



Norwegian University of
Science and Technology

Estimating the Value of Information Using Closed Loop Reservoir Management of Capacitance Resistive Models

**Audun Daniel Dørum
Stensgaard**

Petroleum Geoscience and Engineering

Submission date: June 2016

Supervisor: Reidar B Bratvold, IPT

Norwegian University of Science and Technology

Department of Petroleum Engineering and Applied Geophysics

Acknowledgements

I appreciate very much the help and guidance from my supervisor Professor Reidar Bratvold, who also provided interesting project assignments for both the spezialization project and this thesis. I must also thank Aojie Hong, who took the time to help me in person, over skype and through emails. The implementation of both the ensemble Kalman filter and the optimization, which are the most essential parts of this project, would not have been possible without his help.

Summary

The optimal goal of reservoir management is to achieve the highest possible recovery factor for the lowest possible cost. Changes in injection control have the potential to cause significant gain in net present value. In order to set the injection control that gives the highest possible net present value, it is important to have detailed reservoir description. Gathering information is expensive and it is difficult to determine its worth.

A value-of-information (VOI) analysis can be helpful to identify when and if information should be gathered. This is a powerful method that can be used to identify and exclude alternatives in a decision context.

One of the two purposes in this thesis, is to evaluate the usability of a closed loop reservoir management (CLRM) structure using capacitance resistive models (CRM) to estimate VOI.

A CLRM structure for CRM was implemented in MATLAB. This structure used the ensemble Kalman filter (EnKF) for history matching and the ensemble optimization (EnOpt) for optimizing injection.

The EnKF successfully identified the reservoir flow pattern in reservoir models, whilst the EnOpt successfully increased the objective value after new information became available.

The CLRM with the CRMs were shown to be suitable to estimate VOI, although the slow computer speed limits the usability for more complex reservoir models.

The other purpose of this thesis is to provide information on the CRM, oil fractional flow models, the EnKF, the EnOpt and the CLRM, to ease the way for future research students.

Sammendrag

Målet med reservoarstyring er å oppnå høyest mulig utvinningsgrad for lavest mulig kostnad. Forandringer i injeksjonen har potensial til å utgjøre store forskjeller i inntekter. For å kunne oppnå høyest mulig nåverdi basert på injeksjonskontroll, er det viktig å ha god kunnskap om reservoarbeskrivelsen. Innhenting av slik informasjon er dyrt. Det er i tillegg vanskelig å fastsette hva informasjonen er verdt på forhånd.

En analyse om verdien av informasjon (VOI) kan hjelpe å bestemme når og hvis man bør innhente informasjon.

Ett av to hovedmål i dette prosjektet, er å evaluere bruken av en closed loop reservoir management-struktur ved å bruke *capacitance resistive models* (CRM) for å estimere VOI.

En CLRM-strukture ble implementert i MATLAB. Denne strukturen brukte *ensemble Kalman filter* (EnKF) til historiematching, og *ensemble optimization* (EnOpt) til å optimalisere injeksjonen.

Strømningsmønsteret til en reservoarmodell ble identifisert av EnKF, mens EnOpt ga en høyere objektivverdi etter hvert som mer informasjon ble tilgjengelig. Bruken av CRM ble vurdert til å fungere bra i CLRM-strukturen, men programhastigheten var for treg til å bli brukt for mer komplekse reservoarmodeller.

Det andre hovedmålet i dette prosjektet, er å presentere informasjon om de ulike delene i CLRM-strukturen, slik at oppgaven kan bli brukt som en introduksjon for studenter som vil arbeide med samme temaer.

Table of Contents

Acknowledgements	i
Summary	iii
Sammendrag	iv
Table of Contents	vii
List of Figures	xii
Abbreviations	xiii
1 Introduction	1
1.1 Background	2
2 Statistics	5
2.1 Probability density function	5
2.2 Bayesian updating	6
3 Reservoir Models	9
3.1 First Reservoir, 5 Injectors and 4 Producers	9
3.2 Second Reservoir, 1 Injector and 1 Producer	11
4 Capacitance Resistive Modeling	13
4.1 Introduction	13
4.2 Background	13
4.3 Tank Based Representation	15
4.4 Producer Based Representation	17
4.5 Injector-Producer Pair Based Representation	19
4.6 Oil Fractional Flow Models	21

4.6.1	Gentil's Fractional Flow Model	22
4.6.2	Buckley Leverett Based Fractional Flow Model	24
4.7	Conclusion	26
4.8	CRM and Oil Fraction Formulas	27
5	Kalman Filter	29
5.1	Introduction Kalman Filter	29
5.2	Background Kalman Filter	29
5.3	Linear Kalman Filter	30
5.4	Extended Kalman Filter	31
5.5	Ensemble Kalman Filter	32
5.5.1	Method	32
5.5.2	Limitations of the Ensemble Kalman Filter	35
5.6	Examples	35
5.6.1	CRMP with Gentil's Fractional Flow	35
5.6.2	CRMP and CRMIP	39
5.7	Discussion	41
5.8	Conclusion	42
5.9	Ensemble Kalman Filter Algorithm	43
6	Ensemble Optimization	45
6.1	Introduction	45
6.2	Background	45
6.3	Method	46
6.4	Example Using the CRMP and Gentils Fractional Flow Model	48
6.4.1	Method	48
6.4.2	Results	49
6.4.3	Discussion	50
6.5	Conclusion	51
6.6	Ensemble Optimization Algorithm	52
7	Closed Loop Reservoir Management	53
7.1	Introduction	53
7.2	Background	53
7.2.1	Value of Information	54
7.3	CLRM for VOI	55
7.3.1	CLRM for VOI Structure	56
7.4	Case with CRMT and BLBFFM	58
7.4.1	Methods	58

7.4.2	Results	59
7.4.3	Discussion	60
7.5	Example CRMP and Gentils Fractional Flow Model	60
7.5.1	Methods	60
7.5.2	Results	61
7.5.3	Discussion	61
7.6	Implementation Challenges	62
7.7	Conclusion	62
8	Further Work	65
	References	67
	Appendix A	73
	Appendix B	87

List of Figures

2.1	Normal or Gaussian distribution with $E(\text{red}) > E(\text{green}) = E(\text{blue})$ and $\text{Var}(\text{red}) > \text{Var}(\text{green}) > \text{Var}(\text{blue})$	6
3.1	Reservoir with 5 injection wells and 4 production wells. The reservoir contains two high permeability streaks, between I1 and P1, and between I3 and P4. . . .	9
3.2	Injection rates	10
3.3	Production rates estimated by an Eclipse simulation with $P_{ri} = 1450$ psia. These rates will be used as real observed data.	11
3.4	Reservoir with 1 injection well and 1 production well.	11
3.5	Injection rate.	12
3.6	Production rate estimated by an Eclipse simulation with $P_{ri} = 1450$ psia. . . .	12
4.1	Schematic representation of the CRMT. This image was originally used in Weber (2009)[1].	15
4.2	Estimated rates by the CRMT and observed rates. ($R^2 = 0.8697$).	17
4.3	Schematic representation of the control volume around a producer, as used by the CRMP. This image was originally used in Weber (2009)[1].	18
4.4	Estimated rates by the CRMP and observed rates. ($R^2 = 0.9184$).	19
4.5	Approximate connectivities by the CRMP	20
4.6	Schematic representation of the control volume between an injector-producer pair, as used by CRMIP. This image was originally used in Weber (2009)[1]. . .	20
4.7	Estimated rates by the CRMIP and observed rates. ($R^2 = 0.9975$).	22
4.8	Liquid production rates for each individual production well. Orange rates are the estimated rates, while the blue lines are the observed (Eclipse simulated) rates.	22
4.9	Logarithmic plot water oil ratio and cumulative water injection for the reservoir in figure 3.1 on page 9.	24
4.10	Estimated and observed oil rate using the CRMP. ($R^2 = 0.8911$)	24

4.11	Estimated and observed values for both liquid and oil production rate in producer 4 for the reservoir in figure 3.1 on page 9.	24
4.12	Estimated and observed liquid and oil production rates for the second reservoir (figure 3.4 on page 11).	26
5.1	Connectivities f_{ij} plotted against time. $N = 10000$	37
5.2	Time constants τ_j plotted against time. $N = 10000$	37
5.3	The average α_j plotted against time. $N = 10000$	37
5.4	The average β_j plotted against time. $N = 10000$	38
5.5	Total production rate after history matching plotted against time. The slight mismatch in oil rate at around 2000 days is caused by a rapid increase in injector 1.	38
5.6	Production rate at producers P1 and P4 after history matching plotted against time.	38
5.7	Liquid and oil rate after history matching with the EnKF. The black lines are the true observations of liquid rate and oil rate, while the dots are the rates when using the average model parameter values. The top left plots shows the initial rates without any history matching. The top right, the lower left and the lower right shows the rates after 10, 20 and 30 time steps of history matching, respectively. $N = 1000$	39
5.8	The average difference (error) in connectivity plotted against the number of time steps used in the EnKF history matching. $N = 10$	40
5.9	The average difference (error) in connectivity plotted against the number of time steps used in the EnKF history matching. $N = 100$	40
5.10	The average difference (error) in connectivity plotted against the number of time steps used in the EnKF history matching. $N = 1000$	41
6.1	Model parameters generated by a VBA implementation of CRMP and Gentil's fractional flow model.	49
6.2	Optimized injection rates.	50
6.3	Total production rate and total oil rate before and after optimization.	50
7.1	Closed loop reservoir control management. This figure was originally used in Brouwer et al. (2004)[2].	54
7.2	Closed loop reservoir management structure used to estimate expected value of information. The figure was originally used in Barros et al. (2015)[3]. The step numbers were not included in the original figure and was edited in later to be used in this master thesis.	56
7.3	Expected Value of Information (black dots) and Expected Value of Clairvoyance (blue dots) plotted against the day in which the information was gathered. . . .	59

7.4	Expected Value of Information (black dots) and Expected Value of Clairvoyance (blue dots) for information of producer 1 plotted against the day in which the information was gathered. $N = 30$, $N_r = 11$	61
8.1	Production rates without any knowledge of the model parameters. For graphs where observed liquid rate are not visible, it is approximately equal to the estimated oil rate.	73
8.2	Production rates after history matching with the EnKF for the first 5 time steps. For graphs where observed liquid rate are not visible, it is approximately equal to the estimated oil rate.	74
8.3	Production rates after history matching with the EnKF for the first 10 time steps. For graphs where observed liquid rate are not visible, it is approximately equal to the estimated oil rate.	75
8.4	Production rates after history matching with the EnKF for the first 20 time steps. For graphs where observed liquid rate are not visible, it is approximately equal to the estimated oil rate.	76
8.5	Production rates after history matching with the EnKF for all 60 time steps. . .	77
8.6	CRMP connectivity solved by Excels GRG nonlinear solver.	78
8.7	CRMIP connectivity solved by Excels GRG nonlinear solver.	78
8.8	The initial range of liquid rate (blue) and oil rate (red) when using the EnKF. The black lines are the true observations of liquid rate and oil rate, while the dots are the rates when using the average model parameter values. $N = 1000$.	79
8.9	Liquid rate (blue) and oil rate (red) when using the EnKF after 10 time steps. The black lines are the true observations of liquid rate and oil rate, while the dots are the rates when using the average model parameter values. $N = 1000$.	80
8.10	Liquid rate (blue) and oil rate (red) when using the EnKF after 20 time steps. The black lines are the true observations of liquid rate and oil rate, while the dots are the rates when using the average model parameter values. $N = 1000$.	81
8.11	Liquid rate (blue) and oil rate (red) when using the EnKF after 30 time steps. The black lines are the true observations of liquid rate and oil rate, while the dots are the rates when using the average model parameter values. $N = 1000$.	82
8.12	Liquid rate (blue) and oil rate (red) when using the EnKF after 40 time steps. The black lines are the true observations of liquid rate and oil rate, while the dots are the rates when using the average model parameter values. $N = 1000$.	83
8.13	Liquid rate (blue) and oil rate (red) when using the EnKF after 50 time steps. The black lines are the true observations of liquid rate and oil rate, while the dots are the rates when using the average model parameter values. $N = 1000$.	84

