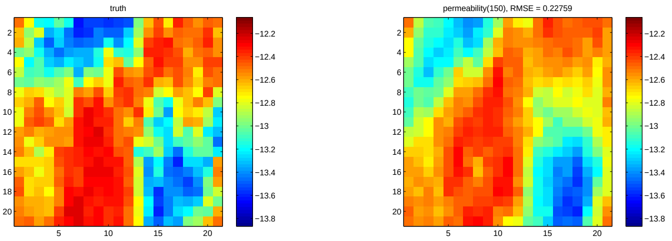
4.2.7 History matching using EnMSF

This section presents the results from different EnMSF runs in two problems where an ensemble square root filter is a benchmark.

The true permeability for the first simulation is shown in Figure 4.30(a). The pressure measurements were obtained from five wells: a center injector and four producers in the corners of the field.

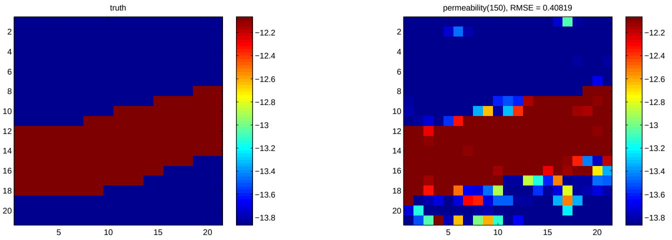
The true permeability for the second simulation is shown in Figure 4.30(c). The pressure measurements were obtained from the wells located along the left (injectors) and the right (producers) edge of the field, in total 42 wells. The domains' size is 21×21 cells.

In each example the data was collected once a month, 12 times. Both examples use 150 replicates of the permeability fields and a measurement error of $10^4 Pa$. A set of initial replicates for each example has characteristics similar to the corresponding truth.



(a) The truth (non-binary).

(b) EnSRF for truth 4.30(a), 150 repl., $RMSE = 0.22759$.



(c) The truth (binary).

(d) EnSRF for truth 4.30(c), 150 repl., $RMSE = 0.40819$.

Figure 4.30: The true permeability fields and EnSRF results for the two study cases.

On the right from the true fields in Figure 4.30 there are the results of assimilation using the EnSRF. The root mean square error is indicated. In both examples the filter performs very well.

The EnMSF uses a tree where each coarser scale node has three children and there are 49 pixels at each fine scale node. All updates are done on values of permeability after a log-transformation, [24].

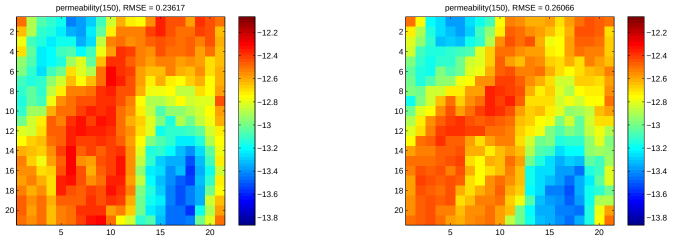
Let us first look at the results where *cdim* and the numbering scheme have been manipulated. Figures 4.31 and 4.32 contain several assimilation results using different settings in EnMSF. Both Figures are organized as follows:

- 4.31(a), 4.32(a) - no *cdim* truncation + feature based numbering templates,
- 4.31(b), 4.32(b) - $cdim = 20$ + feature based numbering templates,

- 4.31(c), 4.32(c) - $cdim = 20$ + numbering along columns,
- 4.31(d), 4.32(d) - $cdim = 1$ + feature based numbering templates.

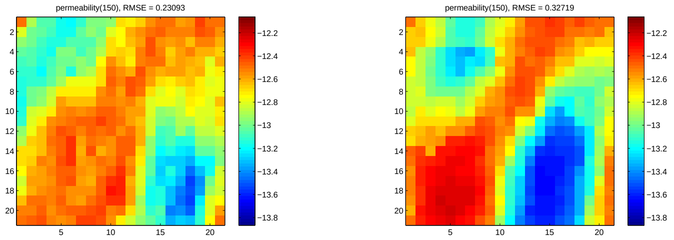
It will test what is the impact of the same coarsening operations on problems with different characteristics.

The feature based numbering template varies for each problem (it is constructed on the basis of the respective true permeability). The binary problem uses a template where the pixels are first numbered in the channel row wise from left to right, and then outside the channel. To construct a feature based numbering scheme for the non-binary problem, a threshold for permeability was set. Then, all the greater true permeability pixels take a value 'channel occurs' and the rest - 'no channel'. Now, the template can be built like for the digital case.



(a) No truncation, feature based numbering, $RMSE = 0.23617$.

(b) $cdim = 20$, feature based numbering, $RMSE = 0.26066$.



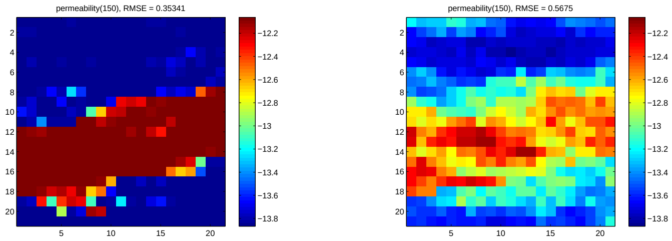
(c) $cdim = 20$, column wise numbering, $RMSE = 0.23093$.

(d) $cdim = 1$, feature based numbering, $RMSE = 0.32719$.

Figure 4.31: EnMSF for the non-binary truth 4.30(a).

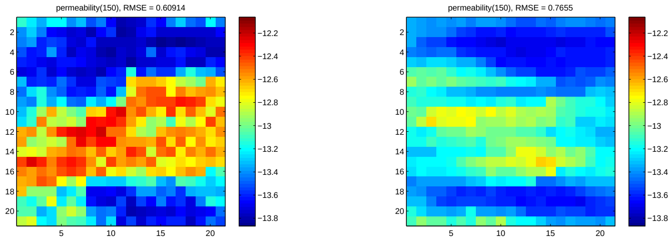
For both cases when the tree used no truncation (4.31(a), 4.32(a)) the results are almost identical to the EnSRF as expected. For example, in the non-binary case, permeability on the right boundary closer to the south-east well is smoothed unlike the true pattern. (Note: A numbering scheme for a non-truncated tree does not make any difference (no plots shown).) The truncation $cdim = 20$ using the template breaks this pattern, 4.31(b), making it more similar to the truth; the column wise numbering, 4.31(c), is not as good. Nevertheless, the results in general are very good. Additionally, the most severely coarsened case, 4.31(d) with $cdim = 1$, performs well. It might be due to strong correlations between the permeability and pressures in this case.

The binary case appears to be more sensitive to the truncation. In 4.32(b) and 4.32(c) the placement of the channel is approximately detected but the values of permeability are not correct. For $cdim = 1$, 4.32(d), it has only a vague recognition of the feature.



(a) No truncation, feature based numbering, $RMSE = 0.35341$.

(b) $cdim = 20$, feature based numbering, $RMSE = 0.5675$.



(c) $cdim = 20$, column wise numbering, $RMSE = 0.60914$.

(d) $cdim = 1$, feature based numbering, $RMSE = 0.7655$.

Figure 4.32: EnMSF for the binary truth 4.30(c).

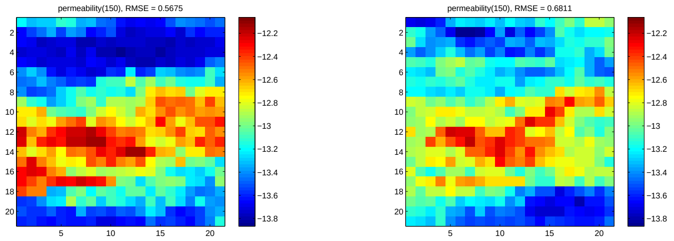
Since this example is larger, another tree could be constructed for comparison. Table 4.2 contains the tree parameters used for assimilations, results of which are Figures 4.33(a) and 4.33(b). Both trees coarsen the middle scale to about $\frac{1}{7}$th but the root nodes have different truncations applied. Both use the same feature based numbering scheme. Most likely the truncation at the root node caused the results to vary, and not the topology of the tree.

	4.33(a)	4.33(b)
$\sharp$ children	3	7
$\sharp$ pixels $\times$ $\sharp$ finest scale nodes $= 441$	49×9	9×49
($\sharp$ states $\times$ $\sharp$ nodes) at the middle scales	20×3	9×7
$\sharp$ states at the root node	20	9

Table 4.2: Tree parameters for two different cases.

Since the neighborhood radius is not significant (or expensive) when the tree is not truncated or is mildly truncated (not shown here), the neighborhood truncation to 1 with $cdim = 1$ is shown as an extreme case. It is not realistic to use these settings in any application but it is presented here for the sake of completeness.

The plots in Figure 4.34 were generated using $cdim = 1$. Figures 4.34(a) and 4.34(b)were generated without the neighborhood coarsening but use different numbering schemes.

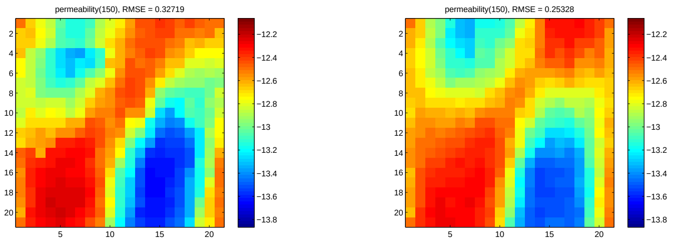


(a) Again Figure 4.32(b): *cdim* = 20, feature based numbering.

(b) *cdim* = 9, feature based numbering.

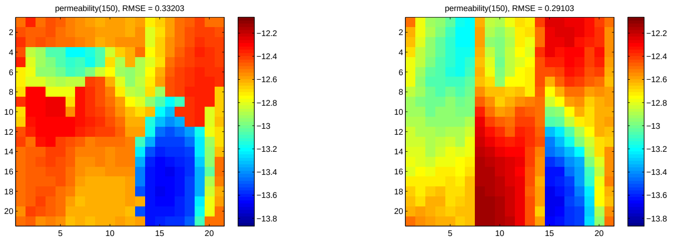
Figure 4.33: EnMSF with different trees, the details of the tree parameters are contained in Table 4.2.

Here, it is visible that the column wise numbering prefers the features in north-south direction. Figures 4.34(c) and 4.34(d) use the respective numbering schemes and additionally a severe neighborhood truncation was applied. The column numbering clearly indicates the division into three nodes of the tree.



(a) Again Figure 4.31(d): *cdim* = 1, full neighborhood, feature based numbering.

(b) *cdim* = 1, full neighborhood, column wise numbering.



(c) *cdim* = 1, neighborhood 1, feature based numbering.

(d) *cdim* = 1, neighborhood 1, column wise numbering.

Figure 4.34: EnMSF with the neighborhood coarsening.

4.3 Conclusions

Chapter 4 described comprehensive research results on the application of the ensemble multiscale filter. The filter is a Kalman type filter that builds the sample covariance matrix on the basis of a tree structure given an ensemble of realizations. After the tree is built, the measurements can be placed at the nodes and the update performed. Typically, a covariance derived from a sample contains spurious correlations due to a finite sample size, and even physically very distant areas can show dependencies. We investigated how ensemble multiscale filter deals with spurious correlations with built-in localization tools.

First, Section 4.1, presented the filter as an interpolation method (no time update) where very noisy measurements of the true domain were available together with an ensemble of initial realizations. We compared the ensemble multiscale filter with the classical ensemble Kalman filter that acted as a benchmark. We showed that the multiscale method performed updates differently, depending on the numbering scheme used, and extracted the information in varying ways.

In Section 4.2 the same setup of the interpolation problem but with different measurements was used. Here, less noisy measurements were available along three equally-spaced vertical lines. It was clearly visible how the ensemble multiscale filter propagates information to pixels that are close to each other in the tree which does not have to reflect physical closeness. We concluded that the numbering of the pixels is important and proceeded with more technical aspects of the investigated filter. We looked at the parameters of the tree that determine the level of the coarsening applied to the covariance matrix. Our conclusion was that the ensemble multiscale filter's performance will depend more on the level of coarsening (mostly parameter *cdim*) than the tree topology. More importantly, the coarsening with an appropriate numbering scheme can act as a localization method.

The filter can be computationally challenging due to its classical ensemble Kalman filter update scheme that is built into the algorithm. Then, if the tree has few children per node or many scales, the computation of the local covariance matrices or building the upward/downward connections might be expensive. Nevertheless, since the EnMSF never stores the full covariance matrix, it is expected to be more efficient than the traditional EnKF.

We recommend the EnMSF as a localization and update tool in several cases. First, in case where there is some knowledge available about local dependencies that will lead to an educated pixel-to-tree assignment procedure. Then, during the tree truncation important correlations will be extracted and kept in the update. Second, in case where there is a vast amount of spatially distributed data available that will need localization techniques due to a possible ensemble collapse. In any case, a truncation in the filter has to be implemented. Feature-based or correlation-based numbering is necessary. Additionally, for large data sets we expect the artefacts created by the tree structure to be less pronounced.

Chapter 5

Feature-based methods

In this chapter a method is developed for global feature deformation. This method is applied in reservoir engineering and groundwater modeling data assimilation. Combining knowledge from image processing and grid generation we come up with a grid deformation method parameterizing an image domain. First, some basic notions are defined. Then, the method is explained and applied to a 2D case. Finally, a straightforward extension to a 3D case is shown.

5.1 Grid distortion for a 2D reservoir model

5.1.1 Introduction

Data assimilation (or computer-assisted history matching) combines theoretical knowledge about a physical process with observed data. Different data assimilation techniques can be classified into two main groups: variational or sequential methods. Variational methods assimilate all the available data over the whole time interval at once through a minimization of an objective function. Sequential methods assimilate data at a particular time, proceed forward in time and assimilate the next available data; they can be derived by minimizing a variance estimate of a conditional probability density of a model given data. Both approaches have advantages and drawbacks, [52], [91].

A special case of data assimilation is parameter estimation problem where only static variables are estimated. Then, additionally, direct search methods (or zero order methods, [48]) can be implemented. These algorithms search through the objective function's domain of feasible solutions not taking into account local gradients. This methodology should be especially profitable in our application due to objective function's high nonlinearity.

Data assimilation is widely used in many branches of industry. It can be applied whenever it is possible to model the underlying physics of a process, and obtain theoretical estimates of variables that are measured in reality. Applications include weather prediction, ocean dynamics and hydrology, [88], [7], [16], or influenza spread, [39].

Reservoir engineering uses data assimilation to improve estimates of subsurface properties from available measurements, [62]. Typically, one wants to estimate a reservoir permeability or porosity field. The measurements can be spatially small-scale (like bottom hole pressures or fluid rates measured in the wells) or large-scale (field-wide seismic, electromagnetic or gravity observations).

Data assimilation for reservoir engineering needs to take into account requirements (constraints) that have to be met due to geological or economical reasons. Geological realism, [49], [10], is one of them. The parameter fields obtained from data assimilation should look geologically correct, that is, the initial subsurface characteristics should be preserved in the history matched estimate. The reason for this requirement is the generally accepted belief that 'geologically realistic' reservoir models have a larger probability to produce reliable forecasts than 'geologically unrealistic' models. Therefore, the main initial geological features should be displaced or bent but not broken. Nevertheless, the prior geological information is often lost, [31], due to neglect of higher order statistics in the data assimilation scheme. Furthermore, the geologically incorrect estimates are often still able to match the production history accurately. Therefore, additional constraints are needed to keep the geological information consistent during the history matching process.

Feature-based methods are used to overcome the problem of geologically unrealistic history matches. They account for shapes/features in a domain of interest and have been investigated from different points of view and for various problems. Here, we focus on reservoirs containing channels (high-permeable passages where liquids travel relatively easily).

The most intuitive approach to the task is to parameterize a channel. If the channel is simple enough, its length, width, starting point and orientation might provide a complete description of the domain, [97], [84]. This kind of parametrization limits the number of variables (degrees of freedom) and ensures a certain consistency of the structure.

Permeability might also be seen as an image to which different image processing tools can be applied. Several methods reviewed below have been proposed to solve the problem of feature estimation, very often restricted to estimation of a channelized field.

- A discrete cosine transform (DCT) originates from jpg file compression; it decomposes an image into a sum of products of basis cosine functions and corresponding DCT coefficients. The application of the DCT has been introduced to data assimilation in reservoir engineering in [34], [35], [36], and later also implemented in [94]. An efficient parametrization of a variable field is provided through DCT coefficient coarsening.

- A level-set method has been applied to a reservoir engineering data assimilation problem, [58], [14], [93], to update contour positions of features. The edges of the features are then modeled as a horizontal cross-section of a surface and an evolution of the level-set function modifies the shape position.

- Field alignment (FA), [67], changes an image by deforming its grid with a vector field. The deforming vector field is regularized by gradient and divergence constraints in an objective function.

The main focus in this chapter is a grid distortion method. It was inspired by the grid generation research described in [82], [81], [80], where the goal is to automatically generate smooth grids for solving differential equations. The method deforms a grid smoothly (like FA) to fit it to given data but does it through a simplified (compared to FA) partial differential equation. The approach to the problem resembles FA, as both methods use deforming vector fields and were created to allow relocation of patterns and their possible deformation. However, unlike FA, grid distortion uses a limited number of parameters. Grid distortion is implemented here within the ensemble Kalman filter framework and also within a pattern search method.

All the methods mentioned above are feature-based methods. Since the field alignment and grid distortion methods are closely related, FA is discussed in more detail in Section 5.1.3. Optimization methods, in particular the ensemble Kalman filter and the pattern search method, are described in Section 5.1.4. Section 5.1.5 introduces the feature-based method developed in this thesis - grid distortion. Results follow in Section 5.1.6, discussion and conclusions in Section 5.1.7.

5.1.2 Basic notions

Vector fields

We want to discuss images and their deformations caused by vector fields, and there are several ways of looking at this type of problems. One can think of a vector field whose domain are the nodes of a regular grid holding the pixels. Then, the deformation of the grid triggers pixel deformation which in turn leads to the image deformation. An interpolation to a regular pixel values is then necessary. This type of warping is shown in Figure 5.1 and was implemented for the grid distortion method.

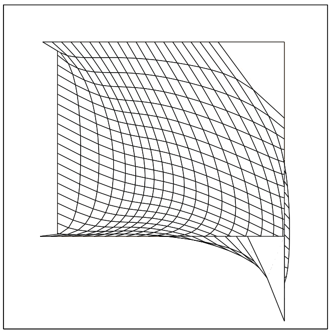


Figure 5.1: A node displacing field (applied in the grid distortion method).

Another approach is demonstrated in Figure 5.2. Here, every pixel contains a vector. The vector field is a search field since it seeks new values for the locations to obtain a deformation of a background image. The location can be a pixel (like in the figure) or a grid node. Alternatively, multiple vectors can be attached at a single pixel location to propagate the pixel's value to several destinations (Figure 5.3). Variations of the approaches could also be considered.

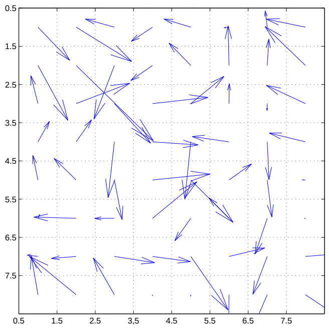


Figure 5.2: A search vector field that can be placed at pixels.

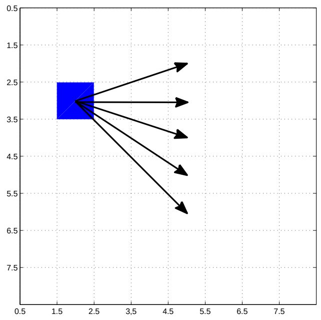


Figure 5.3: An example of a mapping that is not one-to-one.

Interpolation

In grid distortion method an interpolation has to be implemented to obtain the values of pixels in a regular Cartesian grid after the deformation. The simulator in this case cannot handle irregular unstructured grids. Various types of interpolation methods exist. The basic interpolants are: nearest-neighbor, linear, polynomial and spline.

For the 2D grid distortion in Section 5.1, an additional interpolation function is written. It prescribes values to the grid nodes and maps them onto regular pixels after the distortion. In the case when a pixel has not received a value from a node, a weighted mean of neighboring pixel values is assigned. Since the realizations in our applications are binary, there is a threshold value specified.

For the 3D grid distortion in Section 5.2, the nearest-neighbor interpolation is implemented. Even though it makes the problem more non-smooth, it keeps the realizations looking like geological features which is desirable in the presented examples.

Problem formulation

Let us formulate a data assimilation problem in the form of minimization problem where the objective function J is

$$J(\mathbf{q}) = (\mathbf{f}^* - \mathbf{f}(\mathbf{q}))^T(\mathbf{f}^* - \mathbf{f}(\mathbf{q})).$$

J is a function of a vector field $\mathbf{q}$ and contains only a squared measurement mismatch term in the simplest form. The actual data are represented as a vector $\mathbf{f}^*$, $\mathbf{f}(\mathbf{q})$ is a vector of predicted data.

The connection between the data and the vector field, i.e. function $\mathbf{f}$, can have different forms. If one wants to solve an image recognition problem, $\mathbf{f}(\mathbf{q})$ represents the distorted image values at pixel locations, [95]. It is the simplest application one can start with since it does not include any aditional transformation or time.

Complexity can be added by making f a function of a parameter field (for example, a pressure response in a steady-state system to a permeability field). One step further is a data assimilation application where observations are collected over time.

5.1.3 Field Alignment

Field alignment, [67], is a method developed for image deformation purposes. It can be used for image recognition, [95], or feature-based data assimilation.

Let us have a discrete Cartesian grid (ξ, η), Figure 5.4. Each grid node j has a displacement vector (all vectors are column vectors) $\underline{\mathbf{q}}_j$ with coordinates $[q_j^1, q_j^2]^T$, $j = 1, \ldots, N$. A collection of the vectors for all the grid nodes j gives a discrete vector field $\mathbf{q}$:

$$\mathbf{q} = \begin{bmatrix} \underline{\mathbf{q}}_1 \\ \underline{\mathbf{q}}_2 \\ \vdots \\ \underline{\mathbf{q}}_N \end{bmatrix}.$$

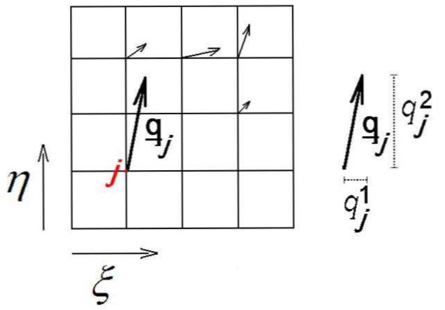


Figure 5.4: Cartesian grid (ξ, η) with displacement vector $\underline{\mathbf{q}}_j = [q_j^1, q_j^2]^T$ at node j.

One wants to minimize an objective function containing an observation mismatch and a background mismatch:

$$J(\mathbf{x}(\mathbf{q})) = \frac{1}{2}(\mathbf{y} - \mathbf{h}(\mathbf{x}(\mathbf{q})))^T \mathbf{R}^{-1}(\mathbf{y} - \mathbf{h}(\mathbf{x}(\mathbf{q}))) + \frac{1}{2}(\mathbf{x}(\mathbf{q}) - \mathbf{x}^b)^T \mathbf{B}^{-1}(\mathbf{x}(\mathbf{q}) - \mathbf{x}^b),$$

where $\mathbf{x}^b$ is a background image and $\mathbf{B}$ is a background error covariance matrix. The measurements are denoted as the vector $\mathbf{y}$, $\mathbf{x}(\mathbf{q})$ is a distorted template (base case) image, $\mathbf{h}(\mathbf{x}(\mathbf{q}))$ are predicted measurements that depend implicitly on the distorting vector field $\mathbf{q}$, $\mathbf{R}$ is a measurement error covariance matrix of the zero-mean noise term $\boldsymbol{\nu}$ such that:

$$\mathbf{y} = \mathbf{h}(\mathbf{x}(\mathbf{q})) + \boldsymbol{\nu}.$$

In case of an image recognition application, the observations $\mathbf{y}$ are the target image and $\mathbf{h}$ is an identity. If the method is used for data assimilation in a physical process then a dynamic model needs to be included. In that case $\mathbf{h}(\mathbf{x}(\mathbf{q}))$ includes a physical model that predicts the values of observations.

To regularize the vector field $\mathbf{q}$ two quadratic penalty terms can be added to the objective function. The first one constrains gradients of the vector field, the second one its divergence. Constraining the gradients will not allow sharp jumps from one vector to its direct neighbor. Minimizing divergence should ensure no excessive local expansion/contraction of the image. The regularization terms are of the form, [67]:

$$\begin{aligned} L(\mathbf{q}) = & \\ & \frac{w_1}{2} \sum_{j=1}^{N} tr \left\{ \left[\frac{\partial \underline{\mathbf{q}}_j}{\partial(\xi, \eta)} \right] \left[\frac{\partial \underline{\mathbf{q}}_j}{\partial(\xi, \eta)} \right]^T \right\} \\ & + \frac{w_2}{2} \sum_{j=1}^{N} [\nabla \cdot \underline{\mathbf{q}}_j]^2. \end{aligned}$$

Here, w_1 and w_2 are weights, summation is done over all grid nodes indexed by j, tr is the trace of a matrix, $\cdot$ denotes the inner product, and

$$\frac{\partial \underline{\mathbf{q}}_j}{\partial(\xi, \eta)} = \begin{bmatrix} \frac{\partial q_j^1}{\partial \xi} & \frac{\partial q_j^1}{\partial \eta} \\ \frac{\partial q_j^2}{\partial \xi} & \frac{\partial q_j^2}{\partial \eta} \end{bmatrix}, \quad \nabla \cdot \underline{\mathbf{q}}_j = \frac{\partial q_j^1}{\partial \xi} + \frac{\partial q_j^2}{\partial \eta}.$$

represent the gradients and the divergence, respectively. Using finite difference approximation it is easy to derive a fully discrete version of these expressions that comes into the objective function as:

$$J(\mathbf{x}(\mathbf{q})) = \frac{1}{2}(\mathbf{y}-\mathbf{h}(\mathbf{x}(\mathbf{q})))^T\mathbf{R}^{-1}(\mathbf{y}-\mathbf{h}(\mathbf{x}(\mathbf{q})))+\frac{1}{2}(\mathbf{x}(\mathbf{q})-\mathbf{x}^b)^T\mathbf{B}^{-1}(\mathbf{x}(\mathbf{q})-\mathbf{x}^b)+L(\mathbf{q}). \quad (5.1)$$

Equation (5.1) can be solved through a so-called *two-step* method, [67], by computing gradients of J with respect to $\mathbf{q}$ and $\mathbf{x}(\mathbf{q})$ and setting them equal to zero. The first step of the method aligns features by deforming the grid. The second step introduces an amplitude adjustment, in other words a pixel-based update on the aligned field.

Even though the *two-step* approach can be implemented as an ensemble scheme in history matching, it needs a model derivative if used with a physical simulation model, [67]. This could be a major obstacle to implementing the field alignment method. Therefore, gradient-free approaches are chosen for the implementation of the grid distortion method.

5.1.4 Ensemble sequential data assimilation and direct search

A sequential data assimilation scheme and a pattern search method are applied in this work due to a relatively simple implementation process that does not require a derivative specification. This section describes the problem formulation, the ensemble sequential approach and the pattern search.

Most data assimilation formulations are based on a state-space approach. Let $\mathbf{x}_k$ be a state vector of dynamic variables. Then:

$$\begin{cases} \mathbf{x}_{k+1} = \mathbf{f}_{k\to k+1}(\mathbf{m}, \mathbf{x}_k) + \boldsymbol{\varepsilon}_{k+1}, \\ \mathbf{y}_{k+1} = \mathbf{h}(\mathbf{x}_{k+1}) + \boldsymbol{\nu}_{k+1}, \end{cases} \quad (5.2)$$

where subscript k indicates discrete time, $\mathbf{f}_{k\to k+1}$ is a model for time-evolution that depends on the static parameters $\mathbf{m}$, and $\boldsymbol{\varepsilon}$ is a model error term. Initial condition $\mathbf{x}_0$ and a set of initial parameters are given, $\mathbf{y}_k$ indicates the measurements, $\mathbf{h}$ is the measurement operator that could depend on time, and $\boldsymbol{\nu}$ is a zero mean normally distributed random variable representing noise with error covariance matrix $\mathbf{R}$.

Let the state vector contain dynamic variables and/or static model parameters. Assume that the model error can be ignored, that is, $\boldsymbol{\varepsilon}_{k+1} = \mathbf{0}$ for all k, and that one is interested in estimating the static variables $\mathbf{m}$ only. Then the state-space approach modifies to:

$$\begin{cases} \mathbf{m}_{k+1} = \mathbf{I}\mathbf{m}_k, \\ \mathbf{y}_{k+1} = \mathbf{h}(\mathbf{f}_{k\to k+1}(\mathbf{m}_k, \mathbf{x}_k)) + \boldsymbol{\nu}_{k+1}. \end{cases} \quad (5.3)$$

If the static variable can be represented by a smaller set of parameters $\boldsymbol{\alpha}$, $\mathbf{m} = \mathbf{m}(\boldsymbol{\alpha})$, then:

$$\begin{cases} \boldsymbol{\alpha}_{k+1} = \mathbf{I}\boldsymbol{\alpha}_k, \\ \mathbf{y}_{k+1} = \mathbf{h}(\mathbf{f}_{k\to k+1}(\mathbf{m}(\boldsymbol{\alpha}_k), \mathbf{x}_k)) + \boldsymbol{\nu}_{k+1}. \end{cases} \quad (5.4)$$

Here, $\mathbf{I}$ is an identity matrix that indicates that the state is constant in time, and $\mathbf{h}(\mathbf{f}_{k\to k+1}(\mathbf{m}(\boldsymbol{\alpha}_k), \mathbf{x}_k))$ denotes a series of operations: first the model static variables are computed from a given parameter set, then the model $\mathbf{f}_{k\to k+1}$ propagates the dynamic variables in time, and finally $\mathbf{h}$ extracts the observations. Formulation (5.4) will further be applied in our study.

The main idea behind the ensemble Kalman filter is to represent a probability distribution of state variables by an ensemble. The ensemble is a collection of possible realizations (replicates) of the variables in the data assimilation process and is used to compute a sample (cross-)covariance matrix. Sequentially, first the process is integrated in time, then, the data are assimilated, and updated replicates are forwarded in time again. Various ensemble Kalman filter flavors can be found in the literature, [1], [20]. The ensemble square root filter (EnSRF) implementation, [18], [70], is used due to its computational efficiency. In the current setup the EnSRF becomes a parameters estimator since all the dynamic variables have been excluded from the state vector.

For comparison, a pattern search (Latin Hypercube Sampling) technique is implemented (see Matlab documentation) that is especially suitable for estimating a small number of variables. It is not based on local gradient changes (hence the straightforward implementation) but usually requires a considerable time to find a minimum.

5.1.5 Grid distortion

The notion of a grid is ubiquitous in mathematics and engineering. Any equation used for modeling must first be discretized on a numerical grid. In more advanced reservoir simulators flow equations can be solved on various types of grids: Cartesian, unstructured, etc. A grid can also be deformed to fit a given shape, like in [80] where the goal is to generate a smooth 'orthogonal' grid for engine combustion chamber simulation (Figure 5.5).

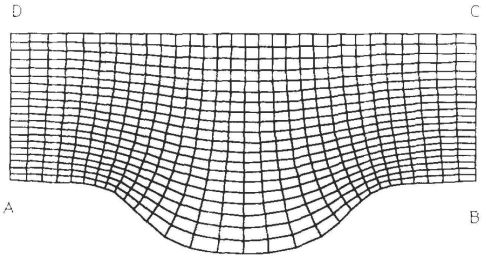
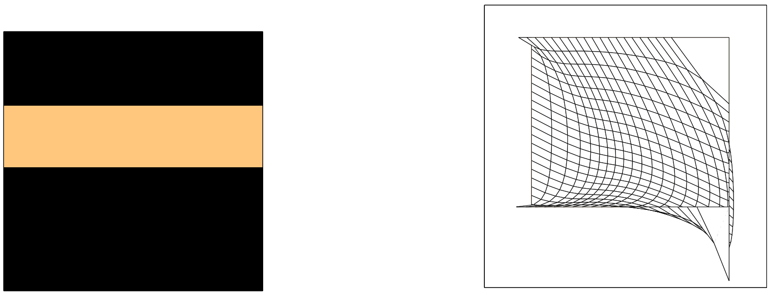


Figure 5.5: Fitting a grid to a shape of an engine combustion chamber, [80].

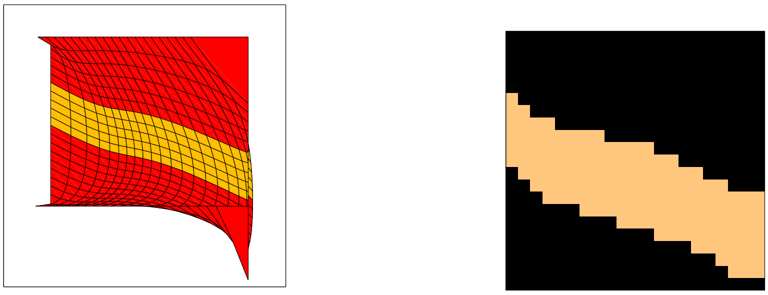
Similarly, in reservoir engineering applications, [76], a grid can be adjusted to fit the top and bottom reservoir horizons, or specify the position of a fault, [75].

Looking at a digital image, we actually look at a set of pixels. This set of pixels can be seen as embedded in (held by) a grid and the image could be deformed by perturbing the grid. Figure 5.6 shows an example of a simple image deformation by changes in the grid. Figure 5.6(a) is an image to be deformed (which will be referred to as a *base case*). Figure 5.6(b) is a deformed grid. Figure 5.6(c) is a deformed grid with the deformed image on top of it. Finally, Figure 5.6(d) represents the deformed image mapped back onto the original Cartesian grid. This backtransformation is necessary because our simulator requires a Cartesian grid. The base case is an image generally representing one of the features that is expected to be seen in the feature field. To allow independent feature



(a) Image to be deformed.

(b) Distorted grid.



(c) Distorted grid with deformed image.

(d) Deformed image mapped back onto the original undistorted Cartesian grid.

Figure 5.6: An image deformed by a grid.

distortion, each shape has one base case representing it. The base cases are not subject to estimation procedures and are set constant beforehand.

The aim of the deformation is to translate the grid smoothly to assure a seamless transition of features. The grid then behaves like an elastic net, and deformations do not corrupt the shapes. Below, the grid distortion method is described in detail. First, the motivation and the relevant equations are presented. Then, grid distortion in the context of data assimilation is described.

Motivation

Grid generation is a well researched area, [82], where the task is to automatically generate curvilinear grids for solving differential equations on variously shaped domains and with specified boundary requirements, for which it is advantageous to have orthogonal smooth grids. First, we present several grid generation equations in increasing order of complexity, and then we compare the grid distortion method to the field alignment method.

Let us consider two-dimensional grids where ξ and η are Cartesian coordinates, and $\mathcal{X}$ and $\mathcal{Y}$ are curvilinear coordinates. The same type of reasoning would apply to a three-dimensional grid. In 2D Laplace's equations are:

$$\Delta\mathcal{X} = \frac{\partial^2\mathcal{X}}{\partial\xi^2} + \frac{\partial^2\mathcal{X}}{\partial\eta^2} = 0,$$

$$\Delta\mathcal{Y} = \frac{\partial^2\mathcal{Y}}{\partial\xi^2} + \frac{\partial^2\mathcal{Y}}{\partial\eta^2} = 0,$$

where boundary conditions are given and the equation solves for smooth grid coordinate lines that become orthogonal away from the boundaries. The boundary conditions can be Dirichlet (in which case the lines are fixed at the boundary and their angle can vary) or Neumann (in which case the lines move along the boundary and have a specified angle). The boundary conditions can also be mixed, but it is not possible to specify both the location and the angle.

Poisson's equation,

$$\Delta\mathcal{X} = P,$$

$$\Delta\mathcal{Y} = Q,$$

is used to control the coordinate line spacing through the control (or distortion) functions P and Q, with the same rules for the boundary conditions. In [81] P and Q are, for example, specified in the form of exponentials to create points of attraction in the curvilinear or Cartesian coordinates. There, it is also noted that a modification to Poisson's equation can be used in the form of a general diffusion equation:

$$\nabla\cdot(\mathcal{K}\ \nabla\mathcal{X}) = 0,$$

$$\nabla\cdot(\mathcal{K}\ \nabla\mathcal{Y}) = 0,$$

where $\mathcal{K}$ is responsible for controlling the coordinates. The equations give:

$$\Delta\mathcal{X} = -\frac{1}{\mathcal{K}}(\nabla\mathcal{K}\cdot\nabla\mathcal{X}), \tag{5.5}$$

$$\Delta \mathcal{Y} = -\frac{1}{\mathcal{K}}(\nabla \mathcal{K} \cdot \nabla \mathcal{Y}),$$

that is

$$\frac{\partial^2 \mathcal{X}}{\partial \xi^2} + \frac{\partial^2 \mathcal{X}}{\partial \eta^2} = -\frac{1}{\mathcal{K}}\left(\frac{\partial \mathcal{K}}{\partial \xi}\frac{\partial \mathcal{X}}{\partial \xi} + \frac{\partial \mathcal{K}}{\partial \eta}\frac{\partial \mathcal{X}}{\partial \eta}\right),$$

$$\frac{\partial^2 \mathcal{Y}}{\partial \xi^2} + \frac{\partial^2 \mathcal{Y}}{\partial \eta^2} = -\frac{1}{\mathcal{K}}\left(\frac{\partial \mathcal{K}}{\partial \xi}\frac{\partial \mathcal{Y}}{\partial \xi} + \frac{\partial \mathcal{K}}{\partial \eta}\frac{\partial \mathcal{Y}}{\partial \eta}\right).$$

The right hand side is determined by $\mathcal{K}$, the slopes of $\mathcal{K}$ and the slopes of the coordinates.

In [80] a covariant Laplace operator was proposed:

$$\frac{\partial}{\partial \xi}\left(\mathcal{K}\frac{\partial \mathcal{X}}{\partial \xi}\right) + \frac{\partial}{\partial \eta}\left(\frac{1}{\mathcal{K}}\frac{\partial \mathcal{X}}{\partial \eta}\right) = 0,$$

$$\frac{\partial}{\partial \xi}\left(\mathcal{K}\frac{\partial \mathcal{Y}}{\partial \xi}\right) + \frac{\partial}{\partial \eta}\left(\frac{1}{\mathcal{K}}\frac{\partial \mathcal{Y}}{\partial \eta}\right) = 0,$$

where $\mathcal{K}$ is given and depends on the magnitudes of the slopes of $\mathcal{X}$ and $\mathcal{Y}$. The equations can be rewritten as:

$$\frac{\partial^2 \mathcal{X}}{\partial \xi^2} + \frac{1}{\mathcal{K}^2}\frac{\partial^2 \mathcal{X}}{\partial \eta^2} = -\frac{1}{\mathcal{K}}\left(-\frac{1}{\mathcal{K}^2}\frac{\partial \mathcal{K}}{\partial \xi}\frac{\partial \mathcal{X}}{\partial \xi} + \frac{\partial \mathcal{K}}{\partial \eta}\frac{\partial \mathcal{X}}{\partial \eta}\right),$$

$$\frac{\partial^2 \mathcal{Y}}{\partial \xi^2} + \frac{1}{\mathcal{K}^2}\frac{\partial^2 \mathcal{Y}}{\partial \eta^2} = -\frac{1}{\mathcal{K}}\left(-\frac{1}{\mathcal{K}^2}\frac{\partial \mathcal{K}}{\partial \xi}\frac{\partial \mathcal{Y}}{\partial \xi} + \frac{\partial \mathcal{K}}{\partial \eta}\frac{\partial \mathcal{Y}}{\partial \eta}\right).$$

Here, the left hand side of the equations is not the Laplacian. It might be more difficult to implement but is certainly an interesting alternative to consider.