-
- 8.14 Liquid rate (blue) and oil rate (red) when using the EnKF after 60 time steps. The black lines are the true observations of liquid rate and oil rate, while the dots are the rates when using the average model parameter values. $N = 1000$. 85
- 8.15 Expected VOI (black) and expected VOC (blue) estimation using CRMIP and Gentils fractional flow model for information from producer 1. The bad result shows the necessity of large enough model parameter sets. $N = 10, N_r = 11$. 86

Abbreviations

α	fitting parameter or step length
β	fitting parameter for fractional flow
c_t	total compressibility
d_{ij}	interwell distance
f	connectivity
f_o	oil fraction
f_w	water fraction
I	water injection rate
J	productivity index or objective
k	permeability
m	relative permeability curve exponent
M	endpoint mobility ratio
μ	viscosity
n	relative permeability curve exponent
N_{inj}	number of injection wells
N_{prod}	number of production wells
N_t	number of total time steps
p_{wf}	bottom hole flowing pressure
q	rate
q_o	oil production rate
q_w	water production rate
R^2	coefficient of determination
S	normalized water saturation
S_{or}	residual oil saturation
S_w	water saturation
S_{wir}	irreducible water saturation
t	time
τ	time constant
V_P	pore volume
W	cumulative water injected
b	discount rate
x	state
z	measurement
K	Kalman gain
u	control vector

Introduction

In the last decades there have been a huge increase in computational development, which has further increased the usability of reservoir simulation. It now is an indispensable tool for supporting reservoir management decisions.

In order to make good predictions of future reservoir behavior, there is a need for detailed description and value of reservoir parameters. Gathering information through means such as coring, seismic surveys or using logging tools is expensive and it is difficult to determine how much the information is worth.

History matching provides a powerful and commonly used means for gathering information related to both the static and dynamic parameters of reservoir simulators and their use for production prediction. In parallel with increasing computational power, there has been tremendous improvements in the solutions and use of algorithms to solve the inverse problem[4, 5, 6, 7]. These methods have been demonstrated to reliably provide high-quality history matches resulting in robust determination of the reservoir simulation input parameters.

In parallel with the development of improved history matching and optimization methods, the oil gas industry has seen an increase in the use of value-of-information (VOI) analysis to determine if and what type of information should be gathered to reduce key uncertainties related to decision making[8, 9, 10, 11]. However, with a few exceptions, VOI analysis has not been applied to information gathering from history matching.

A VOI analysis of history matching requires probabilistic optimization. For such an optimization approach to be robust, the reservoir simulator must be run a large number of times for each test of optimal values.

Commonly used grid-based simulators, such as Eclipse, are computationally expensive and it may not be feasible in a VOI analysis.

There are two main goals of this thesis. The first is to evaluate the usability of capacitance resistive models in closed loop reservoir management and for estimating the value of information from history matching. The second is to ease the way for future research students by

providing information about the capacitance resistive models, oil fractional flow models, the ensemble Kalman filter, ensemble optimization and a closed loop reservoir management structure.

The rest of the chapters in this thesis consists of the following.

Chapter 2 consists of some basic fundamental statistics.

In chapter 3, two versions of a reservoir model will be presented, using Eclipse.

Chapter 4 describes the different versions of the capacitance resistive models, two different oil fraction models, and how they are solved.

Chapter 5 introduces some background on Kalman filtering, presents the structure of the linear Kalman filter and briefly the extended Kalman filter. In section 5.5, there is some more information on the ensemble Kalman filter (EnKF). The structure of the EnKF can be found in section 5.5.1, and a few examples using the EnKF will be provided.

Chapter 6 presents the ensemble optimization technique, along with an example.

In chapter 7 presents a closed loop reservoir management structure and then demonstrate its use with regards to value of information. The structure will be implemented with different CRMs and different oil fractional flow models.

Much of the information in this thesis, especially in chapter 4, is based on my specialization project[12] of last semester.

1.1 Background

There are four main topics that needs to be researched to successfully solve the project goals.

1. As part of my specialization project, fall 2015[12], I implemented the tank-based, the producer based and the injector-producer pair based representations of the capacitance resistive model, as well as the Gentil fractional flow model and the Buckley Leverett based fractional flow model in Excel using VBA. The implemented models were then used to evaluate the capability for predicting future production.

Early work on capacitance resistive models in the oil and gas industry, can be traced back to Albertoni and Lake (2003)[13], who presented a method that described reservoir flow patterns through interwell connectivities, using only injection and production data. This method has later been expanded and improved[14, 15, 16] and may today be helpful for real life day-to-day reservoir management.

2. The ensemble Kalman filter (EnKF) is a recursive filtering technique, first introduced in Evensen (1994)[6], which is based on the linear Kalman filter (KF), but uses ensembles to be able to solve for nonlinear systems. It is a promising method that can be used to history match model parameters with relatively low computational costs compared to other alternative methods[17], and can therefore include a larger amount of variables.

3. Ensemble optimization (EnOpt) uses an ensemble of control variables and covariance matrices to determine optimal injection control, based on a defined objective, most common net present value.
4. Barros et al. 2015[3] proposed a method where a closed loop reservoir management structure was used to estimate value of information (VOI). They presented a way to estimate VOI without knowing how the model parameters related to each other. They only assumed that the individual parameters followed a normal distribution with a known mean and a known standard deviation.

This structure is used in this thesis for the same purpose. The paper by Barros et al. (2015)[3], suggest a novel and innovative way of assessing VOI for production data. In this work we are extending the approach through the use of a closed loop structure together with capacitance resistive models.

Statistics

It is important to have an understanding of basic statistics before working with Kalman filtering. This chapter will provide information of basic statistics.

2.1 Probability density function

The probability density function $p(x)$ for a continuous random variable x must be constrained by[18]:

$$p(x) \geq 0 \quad \int_{\Omega} p(x)dx = 1 \quad (2.1)$$

where Ω is the sample space of x .

For a discrete variable x , the distribution must follow:

$$p(x) \geq 0 \quad \sum_{x \in \Omega} p(x) = 1 \quad (2.2)$$

Gaussian distribution, also called normal distribution is a fundamental assumption in the use of Kalman filtering and can be written as:

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (2.3)$$

where the mean μ for continuous case is:

$$E(X) = \mu = \int_{x \in \Omega} xp(x)dx \quad (2.4)$$

and for discrete case is:

$$E(X) = \sum_{x \in \Omega} xp(x) \quad (2.5)$$

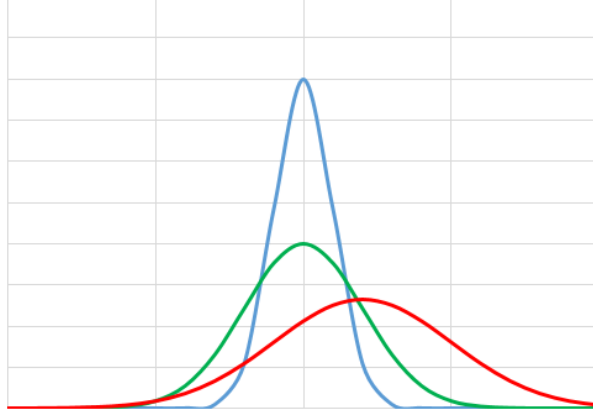


Figure 2.1: Normal or Gaussian distribution with $E(\text{red}) > E(\text{green}) = E(\text{blue})$ and $\text{Var}(\text{red}) > \text{Var}(\text{green}) > \text{Var}(\text{blue})$.

The standard deviation is noted as σ , and the variance is defined as:

$$\text{Var}(x) = \sigma^2 = \int_{x \in \Omega} (x - \mu)^2 p(x) dx \quad (2.6)$$

or

$$\text{Var}(X) = E[(X - E(X))^2] = E(X^2) - E(X)^2 \quad (2.7)$$

For a a probability density function consisting of two random variables X and Y , the covariance becomes:

$$\text{Cov}(X, Y) = E[(X - E(X))(Y - E(Y))^T] \quad (2.8)$$

2.2 Bayesian updating

Bayesian updating is maybe the best known iterative procedure that can be used for a history matching. It is a way of updating a predicted state, based on new information using probability density functions.

Bayes' theorem is defined[19] as:

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)} \quad (2.9)$$

where $P(B)$ is the prior probability of an event B . $P(B|A)$ is the posterior probability of B given A , $P(A|B)$ is the likelihood of A given B and $P(A)$ is the probability of event A .

Bayes' theorem can be used iteratively when more information of outcomes becomes available. If A was not a single event but instead multiple events $A = A_1, A_2, \dots, A_N$, then the estimate of B may be updated for each event iteratively, by using the posterior as the new prior

for the next step.

Bayes theorem follows the behavior[20]:

$$P(B|A) \propto P(A|B)P(B) \quad (2.10)$$

To implement Bayes updating, one first have to define a prior probability $P(B)$. The likelihood $P(A_1|B)$ is here defined as a Gaussian density function around B . The posterior is estimated by normalizing $P(A_1|B)P(B)$. The new prior is set $P(B) = P(B|A_1)$, and the next iteration becomes: $P(B|A_2) \propto P(A_2|B)P(B)$. This process is repeated until A_N .

When working with large data set and many state variables, directly applying the Bayesian approach as discussed here is not computationally feasible. Techniques like Kalman filtering have been derived with the purpose of reducing the computational requirements.

Reservoir Models

In this work, two different scenarios of a simple reservoir model has been implemented in Eclipse. The reason for implementing these in Eclipse, is to use the resulting production as a true reservoir production. The reservoir models will be used in later chapters for the purpose of history matching, optimization and closed loop reservoir control. This chapter provides a description of each scenario, as well as the base case injection. Most of the information and illustrations in this chapter was developed as a part of the specialization project TPG4530 at NTNU during fall of 2015[12].

3.1 First Reservoir, 5 Injectors and 4 Producers

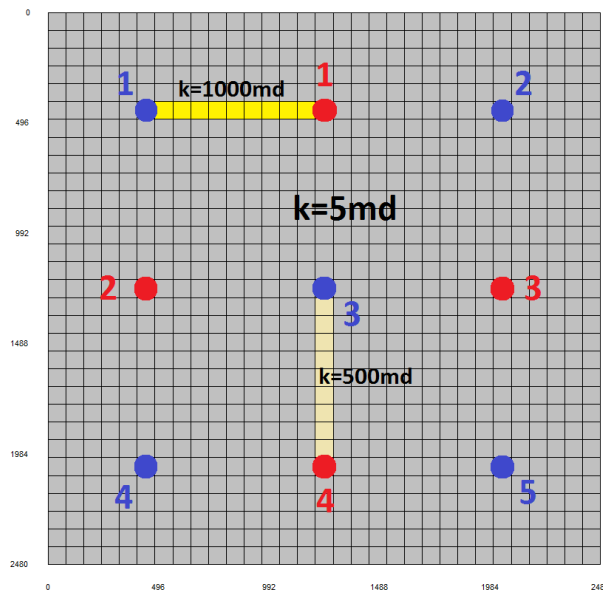


Figure 3.1: Reservoir with 5 injection wells and 4 production wells. The reservoir contains two high permeability streaks, between I1 and P1, and between I3 and P4.