We want to show how the grid generation methodology is similar to field alignment in image processing. Note that $\mathcal{X} = \xi + q^1_{[j]}$, where $q^1_{[j]}$ are x-coordinates of a distorting vector field $\mathbf{q}$ as defined for the field alignment method in the previous section ($q^2_{[j]}$ will denote y-coordinates). This gives $\frac{\partial^2 \mathcal{X}}{\partial \xi^2} = \frac{\partial^2 q^1_{[j]}}{\partial \xi^2}$, $\frac{\partial^2 \mathcal{X}}{\partial \eta^2} = \frac{\partial^2 q^1_{[j]}}{\partial \eta^2}$ and therefore $\Delta \mathcal{X} = \Delta q^1_{[j]}$. Note, however, that the boundary conditions for these problems differ. We will show how a Poisson's-type equation arises naturally in the field alignment context as described in [67]. Recall the objective function from Equation (5.1):

$$J(\mathbf{x}(\mathbf{q})) = \frac{1}{2}(\mathbf{y} - \mathbf{h}(\mathbf{x}(\mathbf{q})))^T \mathbf{R}^{-1}(\mathbf{y} - \mathbf{h}(\mathbf{x}(\mathbf{q}))) + L(\mathbf{q}),$$

where the background mismatch term is ignored. Let the function $L(\mathbf{q})$ be rewritten as:

$$L(\mathbf{q}) = \\ \frac{w_1}{2}\sum_{j=1}^{N} tr\left\{\left[\frac{\partial \underline{\mathbf{q}}_j}{\partial(\xi,\eta)}\right]\left[\frac{\partial \underline{\mathbf{q}}_j}{\partial(\xi,\eta)}\right]^T\right\} + \frac{w_2}{2}\sum_{j=1}^{N}[\nabla \cdot \underline{\mathbf{q}}_j]^2 = \\ \frac{1}{2}\iint\left[\left(\frac{\partial q^1_{[j]}}{\partial \xi}\right)^2 + \left(\frac{\partial q^1_{[j]}}{\partial \eta}\right)^2 + \left(\frac{\partial q^2_{[j]}}{\partial \xi}\right)^2 + \left(\frac{\partial q^2_{[j]}}{\partial \eta}\right)^2\right] d\xi d\eta,$$

where we take $w_1 = 1$ and $w_2 = 0$ for simplicity. This gives

$$\begin{aligned} &J(\mathbf{x}(\mathbf{q})) = \\ &\quad \frac{1}{2}(\mathbf{y} - \mathbf{h}(\mathbf{x}(\mathbf{q})))^T \mathbf{R}^{-1}(\mathbf{y} - \mathbf{h}(\mathbf{x}(\mathbf{q}))) + \\ &\quad \frac{1}{2}\iint \left[\left(\frac{\partial q^1_{[j]}}{\partial \xi}\right)^2 + \left(\frac{\partial q^1_{[j]}}{\partial \eta}\right)^2 + \left(\frac{\partial q^2_{[j]}}{\partial \xi}\right)^2 + \left(\frac{\partial q^2_{[j]}}{\partial \eta}\right)^2 \right] d\xi d\eta \end{aligned}$$

and hence:

$$\frac{\partial J}{\partial (q^1_{[j]})} = -\left(\frac{\partial \mathbf{x}}{\partial (q^1_{[j]})}\right)^T \left(\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right)^T \mathbf{R}^{-1}(\mathbf{y} - \mathbf{h}(\mathbf{x}(\mathbf{q}))) + \frac{\partial^2 q^1_{[j]}}{\partial \xi^2} + \frac{\partial^2 q^1_{[j]}}{\partial \eta^2},$$

$$\frac{\partial J}{\partial (q^2_{[j]})} = -\left(\frac{\partial \mathbf{x}}{\partial (q^2_{[j]})}\right)^T \left(\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right)^T \mathbf{R}^{-1}(\mathbf{y} - \mathbf{h}(\mathbf{x}(\mathbf{q}))) + \frac{\partial^2 q^2_{[j]}}{\partial \xi^2} + \frac{\partial^2 q^2_{[j]}}{\partial \eta^2}.$$

Letting the derivatives of the objective function be equal to zero leads to:

$$\Delta q^1_{[j]} = \left(\frac{\partial \mathbf{x}}{\partial (q^1_{[j]})}\right)^T \left(\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right)^T \mathbf{R}^{-1}(\mathbf{y} - \mathbf{h}(\mathbf{x}(\mathbf{q}))),$$

$$\Delta q^2_{[j]} = \left(\frac{\partial \mathbf{x}}{\partial (q^2_{[j]})}\right)^T \left(\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right)^T \mathbf{R}^{-1}(\mathbf{y} - \mathbf{h}(\mathbf{x}(\mathbf{q}))).$$

Following [67], $\frac{\partial \mathbf{x}}{\partial (q^1_{[j]})}$ and $\frac{\partial \mathbf{x}}{\partial (q^2_{[j]})}$ comprise the gradient $\nabla \mathbf{x}$ of $\mathbf{x}$, and finally we get:

$$\Delta \mathbf{q} = (\nabla \mathbf{x})^T \left(\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right)^T \mathbf{R}^{-1}(\mathbf{y} - \mathbf{h}(\mathbf{x}(\mathbf{q}))). \tag{5.6}$$

Equations (5.5) and (5.6) are not Poisson's equations since their right-hand sides depend on the unknown variable. Nevertheless, in [67] it is shown that Equation (5.6) can be solved iteratively as Poisson's equation holding the right-hand side fixed to the value from the previous iteration.

We want to simplify the solution process and aim to find a right-hand side of Poisson's equation in a form that is easy to parameterize.

Grid distortion equations

Let again $\mathcal{X}$ and $\mathcal{Y}$ be curvilinear coordinates, and ξ and η Cartesian coordinates. Then a distorted grid is a solution for $\mathcal{X}$ and $\mathcal{Y}$ to the equations

$$\frac{\partial^2 \mathcal{X}}{\partial \xi^2} + \frac{\partial^2 \mathcal{X}}{\partial \eta^2} = P, \tag{5.7}$$

$$\frac{\partial^2 \mathcal{Y}}{\partial \xi^2} + \frac{\partial^2 \mathcal{Y}}{\partial \eta^2} = Q. \tag{5.8}$$

Functions P and Q on the right hand sides of the equations are called distortion functions (control functions) and they drive the coordinate transformation. They represent the type of smooth deformation one wants to achieve through the coordinate transformation. The distortion functions are parameterized to reduce the number of variables to be estimated.

If Equations (5.7) and (5.8) are discretized on a regular Cartesian grid, distortion functions become distortion matrices (referred to as $\mathbf{P}$ and $\mathbf{Q}$). The second order derivatives are approximated by

$$\frac{\partial^2 \mathcal{X}}{\partial \xi^2} = \frac{\mathcal{X}(i+\tau, j) - 2\mathcal{X}(i,j) + \mathcal{X}(i-\tau, j)}{\tau^2},$$

where $\tau = 1$ since integer grid coordinates are used and (i, j) travels through all the grid nodes. The other terms in (5.7) and (5.8) are discretized in a similar fashion.

Consider an image of size $(N_\eta + 1) \times (N_\xi + 1)$ pixels in η- and ξ-axis direction, respectively. There are $(N_\eta + 2) \times (N_\xi + 2)$ grid nodes in the grid representing the given image. Let us assume that the grid at its boundaries is fixed and that one needs to find the curvilinear coordinates of the remaining $N_\eta \times N_\xi$ nodes. Equation (5.7) can be expressed in the form:

$$\mathbf{A}\boldsymbol{\mathcal{X}} = \mathbf{b}_x, \tag{5.9}$$

where $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{(N_\eta \cdot N_\xi)\times 1}$ is the unknown column vector of curvilinear coordinates $\mathcal{X}$. The origin is chosen at the lower left corner of the node domain. The fixed boundary values are: zeros at the left edge, $N_\xi + 1$ at the right edge, while at the top and bottom the values increase from left to right from zero to $N_\xi + 1$ with increment one. Let us define a matrix $\mathbf{B}_x$ that accounts for the boundary condition:

$$\mathbf{B}_x = \begin{pmatrix} 1 & 2 & \cdots & N_\xi - 1 & 2N_\xi + 1 \\ 0 & 0 & \cdots & 0 & N_\xi + 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & N_\xi + 1 \\ 1 & 2 & \cdots & N_\xi - 1 & 2N_\xi + 1 \end{pmatrix}.$$

Matrix $\mathbf{B}_x$ is of the same size as the distortion matrix $\mathbf{P}$: $N_\eta \times N_\xi$. Let us define $\underline{\mathbf{P}}$ and $\underline{\mathbf{B}_x}$ to be column vectors of size $(N_\eta \cdot N_\xi) \times 1$ reshaped from matrices $\mathbf{P}$ and $\mathbf{B}_x$, respectively. During the discretization the fixed boundary values are moved to the right-hand side of the equation where one gets $\mathbf{b}_x = \underline{\mathbf{P}} - \underline{\mathbf{B}_x}$. At the left-hand side $\mathbf{A}$ is a sparse $(N_\eta \cdot N_\xi) \times (N_\eta \cdot N_\xi)$ pentadiagonal finite-difference matrix.

For Equation (5.8) the only difference is the boundary condition. We want to solve

$$\mathbf{A}\boldsymbol{\mathcal{Y}} = \mathbf{b}_y. \tag{5.10}$$

Let $\underline{\mathbf{Q}}$ be a column vector reshaped from the distortion matrix $\mathbf{Q}$. Let $\underline{\mathbf{B}_y}$ be a column vector reshaped from $N_\eta \times N_\xi$ matrix:

$$\mathbf{B}_y = \begin{pmatrix} 2N_\eta + 1 & N_\eta + 1 & \cdots & N_\eta + 1 & 2N_\eta + 1 \\ N_\eta - 1 & 0 & \cdots & 0 & N_\eta - 1 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ 2 & 0 & \cdots & 0 & 2 \\ 1 & 0 & \cdots & 0 & 1 \end{pmatrix}.$$

Then $\mathbf{b}_y = \underline{\mathbf{Q}} - \underline{\mathbf{B}_y}$.

Grid Distortion in Data Assimilation

The ensemble Kalman filter is directly applicable to reservoir engineering problems in the form of Equations (5.3). Typically, the ensemble of states contains uncertain static variables like permeability. Each cell value in the permeability field can be estimated in the data assimilation process (here referred to as pixel-based estimation). Unfortunately, in reservoir engineering applications there are few data available. There might be only several measurements available to estimate hundreds or thousands of state values, which makes the problem severely ill-posed.

Additionally, Kalman filters consider only first and second moments of the given distribution. This results in smooth estimates that might not always be desirable. One might want to account for specific features introduced in the initial permeability ensemble which the filter loses during the history match. Both problems (ill-posedness and not-preserved initial information) are approached by the feature-based methods.

Let us explain how grid distortion fits into the sequential ensemble data assimilation framework. The state vector in Equation (5.4) contains the distortion parameters. Uncertainty is assumed in the initial estimate of the distortion parameters only. Operator $\mathbf{m}(\alpha)$ represents a sequence of operations that create a permeability field from the distortion parameters. To compute predicted measurements, several steps are required. First, the distortion parameters ($\boldsymbol{\alpha}$) need to be converted to distortion matrices $\mathbf{P}$ and $\mathbf{Q}$. The equations are solved for curvilinear coordinates, base cases are distorted and mapped back to a Cartesian grid. The margins from grid embedding need to be cut off and the results from different base cases need to be merged. The resulting images of permeability are the input for the reservoir simulator $\mathbf{f}$. Operator $\mathbf{h}$ extracts the measurement predictions for the next update time step. The measurements are contaminated with noise. The distortion parameters are updated every measurement time step to create an updated ensemble of channel fields. The base cases do not change throughout the whole process.

To generate an ensemble of possible permeability realizations, the grid distortion with randomly generated coefficients is repeatedly applied to the base cases. Figure 5.7 represents the replicate generation process for one base case. On top is the base case which, together with the distortion parameters, creates realizations of fields with a horizontal channel.

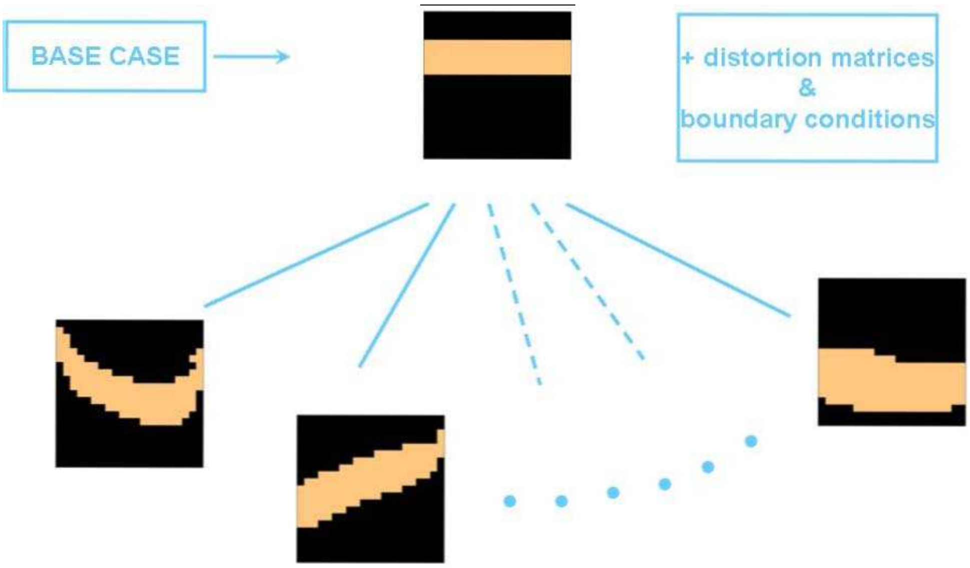


Figure 5.7: Generation of replicates with the grid distortion method.

The full sequential data assimilation algorithm is described below.

Algorithm 1 - Sequential ensemble data assimilation for parameter estimation in grid distortion

- Initialize - load measurements, initial distortion parameters $\boldsymbol{\alpha}$ for all ensemble members, and N_f base cases.

Repeat for the total number of measurement times, for all ensemble members:

1. create distortion matrices $\mathbf{P}(\boldsymbol{\alpha})$, $\mathbf{Q}(\boldsymbol{\alpha})$,
2. solve Equations (5.9) and (5.10) for $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{Y}}$ for each of the N_f base cases,
3. distort the base cases with computed distorted grids,
4. map distorted images to Cartesian grid,
5. cut off the margins from embedding,
6. merge the results from different base cases,
7. simulate the forward reservoir model with the new permeability fields and get predicted data at the next measurement time, Equation (5.4),
8. assimilate data to the parameters (using, for example, EnSRF).

Additionally to pixel- and feature-based sequential data assimilation, a pattern search method was implemented. Pattern search requires a definition of an objective function and an initial condition. The objective function is a weighted measurement mismatch over the whole time interval, the initial condition is zero for all the parameters. The base case is identical to the one used in the sequential method. The pattern search scheme is presented below. Note that points 1-6 are identical with Algorithm 1.

Algorithm 2 - Pattern search for grid distortion

- Initialize - load measurements, initial distortion parameters $\boldsymbol{\alpha}$ and N_f base cases.

Repeat until stopping criteria have been met:

1. create distortion matrices $\mathbf{P}(\boldsymbol{\alpha})$, $\mathbf{Q}(\boldsymbol{\alpha})$,
2. solve Equations (5.9) and (5.10) for $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{Y}}$ for each of the N_f base cases,
3. distort the base cases with computed distorted grids,
4. map distorted images to Cartesian grid,
5. cut off the margins from embedding,
6. merge the results from different base cases,
7. simulate the forward reservoir model with the new permeability field and get predicted data at all measurement times, Equation (5.4),
8. compute the objective function value.

Pattern search has been applied only for the grid distortion parametrization since it cannot handle as many variables as there are in the pixel domain. Therefore, it is presented in the grid distortion result section.

5.1.6 Twin Experiment

The section is organized as follows. First, the study case is described. Then, the data assimilation results are presented in two subsections following first the pixel-based approach and then the feature-based approach.

As an example we consider a domain with N_f features. Then, the grid distortion needs to be able to handle N_f separate characteristics. Each feature is distorted by Poisson's equations with its own set of distortion functions P and Q. From the image processing and grid generation literature we conclude that we can construct (and parameterize) the distortion functions such that they reflect the type of deformation of the underlying features. We want the deformation to be global and smooth since small local changes might be insignificant for the given type of data. We choose each distortion function, P and Q in Equations (5.7) and (5.8), to be an independent paraboloid $a(\xi - c)^2 + b(\eta - d)^2$, where ξ and η are Cartesian coordinates, a and b are independent normally distributed parameters with zero mean and standard deviation $6 \cdot 10^{-5}$, and c and d (deviations from the center of the domain) are independent normally distributed with zero mean and standard deviation 10. Higher standard deviations for the parameters led to shapes being removed from the domain. We want to mention here that the search interval in the pattern search method for the distortion parameters is set to $[-33 \cdot 10^{-5}, 33 \cdot 10^{-5}]$ for

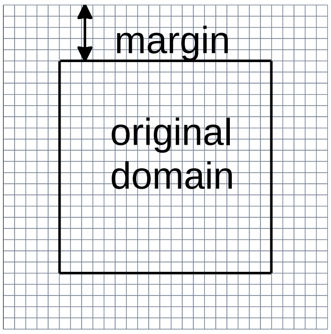


Figure 5.8: Grid embedding provides flexibility at the boundaries of the original grid domain.

a's and b's, and $[-33, 33]$ for c's and d's. The intervals are wider than twice the standard deviation that was used to generate the replicates in the initial ensemble.

Paraboloids are simple enough to reflect only curving and smooth bending which should be able to fit the position of the feature. We can think of it as a local shape fitting problem, a part of a larger domain. For larger domains more elaborate distortion functions should be used, for example higher order polynomials or spline interpolation.

The initial grid is embedded in a larger grid (outer grid) with a given margin equal to N_m pixels in each direction, Figure 5.8. This outer grid is fixed at its boundaries. It provides flexibility at the edges of the original grid and does not introduce any more parameters (the margin width is fixed).

Since there are N_f features (e.g., channels) to be estimated, each one of them is modeled separately. That is, there are N_f base cases, each assigned to model one of the channels with a separate set of parameters. That gives in total only $8 \cdot N_f$ parameters coming from the distortion functions that are estimated in the data assimilation process in Equation (5.4).

The realizations of each of the N_f features are merged before being used as input to a reservoir simulator. That is, the shapes are gathered in one domain and given an arbitrary inside value, while the background value is also prescribed; note that overlapping features do not add up their values.

Data assimilation for physical processes always requires a model/simulator of the underlying phenomenon. An in-house reservoir simulator, [37], is used for the results in this section. It is based on mass balance equations and a two-phase version of Darcy's law for slightly compressible two-phase (oil-water) flow, neglecting gravity and capillary pressure effects. The background of reservoir simulation can be found in, e.g., [63], [5].

An oil saturated 2D reservoir is waterflooded and produced for 24 months, water breakthrough occurs later. The wells operate on a fixed rate (0.001 m^3/s for all wells). Porosity is assumed known, constant and equal to 0.3. All the other reservoir and fluid properties can be found in Table 5.1.

The domain size is 49×49 pixels. Bottom hole pressure in Pa is measured in the six wells displayed in Figure 5.9, where the true permeability is shown. Low log10-permeability (black) is equal to -13.5 and high is equal to -12.06 m^2. The truth was generated with the help of the *snesim* algorithm, [79]. The six measurement points in space provide data every month for two years. The noise standard deviation is assumed to

Variable	**Value**	**SI units**
Gridblock height	2	m
Gridblock length/width	1500/49	m
Oil dynamic viscosity	$0.5 \cdot 10^{-3}$	$Pa \cdot s$
Water dynamic viscosity	$1.0 \cdot 10^{-3}$	$Pa \cdot s$
Oil compressibility	$1.0 \cdot 10^{-9}$	Pa^{-1}
Rock compressibility	$1.0 \cdot 10^{-9}$	Pa^{-1}
Water compressibility	$1.0 \cdot 10^{-9}$	Pa^{-1}
Initial reservoir pressure	$3 \cdot 10^{7}$	Pa
Endpoint relative permeability of oil	0.9	–
Endpoint relative permeability of water	0.6	–
Corey exponent, oil	2.0	–
Corey exponent, water	2.0	–
Residual oil saturation	0.2	–
Connate water saturation	0.2	–
Porosity	0.3	–
Well bore radius for all wells	$4.5 \cdot 0.0254$	m

Table 5.1: Reservoir and fluid properties for the twin experiment.

be $7 \cdot 10^6$ Pa. The ensemble contains $n_e = 50$ replicates. Their prior mean and first nine prior replicates are shown in Figures 5.10 and 5.11, respectively. The prior ensemble was generated with the grid distortion method and is used in both pixel- and feature-based sequential methods.

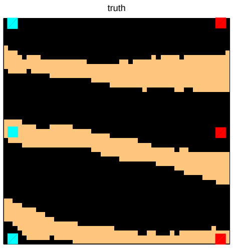


Figure 5.9: The true permeability field. Three injectors at the left-hand side boundary, three producers at the right-hand side boundary.

Each prior replicate is simulated in time without data assimilation. Bottom hole pressures are collected at observation times from each well. Figure 5.12 contains six plots, each representing one well. The left column is for the injectors, and the right column is for the producers. Simulated pressure observations from 50 replicates (blue dashed lines) are plotted against the true measurements (red continuous line).

Figure 5.13 displays prior water-production (in $[m^3/s]$) predictions for each of the 50 ensemble members. Colors indicate different producers. The thick dashed line is the true water-production in each of the three producers. The first water breakthrough occurs after the last assimilation time (24 months indicated on the graph with a vertical dashed line).

A root mean square (rms) error of the water breakthrough time (expressed in months) is computed for each well:

$$E_{rms} = \sqrt{\frac{1}{n_e}\sum_{i=1}^{n_e}(t_{wb,t} - t_{wb,i})^2},$$

where $t_{wb,t}$ and $t_{wb,i}$ are true water breakthrough time and simulated water breakthrough time $[month]$ of replicate i, respectively. Each water-production plot indicates corresponding root mean square errors of the water breakthrough times of the underlying ensemble. If water breakthrough did not occur until month 70, the water breakthrough time is taken to be equal to 70.

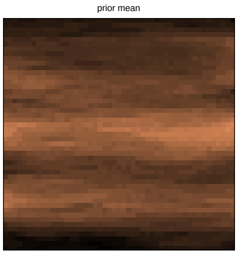


Figure 5.10: Prior permeability mean.

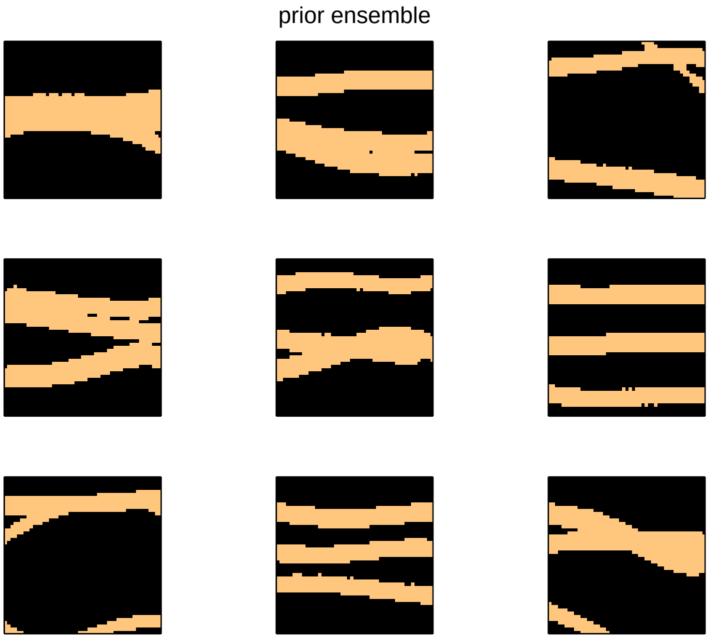


Figure 5.11: First 9 prior permeability replicates.

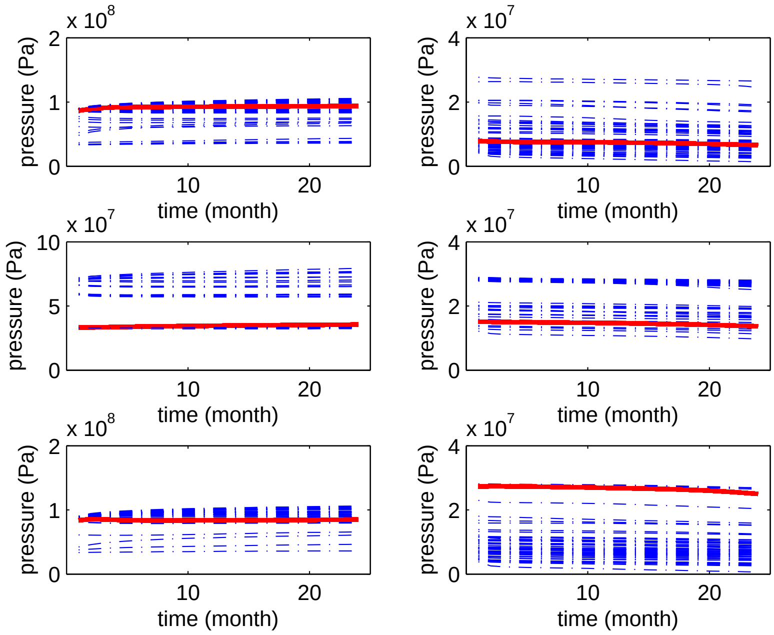


Figure 5.12: Prior ensemble. Bottom hole pressure from six wells; left column: injectors; right column: producers. Red continuous line - true bottom hole pressure. Blue dashed lines - prior ensemble bottom hole pressure forecast.

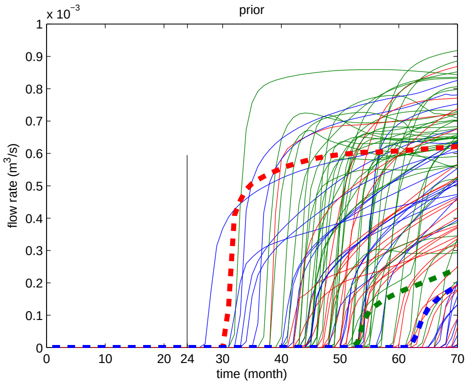


Figure 5.13: Water-production $[m^3/s]$ in three wells for the prior ensemble. Thick dashed: truth. Thin solid: ensemble. Vertical line: last assimilation time. E_{rms}(red) = 32.8, E_{rms}(green) = 9.86, E_{rms}(blue) = 14.34.

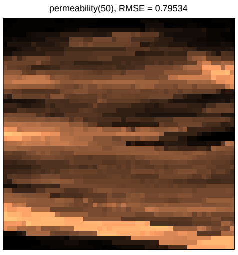


Figure 5.14: Pixel-based posterior mean.

Pixel-based approach

The pixel-based approach is the ensemble square root filter, [18], applied directly to pixel values of permeability. It is assumed that the permeability is the only uncertain parameter, that is, the state vector contains permeability values only. Variable transformation [24] is applied to keep the values of permeability within reasonable bounds during Kalman filtering.

The posterior permeability mean and the first nine replicates are shown in Figures 5.14 and 5.15, respectively. The posterior estimate clearly shows the lowest channel. The upper features are more pronounced in the near-well area. This is due to the fact that the lowest channel contains the lowest producer and the water travels faster towards it, even though the water breakthrough does not occur in the assimilation time. The continuity of shapes is not kept in general.

The posterior replicates are run from time 0, bottom hole pressure is collected at observation times and plotted against the true observations, see Figure 5.16, which should be compared to the prior runs in Figure 5.12. The uncertainty of the estimates is visibly decreased.

Figure 5.17 displays water-production for the posterior pixel-based ensemble. The filter estimated the first water breakthrough to occur at least 8 months too late.

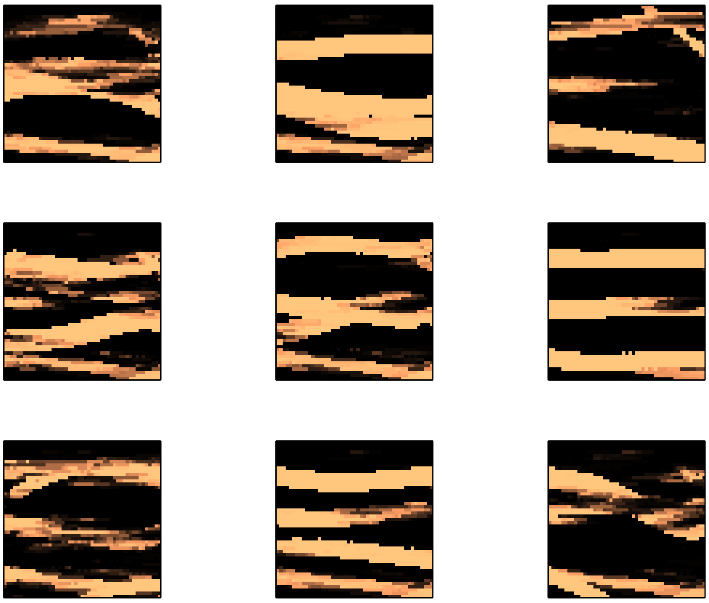


Figure 5.15: Pixel-based posterior replicates.

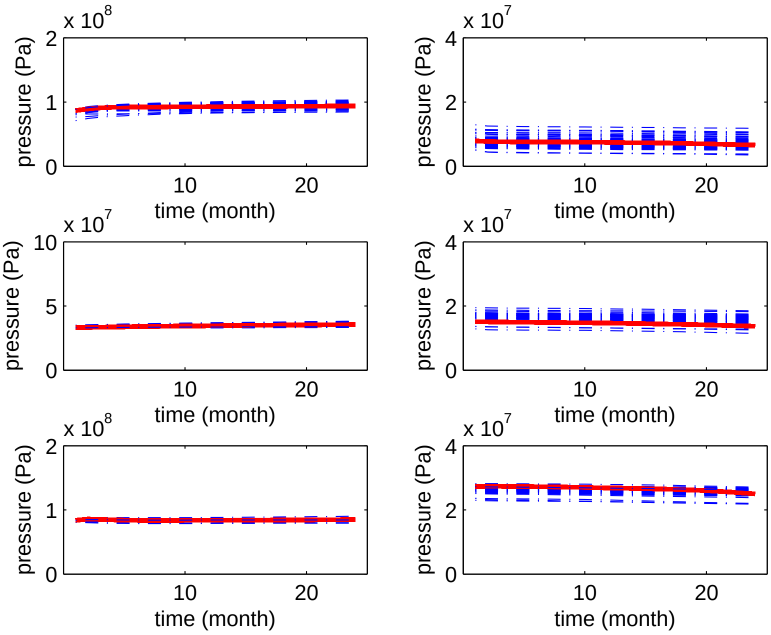


Figure 5.16: Pixel-based posterior estimate. Bottom hole pressure from six wells; left column: injectors; right column: producers. Red continuous line - true bottom hole pressure. Blue dashed lines - posterior ensemble bottom hole pressure forecast.

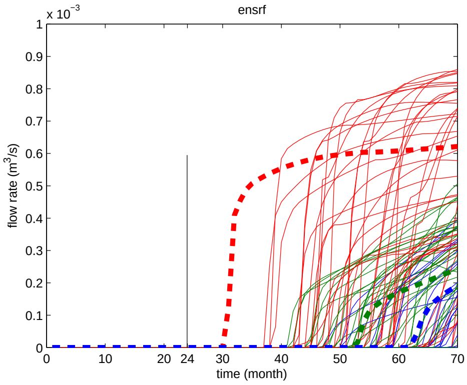


Figure 5.17: Water-production $[m^3/s]$ in three wells for the posterior pixel-based ensemble. Thick dashed: truth. Thin solid: ensemble. Vertical line: last assimilation time. $E_{rms}(\text{red}) = 26.35$, $E_{rms}(\text{green}) = 12.09$, $E_{rms}(\text{blue}) = 7.57$.

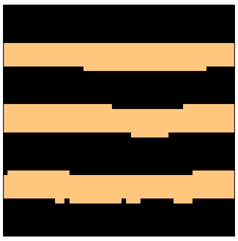


Figure 5.18: Base case permeability field.

Feature-based approach

The feature-based approach is the ensemble square root filter applied to the distortion parameters. It is assumed that the parameters carry the only uncertainty (and therefore indirectly the permeability is uncertain). It makes the state vector to be of size $8 \cdot N_f = 8 \cdot 3 = 24$ ($N_f = 3$ there are three channels and 8 parameters from the grid distortion functions).

There are $N_f = 3$ base cases, each one representing one channel. The margin for grid embedding is equal to $N_m = 10$. The horizontal lines in the base cases are between the rows: 9-13, 23-27, 37-41 in the original domain which is 49×49 pixels. The whole base case template is 69×69 pixels. Figure 5.18 shows a realization with all the parameters equal to zero. It is the initialization for the pattern search method.

The posterior permeability estimates are shown in Figure 5.19 (permeability ensemble mean), Figure 5.20 (distortion parameter mean applied to the base case), and Figure 5.21 (replicates). The method always maintains continuity of the structures from the left to the right boundary of the domain. Figure 5.22 shows how the first replicate evolved in time along the sequential data assimilation process.

The posterior replicates are run from time 0 and the bottom hole pressure is collected at observation times and plotted against the true observations, Figure 5.23. Figure 5.23 should be compared to the prior runs in Figure 5.12 and the pixel-based method in Figure 5.16. The posterior uncertainty is decreased for both methods and for all the wells,

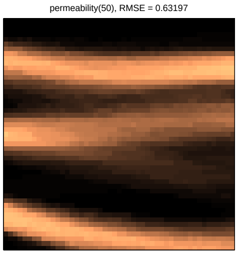


Figure 5.19: Feature-based posterior mean.

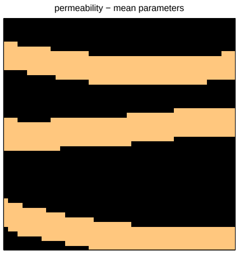


Figure 5.20: The mean of posterior parameters applied to the base case.

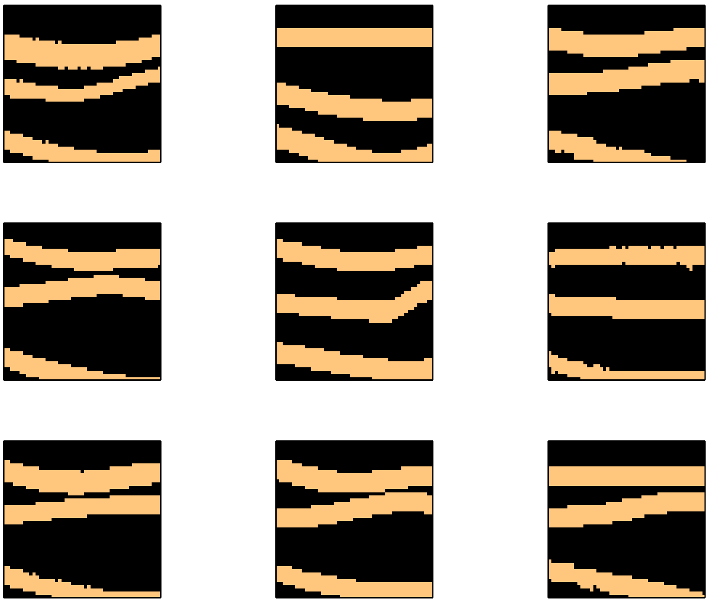


Figure 5.21: Feature-based posterior replicates.

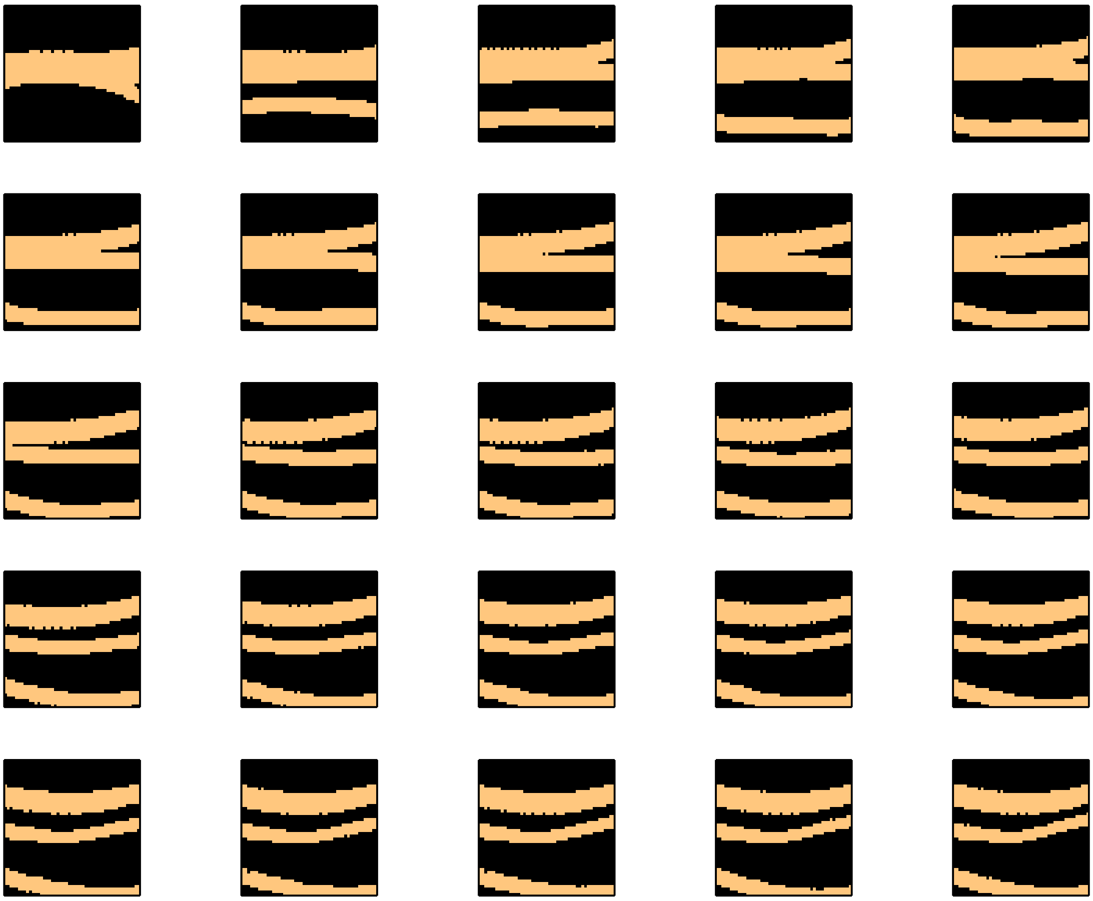


Figure 5.22: In reading order, the evolution of the first replicate in time along the sequential data assimilation process, initial and 24 updated permeability states.

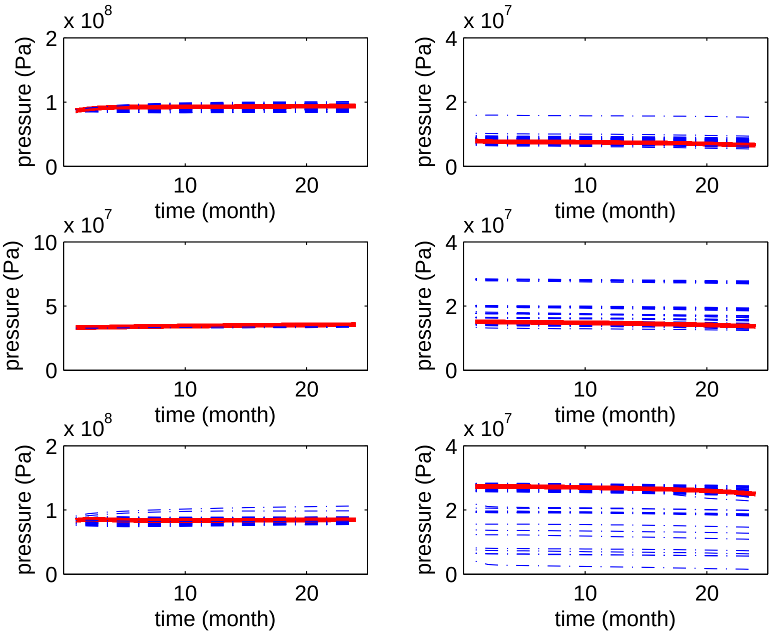


Figure 5.23: Feature-based posterior estimate. Bottom hole pressure from six wells; left column: injectors; right column: producers. Red continuous line - true bottom hole pressure. Blue dashed lines - posterior ensemble bottom hole pressure forecast.

but the pixel-based method better matched the production data.

Figure 5.24 shows the water-production for the posterior feature-based ensemble. The water breakthrough contains the truth within the ensemble bounds and it does not seem over-confident about the estimate.

Figure 5.25 shows the objective function values for the 50 ensemble members. Prior large values (circles) can be compared to the posterior pixel- and feature-based values. Here, the pixel-based posterior estimates (squares) are smaller in some cases than the feature-based posterior estimates (stars). For example, posterior feature-based replicates number 3 and 6 are the ones with larger objective values. They are pictured in Figure 5.21 row 1 column 3, and row 2 column 3, respectively. The large objective value in the replicate 3 is caused by the mismatch in the lower right corner where the channel does not reach the producer; for replicate 6, the center producer is in a high permeability area unlike in the truth. In Figure 5.25 'ensemble member' number 51 represents the objective function value of the mean. We can see the objective function value for the prior mean from Figure 5.10 (circle), the posterior pixel-based mean from Figure 5.14 (square) (which is the smallest value in this plot), the posterior feature-based permeability mean from Figure 5.19 (small star), and the posterior feature-based parameter mean from Figure 5.20 (big star). The large value of the big star indicates how nonlinear the relationship is between the observations and the distortion parameters and how for severely nonlinear problems a mean is not necessarily a good estimate. For pixel-based results the mean is a good estimate due to the larger number of degrees of freedom which can give a better fit. This high nonlinearity makes the problem more appropriate for direct parameter search

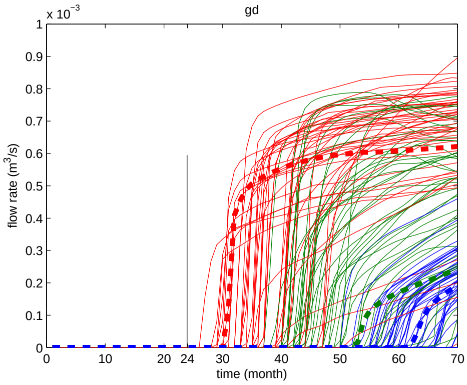


Figure 5.24: Water-production $[m^3/s]$ for the posterior feature-based ensemble. Thick dashed: true. Thin solid: ensemble. Vertical line: last assimilation time. E_{rms}(red) $= 12.93$, E_{rms}(green) $= 9$, E_{rms}(blue) $= 5.51$.

methods.