This is the same model used by Albertoni and Lake (2003)[13], Yousef et al. (2006)[14], Liang (2010)[15], Sayarpour (2008)[16], Sayarpour et al. (2009)[21] and Weber[1].

The reservoir is $2048 \times 2048 \times 60 \text{ ft}^3$ and consists of 5 vertical injection wells and 4 vertical production wells. The permeability is constant throughout the reservoir, except for two high permeability streak between injector 1 and producer 1, and between injector 3 and producer 4. The porosity is 18% and constant. The rock compressibility is $3 \times 10^{-6} \text{ psi}^{-1}$, the water compressibility is $1 \times 10^{-6} \text{ psi}^{-1}$ while the oil compressibility is generated by Eclipse using different PVT parameters. The wells produced at a constant bottom hole pressure of 250 psia. Two different initial reservoir pressure are tested: $P_{ri} = 1450 \text{ psia}$ and $P_{ri} = 250 \text{ psia} = \text{producing well pressure}$.

The original injection rates for the injectors used in the Eclipse simulation are shown in figure 3.2.

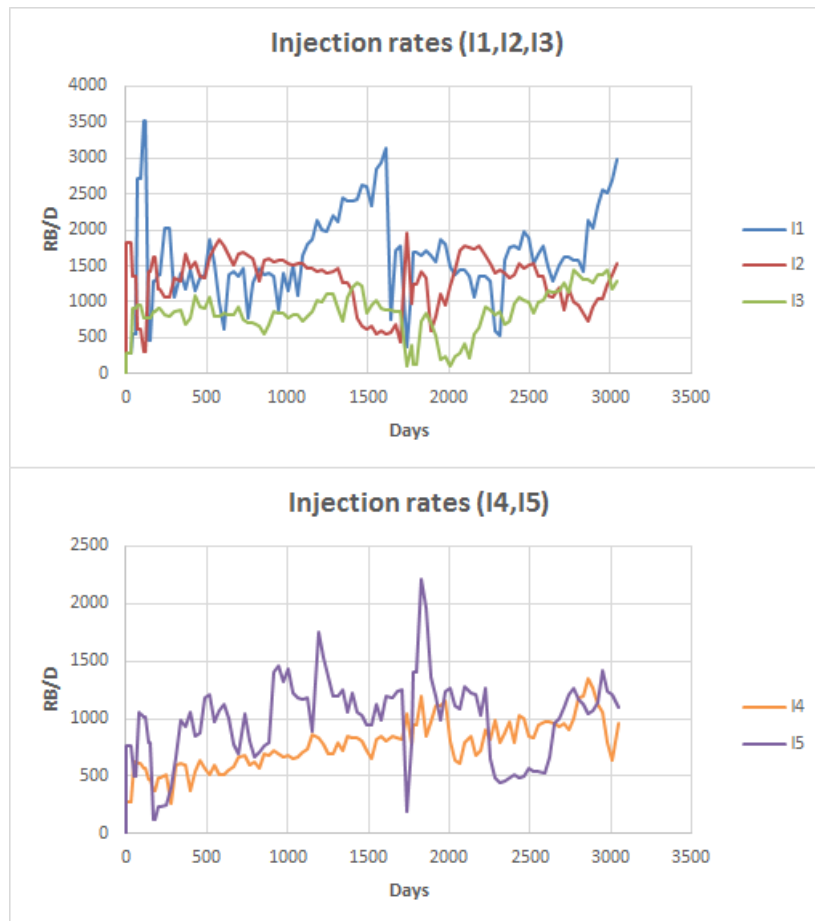


Figure 3.2: Injection rates

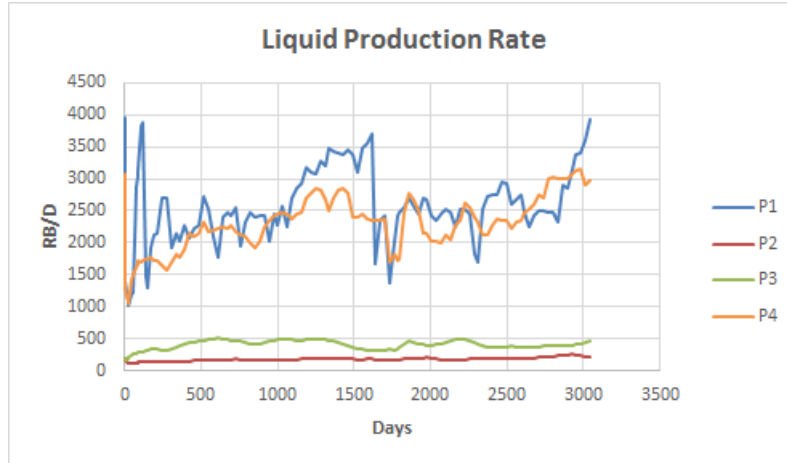


Figure 3.3: Production rates estimated by an Eclipse simulation with $P_{ri} = 1450$ psia. These rates will be used as real observed data.

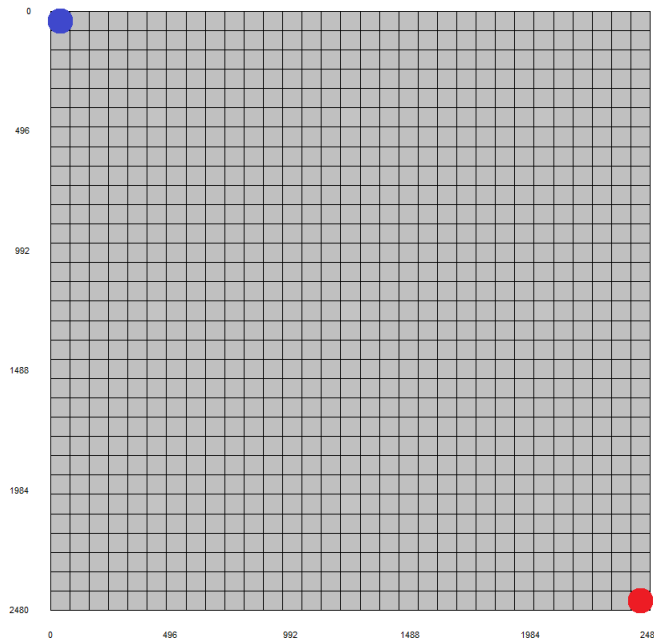


Figure 3.4: Reservoir with 1 injection well and 1 production well.

3.2 Second Reservoir, 1 Injector and 1 Producer

The second scenario is similar to a reservoir model used in Sayarpour et al. (2011)[22]. The horizontal and vertical permeability are 40 md and 4 md, respectively. The rest of the reservoir characteristics are the same as the reservoir model described in section 3.1. The production well is producing at a constant bottom hole pressure of 250 psia. The injection rate used in the Eclipse simulation is depicted in figure 3.5 on the following page, which resulted in the production rates shown in figure 3.6.

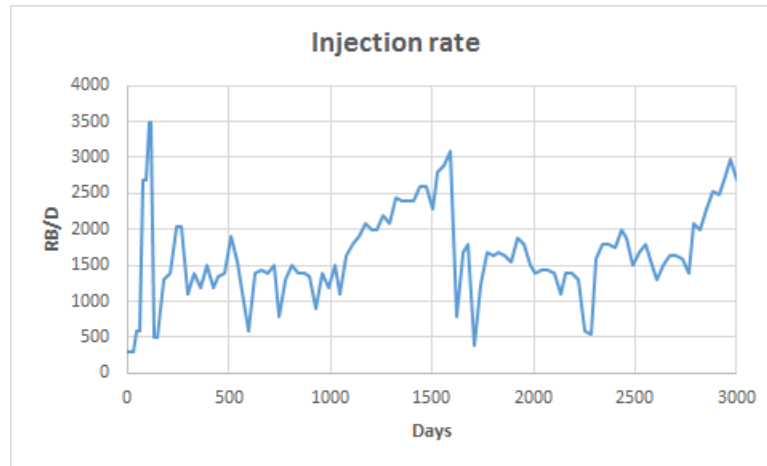


Figure 3.5: Injection rate.

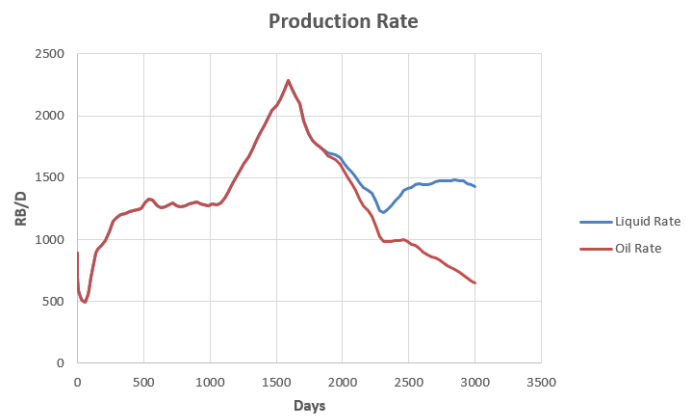


Figure 3.6: Production rate estimated by an Eclipse simulation with $P_{ri} = 1450$ psia.

Capacitance Resistive Modeling

4.1 Introduction

Albertoni and Lake 2003[13] introduced a method for evaluating interwell connectivities, to describe the flow pattern in a reservoir based on only injection and production rates. Several other studies including Yousef et al. 2006[14], Liang 2010[15] and Sayarpour 2008[16] have further extended and improved this method, which today is known as capacitance resistive modelling (CRM). CRMs are fast analytical simulators with low computational costs. It can be useful on a day-to-day basis, by being able to quickly update the optimal injection control when new production data becomes available.

The high-cost nature of petroleum exploration and development combined with a low oil price, suggests that developing further understanding and insights in the use of reduced order models such as the CRM, may provide value creation and cost reduction for certain reservoir conditions and decision aspects.

In this chapter we present the different capacitance resistive models including the solution approaches required for generating production forecasts. Following this, we present two approaches based on oil fractional flow methods which are used to forecast oil and water production.

Much of the information and figures in this chapter was first presented in my specialization project[12].

4.2 Background

At the start of the production life of a reservoir, oil is produced mainly due to primary recovery mechanisms, meaning the reservoirs own natural drive, such as fluid and rock expansion, solution gas drive, gravity drainage and aquifer influx[23]. As time goes on, the reservoir pressure, and subsequently the production decreases. In order to maintain production, the field must be

produced using secondary recovery mechanisms, which in this project is through water injection (also called waterflooding). The main purpose of waterflooding is to increase the oil production by filling the void left as the oil is produced, and thereby maintaining the reservoir pressure. By controlling the injection wells' locations and injection rates, waterflooding may be used as a tool to control the flow pattern and subsequently the oil production.

Waterflooding has been a common secondary recovery mechanism in petroleum industry since the the 1930s[23] and several techniques exists for estimating waterflooding performance, including analytical, empirical and numerical methods.

Among the better known are Buckley and Leverett (1942)[24], who introduced the frontal advance equation to estimate linear displacement of water displacing oil and the Dykstra Parsons method[25] which calculates piston displacement of fluid in non-communicating layers.