The grid distortion parametrization contains only a few parameters to be estimated. Additionally, since the parameters are static and they are the only source of uncertainty, the formulated problem is suitable for direct search methods. For comparison, we show how pattern search works with grid distortion parametrization, Figure 5.26. The objective function goes down to 0.815 in the case presented here. Pattern search methods involve slight perturbations along the minimization procedure, therefore, the results might differ for multiple runs. The objective function values for the posterior realizations from sequential data assimilation ensembles are of a similar magnitude.

The channels in the permeability field obtained by the pattern search method with grid distortion, Figure 5.26, are not as smooth as expected. The kinky edges of the shapes come from the fact that the grid is strongly stretched and the interpolation procedure produces artefacts. If we look at the three grids (one for each feature) before interpolation, Figure 5.27, we can see that the transformation itself is smooth. There, the full grids with the margin equal to 10 are shown, therefore, after interpolation and margin trimming, we receive images from the center of these grids. Precisely, the grid for x- and y-coordinates from the interval $[11, 60]$.

The objective function for the pattern search method is a measurement mismatch, therefore, the pressure data are almost perfectly fitted. The water production curves for the estimated permeability field in Figure 5.26 are shown in Figure 5.28. The estimated parameters indicate a late water breakthrough.

Both techniques, pixel- and feature-based, improved the bottom hole pressure match in the wells. The first water breakthrough time was better estimated by the grid distortion method. The advantage of the feature-based method over the pixel-based method is that the channels are preserved in the posterior permeability replicates. Moreover, they can be parameterized with just a few parameters which makes the estimation problem feasible for alternative minimization techniques.

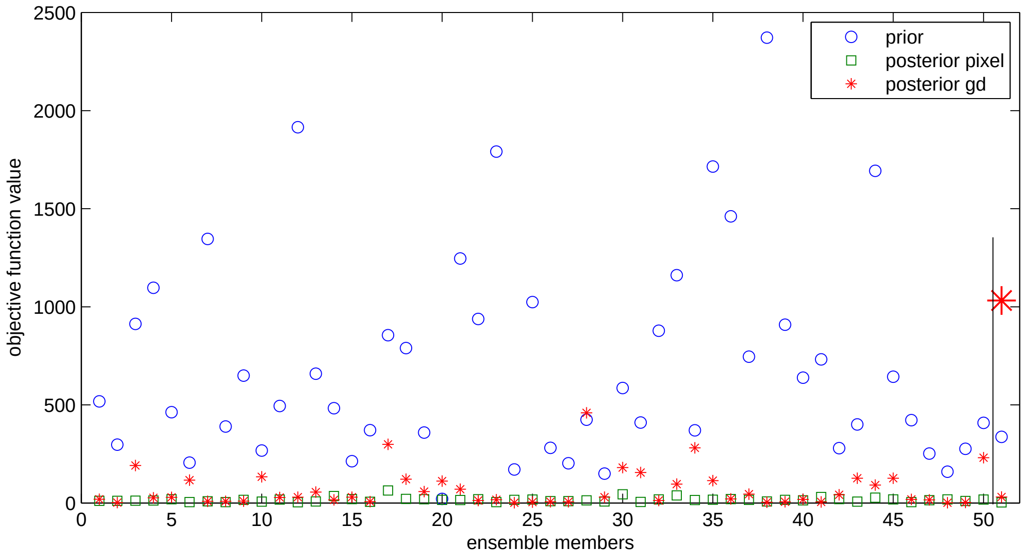
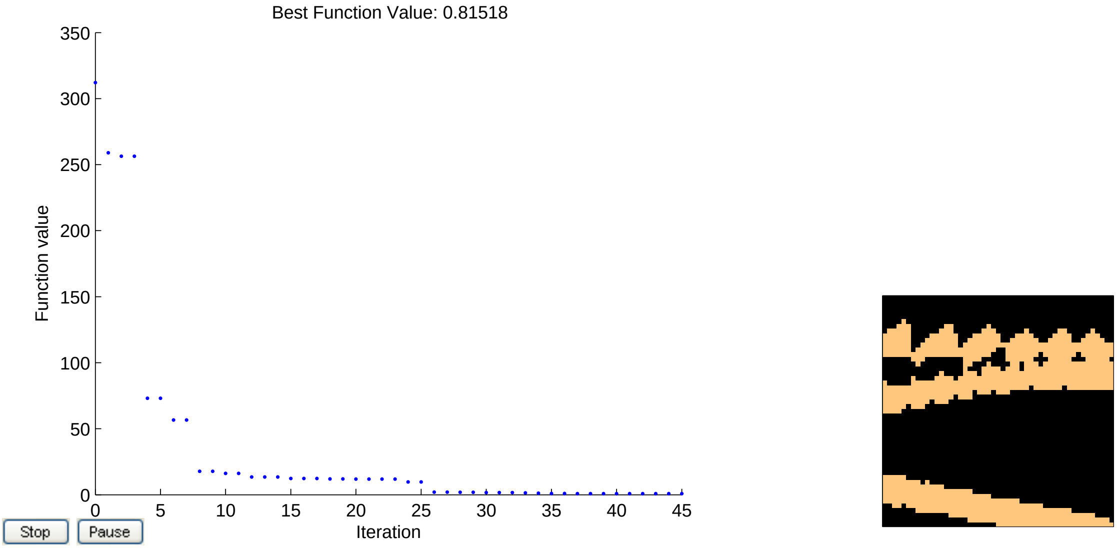


Figure 5.25: The values of the objective function for prior and posterior permeability replicates.



(a) Objective function, pattern search method for grid distortion.

(b) Posterior permeability estimate with objective value equal to 0.815.

Figure 5.26: Pattern search method with grid distortion.

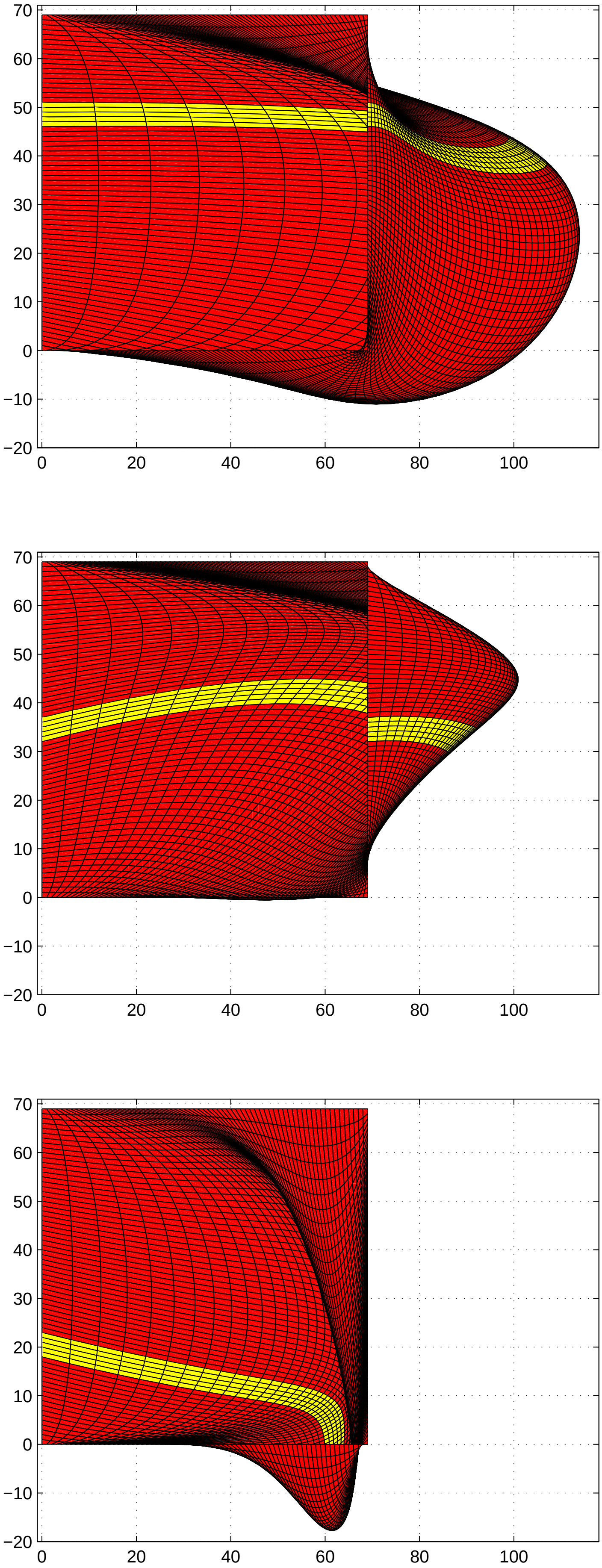


Figure 5.27: Distorted grids for three channels from Figure 5.26.

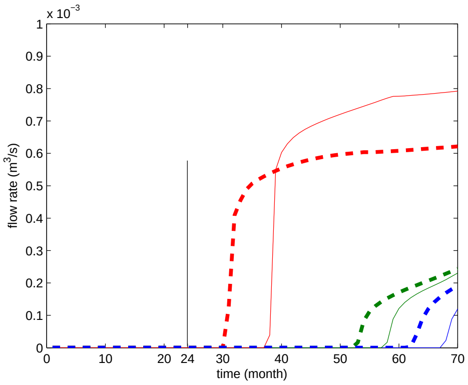


Figure 5.28: Water-production $[m^3/s]$ in three wells for the posterior feature-based permeability estimated using direct search, Figure 5.26. Thick dashed: truth. Thin solid: estimate. Vertical line: last observation time. $E_{rms}(\text{red}) = 8$, $E_{rms}(\text{green}) = 5$, $E_{rms}(\text{blue}) = 5$.

5.1.7 Discussion and Conclusions

In this section a new feature-based method based on grid warping is developed. It parameterizes the uncertain parameter field such that the parameter vector to be estimated becomes about 1% of its original size, and keeps the initially introduced features. The grid distortion method was applied to a 2D synthetic reservoir problem with ensemble Kalman filtering. The positions of three channels present in the domain were estimated and the water-production response was improved. The estimated channels run continuously from one boundary of the domain to the other without breaks that occur in the standard approach. The variability between the posterior replicates is reflected in the variability in the bottom hole pressure and water-production profiles.

The implementation with the ensemble Kalman filter is not essential. Due to the significant reduction in the dimensionality of the problem, grid distortion can be combined with a pattern search minimization method.

Poisson's equations in (5.7) and (5.8) are the basis of the distortion method. The right-hand-side distortion functions can be adjusted to the requirements of an application. Here, the domain is relatively small and it is sufficient to use paraboloids with constrained magnitude that adjust the channels globally. For larger, more complicated domains, it might be worth considering higher order polynomials for the distortion functions or localization if only regional influence is expected.

The distortion functions cannot have too large values to prevent the domain from flipping over. That was the reason to use small standard deviations of $6 \cdot 10^{-5}$ for the parameters in the paraboloids. More rigorous rules for defining the functions should be the topic of follow-up research that can still gain from insights in grid generation and image processing techniques. Further, the method should be applied to different types of problems to determine its strengths and faults. A 3D application is developed next.

5.2 Grid distortion for a 3D groundwater flow model

This section extends the grid distortion method to three dimensions. In Section 5.2.1 the 3D extension is presented. Thereafter, two applications are shown. First, a 3D groundwater flow model is used to test the concept. Thereafter, the grid distortion method is used with a 3D multiphase reservoir simulator. In both cases, we choose to use a pattern search method, Section 3.5, to estimate the grid distortion parameters.

For an alternative method of feature estimation with the ensemble Kalman filter in the groundwater problem see [98]. Here, the authors propose a method termed normal score ensemble Kalman filter that transforms channelized conductivity fields into univariate Gaussian variables before the update step is performed. After updating the output is backtransformed to the feature domain.

5.2.1 The 3D grid distortion method

Here, we present the grid distortion method for a 3D problem. Analogically to Section 5.1.5, let $\mathcal{X}$, $\mathcal{Y}$ and $\mathcal{Z}$ be curvilinear 3D coordinates, and ξ, η and ζ Cartesian coordinates. Then, a distorted grid is a solution for $\mathcal{X}$, $\mathcal{Y}$ and $\mathcal{Z}$ to Poisson's equations:

$$\frac{\partial^2 \mathcal{X}}{\partial \xi^2} + \frac{\partial^2 \mathcal{X}}{\partial \eta^2} + \frac{\partial^2 \mathcal{X}}{\partial \zeta^2} = P,$$
$$\frac{\partial^2 \mathcal{Y}}{\partial \xi^2} + \frac{\partial^2 \mathcal{Y}}{\partial \eta^2} + \frac{\partial^2 \mathcal{Y}}{\partial \zeta^2} = Q,$$
$$\frac{\partial^2 \mathcal{Z}}{\partial \xi^2} + \frac{\partial^2 \mathcal{Z}}{\partial \eta^2} + \frac{\partial^2 \mathcal{Z}}{\partial \zeta^2} = R.$$

Functions P, Q and R on the right hand sides of the equations are distortion functions and they drive the coordinate transformation. They are parameterized to reduce the number of variables to be estimated. The parametrization reflects a belief in a type of deformation that could drive the change of features in the domain. This allows the method to be adjusted to different types of applications. The boundary conditions need to be given. The setup is analogous to the 2D version of the method. Therefore, the boundary is fixed at the Cartesian grid nodes and the domain is embedded in a larger grid with given margin to remove the boundary effects.

5.2.2 3D groundwater flow model

We choose to start the 3D grid distortion experiments with a 3D model that is easily accessible and relatively simple compared to complex reservoir models. A 3D groundwater flow model, [69], is presented here.

Let a voxel be a $1 \times 1 \times 1$ three-dimensional pixel. Consider a rectangular cuboid composed of $n_x \times n_y \times n_z$ voxels (in x-, y- and z-axis direction, respectively), where a conductivity value is prescribed in each voxel. If voxel v_1 has conductivity K_1 and voxel v_2 has conductivity K_2, and v_1 and v_2 share a face, then the conductivity between the two voxels is equal to: $K_{12} = \sqrt{K_1 \cdot K_2}$. Let us call K_x, K_y and K_z a conductivity in x-, y- and z-axis direction, respectively. Then, the head $h = h(x, y, z, t)$ at point (x, y, z) at time t can be expressed by the following equation:

$$\frac{\partial}{\partial x}\left(K_x \frac{\partial h}{\partial x}\right) + \frac{\partial}{\partial y}\left(K_y \frac{\partial h}{\partial y}\right) + \frac{\partial}{\partial z}\left(K_z \frac{\partial h}{\partial z}\right) = S_s \frac{\partial h}{\partial t},$$

where S_s is a specific storage coefficient.

Boundary conditions have to be specified for the domain. Figure 5.29 depicts a cuboid with two indicated faces for which high or low head is prescribed. The other faces work under no-flow conditions.

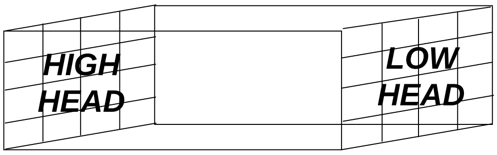


Figure 5.29: Representation of the domain.

Initial condition for the head has to be given.

A source/sink term Q representing a well (or wells) can be introduced:

$$\frac{\partial}{\partial x}\left(K_x\frac{\partial h}{\partial x}\right)+\frac{\partial}{\partial y}\left(K_y\frac{\partial h}{\partial y}\right)+\frac{\partial}{\partial z}\left(K_z\frac{\partial h}{\partial z}\right)+Q=S_s\frac{\partial h}{\partial t}.$$

Vertical wells pump and/or collect data (in the case of monitoring wells) from each layer.

The equation is discretized on a rectangular grid and after discretization, K_x, K_y and K_z are defined at the faces of voxels, and h is defined in the voxels. The equation is solved explicitly with central and forward finite difference schemes.

The groundwater flow model is a single-phase model and can be derived directly from Equation (2.5) with no gravity:

$$\frac{\partial}{\partial x}\left(\frac{k}{\mu_w}k_{rw}\frac{\partial p}{\partial x}\right)+\frac{\partial}{\partial y}\left(\frac{k}{\mu_w}k_{rw}\frac{\partial p}{\partial y}\right)+\frac{\partial}{\partial z}\left(\frac{k}{\mu_w}k_{rw}\frac{\partial p}{\partial z}\right)+Q_w=$$
$$\phi\left[S_w(c_w+c_r)\frac{\partial p}{\partial t}+\frac{\partial S_w}{\partial t}\right],$$

where we are interested only in the motion of water with no oil present, therefore, $S_w=1$ and $k_{rw}=1$. The groundwater flow equation arises naturally given the relationships:

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

$$h=\frac{p}{\rho g}+h^*,$$

$$\rho g\phi(c_w+c_r)=S_s,$$

where ρ is the water density, g is the gravity acceleration, and h^* is a height above a datum.

5.2.3 Experiment setup

A list in Table 5.2 summarizes the experimental setup of the 3D groundwater model.

Figure 5.30 shows the true head contours mapped on the x-y plane at the last time step. The vertical pumping well is located in the center of the domain and it is open to flow in all the 18 inner horizontal layers (top and bottom layers are no-flow). There are eight monitoring wells spread over the domain, circles in Figure 5.31, each collecting head data from every inner horizontal layer at all the 500 time steps. There is no measurement noise. The data over time from all the wells is shown in Figure 5.32. Finally, the true conductivity inner layers are shown in Figure 5.33. It was constructed using the 3D grid distortion method with 18 parameters per shape (in total $18 \cdot 2$ in this case) where the distortion functions are three dimensional second order polynomials. The initial conductivity (which is also a base case) is shown in Figure 5.34.

- Domain size is 900 × 900 × 250 $[m \times m \times m]$.
- Number of gridblocks is 30 × 30 × 20.
- Voxel size is 30 × 30 × 12.5 $[m \times m \times m]$.
- Specific storage coefficient is $10^{-2.2}$ $[m^{-1}]$.
- Pumping rate per well opening is $2.1/30^2/12.5/18$ $[s^{-1}]$.
- Background low conductivity is $3.2 \cdot 10^{-5}$ $[m/s]$.
- Channel high conductivity is $3 \cdot 10^{-4}$ $[m/s]$.
- Horizontal conductivity is equal to 10% of vertical conductivity, $K_z = 0.1 \cdot K_x$.
- Head boundary condition is equal to 200 and 100 at high head and low head end, respectively, see Figure 5.29.
- Initial head is a steady-state head for homogeneous conductivity, hence the head initial condition is inconsistent with the underlying conductivity.
- There are 500 time steps each equal to 1 hour.

Table 5.2: Variable setup for the groundwater problem.

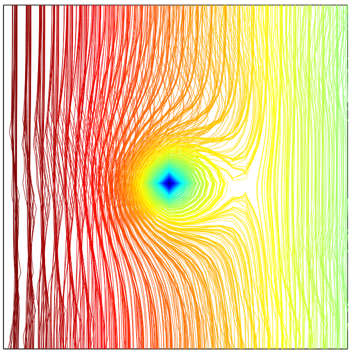


Figure 5.30: The true head contours mapped on the x-y plane at the last time step.

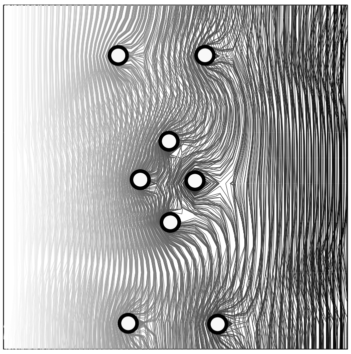


Figure 5.31: The location of the 8 monitoring wells (circles). Contour lines are not relevant.

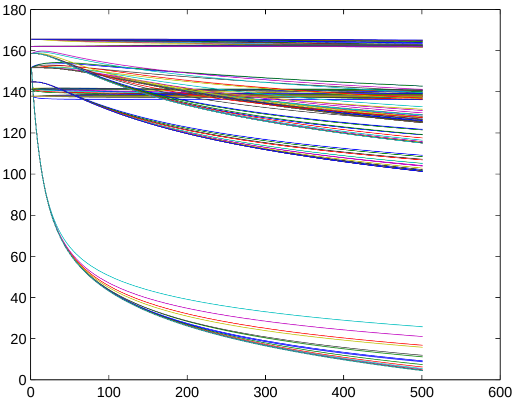
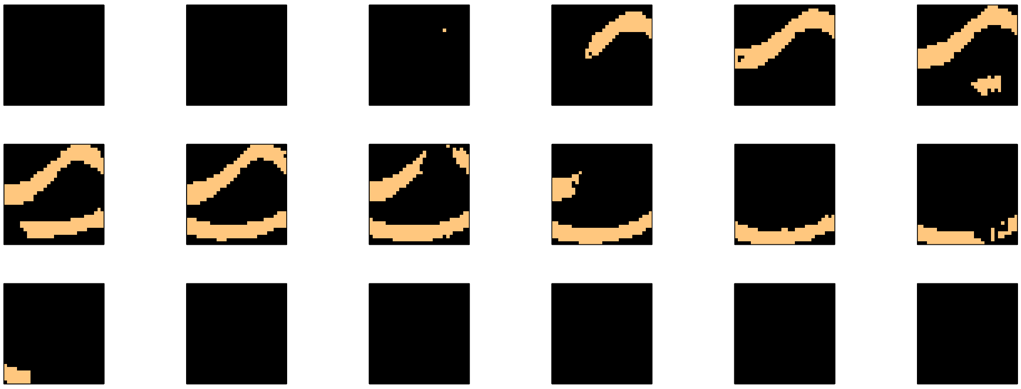
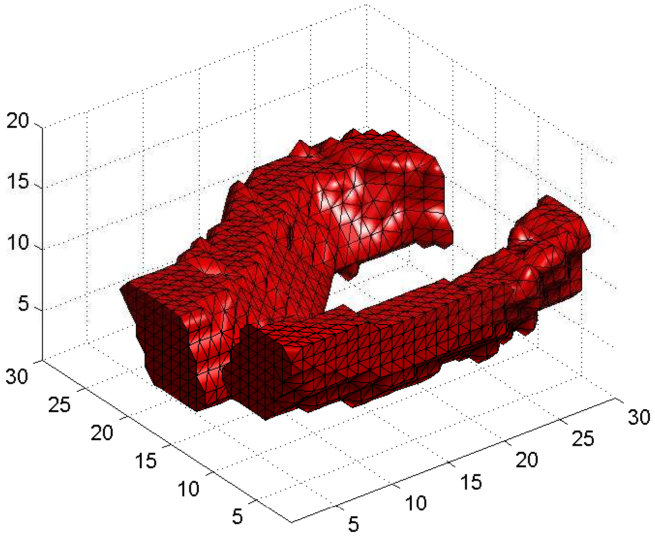


Figure 5.32: Head observations over time from all the wells $[m]$. The stronger head decline observations come from the producer (towards the bottom of the plot). The observations with smaller variability come from the monitoring wells (towards the top of the plot).

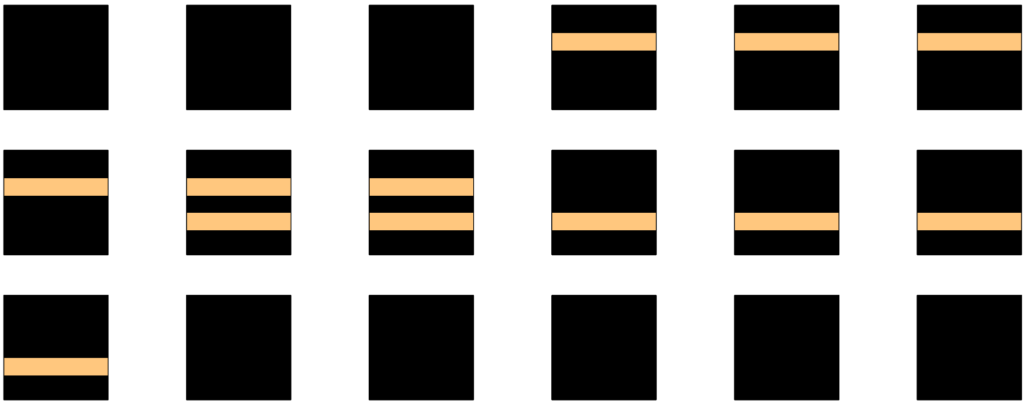


(a) The layers in reading order starting from the bottom.

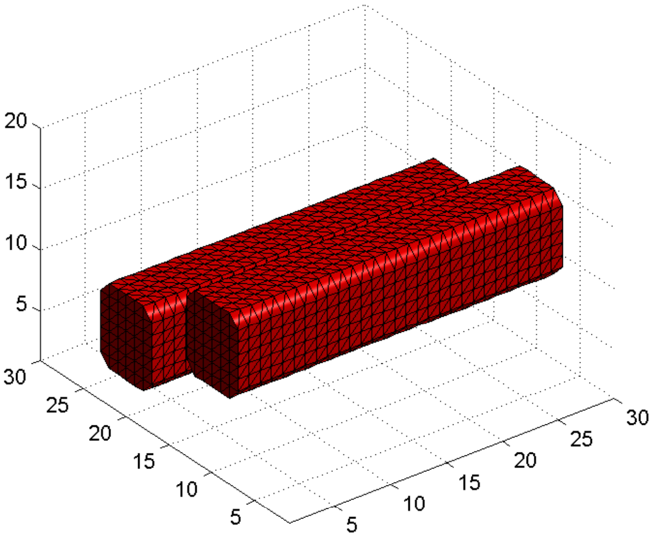


(b) The 3D view.

Figure 5.33: The true conductivity.



(a) The layers in reading order starting from the bottom.



(b) The 3D view.

Figure 5.34: The initial conductivity. The base case with zero distortion.

5.2.4 Discussion and results

Current chapter begins with Section 5.1 where the 2D grid distortion method is introduced, developed and thoroughly investigated. An extension of the dimension in the 3D method is simple, on the one hand, since it does not introduce any complications in the method itself, only an additional derivative term. On the other hand, in three-dimensional space the dependence between cells becomes more complex and there are more degrees of freedom when seeking the solution. Therefore, even though we might be certain that the grid distortion is an effective method, we need to investigate if the additional dimension is not making the stated problem too difficult to solve.

The 3D grid distortion method is applied to the 3D groundwater flow model. The task is to find the true conductivity pattern that lies within the span of possible solutions given the head response in 9 wells, eight of which are monitoring wells. A pattern search method (Latin Hypercube Sampling, refer to Matlab documentation) is implemented to find the grid distortion parameters where a search interval is defined to have radius equal to about two around the true solution. The objective function is a sum of squared measurement mismatch terms.

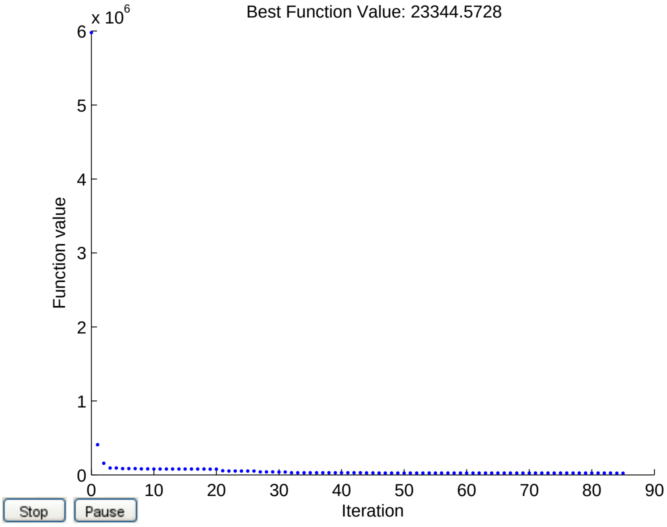
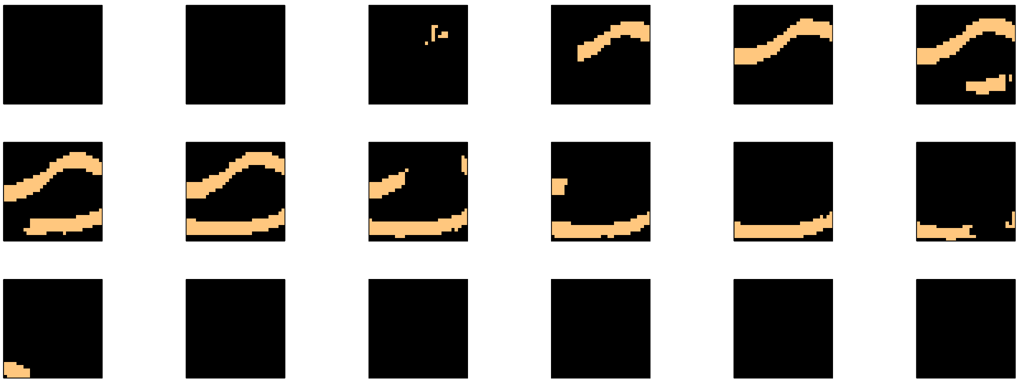


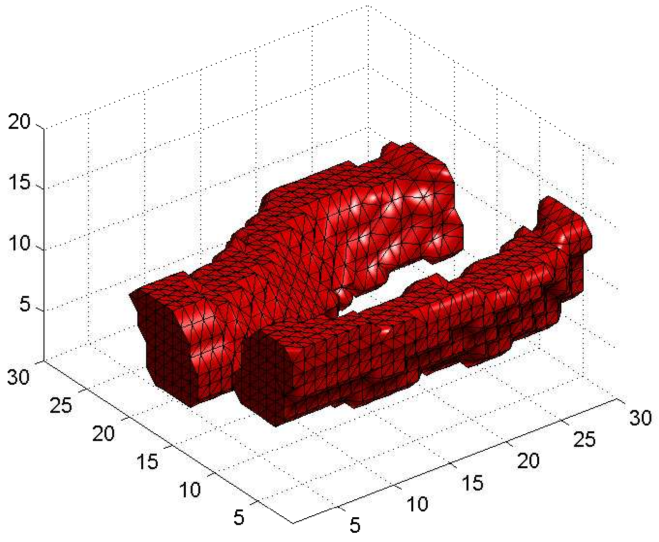
Figure 5.35: Objective function decay in pattern search method.

Figure 5.35 presents the objective function values which go down to 23344 starting at $6 \cdot 10^6$. (Note that, since there is no measurement noise, it would be possible for the objective function to reach 0.) The posterior conductivity is shown in Figure 5.36. The position of the channels is correct but it is not identical on a pixel-by-pixel basis. These small differences influence the value of the objective function. The pattern search method has not been able to find the exact solution due to the nature of the search algorithm but still it has performed very well.

We need to note that this result has been obtained with relatively narrow search interval. It comes from the fact that Latin hypercube sampling aims at drawing almost uniformly distributed points in the search interval. Since the true solution was randomly sampled from a Gaussian zero-mean distribution with small variance, we expect to see rather zero concentrated small numbers, and we allow the search algorithm to be able to



(a) The layers in reading order starting from the bottom.



(b) The 3D view.

Figure 5.36: The posterior conductivity estimate.

find similar solutions. Therefore, even though the true solution is in the search space and the problem seems to be easily solvable, it might not be possible to find the exact answer. Either way, the performance of the method is excellent. In the next application we will study a more realistic case where the true solution is not in the search space.

5.3 Grid distortion for a 3D reservoir model

This section presents the research findings on the 3D grid distortion method implemented in a 3D 2-phase reservoir simulator. We use 3D black oil simulator for water flooding. Given the true permeability field, modeled with $480m \times 480m \times 28m$, $60 \times 60 \times 7$ grid blocks where the layers are shown from the bottom in reading order in Figure 5.37, we simulate[1] observations of bottom hole pressure, liquid rate, oil rate and water rate for two hundred time steps of one day each. A 3D view with active cells, and well locations is shown in Figure 5.38, the low permeability value is 300 mD, and high 6100 mD in the horizontal, and ten times less in the vertical direction. Table 5.3 summarizes other reservoir and fluid properties of this experiment.

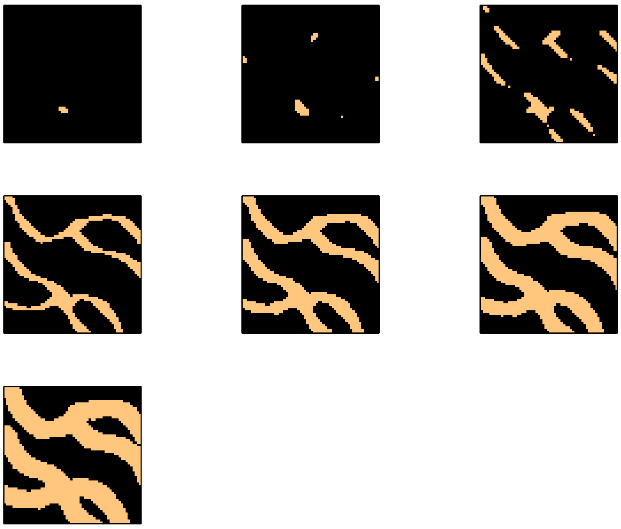


Figure 5.37: The true permeability layers.

We choose the objective function to be a sum of squared differences between the observed and predicted data. One evaluation of the objective function given the grid distortion parameters takes a few minutes (3-6 minutes) and its most time-consuming part is the construction of the permeability field. It is due to the fact that for one permeability field to be built we need to solve Poisson's equation for each dimension and feature. The grid distortion uses four base cases, each one with one horizontal channel, trying to follow our expectations with respect to the true shapes. This gives $3 \cdot 4 = 12$ Poisson's equations for large domains. Even though the grid is $60 \times 60 \times 7$, a margin is added for the flexibility at the permeability boundaries. The margin is equal to 7 which gives fields of size $74 \times 74 \times 21$ to be solved for in Poisson's equation, which might be considered computationally intense.

To show that the model is sensitive to small permeability changes we run it with the true permeability where the two bottom layers have been made uniformly low permeable, Figure 5.39. In this case the objective function value equals to $4.33 \cdot 10^6$. It indicates that

---

[1] Thanks to G. van Essen, M. Kaleta and M. Glegola.

Variable	**Value**	**SI units**
Gridblock height	4	m
Gridblock length/width	8	m
Oil dynamic viscosity	$5 \cdot 10^{-3}$	$Pa \cdot s$
Water dynamic viscosity	$1 \cdot 10^{-3}$	$Pa \cdot s$
Oil compressibility	$1.0 \cdot 10^{-10}$	Pa^{-1}
Rock compressibility	0	Pa^{-1}
Water compressibility	$1.0 \cdot 10^{-10}$	Pa^{-1}
Initial reservoir pressure	$4 \cdot 10^{7}$	Pa
Endpoint relative permeability of oil	0.9	–
Endpoint relative permeability of water	0.75	–
Corey exponent, oil	1	–
Corey exponent, water	1	–
Residual oil saturation	0.1	–
Connate water saturation	0.1	–
Porosity	0.2	–

Table 5.3: Reservoir and fluid properties for the 3D experiment.

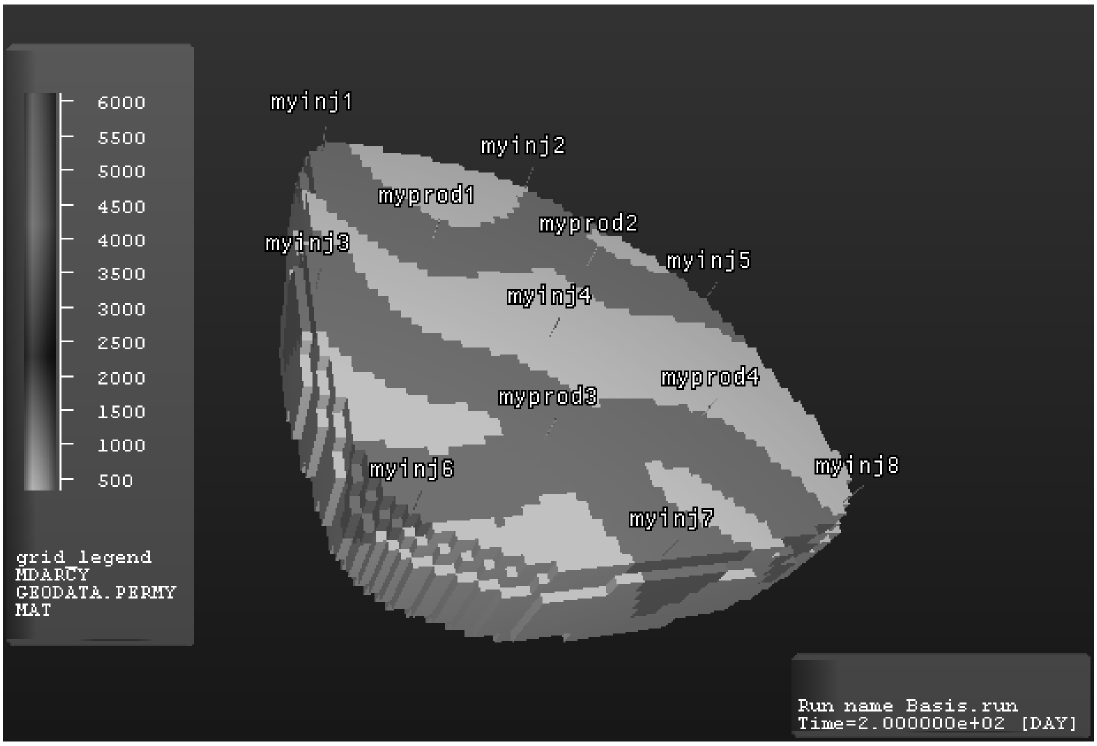


Figure 5.38: The true permeability in 3D with indicated wells.

values of the objective function with a similar magnitude should not be considered very large.

We want each of the four base cases represent one horizontal channel wider at the top and narrower towards the bottom. Let each distortion function be a constant. This way, we estimate 3 parameters (one for each $\mathcal{X}$-, $\mathcal{Y}$- and $\mathcal{Z}$-coordinate) for each of the 4 base cases, that gives only 12 parameters in total. The problem is simple and the question is if the parametrization is sufficient for the level of complexity of the permeability field. The initial guess for the parameter values is 0 and Figure 5.40 shows the corresponding initial permeability field with the objective function value $1.037 \cdot 10^{10}$. We run the pattern search method, Section 3.5, with search interval $[-5, 5]$. The algorithm terminates due to a small mesh size and inability to proceed. The objective function is shown in Figure 5.41 and a corresponding posterior estimate in Figure 5.42. Clearly, the algorithm does not seem very effective.

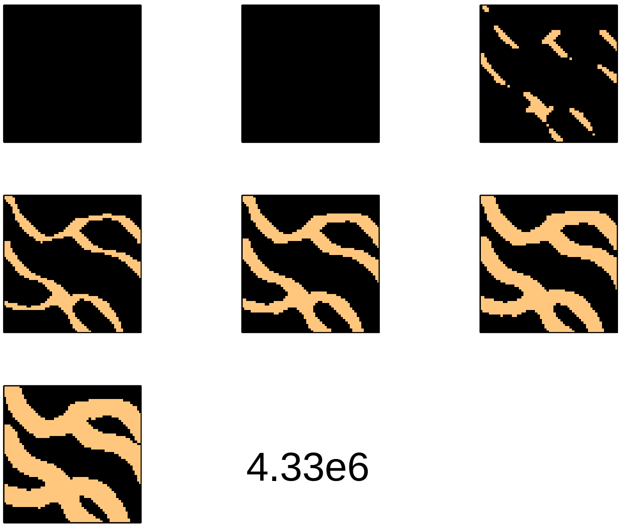


Figure 5.39: True permeability with uniform two bottom layers giving the objective value of $4.33 \cdot 10^6$. Compare to Figure 5.37.

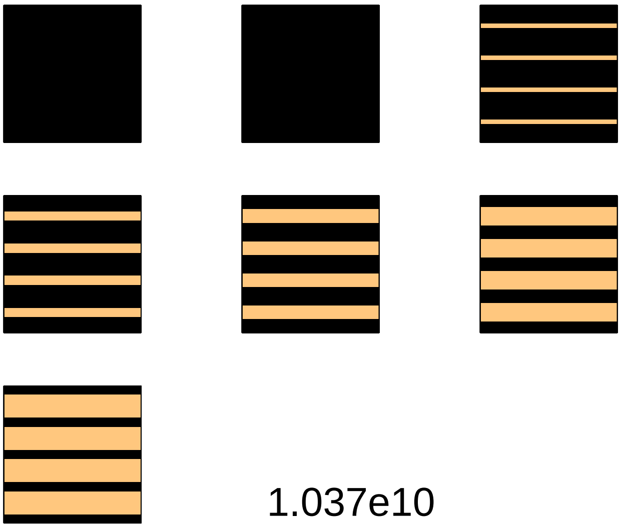


Figure 5.40: The base case permeability and initial guess giving the objective value of $1.037 \cdot 10^{10}$.