However, the underlying assumptions in these models, such as diffuse flow and no capillary dispersion at the front (Buckley-Leverett) and piston displacement and constant pressure drop for all layers (Dykstra-Parsons), makes it difficult to account for the heterogeneity often experienced in real reservoirs.

According to the Petroleum Engineering Handbook (2006)[23], the most important aspect when evaluating waterflooding is to understand the reservoir rocks. Therefore, being able to describe the reservoir flow during a waterflood by using only a few parameters is the motivation behind the development of the CRMs.

Sayarpour et al. (2009)[21] argues that there is a lack of usability for more complex numerical simulator that may be filled by the CRM. The main advantage for the CRM compared to higher order simulators (i.e. Eclipse) is that the need for input data and detailed reservoir description is significantly reduced. This reduces both the time required to gather information and the computational requirements.

Work on the CRM started at the University of Texas. The vast majority of research done on this subject is connected to the university either through master and PhD students, or professors.

Albertini and Lake (2003)[13] proposed a technique that used multivariate linear regression combined with a diffusivity filter to estimate liquid production by looking at the reservoirs heterogeneity such as faults and discontinuities that provide permeability changes or barriers. They quantified weighing coefficients that signified the flow pattern of injected water. The results were compared with a higher order reservoir simulator and indicated a good match with permeability trends and barriers. Yousef et al. (2006)[14] improved the model by introducing a time constant to signify response delay instead of the diffusivity filter.

Gentil (2005)[26] provided more insight to the underlying assumptions and the physical

meaning of the parameters in the model described in Albertoni and Lake (2003)[13]. He also extended the basic CRM by introducing a fractional flow method. This method, noted as Gentil's fractional flow model, was used by Liang (2010)[15] for prediction of oil production.

Sayarpour (2008)[16] introduced analytical solutions for the governing mass balance equation, which was adjusted to three different control volumes, tank based, producer based and injector-producer pair based. He showed that the CRM could be used to optimize oil production by changing the injection rates.

Weber (2009)[1] modified Sayarpour model and demonstrated how the CRM can be used to optimize well placement in order to increase recovery.

4.3 Tank Based Representation

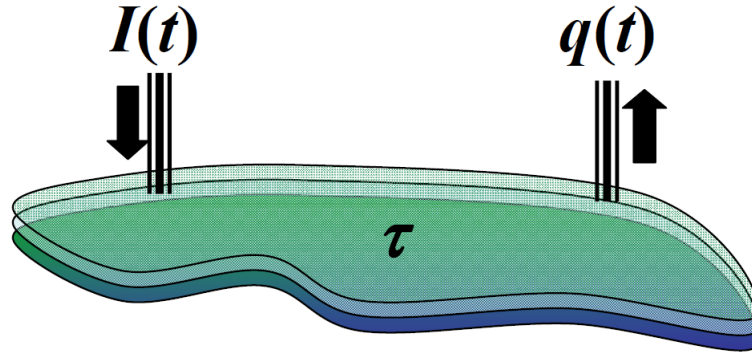


Figure 4.1: Schematic representation of the CRMT. This image was originally used in Weber (2009)[1].

The CRMT uses the entire reservoir as the control volume. Instead of calculating parameters for each well, a pseudo-producing well is used to represent all producing wells at the field, and a pseudo-injector for all the injecting wells (equation 4.1). The control volume is illustrated in figure 4.1.

$$q_{Field} = \sum_{j=1}^{N_{prod}} q_j \quad I_{Field} = \sum_{i=1}^{N_{inj}} I_i \quad (4.1)$$

For a field consisting of one producer and one injector, the governing mass balance differential equation at reservoir conditions is[16]:

$$c_t V_P \frac{d\bar{p}}{dt} = I(t) - q(t) \quad (4.2)$$

where c_t is the total compressibility, V_p is the pore volume, \bar{p} is the average reservoir pressure, I is the injection rate and q is the liquid production rate. It is worth noting that this equation is based on multiple assumptions including constant temperature, immiscible phases, negligible capillary pressure and constant productivity index[16]. The production rate is defined as[16]:

$$q(t) = J(\bar{p} - p_{wf}) \quad (4.3)$$

where J is the productivity index and p_{wf} is the bottom hole flowing pressure. By combining equation 4.2 on the previous page and 4.3 (as done by Yousef et al.(2006)[14]), and assuming a constant productivity index, we derive equation 4.4.

$$\frac{dq(t)}{dt} + \frac{1}{\tau}q(t) = \frac{1}{\tau}I(t) - J\frac{dp_{wf}}{dt} \quad (4.4)$$

where the time constant τ is defined as:

$$\tau = \frac{c_t V_p}{J} \quad (4.5)$$

The time constant τ , is one of the fundamental parameters in the CRM. τ has units of time and is assumed to be constant during production.

Sayarpour[16] showed the discrete analytical one time interval solution of equation 4.4 from time t_{k-1} to t_k to be:

$$q(t_k) = q(t_{k-1})e^{-\frac{\Delta t_k}{\tau}} + (1 - e^{-\frac{\Delta t_k}{\tau}})(I_k - J\tau \frac{\Delta P_{wf,k}}{\Delta t_k}) \quad (4.6)$$

where the injection rate I_k is kept constant between t_{k-1} and t_k .

In order to use equation 4.6 in a field application with more than one production well, the CRMT must assume constant bottom hole pressure at the pseudo-producer, and the equation is modified to:

$$q_F(t_k) = q_F(t_{k-1})e^{-\frac{\Delta t_k}{\tau_F}} + (1 - e^{-\frac{\Delta t_k}{\tau_F}})I_k f_F \quad (4.7)$$

where f_F is the field connectivity, one of the parameters solved for by the CRM. The physical interpretation of the connectivity is the fraction of the injected water that reaches a producer. This implies a $f_F < 1$ means some of the injected water will leave the control volume and not be produced. The value of f_F is limited to be between 0 and 1.

τ has a lower boundary of zero. This means for a large tau, then $q_F(t_k) = q_F(t_{k-1})$, while when $\tau \rightarrow 0$, all the available oil will be immediately produced.

Solving the CRMT

The CRMT needs to be solved for three unknowns; τ , f and $q(t = 0)$. This is done by minimizing the error given in equation 4.8.

$$error = \sum_{k=1}^{N_t} (q_{obs,k} - q_{est,k})^2 \quad (4.8)$$

where $q_{obs,k}$ is the total observed liquid rate at time step k and $q_{est,k}$ is the total liquid rate at time step k as estimated by the CRMT. In this project, Excels Generalized Reduced Gradient (GRG) Nonlinear solver, was used to minimize the sum of squared differences in equation 4.8 on the preceding page.

Before solving the CRMT, initial values of τ , f and $q(t = 0)$ must be provided. The simplicity of the CRMT, by only needing to solve for three parameters, yields good results by simply choosing the initial values $\tau = 1$, $f = 1$ and $q(t = 0) = 0$.

Consider the reservoir model in figure 3.1 on page 9 with $P_{ri} = 1450$ psia and the resulting liquid production in figure 3.3. Figure 4.2 shows the total liquid production as estimated by the CRMT when minimizing the error in equation 4.8 using the nonlinear GRG solver in Excel.

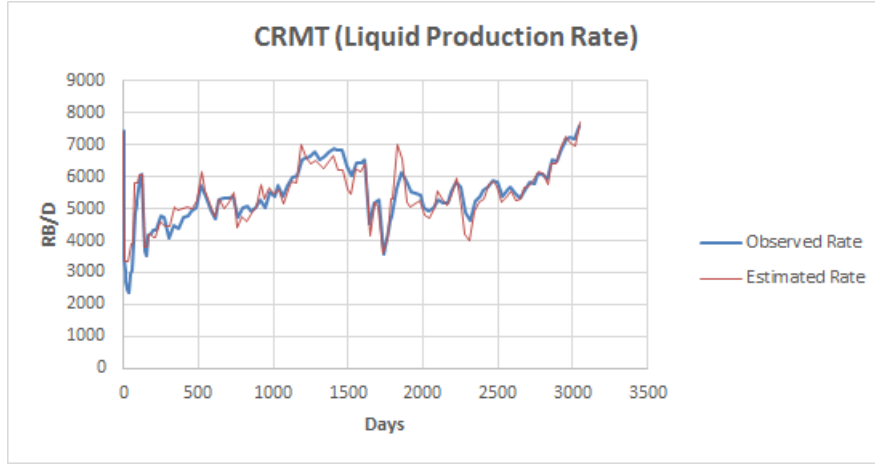


Figure 4.2: Estimated rates by the CRMT and observed rates. ($R^2 = 0.8697$).

4.4 Producer Based Representation

For the CRM with producer based representation, the control volume is specified around a given producer (figure 4.3 on the next page).

By using the control volume around a producer j , Liang (2010)[15] modified the governing differential equation (equation 4.4 on the facing page) to:

$$\frac{dq_j(t)}{dt} + \frac{1}{\tau_j}q_j(t) = \frac{1}{\tau_j} \sum_{i=1}^{N_{inj}} f_{ij}I_i(t) - J_j \frac{dp_{wf,j}}{dt} \quad (4.9)$$

where q_j is the production rate for producer j , τ_j is the time constant for the control volume around producer j , f_{ij} (equation 4.10) is the connectivity associated with each producer-injector pairs and assumed constant.

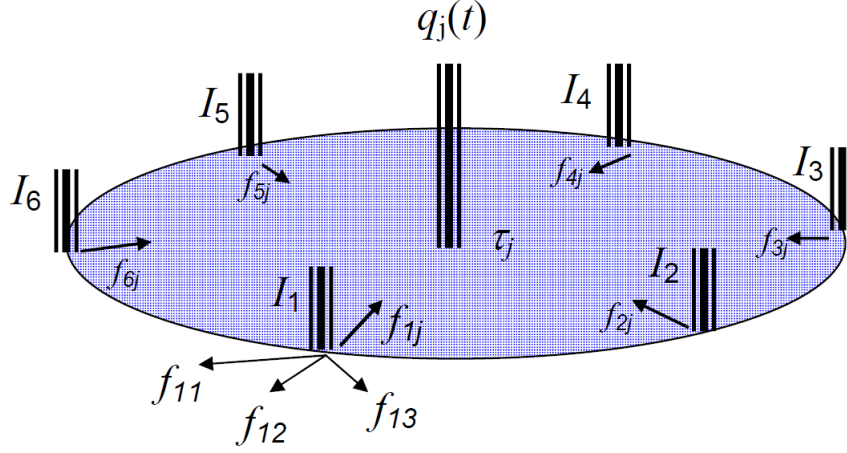


Figure 4.3: Schematic representation of the control volume around a producer, as used by the CRMP. This image was originally used in Weber (2009)[1].

$$f_{ij} = \frac{q_{ij}(t)}{I_i(t)} \quad (4.10)$$

I_i is injection rate from injector i , J_j is the productivity index for producer j and $p_{wf,j}$ is the bottom hole pressure at producer j . As in the CRMT, the sum of connectivities from an injector i , is limited by an upper bound of 1 and lower bound of 0.

The analytical solution of equation 4.9 on the previous page, as presented in Sayarpour (2008)[16] and Weber (2009)[1], is:

$$q_j(t_k) = q_j(t_{k-1})e^{-\frac{\Delta t_k}{\tau_j}} + (1 - e^{-\frac{\Delta t_k}{\tau_j}}) \left(\sum_{i=1}^{N_{inj}} (f_{ij} I_{i,k}) - J_j \tau_j \frac{\Delta p_{wf,j,k}}{\Delta t_k} \right) \quad (4.11)$$

This is the one-time interval solution for a step change in injection rate, linear change in bottom hole pressure and assuming a constant productivity index J .

If the bottom hole pressure is constant, which is the case for the examples used in this project, the equation can be simplified to equation 4.12. For cases where the bottom hole pressure is unknown, the ΔP_{wf} is also assumed to be zero.

$$q_j(t_k) = q_j(t_{k-1})e^{-\frac{\Delta t_k}{\tau_j}} + (1 - e^{-\frac{\Delta t_k}{\tau_j}}) \sum_{i=1}^{N_{inj}} f_{ij} I_{i,k} \quad (4.12)$$

Solving the CRMP

The CRMP calculates a rate value for each producer at each time step t_k . This means that we now minimize the equation 4.13 on the next page, by solving for τ_j , f_{ij} and $q_j(t = 0)$.

$$error = \sum_{k=1}^{N_t} \sum_{j=1}^{N_{prod}} (q_{obs,j,k} - q_{est,j,k})^2 \quad (4.13)$$

The initial values of τ_j before solving, should equal τ_{CRMT} , meaning the CRMT must be used first. Since the connectivities f_{ij} for a homogeneous reservoir reflect the interwell distance and $\sum_{j=1}^{N_{prod}} f_{ij} \leq 1$, the initial f_{ij} suggested by Sayarpour[16], and used in this project are:

$$f_{ij} = \frac{1/d_{ij}}{\sum_{j=1}^{N_{prod}} 1/d_{ij}} \quad (4.14)$$

where d_{ij} is the distance between injector i and producer j .

The initial values before solving for production rate at time=0, are set as:

$$q_j(t = 0) = q_{obs,j,k=1} \quad (4.15)$$

where $q_{obs,j,k=1}$ is the observed rate at producer j at the first time step.

Solving the CRMP (eq. 4.13) using the liquid production rates in 3.3 on page 11 results in the estimated total production rate in figure 4.4. The resulting flow pattern, governed by the connectivities, can be seen in figure 4.5 on the following page. Considering the permeability streaks in the reservoir model, the flow pattern is reasonable.

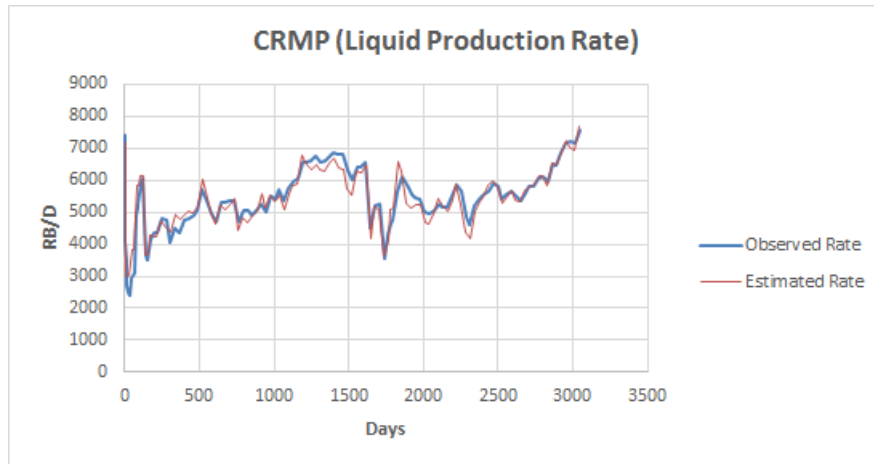


Figure 4.4: Estimated rates by the CRMP and observed rates. ($R^2 = 0.9184$).

4.5 Injector-Producer Pair Based Representation

For the CRMIP, the reservoir is approached by considering control volumes between each injector-producer well pair, as illustrated in figure 4.6 on the next page. In other words, for each combination of a producer j and injector i , a different control volume is used.

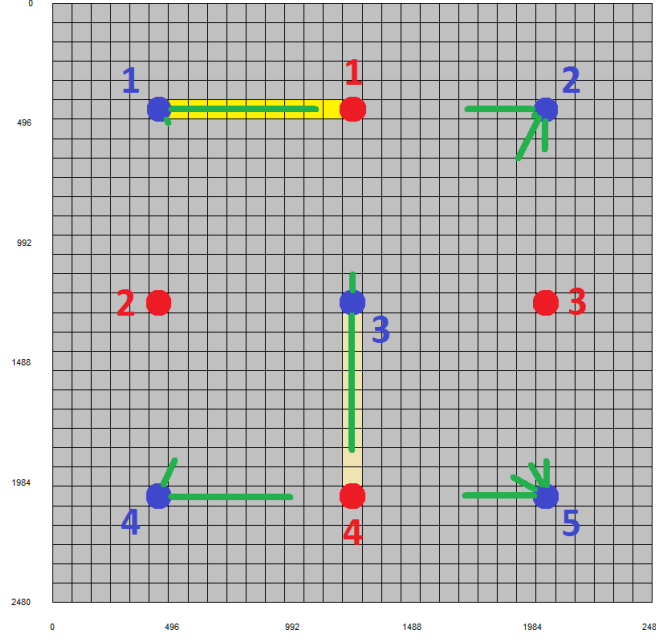


Figure 4.5: Approximate connectivities by the CRMP

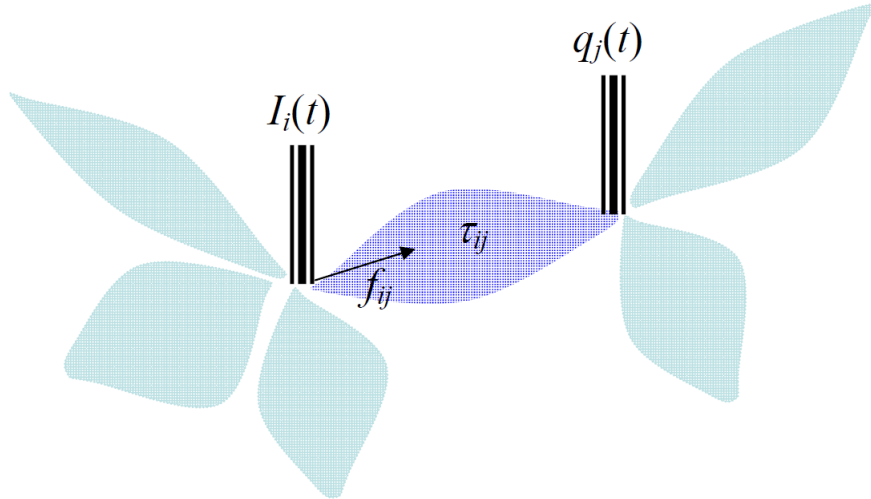


Figure 4.6: Schematic representation of the control volume between an injector-producer pair, as used by CRMIP. This image was originally used in Weber (2009)[1].

The governing differential equation, as presented in Yousef et al. (2006), is then:

$$\frac{dq_{ij}(t)}{dt} + \frac{1}{\tau_{ij}}q_{ij}(t) = \frac{1}{\tau_{ij}}f_{ij}i_i(t) - J_{ij}\frac{dp_{wf,j}}{dt} \quad (4.16)$$

The main difference from CRMP is that there is now a time constant τ for each injector-producer pair.

Sayarpour[16] and Weber[1] showed that the discrete analytical solution of the CRMIP id:

$$q_{ij}(t_k) = q_{ij}(t_{k-1})e^{-\frac{\Delta t_k}{\tau_{ij}}} + (1 - e^{-\frac{\Delta t_k}{\tau_{ij}}})(f_{ij}I_{i,k} - J_{ij}\tau_{ij}\frac{\Delta p_{wf,j,k}}{\Delta t_k}) \quad (4.17)$$

When assuming constant bottom hole pressure, the equation simplifies to:

$$q_{ij}(t_k) = q_{ij}(t_{k-1})e^{-\frac{\Delta t_k}{\tau_{ij}}} + (1 - e^{-\frac{\Delta t_k}{\tau_{ij}}})f_{ij}I_{i,k} \quad (4.18)$$

Solving the CRMIP

The CRMIP estimates a rate value for each injector-producer pair at time step t_k , by solving for τ_{ij} , f_{ij} and $q_{ij}(t = 0)$. Since there is only one observed rate for each producer, the CRMIP error is given by equation 4.19.

$$error = \sum_{k=1}^{N_t} \sum_{j=1}^{N_{prod}} (q_{obs,j,k} - q_{est,j,k})^2 \quad (4.19)$$

where

$$q_{est,j,k} = \sum_{i=1}^{N_{inj}} q_{ij,k} \quad (4.20)$$

where $q_{ij,k}$ are the estimated rates for producer j at time step t_k from injector i .

The initial connectivities set before solving are the same as the initial values for CRMP (see equation 4.14).

The initial production rate values used before solving are:

$$q_{ij}(t = 0) = \frac{q_{obs,j,k=1}}{N_{inj}} \quad (4.21)$$

where N_{inj} are the total number of injectors.

Using the CRMT with the liquid production rates in figure 3.3 results in the total production rate in figure 4.7 on the next page. This is a significantly better match than both for the CRMP and the CRMT.

The individual well rates using the CRMIP, can be seen in figure 4.8.

4.6 Oil Fractional Flow Models

Since the CRM methods only estimate the total liquid production, it is necessary to implement a method for estimating the oil rate.

Throughout the production time of a field, the oil fraction of the rate will vary. During primary recovery the field produces by natural reservoir drive, such as natural gas/water-drive and gravity drainage, and the oil fraction tends to be high. As secondary recovery takes effect, such as waterflooding, the oil fraction starts to decrease after water breakthrough. In tertiary

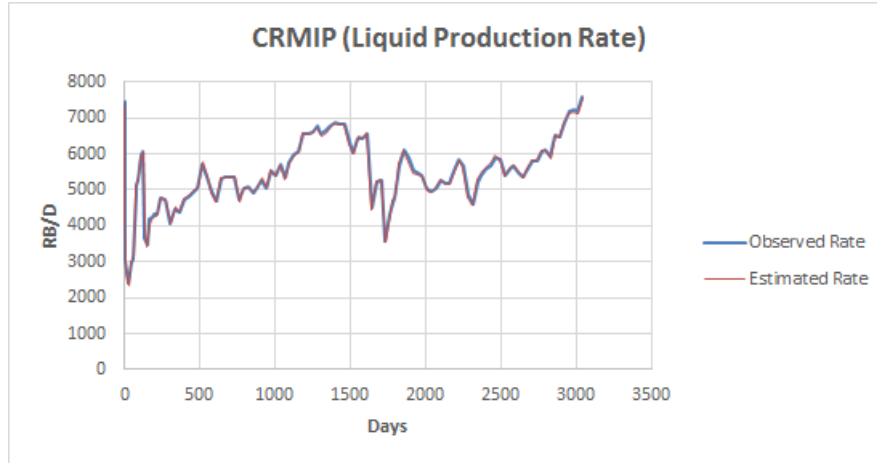


Figure 4.7: Estimated rates by the CRMIP and observed rates. ($R^2 = 0.9975$).