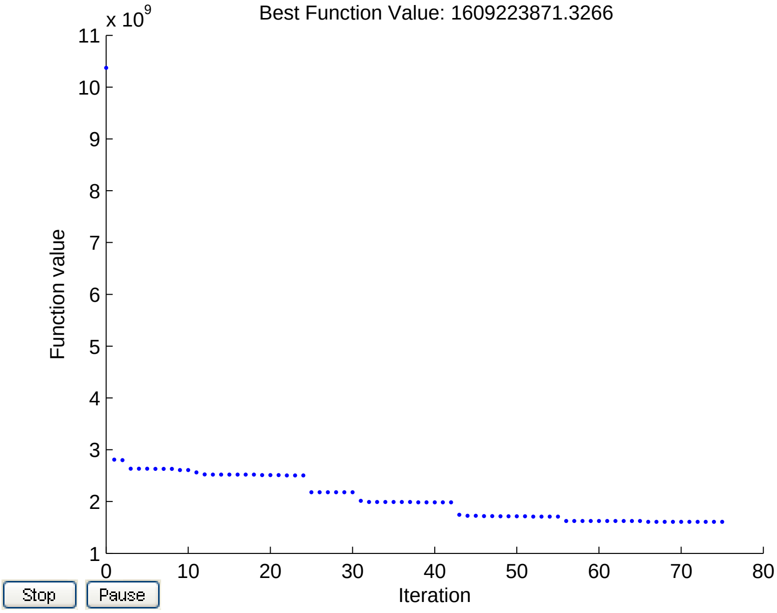


Figure 5.41: Pattern search method objective function with simple constant distortion functions.

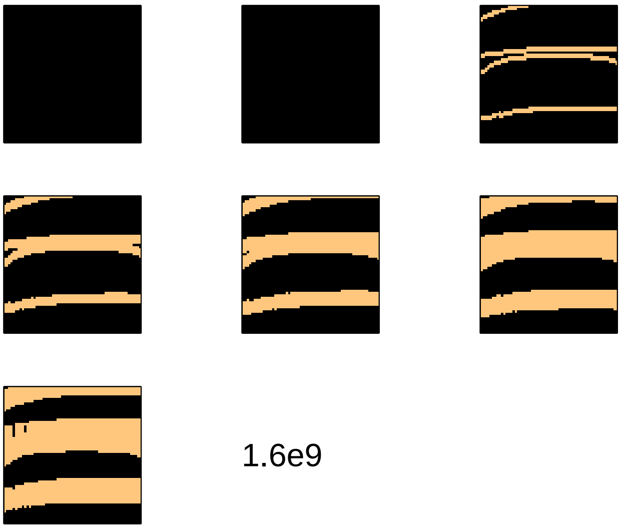


Figure 5.42: Pattern search method final estimate from the objective in Figure 5.41 with value $1.6 \cdot 10^9$ with simple constant distortion functions.

Now we modify the distortion functions slightly to allow more feature deformation. We choose 9 parameters per coordinate direction, that is a total of 27 parameters that control one single feature. This gives 108 (four channels give $27 \cdot 4$) parameters fully describing the 3D permeability field, compared to 25200 total number of all grid cells. The 9 parameters make up a right-hand-side distortion function; they are assigned to vertical panels of the 3D domain. We want to estimate the 108 grid distortion parameters through direct search specifying the feasible interval for parameter values to be $[-\frac{1}{2}, \frac{1}{2}]$. The initial guess for the parameter values is 0. Figure 5.43 shows the pattern search objective function, and Figure 5.44 is the result of the final iteration with value $9.359 \cdot 10^7$. The search took about seven days[2] but could have been stopped earlier since the last iterations did not improve the objective value any more. The posterior permeability estimate contains the channel structure that bends in a similar way as the true features due to the distortion function parametrization. The channel locations are not found exactly but it seems the algorithm distinguishes the north and south part of the structure and some well connections are restored, lowering the objective function value.

---

[2] The experiments have been run on a single desktop workstation with $3200 MHz$ processor and $24 GB$ RAM.

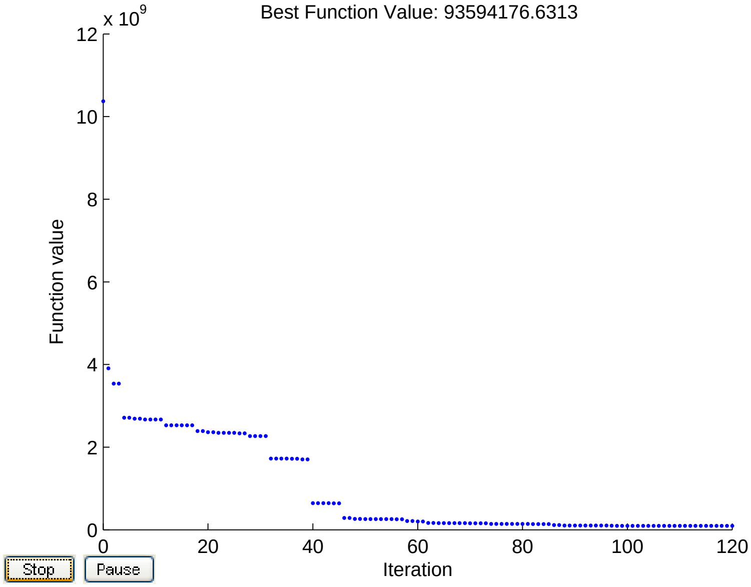


Figure 5.43: Pattern search method objective function with more complex distortion functions.

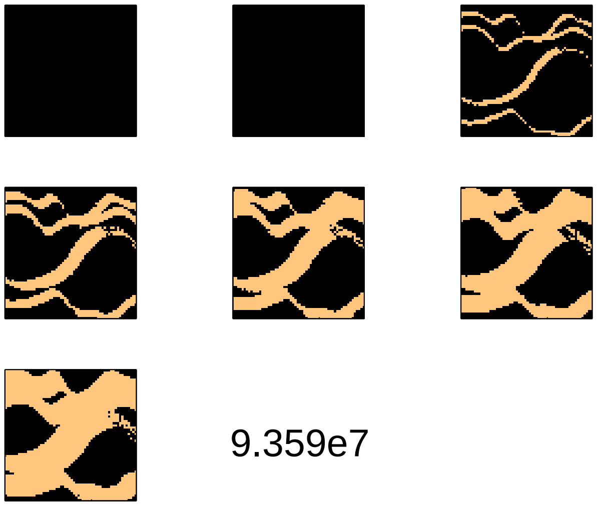


Figure 5.44: Pattern search method final estimate from the objective in Figure 5.43 with value $9.359 \cdot 10^7$.

5.3.1 Discussion and conclusions

We compare the results from two experiments, one with the simplest possible distortion function forms, and the other with slightly more complex representation. We see that the former does not allow significant channel distortion and gives not satisfactory results. The latter distortion function representations allow multiply-bending channels. These prove to be more effective and take the objective function values down to a small number. Even though the channels seem to have a wrong orientation, they create a relatively good permeability field as far as the objective function is concerned.

The grid distortion performs well with images that represent facies of a homogeneous nature since it has not yet been developed to a point where it could represent heterogeneous fields. Some knowledge about the shape characteristics appears to be useful for specification of the distortion functions. The distortion functions and the base cases fully describe the domain, they need to allow a construction of geological features we look for.

Due to a time consuming grid distortion procedure for the large 3D cases, it is not easy to implement the classical ensemble Kalman filter algorithm in these large examples. It takes around 3 minutes to create one permeability field in the grid distortion method and run it forward in the large reservoir model. Assuming 100 possible ensemble members and 200 time steps, the time to run the full experiment is $3 \cdot 100 \cdot 200min = 1000h \approx 41.7days$. Clearly, in the current computational conditions it is not a feasible methodology. An alternative could be the asynchronous ensemble Kalman filter, [71], or the ensemble smoother, [78], or the possibility to implement the ensemble computation in parallel.

Chapter 6

Conclusions and recommendations

This thesis combined research in image processing, data assimilation and reservoir engineering. Two different image processing studies led to two main directions in this thesis, both embedded in parameter estimation for reservoir engineering applications, both still worth exploring further. The leading topic was feature-based modeling, which is particularly suited for fields that show leading patterns and shapes.

First, an ensemble multiscale filter was investigated. The ensemble multiscale filter is an update tool that partitions the domain and assigns it to leaf nodes of a tree. The tree represents the connections and expected dependencies between the parts of the domain. The upper scales are built from the strongest relationships between the children nodes. This upscaling (pyramid) is common in image processing techniques. It was shown that the tree structure used in the covariance representation is an efficient localization tool. We investigated several numbering schemes, i.e. procedures of assigning the variables to the tree nodes. It revealed that the order with which the assignment is performed is an important factor for covariance representation. Moreover, it can account for stronger correlations along features which means that the structural information can be embedded in the filter.

Still, it would be beneficial to test EnMSF in a larger application with more variables and/or measurements. The tree assignment could be made automatic, based for example on the features in the field. Then, the subtrees could directly reflect separate features that would be connected higher up the tree. Additionally, the numbering inside the features could be made in clusters.

Second, the grid distortion method for reservoir history matching was developed in this thesis. We were looking for a technique that would smoothly distort some prescribed features without breaking them. It is an important issue in reservoir engineering history matching where there is usually few data available and many variables to estimate, which creates many degrees of freedom. Therefore, we wanted our technique to parameterize the feature field with very few parameters. We chose to work with the mesh instead of the pixel values directly, and used Poisson's equation for grid deformation. The parametrization of distortion functions in Poisson's equations provided an indirect parametrization of the permeability fields. These parameters were estimated by optimization algorithms. We showed that grid distortion efficiently defines the domain with just a few parameters. The shapes are preserved along the optimization, and their deformation is global rather than pixel based. This small number of parameters made the grid distortion suitable for parameter estimation methods based on the objective function minimization. In our applications the drawback of having a small number of parameters from grid distortion is the severe nonlinearity that is created between the parameters and observations.

The grid distortion method could be enriched with an additional amplitude adjustment procedure possibly allowing heterogeneous fields. Other partial differential equations could be used in place of Poisson's equation investigated in this thesis. For even larger fields, the grid distortion method could be used locally to perform smooth small-scale changes. Other feature types and distortion function representations could be implemented. On top of the parametrization, various optimization techniques can be tested.

To combine the two ideas from this thesis, EnMSF update could be applied as an amplitude adjustment method along features specified in grid distortion. This focused update inside the feature driven fields would make the methods benefit from each other.

Bibliography

[1] Aanonsen, S. I., G. Nævdal, D. S. Oliver, A. C. Reynolds, and B. Vallès, *The ensemble Kalman filter in reservoir engineering - a review*, SPE Journal, SPE 117274, 2009.

[2] Alexander, G. D., J. A. Weinman, and J. L. Schols, *The Use of Digital Warping of Microwave Integrated Water Vapor Imagery to Improve Forecasts of Marine Extratropical Cyclones*, Monthly Weather Review, 126, 1469-1496, 1998.

[3] Anderson, J. L., *An adaptive covariance inflation error correction algorithm for ensemble filters*, Tellus, 59A, 210-224, 2007.

[4] Aziz, K., L. Durlofsky, and H. A. Tchelepi, *Notes on reservoir simulation*, Stanford University, 2005.

[5] Aziz, K., and A. Settari, *Petroleum reservoir simulation*, Applied Science Publishers, London, UK, 1979.

[6] Beezley, J. D., and J. Mandel, *Morphing Ensemble Kalman Filters*, arXiv:0705.3693v4 [math.DS], 2007.

[7] Bertino, L., G. Evensen, and H. Wackernagel, *Sequential Data Assimilation Techniques in Oceanography*, International Statistical Review / Revue Internationale de Statistique, v. 71, no 2, 223-241, 2003.

[8] Burgers, G., P. J. van Leeuwen, and G. Evensen, *Analysis Scheme in the ensemble Kalman filter*, Monthly Weather Review, v. 126(6), 1719-1724, 1998.

[9] Cabeen, K., and P. Gent, *Image Compression and the Discrete Cosine Transform*, http://online.redwoods.cc.ca.us/instruct/darnold/LAPROJ/Fall98/PKen/dct.pdf.

[10] Caers, J., *Petroleum Geostatistics*, Society of Petroleum Engineers, 2005.

[11] Caers, J., *Modeling Uncertainty in the Earth Sciences*, Wiley, 2011.

[12] Courtier, P., J. Derber, R. Errico, J. -F. Louis, and T. Vukicevic, *Important literature on the use of adjoint, variational methods and the Kalman filter in meteorology*, Tellus 45A(5), 342-357, 1993.

[13] Dee, D. P., *Simplification of the Kalman filter for meteorological data assimilation*, Q. J. R. Meteorol. Soc., v. 117, 365-384, 1991.

[14] Dorn, O., and R. Villegas, *History matching of petroleum reservoirs using a level set technique*, Inverse Problems, 24, 035015, 2008.

[15] Emerick, A. A., and A. C. Reynolds, *History matching a field case using the ensemble Kalman filter with covariance localization*, SPE 141216, SPE Reservoir Simulation Symposium, The Woodlands, Texas, U.S.A., 21-23 February 2011.

[16] Evensen, G., *Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics*, J. Geophys. Res., v. 99, pp. (10,143-10,162), 1994.

[17] Evensen, G., *The Ensemble Kalman Filter: theoretical formulation and practical implementation*, Ocean Dynamics, 53: 343-367, 2003.

[18] Evensen, G., *Sampling strategies and square root analysis schemes for the EnKF*, Ocean Dynamics, v. 54, 539-560, 2004.

[19] Evensen, G., *The ensemble Kalman filter for combined state and parameter estimation: Monte Carlo techniques for data assimilation for large systems*, Control Systems, IEEE, v. 29, issue 3, pp. 83 - 104, 2009.

[20] Evensen, G., *Data assimilation: the ensemble Kalman filter*, 2nd ed., Springer, Berlin, 2009.

[21] Evensen, G., and P. J. van Leeuwen, *Assimilation of geosat altimeter data for the Agulhas current using ensemble Kalman filter with quasi-geostrophic model*, Monthly Weather Review, v. 124, 85-96, 1996.

[22] Frakt, A. B., and A. S. Willsky, *Computationally efficient stochastic realization for internal multiscale autoregressive models*, Multidimensional systems and signal processes, v. 12, 109-142, 2001.

[23] Gao, G., and A. C. Reynolds, *Quantifying uncertainty for the PUNQ-S3 problem in a Bayesian settings with RML and EnKF*, SPE Reservoir simulation Symposium, SPE 93324, 2005.

[24] Gao, G., and A. C. Reynolds, *An Improved Implementation of the LBFGS Algorithm for Automatic History Matching*, SPE Journal, v. 11, no 1, 5-17, March 2006.

[25] Ghil, M., and P. Malanotte-Rizzoli, *Data assimilation in meteorology and oceanography*, Adv. Geophys., v. 33, 141-266, 1991.

[26] Gilleland, E., J. Lindström, and F. Lindgren, *Analyzing the Image Warp Forecast Verification Method on Precipitation Fields from the ICP*, Wea. Forecasting, 25, 1249-1262, 2010.

[27] Golub, G. H., and C. F. van Loan, *Matrix computations*, The Johns Hopkins University Press, 1983, 1989, 1996.

[28] Gu, Y., and D. S. Oliver, *History matching of the PUNQ-S3 reservoir model using the Ensemble Kalman filter*, SPE Annual Technical Conference and Exhibition, SPE 89942, 2004.

[29] Hamill, T. M., J. S. Whitaker, and C. Snyder, *Distance-dependent filtering of background error covariance estimates in an ensemble Kalman filter*, Monthly Weather Review, v. 129, 2001.

[30] Hanea, R. G., G. J. M. Velders, and A. W. Heemink, *Data assimilation of ground level ozone in Europe with a Kalman filter and chemistry transport model*, J. Geophys. Res., v. 109, 1-19, 2004.

[31] Hohl, D., E. A. Jimenez, and A. Datta-Gupta, *Field experiences with history matching an offshore turbiditic reservoir using inverse modeling*, SPE 101983, paper presented at SPE Annual Technical Conference and Exhibition, San Antonio, Texas, U.S.A., 24-27 September 2006.

[32] Houtekamer, P., and H. L. Mitchell, *Data assimilation using an Ensemble Kalman Filter technique*, Monthly Weather Review, v. 126, 796-811, 1998.

[33] Houtekamer, P., and H. L. Mitchell, *A sequential ensemble Kalman filter for atmospheric data assimilation*, Monthly Weather Review, v. 129, 123-137, 2001.

[34] Jafarpour, B., and D. McLaughlin, *History matching with an ensemble Kalman filter and discrete cosine parameterization*, Computational Geosciences, v. 12 (2), 227-244, 2008.

[35] Jafarpour, B., and D. McLaughlin, *Reservoir characterization with discrete cosine transform. Part 1: Parameterization*, SPE Journal, 14(1), 182-188, 2009.

[36] Jafarpour, B., and D. McLaughlin, *Reservoir characterization with discrete cosine transform. Part 2: History Matching*, SPE Journal, 14(1), 188-201, 2009.

[37] Jansen, J. D., *Systems theory for reservoir management. Lecture notes for course AES1490*, Delft University of Technology, 2011.

[38] Jazwinski, A. H., *Stochastic processes and filtering theory*, Dover Publications, Inc., 2007.

[39] Jégat, C., F. Carrat, C. Lajaunie, and H. Wackernagel, *Early detection and assessment of epidemics by particle filtering*, GeoENV VI - Geostatistics for Environmental Applications, A. Soares, M. J. Pereira, and R. Dimitrakopoulos, Eds., Springer, pp. 23-35, 2008.

[40] Kalman, R. E., *A new approach to linear filtering and prediction problems*, ASME, 59-IRD-11, 1960.

[41] Kaluza, R., A. Kostant, and W. A. Woyczynski, *Through a reporter's eyes: The Life of Stefan Banach*, Birkhäuser Boston, 1996.

[42] Keil, C., and G. C. Craig, *A Displacement and Amplitude Score Employing an Optical Flow Technique*, Wea. Forecasting, 24, 1297-1308, 2009.

[43] Khayam, S. A., *The Discrete Cosine Transform (DCT): Theory and Application*, http://www.egr.msu.edu/waves/people/Ali_files/DCT_TR802.pdf.

[44] Krantz, S. G., *How to teach mathematics*, American Mathematical Society, 1999.

[45] Lawniczak, W., R. Hanea, A. Heemink, and D. McLaughlin, *Multiscale ensemble filtering for reservoir engineering applications*, Computational Geosciences, 13:245-254, 2009.

[46] Lawniczak, W., R. G. Hanea, A. W. Heemink, D. McLaughlin, and J. D. Jansen, *History matching using a multiscale ensemble Kalman filter*, 11th European Conference on Mathematics in Oil Recovery, conference proceedings, Bergen, Norway, 2008.

[47] Le Dimet, F.-X., and O. Talagrand, *Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects*, Tellus A, v. 38A, issue 2, pp. 97-110, March 1986.

[48] Lewis, R. M., V. Torczon, and M. W. Trosset, *Direct search methods: then and now*, Journal of Computational and Applied Mathematics, v. 124, issues 1-2, pp. 191-207, 2000.

[49] Li, H., and J. Caers, *Geological modeling and history matching of multi-scale flow barriers in channelized reservoirs: methodology and application*, Petroleum Geoscience, v. 17, no 1, pp. 17-34, February 2011.

[50] Liu, N., and D. S. Oliver, *Critical evaluation of the Ensemble Kalman filter on history matching of geological facies*, SPE Reservoir Simulation Symposium, SPE 92867, 2005.

[51] Livings, D. M., S. L. Dance, and N. K. Nichols, *Unbiased ensemble square root filters*, Physica D, v. 237, issue 8, pp. 1021-1028, 2008.

[52] Lorenc, A. C., *The potential for the ensemble Kalman filter for NWP: a comparison with 4D-Var*, Quart. J. Roy. Meteor. Soc., v. 129, 3183 - 3203, 2003.

[53] Mandel, J., J. D. Beezley, L. Cobb, and A. Krishnamurthya, *Data driven computing by the morphing fast Fourier transform ensemble Kalman filter in epidemic spread simulations*, Procedia Computer Science, v. 1, issue 1, pp. 1221-1229, ICCS 2010, May 2010.

[54] Mandel, J., J. D. Beezley, and V. Y. Kondratenko, *Fast Fourier Transform Ensemble Kalman Filter with Application to a Coupled Atmosphere-Wildland Fire Model*, arXiv:1001.1588v1 [physics.ao-ph], 2010.

[55] Margulis, S. A., D. McLaughlin, D. Entekhabi, and S. Dune, *Land data assimilation and estimation of soil moisture using measurements from the southern great plains 1997 field experiment*, Water Resources Research, v. 38, 2002.

[56] McKay, M. D., W. J. Conover, and R. J. Beckman, *A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output From a Computer Code*, Technometrics, 21, 239-245, 1979.

[57] Mitchell, H. L., P. L. Houtekamer, and G. Pellerin, *Ensemble size, balance, and model-error representation in an ensemble Kalman filter*, Monthly Weather Review, v. 130, 2002.