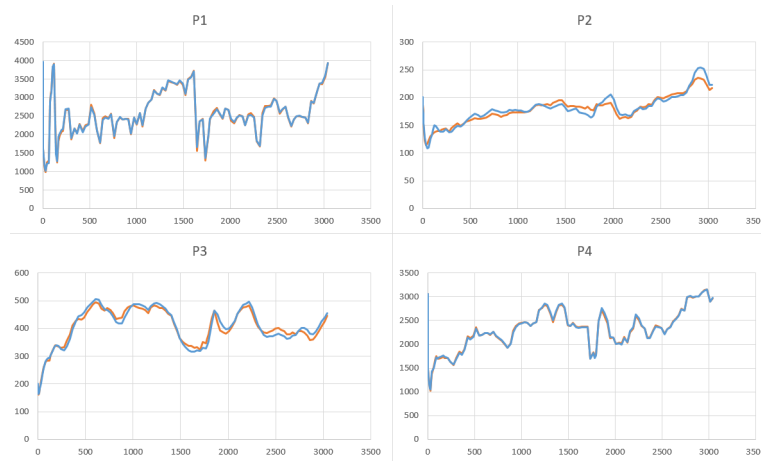


Figure 4.8: Liquid production rates for each individual production well. Orange rates are the estimated rates, while the blue lines are the observed (Eclipse simulated) rates.

recovery, such as CO_2 injection, the oil fraction will increase before it eventually decreases to zero. If no tertiary recovery methods are used, the oil fraction will continue to decline to zero.

As mentioned in section 4.2, several different models have been developed to estimate the fractional flow. Oil fractional flow models can be combined with the CRMs by using the estimated total liquid production rate by the CRMs as input to estimate the oil rate. In this project rapport two models will be presented: the Buckley-Leverett based fractional flow model (BLBFFM) and Gentil's fractional flow model.

4.6.1 Gentil's Fractional Flow Model

Gentil's fractional flow model is an empirical fractional flow model developed by Gentil[26], and first used by Liang[15] to predict oil production. The oil fraction of the production rate is defined as:

$$f_o = \frac{1}{1 + \alpha W^\beta} \quad (4.22)$$

where W is the cumulative water injected that reaches a producer, while α and β are fitting parameters of the model. Combining the fractional flow model with a liquid production rate $q_j(t)$, results in equation 4.23,

$$q_{o,j}(t) = \frac{q_j(t)}{1 + \alpha_j W_j^{\beta_j}} \quad (4.23)$$

where $q_{o,j}(t)$ is the oil rate and $q_j(t)$ is the liquid production rate for producer j at time t . W_j is the cumulative water injected at time t that reaches producer j .

Solving Gentil's Fractional Flow Model

For Gentil's fractional flow, we use the same procedure for solving as for CRMT (section 4.3), CRMP (section 4.4) and CRMIP (section 4.5), which is to minimize:

$$error = \sum_{k=1}^{N_t} (q_{obs} - q_{est})^2 \quad (4.24)$$

Since this model processes output signals (production data), there can be multiple solution for a system. In order to avoid this, it is important to set good initial model parameter values.

Sayarpour[16] estimated the initial values by first writing equation 4.23 on a logarithmic form:

$$\log(WOR) = \log\left(\frac{q}{q_o} - 1\right) = \log(\alpha) + \beta * \log(W) \quad (4.25)$$

which, when plotted in a log-log plot, can be used to estimate α and β .

However, a plot of $\log(WOR)$ versus $\log(W)$ is not always linear due to primary recovery effects (e.g. figure 4.9 on the next page). To avoid these effects, the late linear behavior is used to estimate the initial values. As seen in figure 4.9 on the following page, when simulating a reservoir, the late linear log-log behavior may not in reality be linear but the plot is helpful in determining (estimating) the model parameters.

The parameters must be constrained by $\alpha > 0$ and $\beta \geq 0$.

Using the CRMP combined with Gentil's fractional flow for the first reservoir (figure 3.1) with $P_{ri} = 1450$ psia, resulted in the rates seen in figure 4.10 and 4.12.

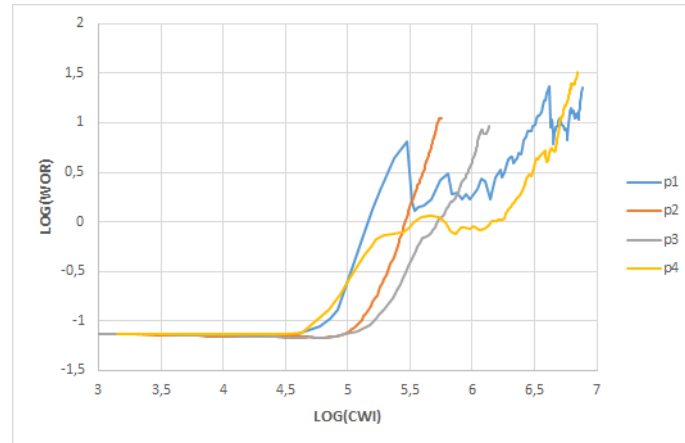


Figure 4.9: Logarithmic plot water oil ratio and cumulative water injection for the reservoir in figure 3.1 on page 9.

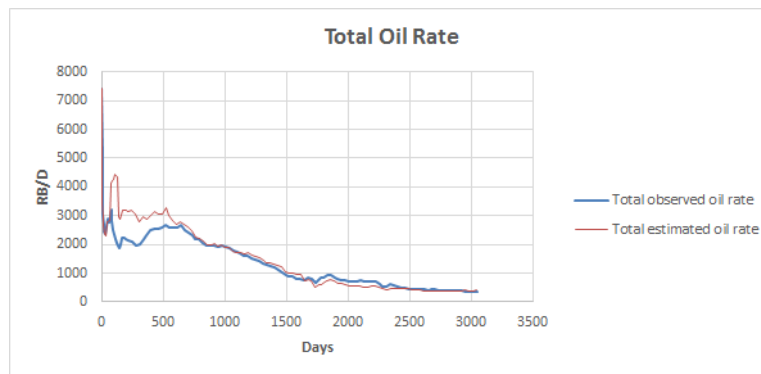


Figure 4.10: Estimated and observed oil rate using the CRMP. ($R^2 = 0.8911$)

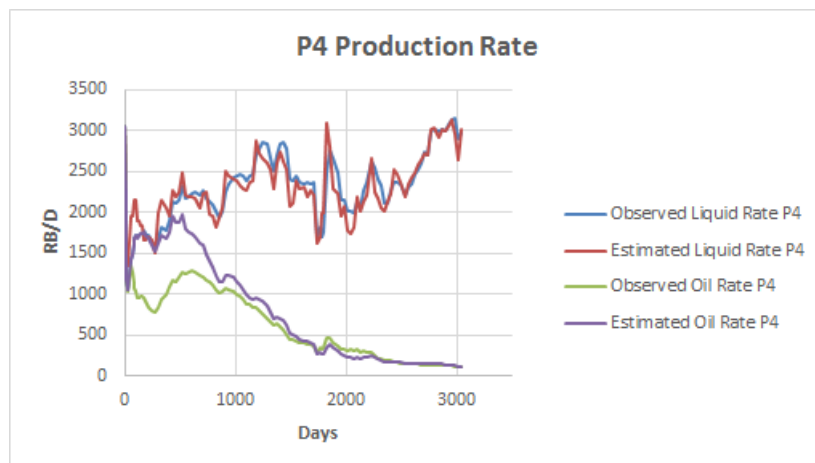


Figure 4.11: Estimated and observed values for both liquid and oil production rate in producer 4 for the reservoir in figure 3.1 on page 9.

4.6.2 Buckley Leverett Based Fractional Flow Model

This model is based on a fractional flow model (equation 4.26 on the next page) defined in Leverett (1941)[27].

$$f_w = \frac{1 + \frac{kk_{ro}A}{q\mu_o} \left(\frac{\delta p_c}{\delta x} - \Delta \rho g \sin \alpha \right)}{1 + \frac{k_{ro}\mu_w}{\mu_o k_{rw}}} \quad (4.26)$$

Sayarpour[16] showed that by assuming negligible capillary pressure and gravity effects, and using a power-law model for relative permeability curves, the oil fractional flow could be written as:

$$f_o(S) = 1 - \left(1 + \frac{(1-S)^m}{MS^n} \right)^{-1} \quad (4.27)$$

where M is the endpoint mobility ratio, m and n are relative permeability curve exponents and S is the normalized water saturation defined as:

$$S(t) = \frac{S_w(t) - S_{wir}}{1 - S_{or} - S_{wir}} \quad (4.28)$$

where S_{wir} is the irreducible water saturation, S_{or} is the residual oil saturation and $S_w(t)$ is the water saturation at time t defined as:

$$S_w(t_k) = S_w(t_{k-1}) + \frac{I^*(t_k) - q_w(t_{k-1})}{V_P} \Delta t_k \quad (4.29)$$

where $I^*(t_k)$ is the effective water injection rate, which equals $f \cdot I(t_k)$ when using the CRMT. The V_P is the pore volume, while $q_w(t_{k-1})$ is the water production rate at time t_{k-1} . Using q_w at the previous time step, enables the model to be used for prediction.

When using BLBFFM, we first estimate equation 4.29. The resulting $S_w(t_k)$ is inserted in equation 4.28. The $S(t)$ from equation 4.28 is used to calculate f_o , which when multiplied with the estimated liquid rate from the CRM, produces q_o .

Solving Buckley Leverett Based Fractional Flow Model

The BLBFFM has seven unknowns; M , m , n , S_{wir} , S_{or} , $S_w(0)$ and V_P . These parameters are estimated by minimizing the error (equation 4.24) between observed and estimated oil rate.

There are however, some constraints in the models parameter. All parameters must be greater than zero, $M, S_{wir}, S_{or}, S_w(0) \leq 1$ and $S_w(0) \geq S_{wr}$.

Figure 4.12 shows the production rate for the reservoir described in section 3.2 with $P_{ri} = 1450$ psia. In this case, the porosity was determined to be around 18%, the irreducible water saturation to be 34% and the residual oil saturation to be 18%, similar to the results in Sayarpour et al. (2011)[22].

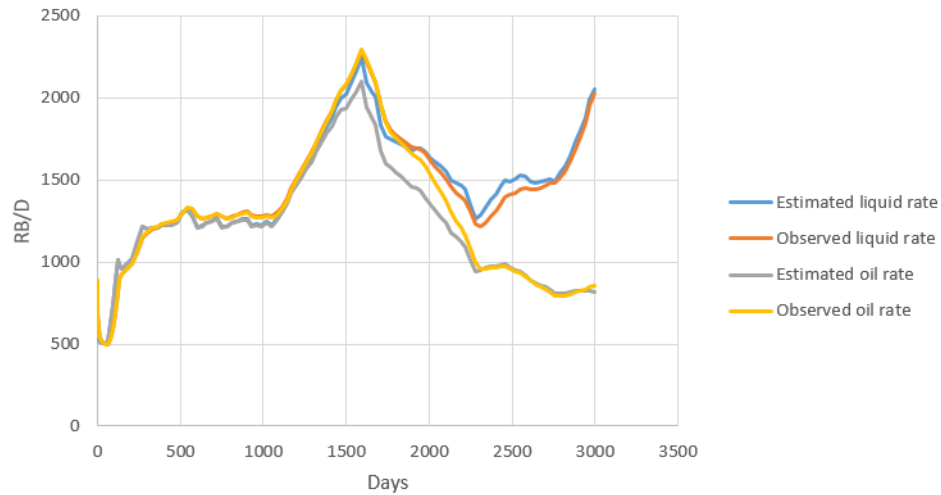


Figure 4.12: Estimated and observed liquid and oil production rates for the second reservoir (figure 3.4 on page 11).

4.7 Conclusion

The CRMs provides good estimate of actual production after primary recovery. The CRMs are able to describe flow pattern with two different types of variables (τ and f) using only production data, resulting in significantly lower computational costs compared to higher order reservoir simulators such as Eclipse. The simplicity of the models makes them easy to implement in for example Excel and MATLAB.

The results produced in this chapter were verified by previous literature work on the CRM.

4.8 CRM and Oil Fraction Formulas

Liquid rate:	
CRMT	$q(t_k) = q(t_{k-1})e^{-\frac{\Delta t_k}{\tau}} + (1 - e^{-\frac{\Delta t_k}{\tau}})(I_k f_F - J\tau \frac{\Delta P_{wf,k}}{\Delta t_k})$
CRMP	$q_j(t_k) = q_j(t_{k-1})e^{-\frac{\Delta t_k}{\tau_j}} + (1 - e^{-\frac{\Delta t_k}{\tau_j}})(\sum_{i=1}^{N_{inj}} (f_{ij} I_{i,k}) - J_j \tau_j \frac{\Delta p_{wf,j,k}}{\Delta t_k})$
CRMIP	$q_{ij}(t_k) = q_{ij}(t_{k-1})e^{-\frac{\Delta t_k}{\tau_{ij}}} + (1 - e^{-\frac{\Delta t_k}{\tau_{ij}}})(f_{ij} I_{i,k} - J_{ij} \tau_{ij} \frac{\Delta p_{wf,j,k}}{\Delta t_k})$
Oil fraction:	
Gentil	$f_{o,j} = \frac{1}{1 + \alpha_j W_j^{\beta_j}}$
BLBFFM	$S_w(t_k) = S_w(t_{k-1}) + \frac{I^*(t_k) - q_w(t_{k-1})}{V_P} \Delta t_k$ $S(t) = \frac{S_w(t) - S_{wir}}{1 - S_{or} - S_{wir}}$ $f_o(S) = 1 - \left(1 + \frac{(1-S)^m}{MS^n}\right)^{-1}$

Kalman Filter

5.1 Introduction Kalman Filter

This chapter describes three different Kalman filters with particular emphasis on the ensemble Kalman filter. Section 5.2 provides a short historical background of Kalman filtering. Sections 5.3 and 5.4 shows the structure for the linear Kalman filter and the extended Kalman filter. In section 5.5, the ensemble Kalman filter is further investigated including a discussion of the historical background and the structure of the filter. A few examples using the CRMP, CRMIP and Gentil's fraction flow are presented, where the EnKF is used to history match the reservoir introduced in section 3.1.

5.2 Background Kalman Filter

In 1960 Rudolf Emil Kalman introduced a 'New Approach to Linear Filtering an Prediction Problems'[28], which contains what today is known as the Kalman filter (KF). This method is a filtering technique used for linear models. Kalman argued that his approach would sidestep limitations of other filtering techniques, such as the Wiener filter.

The Kalman filter turned out to be very powerful, but was originally only applicable for linear problems. Following Kalman's original work, other researcher have worked on extending the method to nonlinear problems[29].

The extended Kalman filter (EKF) addressed the non-linearity problem by linearizing about the current mean and covariance[30]. However, this approach does not work well for large scale or strongly nonlinear problems[31].

The Ensemble Kalman Filter was first introduced by Evensen[6] and resulted in better accuracy than the extended Kalman filter. Evensen showed that the EnKF is suited for handling nonlinear ocean model[6] [31].

5.3 Linear Kalman Filter

The linear Kalman filter is a recursive filter used to calculate model states of linear systems containing Gaussian measurement noise[29]. The structure presented in this section mainly follows Welch and Bishop[30] and Jensen 2007[31], which both provides relatively easy to read introductions to the Kalman filter.

For the linear Kalman filter, we first estimate the state using the linear stochastic difference equation:

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1} \quad (5.1)$$

where x_k is the state at time step k , x_{k-1} and u_{k-1} are the state and optional control input, respectively, at previous time step. A is a matrix that relates x_{k-1} to x_k , while B is a matrix the relates u_{k-1} to x_k . The measurement z is described by the following equation:

$$z_k = Hx_k + v_k \quad (5.2)$$

where the matrix H relates the state to the measurement. The terms w_k and v_k , are the process noise and the measurement noise, respectively. They are assumed to be independent of each other and follow a normal probability distribution:

$$w_k \sim N(0, Q) \quad v_k \sim N(0, R) \quad (5.3)$$

where Q and R are covariance matrices for the model noise and measurement noise, respectively.

As shown by Welch and Bishop[30], the a priori and a posteriori estimate error are defined as:

$$e_k^- = x_k - \hat{x}_k^-, \quad e_k = x_k - \hat{x}_k \quad (5.4)$$

where \hat{x}_k^- is the a priori state estimated at time step k given knowledge of the process at the prior to time step k , and \hat{x}_k is the a posteriori state estimated at time step k given knowledge of the measurement z_k . The a priori and a posteriori error covariance are respectively given as:

$$P_k^- = E(e_k^- e_k^{-T}), \quad P_k = E(e_k e_k^T) \quad (5.5)$$

The a posteriori state is then updated:

$$\hat{x}_k = \hat{x}_k^- + K(z_k - H\hat{x}_k^-) \quad (5.6)$$

The difference between z_k and $H\hat{x}_k^-$ indicates the error between the actual measurement z_k and the predicted measurement $H\hat{x}_k^-$. This means that if the predicted and actual measurement

are the same, then the updated a posteriori state will equal the predicted state \hat{x}_k^- . The Kalman gain is the matrix K_k that is used to minimize the a posteriori error covariance and is given as:

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1} \quad (5.7)$$

Where R is the measurement error covariance. As explained by Welch and Bishop[30], this means that when $R \rightarrow 0$ then $K_k \rightarrow H^{-1}$, and the Kalman gain would give more weight to the difference between z_k and $H\hat{x}_k^-$ in equation 5.6. If the a priori error covariance $P_k^- \rightarrow 0$, the Kalman gain trusts the predicted measurement $H\hat{x}_k^-$ more compared to the actual measurement z_k , meaning the $K_k \rightarrow 0$.

The linear Kalman filter algorithm will therefore first make a prediction of the a priori state based on the previous time step (equation 5.8), as well as the a priori error covariance matrix (equation 5.9).

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1} \quad (5.8)$$

$$P_k^- = AP_{k-1}A^T + Q \quad (5.9)$$

Then the Kalman gain is estimated (equation 5.7) and the state is updated (equation 5.6). The a posteriori covariance matrix is estimated using equation 5.10).

$$P_k = (I - K_k H) P_k^- \quad (5.10)$$

5.4 Extended Kalman Filter

The extended Kalman filter (EKF) is a modified version of the Kalman filter to include a non-linear relationship between the measurement and the state. The state is now represented by the nonlinear stochastic difference equation[30]:

$$x_k = f(x_{k-1}, u_{k-1}, w_{k-1}) \quad (5.11)$$

and the measurement is:

$$z_k = g(x_k, v_k) \quad (5.12)$$

where as before, x_k is the state at timestep k , u_k is the control function, while w_k and v_k are the process/model and measurement noise. Both f and h are nonlinear functions that describe the relationship between previous and current state, and current state and measurement,

respectively.

The extended Kalman filter follows a similar stepwise procedure as the linear Kalman filter[31]. The difference between the filters is that in the EKF, the model and measurement matrices are Jacobians that are linearized in each step, as shown in equation 5.13 and 5.14.

$$A_k = \frac{\delta f(x)}{\delta x} \bigg|_{x=x_k} \quad (5.13)$$

$$H_k = \frac{\delta g(x)}{\delta x} \bigg|_{x=x_k} \quad (5.14)$$

5.5 Ensemble Kalman Filter

The ensemble Kalman filter (EnKF) is a data assimilation technique first introduced by Geir Evensen in (1994)[6]. According to Aanonsen et al. (2009)[29], it was Lorentzen et al. (2001)[32] that first applied the EnKF in the petroleum industry. Lorentzen et al. (2003)[33] used the EnKF for both synthetic and real data, and argued that the EnKF had promising usability in the petroleum industry. Other studies have also shown the EnKF to be a promising method for history matching reservoir simulation models[34].

The motivation behind the development of the EnKF, is to modify the Kalman filter to solve large-scale nonlinear systems. Although the extended Kalman filter may be used for nonlinear problems, the error covariance can often be challenging to calculate. For a system with m model parameters, the error covariance matrix will contain m^2 unknowns, which in turn means that the updated matrix (equation 5.9 on the previous page) requires $2m$ model integrations[31]. In the EnKF, the computational requirements are reduced, by using model ensembles. The biggest advantage for the EnKF compared to linear KF is therefore its large scale usability for nonlinear systems. This has led to an increase of research using the EnKF for reservoir simulation purposes[34].

According to Pavel Sakov[35] there are no disadvantages in using the EnKF compared to both EK and EKF. For a linear system, the EnKF is converging towards the linear KF[31].

5.5.1 Method

The EnKF is a history matching filter based on model parameters, states and measurements. When used for reservoir simulation, the permeability and porosity in the reservoir grid blocks, can be model parameters. The state could be pressures and saturations in the same grid blocks, and the measurements and observations could be the rate values, WOR or BHP.

To avoid confusion, note that observation and measurement may be used to describe the same variables.

We define the model parameters as \mathbf{m} . The state \mathbf{x}_k is a function of the state at previous time step \mathbf{x}_{k-1} and the model parameters [29, 31]:

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{m}) + \mathbf{w}_k \quad (5.15)$$

where \mathbf{w}_k is the model noise at time step k . The model noise \mathbf{w}_k follows a normal distribution with zero mean and is generated with a predefined standard deviation SD_w . In the examples in this thesis, the model noise is presumed to be zero.

$$\mathbf{w}_k = \mathcal{N}(0, SD_w) \quad (5.16)$$

Using the current state \mathbf{x}_k and the model parameters \mathbf{m} , the measurement is:

$$\mathbf{z}_k = \mathbf{g}(\mathbf{x}_k, \mathbf{m}) + \mathbf{v}_k \quad (5.17)$$

where \mathbf{v}_k is Gaussian measurement noise generated with a predefined standard deviation SD_v .

$$\mathbf{v}_k = \mathcal{N}(0, SD_v) \quad (5.18)$$

In this thesis, the standard deviation SD_v was set at around 10% of the possible range. If the initial production rate was available, the individual wells SD_v was set at around 10% the individual initial rates.

The formulation in equations 5.15 and 5.17 requires that an initial state is given and an initial guess of \mathbf{m} denoted as \mathbf{m}_0 .

$$\mathbf{m} = \mathbf{m}_0 + \delta\mathbf{m} \quad (5.19)$$

where $\delta\mathbf{m}$ is the difference between the actual model parameters and the initial guess.