[58] Moreno, D., S. I. Aanonsen, G. Evensen, and J. A. Skjervheim, *Channel Facies Estimation Based on Gaussian Perturbations in the EnKF*, paper presented at 11th European Conference on the Mathematics of Oil Recovery, Bergen, Norway, 8-11 September 2008.

[59] Nævdal, G., T. Mannset, and E. H. Vefring, *Instrumented wells and near well reservoir monitoring through ensemble Kalman filter*, Proceedings of 8th European Conference on Mathematics of Oil Recovery, 2002a.

[60] Nævdal, G., T. Mannset, and E. H. Vefring, *Near well reservoir monitoring through ensemble Kalman filter*, SPE Improved Oil Recovery Symposium, 2002b.

[61] Nævdal, G., L. M. Johnsen, S. Aanonsen, and E. H. Vefring, *Reservoir Monitoring and Continuous Model Updating Using Ensemble Kalman Filter*, SPE Journal, 66-74, March 2005.

[62] Oliver, D. S., and Y. Chen, *Recent progress on reservoir history matching: a review*, Computational Geosciences, DOI: 10.1007/s10596-010-9194-2, 2010.

[63] Oliver, D. S., A. C. Reynolds, and N. Liu, *Inverse theory for petroleum reservoir characterization and history matching*, Cambridge University Press, 2008.

[64] Osgood, B., *Lecture notes for EE 261. The Fourier Transform and its Applications*, Stanford University.

[65] Ott, E., B. R. Hunt, I. Szunyogh, A. V. Zimin, E. J. Kostelich, M. Corazo, E. Kalnay, D. J. Patil, and J. A. Yorke, *A local ensemble Kalman filter for atmospheric data assimilation*, Tellus A, v. 56, 415-428, 2004.

[66] Pajonk, O., M. Krosche, R. Schulze-Riegert, R. Niekamp, and H.G. Matthies, *Stochastic Optimization Using EA and EnKF - A Comparison*, 11th European Conference on the Mathematics of Oil Recovery, September 2008.

[67] Ravela, S., K. Emanuel, and D. McLaughlin, *Data assimilation by field alignment*, Physica D 230, 127-145, 2007.

[68] Remy, N., A. Boucher, J. Wu, *Applied geostatistics with SGeMS. A user's guide*, Cambridge University Press, Cambridge, UK, 2009.

[69] Rushton, K. R., *Groundwater hydrology: conceptual and computational models*, John Wiley and Sons, England, 2003.

[70] Sakov, P., and P. R. Oke, *Implications of the Form of the Ensemble Transformation in the Ensemble Square Root Filters*, Monthly Weather Review, v. 136, 1042-1053, 2008.

[71] Sakov, P., G. Evensen, and L. Bertino, *Asynchronous data assimilation with the EnKF*, Tellus, 62A, 24-29, 2010.

[72] Sarma, P., *Efficient closed-loop optimal control of petroleum reservoirs under uncertainty*, Stanford University, 2006.

[73] Schulze-Riegert, R., M. Krosche, and O. Pajonk, *Hybrid Optimization Coupling EnKF and Evolutionary Algorithms for History Matching: A Case Example* , SPE 121965-MS, EUROPEC/EAGE Conference and Exhibition, Amsterdam, The Netherlands, 8-11 June 2009.

[74] Segers, A., A. W. Heemink, M. Verlaan, and M. van Loon, *A modified RRSQRT-filter for assimilating data in atmospheric chemistry models*, Environmental Modeling and Software, v. 15, 663-671, 2000.

[75] Seiler, A., S. I. Aanonsen, G. Evensen, and O. Lia, *An elastic grid approach for fault uncertainty modeling and updating using the ensemble Kalman filter*, SPE 130422-PP, paper presented at SPE EUROPEC/EAGE Annual Conference and Exhibition, Barcelona, Spain, 14-17 June 2010, SPE/EAGE 2010.

[76] Seiler, A., J. C. Rivenæs, S. I. Aanonsen, and G. Evensen, *Structural uncertainty modeling and updating by production data integration*, SPE 125352, paper presented at SPE/EAGE Reservoir Characterization and Simulation Conference, Abu Dhabi, UAE, 19-21 October 2009.

[77] Skjervheim, J. A., G. Evensen, S. I. Aanonsen, B. O. Ruud, and T. A. Johansen, *Incorporating 4D seismic data in reservoir simulation models using Ensemble Kalman filter*, SPE Annual Technical Conference and Exhibition, SPE 95789, 2005.

[78] Skjervheim, J.-A., G. Evensen, J. Hove, and J. G. Vabø, *An Ensemble Smoother for assisted History Matching*, SPE Reservoir Simulation Symposium, The Woodlands, Texas, USA, SPE 141929, 2011.

[79] Strebelle, S., *Conditioning Simulation of Complex Structures Multiple-Point Statistics*, Mathematical Geology, v. 34, no 1, 2002.

[80] Tamamidis, P., and D. N. Assanis, *Generation of Orthogonal Grids with Control of Spacing*, Journal of Computational Physics, 94, 437-453, 1991.

[81] Thompson, J. F., Z.U.A. Warsi, and C. W. Mastin, *Review Article. Boundary-Fitted Coordinate Systems for Numerical Solution of Partial Differential Equations-A Review*, Journal of Computational Physics 47, l-108, 1982.

[82] Thompson, J. F., Z.U.A. Warsi, and C. W. Mastin, *Numerical Grid Generation. Foundations and Applications*, Elsevier Science Publishers, 1985.

[83] Tippett, M. K., J. L. Anderson, C. H. Bishop, T. M. Hamill, and J. S. Whitaker, *Ensemble Square Root Filters*, Monthly Weather Review, 131, 1485-1490, 2003.

[84] Van Doren, J., J. D. Jansen, P. M. J. Van den Hof, and O. H. Bosgra, *Structural Identifiability of Grid Block and Geological Parameters in Reservoir Simulation Models*, paper presented at 11th European Conference on the Mathematics of Oil Recovery, Bergen, Norway, 8-11 September 2008.

[85] Van Leeuwen, P. J., *Particle Filtering in Geophysical Systems*, Monthly Weather Review, 137, 4089-4114, 2009.

[86] Van Leeuwen, P. J., *Nonlinear data assimilation in geosciences: an extremely efficient particle filter*, Q. J. R. Meteorol. Soc. 136: 1991-1999, Part B, October 2010.

[87] Verlaan, M., and A. W. Heemink, *Data assimilation schemes for non-linear shallow water flow models*, Adv. Fluid Mechanics, v. 96, 277-286, 1996.

[88] Verlaan, M., and A. W. Heemink, *Tidal flow forecasting using reduced-rank square root filters*, Stochastic Hydro. Hydraul., v. 11, 349-368, 1997.

[89] Wang, K. Y., D. J. Lary, D. E. Shallcross, S. M. Hall, and J. A. Pyle, *A review on the use of the adjoint method in four-dimensional atmospheric-chemistry data assimilation*, Q. J. R. Meteorol. Soc., v. 127, 2181-2204, 2001.

[90] Wen, X. H., and W. H. Chen, *Real-time reservoir model updating using Ensemble Kalman filter*, SPE Reservoir Simulation Symposium, SPE 92991, 2005.

[91] Whitaker, J. S., G. P. Compo, and J. -N. Thépaut , *A comparison of variational and ensemble-based data assimilation systems for reanalysis of sparse observations*, Monthly Weather Review, 137, 1991-1999, 2009.

[92] Willsky, A. S., *Multiresolution Markov models for signal and image processing*, Proceedings of the IEEE, v. 90, 1396-1458, 2002.

[93] Xie, J., Y. Efendiev, and A. Datta-Gupta, *Uncertainty Quantification in History Matching of Channelized Reservoirs Using Markov Chain Level Set Approaches*, SPE 141811, SPE Reservoir Simulation Symposium, The Woodlands, Texas, USA, 21-23 February 2011.

[94] Xie, J., A. Mondal, Y. Efendiev, B. Mallick, and A. Datta-Gupta, *History Matching Channelized Reservoirs Using Reversible Jump Markov Chain Monte Carlo Methods*, SPE 129685, SPE Improved Oil Recovery Symposium, Tulsa, Oklahoma, USA, 24-28 April 2010.

[95] Yang, C., and S. Ravela, *Spectral Control of Viscous Alignment for Deformation Invariant Image Matching*, Proc. ICCV, pp. 1303-1310, Kyoto, Japan, 2009.

[96] Zafari, M., and A. C. Reynolds, *Assessing the uncertainty in reservoir description and performance predictions with the ensemble Kalman filter*, SPE 95750, prepared for presentation at the 2005 SPE Annual Technical Conference and Exhibition held in Dallas, Texas, U.S.A., 9-12 October 2005.

[97] Zhang, F., A. C. Reynolds, and D. S. Oliver, *Evaluation of the Reduction in Uncertainty Obtained by Conditioning a 3D Stochastic Channel to Multiwell Pressure Data*, Mathematical Geology, v. 34, no 6, 2002.

[98] Zhou, H., J. Gomez-Hernandez, H.J. Hendricks Franssen and L. Li, *Handling Non-Gaussian distributions with Ensemble Kalman Filter*, Advances in Water Resources 34(7), 844-864, 2011.

[99] Zhou, Y., D. McLaughlin, D. Entekhabi, and G. H. C. Ng, *An ensemble multiscale filter for large nonlinear data assimilation problems*, Monthly Weather Review, v. 136(2), 678-698, 2007.

[100] Zinner, T., H. Mannstein, and A. Tafferner, *Cb-TRAM: Tracking and monitoring severe convection from onset over rapid development to mature phase using multi-channel Meteosat-8 SEVIRI data*, Meteorol. Atmos. Phys. 101, 191-210, 2008.

Summary

A reservoir simulator mimics the movement of fluids in the presence of each other through a porous medium under some specified conditions (e.g. temperature, depth, initial pressure or initial saturation). It is a numerical model of a real-life physical process, therefore, subject to uncertainty. Some uncertainties can be lowered by improving model-parameter estimates. This is where data assimilation plays an important role. Automated data assimilation, using sophisticated techniques, is a widely researched topic in today's applied science.

We investigated two research topics in data assimilation that are closely connected to the area of image processing. Images are an integral part of reservoir engineering application in the form of property or variable fields. Reservoir engineering, image processing and data assimilation are the leading themes of this thesis.

First, we applied an ensemble multiscale filter as a permeability estimator for one update step and later as a full data assimilation experiment. We summarized by investigating properties of different covariance matrix approximations obtained from the ensemble multiscale filter. We concluded that the filter can be an efficient localizing tool especially for spatially large observations.

The ensemble Kalman filter is a sequential Monte-Carlo approach that uses an ensemble of reservoir models. For realistic large scale applications the ensemble size needs to be kept small due to computational inefficiency. Consequently, the error space is not well covered (poor cross-correlation matrix approximations) and the updated parameter field becomes scattered and loses important geological features. The prior geological knowledge present at the initial time is not found in the final updated parameter any more.

We propose a new approach to overcome some of the ensemble Kalman filter limitations. We show the specifications and results of the ensemble multiscale filter for automatic history matching. The ensemble multiscale filter replaces, at each update time, the prior forecasted covariance with a multiscale tree. The global dependence is preserved via the parent-child relation in the tree. After constructing the tree the Kalman update is performed.

The ensemble multiscale filter is a different way to represent the covariance of an ensemble. The computations are done on a tree structure and are based on an ensemble of possible realizations of the states and/or parameters of interest. The original ensemble is partitioned between the nodes of the finest scale in the tree. A construction of the tree is led by the eigenvalue decomposition. Then, the state combinations with the greatest corresponding eigenvalues are kept on the higher scales.

The properties of the ensemble multiscale filter were presented with a 2D, two phase (oil and water) twin experiment, and the results are compared to the ensemble Kalman filter. The advantages of using the ensemble multiscale filter are: localization in space and scale, adaptability to prior information, and efficiency in case many measurements are available. These advantages make the ensemble multiscale filter a practical tool for data assimilation problems. The updated states/parameters using the ensemble multiscale filter are believed to keep geological structure due to localization property.

A comparison of covariance matrices obtained with different setups used in the EnMSF is presented. This sensitivity study is necessary since there are many parameters in the algorithm which can be adjusted to the needs of an application; they are connected to the tree construction part.

The localization property is discussed based on the example where the filter is run with a simple simulator and a binary ensemble is used (the pixels in the replicates of permeability can have one of only two values).

Second, we developed a grid deformation technique inspired by grid generation and image warping methods. Our implementation was defined such that the deformation was smooth, global and led by just a few parameters. The grid distortion was used to adjust a position of high-permeable channels in the permeability field through data assimilation or a direct search method. We presented two- and three-dimensional versions of the method in reservoir and groundwater flow models, and concluded that the grid distortion proved cost efficient and effective in terms of time and performance.

Data assimilation in hydrocarbon reservoir engineering involves adjusting the reservoir model parameters such that the simulated model response matches the measured historical response within the prescribed uncertainty bounds. During this process the model parameters are often changed to such an extent that the resulting model lacks geological realism. This is in particular the case when the original model contains high-permeability channels, i.e. elongated features extending over large distances, which are broken up after the parameter update.

We propose to avoid such a loss of geological realism with a parameterization based on grid deformation which maintains the continuity of geological features in the initial model. The parameters determine a smooth distortion of the grid that defines the original geological pattern, using deformation functions in the form of Poisson's equation defined on the grid coordinates. The grid distortion drives the underlying geological-image distortion.

We tested the method using a twin experiment involving history matching of production data generated during waterflooding of a simple two-dimensional two-phase (oil-water) reservoir with high-permeable streaks extending over the entire domain. We compared the results against those obtained by using a pixel-based updating method in which the connectivity of the original channels was destroyed during the parameter updating process. On the contrary, in our method geological continuity was successfully maintained. Moreover, we found an improved capacity of the geologically realistic models to predict the future reservoir response compared to a poor predictive capability of the geologically unrealistic models obtained through the pixel-based updating.

Samenvatting

Een reservoirsimulator modelleert de vloeistofbeweging in de aanwezigheid van elkaar door een poreus medium onder bepaalde voorwaarden (bijv. temperatuur, diepte, initiële spanning of initiële verzadiging). Het is een numeriek model van een real-life fysieke proces, dus onzeker. Sommige onzekerheden kunnen worden verlaagd door betere schattingen van modelparameters. Dit is waar data-assimilatie een belangrijke rol speelt. Geautomatiseerde data-assimilatie, gebruikmakend van geavanceerde technieken, is een zeer onderzocht onderwerp in tegenwoordige toegepaste wetenschappen.

We onderzochten twee onderzoekthema's in data-assimilatie, die nauw met het gebied van beeldverwerking verbonden zijn. Beelden zijn een integraal onderdeel van reservoirengineering toepassing in de vorm van eigenschap of variabele velden. Reservoirengineering, beeldverwerking en data-assimilatie zijn hoofdthema's van deze scriptie.

Ten eerste pasten we toe een ensemble multiscale filter als een permeabiliteitschatter voor één bijwerkingstap en later als een volledige data-assimilatie experiment. We vatten samen met onderzoeking van eigenschappen van verschillende schattingen van covariantiematrix afkomstig van het ensemble multiscale filter. Wij concludeerden dat het filter een efficiënte lokalisatiemiddel kan zijn vooral voor ruimtelijk grote observaties.

Het ensemble Kalman filter is een sequentiële Monte-Carlo benadering die gebruikt een ensemble van reservoirmodellen. Voor realistische grootschalige toepassingen moet het ensemblegrootte klein worden gehouden door rekeninefficiëntie. Als gevolg van is de foutruimte niet goed gedekt (slechte schattingen van cross-correlatie matrix) en de bijgewerkte parameterveld wordt verspreid en verliest belangrijke geologische kenmerken. De prior geologische kennis aanwezig op de initiële tijd wordt in de laatste bijgewerkte parameter niet meer gevonden.

We stellen een nieuwe benadering voor om een aantal beperkingen van ensemble Kalman filter te overwinnen. We tonen de specificaties en resultaten van het ensemble multiscale filter voor automatische geschiedenis matching. Het ensemble multiscale filter vervangt, op elke bijwerking tijd, de prior voorspelde covariantie met een multiscale boom. De globale afhankelijkheid wordt door de parent-child relatie in de boom bewaard. Na constructie van de boom wordt de Kalman bijwerking uitgevoerd.

Het ensemble multiscale filter is een andere manier om de covariantie van een ensemble weer te geven. De berekeningen worden gedaan op een boomstructuur en worden gebaseerd op een ensemble van mogelijke realisaties van de beschouwde staten en/of parameters. Het oorspronkelijke ensemble wordt tussen de knoppunten van de fijnste schaal in de boom verdeeld. Een constructie van de boom wordt door de eigenwaardedecompositie geleid. Daarna worden de combinaties van staten met de grootste corresponderende eigenwaarden op de hogere schalen gehouden.

De eigenschappen van het ensemble multiscale filter zijn met een 2D, twee fase (olie en water) dubbel experiment weergegeven, en de resultaten worden met het ensemble Kalman filter vergeleken. De voordelen van gebruik van het ensemble multiscale filter zijn: lokalisatie in de ruimte en schaal, aanpassingsvermogen aan prior informatie, en efficiency indien veel metingen beschikbaar zijn. Deze voordelen maken het ensemble multiscale filter een praktisch middel voor data-assimilatie problemen. Gebruikmakend van het ensemble multiscale filter houden de bijgewerkte staten/parameters geologische structuur vanwege lokalisatieeigendom.

Een vergelijking wordt gepresenteerd van covariantiematrices verkregen met verschillende instellingen gebruikt in de EnMSF. Deze gevoeligheidsanalyse is noodzakelijk omdat er veel parameters in het algoritme zijn die aan de behoeften van een toepassing kunnen worden aangepast; ze worden met de boomconstructie onderdeel verbonden.

De lokalisatieeigendom wordt besproken met behulp van het voorbeeld waarin het filter met een simpele simulator uitgevoerd wordt en een binaire ensemble gebruikt wordt (de pixels in de permeabiliteitreplieken kunnen één van slechts twee waarden hebben).

Ten tweede ontwikkelden we een techniek voor roostervervorming, dat door roostergeneratie en beeldverbuiging methoden geinspireerd wordt. Onze uitvoering is zodanig gedefinieerd dat de vervorming smooth, globaal en geleid door enkele parameters was. De roostervervorming is gebruikt om een positie van hoge permeabiliteitkanalen in het permeabiliteitveld door data-assimilatie of een directe zoekmethode aan te passen. Wij presenteerden twee- en driedimensionale versies van de methode in reservoir en grondwaterstromingsmodellen, en concludeerden dat de roostervervorming in termen van tijd en prestatie kostenefficiënt en effectief is.

Data-assimilatie in koolwaterstof reservoirengineering betrekt aanpassing van het reservoir modelparameters zodanig de gesimuleerde modeluitkomst met de gemeten historische uitkomst binnen de voorgeschreven onzekerheidgrenzen overeenkomt. Tijdens deze processen worden de modelparameters vaak gewijzigd zodanig geologisch realisme aan het resulterende model ontbreekt. Dit is met name het geval als het oorspronkelijke model hoge permeabiliteitkanalen bevat, namelijk uitgestrekte kenmerken over grote afstanden, die na de parameterbijwerking gebroken worden.

We stellen voor om te voorkomen een dergelijk verlies van geologisch realisme door een parametersering gebaseerd op roostervervorming, die de continuiteit van geologische eigenschappen van het initiële model handhaaf. De parameters bepalen een smooth vervorming van de rooster, die de oorspronkelijke geologische patroon definieert, gebruikmakend van vervormingfuncties in de vorm van Poisson vergelijking gedefinieerd op de roostercoördinaten. De roostervervorming bepaalt de onderliggende geologische beeldvervorming.

Wij testten de methode met behulp van een dubbele experiment betrekking geschiedenis matching van productiedata gegenereerd tijdens wateroverstroming van een eenvoudige twee-dimensionale twee-fase (olie-water) reservoir met hoge permeabele strepen verbreiden over the hele domein. We vergeleken de resultaten met die verkregen door gebruik van een pixel-gebaseerde actualiseringmethode waarin de connectiviteit van de oorspronkelijke kanalen tijdens het bewerkingsproces van de parameters vernietigd werd. Integendeel is geologische continuiteit in onze methode met succes gehandhaafd. Bovendien vonden we een verbeterde capaciteit van de geologisch realistische modellen voor het voorspellen van de toekomstige reservoiruitvoer vergelijking met een slechte voorspellingcapaciteit van de geologisch onrealistische modellen verkregen door de pixel-gebaseerde bijwerking.

Curriculum Vitae

Wiktoria Ławniczak was born on September 27, 1983 in Zielona Góra, Poland. She attended a high school in Zielona Góra (V Liceum Ogólnokształcące) and followed a master's program at University of Zielona Góra, both with specialization in mathematics. She obtained a second master's degree from Delft University of Technology in applied mathematics. She continued in applied mathematics as a Ph.D. candidate at Delft University of Technology.