The first step in the EnKF algorithm is to set an initial state ensemble of N realizations, denoted as the updated step:

$$\mathbf{x}_0^u = [\mathbf{x}_0^{1u} \quad \mathbf{x}_0^{2u} \quad \dots \quad \mathbf{x}_0^{Nu}] \quad (5.20)$$

and make an initial guess for the model parameter ensemble:

$$\mathbf{m}_0^u = [\mathbf{m}_0^{1u} \quad \mathbf{m}_0^{2u} \quad \dots \quad \mathbf{m}_0^{Nu}] \quad (5.21)$$

The model noise \mathbf{w}_k is then calculated. The forecasted model parameters, states and mea-

measurements are inserted into a matrix \mathbf{Y} :

$$\mathbf{Y}_k^f = \begin{bmatrix} \mathbf{m}_k^f \\ \mathbf{x}_k^f \\ \mathbf{g}_k^f \end{bmatrix} = \begin{bmatrix} \mathbf{m}_{k-1}^{1u} & \mathbf{m}_{k-1}^{2u} & \dots & \mathbf{m}_{k-1}^{Nu} \\ \mathbf{f}_k(\mathbf{x}_{k-1}^{1u}, \mathbf{m}_{k-1}^{1u}) + \mathbf{w}_k^1 & \mathbf{f}_k(\mathbf{x}_{k-1}^{2u}, \mathbf{m}_{k-1}^{2u}) + \mathbf{w}_k^2 & \dots & \mathbf{f}_k(\mathbf{x}_{k-1}^{Nu}, \mathbf{m}_{k-1}^{Nu}) + \mathbf{w}_k^N \\ \mathbf{g}_k(\mathbf{x}_k^{1f}, \mathbf{m}_{k-1}^{1u}) & \mathbf{g}_k(\mathbf{x}_k^{2f}, \mathbf{m}_{k-1}^{2u}) & \dots & \mathbf{g}_k(\mathbf{x}_k^{Nf}, \mathbf{m}_{k-1}^{Nu}) \end{bmatrix} \quad (5.22)$$

The error covariance matrix for the forecasted step \mathbf{P}_k^f is calculated:

$$\mathbf{P}_k^f = \frac{1}{N-1} [\mathbf{Y}_k^f - E(\mathbf{Y}_k^f)][\mathbf{Y}_k^f - E(\mathbf{Y}_k^f)]^T \quad (5.23)$$

where N is the number of the sets of model parameters, states and measurements used and $E(\mathbf{Y})$ is the expected values of each row in \mathbf{Y} . The observation data are calculated from the true observation \mathbf{d}_{obs} and measurement noise \mathbf{v}_k :

$$\mathbf{d}_k = [\mathbf{d}_{obs} + \mathbf{v}_k^1 \quad \mathbf{d}_{obs} + \mathbf{v}_k^2 \quad \dots \quad \mathbf{d}_{obs} + \mathbf{v}_k^N] \quad (5.24)$$

with the error covariance \mathbf{C}_{dk} for the observation being:

$$\mathbf{C}_{dk} = \frac{1}{N-1} \mathbf{v}_k \mathbf{v}_k^T \quad (5.25)$$

The Kalman gain \mathbf{K} is then computed:

$$\mathbf{K}_k = \mathbf{P}_k^f \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^f \mathbf{H}_k^T + \mathbf{C}_{dk})^{-1} \quad (5.26)$$

where the matrix \mathbf{H}_k is defined so that:

$$\mathbf{H}_k \mathbf{Y}_k^f = [\mathbf{g}_k^{1f} \quad \mathbf{g}_k^{2f} \quad \dots \quad \mathbf{g}_k^N] \quad (5.27)$$

The matrix \mathbf{Y}_k^u , containing the updated states, model parameters and measurements, is then updated:

$$\mathbf{Y}_k^u = \mathbf{Y}_k^f + \mathbf{K}_k (\mathbf{d}_k - \mathbf{H}_k \mathbf{Y}_k^f) \quad (5.28)$$

If the actual measurement \mathbf{d}_k is equal to the predicted measurement $\mathbf{H}_k \mathbf{Y}_k^f$ then the EnKF interprets the predicted (forecasted) \mathbf{Y}_k^f to be correct, and the updated \mathbf{Y}_k^u will therefore be equal to \mathbf{Y}_k^f . If there is a difference between the actual and predicted measurement, the Kalman gain \mathbf{K}_k is used to improve the update of \mathbf{Y} .

The resulting error covariance matrix becomes:

$$\mathbf{P}_k^u = \frac{1}{N-1} [\mathbf{Y}_k^u - E(\mathbf{Y}_k^u)][\mathbf{Y}_k^u - E(\mathbf{Y}_k^u)]^T \quad (5.29)$$

The model noise for the next step w_{k+1} is generated (equation 5.16 on page 33) and equations 5.22 to 5.29 are repeated for each time step with information.

The EnKF works in such a way that for every new observation, which include noise, the EnKF makes a new prediction of the actual parameters, states and measurements. This process is repeated for each time step with new observations.

For a clearer structure of the implementation, look at EnKF algorithm in section 5.9.

5.5.2 Limitations of the Ensemble Kalman Filter

A limitation with this technique, is due to the fact that the EnKF is based on using only covariances when updating[20]. The covariance matrix approximation means that it neglects the probability functions higher than second order moments[29]. This approximation means that the EnKF avoids closure problems[36], but it may be difficult to work with nonlinear systems consisting of non-Gaussian probability density functions. A Gaussian distribution cannot in some instances adequately represent the complexity of the behavior of reservoir characteristics. As a possible solution to handle non Gaussian distribution, Aanonsen et al. (2009)[29] presented two categories for approaches, the first being parametrization of the state variables. In this case, non Gaussian state variables are replaced by state variables that follow a normal distribution. Chen et al. (2009)[37] replaced non Gaussian water saturation with a saturation arrival time that followed a Gaussian distribution. The other is, according to Aanonsen, using iterative filters, such as presented in Reynolds et al. (2006)[38], Gu and Oliver (2007)[39] and Li and Reynolds (2007)[40]. Aanonsen et al. (2009)[29] mentions the particle filter as a possible solution to handling non Gaussian distribution, although it is mainly applicable for small scale models, due to the computer requirements.

Gu (2006)[41] suggested that the EnKF could potentially have problems when the changes in state variables are big.

5.6 Examples

In this section, a few examples is provided using the EnKF with the CRM. To simplify the examples, as well as the rest of this master thesis, only measurement noise v_k and not the model noise w_k will be included.

5.6.1 CRMP with Gentil's Fractional Flow

In this example, the EnKF is used to history match the last 2000 days of production of the reservoir described in section 3.1, both with regards to liquid rate parameters using the CRMP and oil rate parameters using Gentil's fractional flow model.

The example combines a CRMP implementation in Excel, and a EnKF implementation in MATLAB

In all, there are 32 different unknown model parameters. They consist of 20 connectivities (5 injectors, 4 producers), 4 values of τ (one for each producer), 4 α parameters and 4 β parameters. The liquid and oil rates for each production well are treated as measurements, 8 in total, meaning there are no state values.

Since the sum of connectivities from each injector add up to 1, and there are 4 producers, the initial guess for the connectivities was generated with a mean of 0.25 and a standard deviation of 5% of the possible range. All τ parameters were initially set to 1 with a standard deviation of 50.

The initial rate values in the EnKF implementation was set constant. The GRG nonlinear solver in Excel found an estimate of the initial rates, which were then used in the EnKF implementation as true values.

The reason for not using initial rate values as model parameters is due the way the CRMP function (equation 4.11 on page 18) is written. After the first updated time step, the initial rate value does not affect subsequent rate values, and the EnKF history matched initial rate, should not be trusted.

The initial β parameters were generated with a mean of 1. The VBA implementation was used in order to find appropriate standard deviation for the α parameters. This was due to the large range of possible α , as α could for example be in the order of 10^{-2} and 10^{-20} . Without more prior knowledge, the EnKF had difficulties in identifying these parameters.

The development of each of the model parameters are shown in figure 5.1, 5.2, 5.3 and 5.4.

The resulting production rates after the EnKF history matching, can be seen in figure 5.5 and 5.6.

To see how the EnKF guess evolves, all liquid and oil rates based on the different model parameter realizations are plotted in figure 5.7. To see the development for each 10th time step, see figures 8.8 on page 79 to 8.14 in the appendix. Figure 5.7 shows that after 10 time steps, 3 of the 4 wells matched well, with exceptions of P2, which also has the lowest production. After 20 time steps all producers matches well, and at subsequent time steps the EnKF becomes more certain of the model parameters.

At $N = 100$, the EnKF was able to consistently identify the flow pattern in the reservoir, described by the connectivities and the time constants. To help identify the oil rate parameters, N was increased to 1000. $N = 10000$ was finally used, as the oil parameters resulted in a slightly better match, while the liquid rate parameters resulted more or less in the same rate.

The top left of figure 5.7 shows that the production starts with a poor prior parameter ensem-

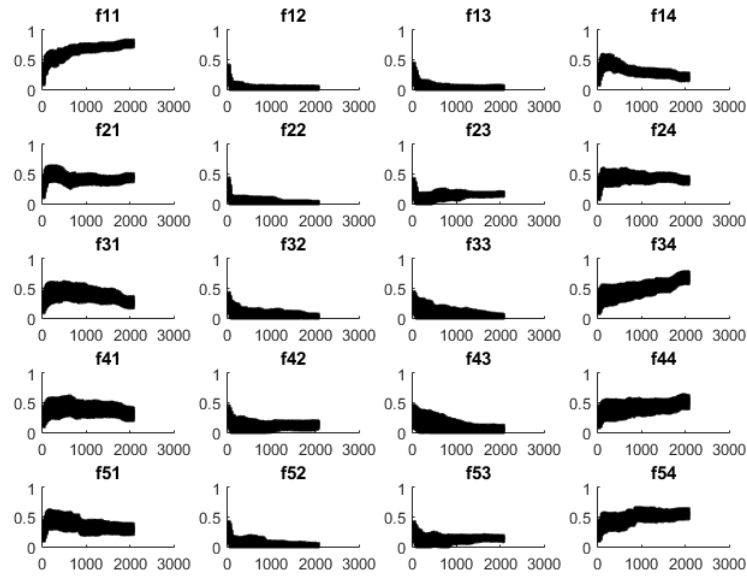


Figure 5.1: Connectivities f_{ij} plotted against time. $N = 10000$

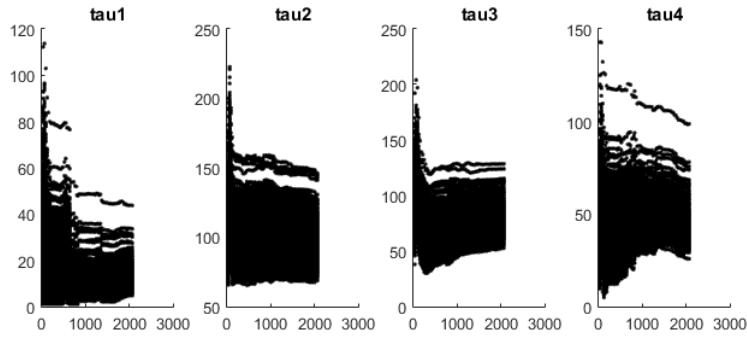


Figure 5.2: Time constants τ_j plotted against time. $N = 10000$

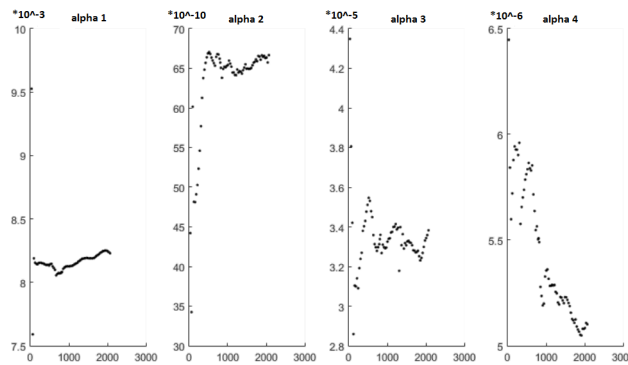


Figure 5.3: The average α_j plotted against time. $N = 10000$

ble. Especially for the producer P2, it takes some time for the EnKF to make a good prediction. After 20 time steps, all producer have found a close estimation of the true variables.

As seen in figure 3.1 on page 9 and 4.5 on page 20, the two main characteristics of the

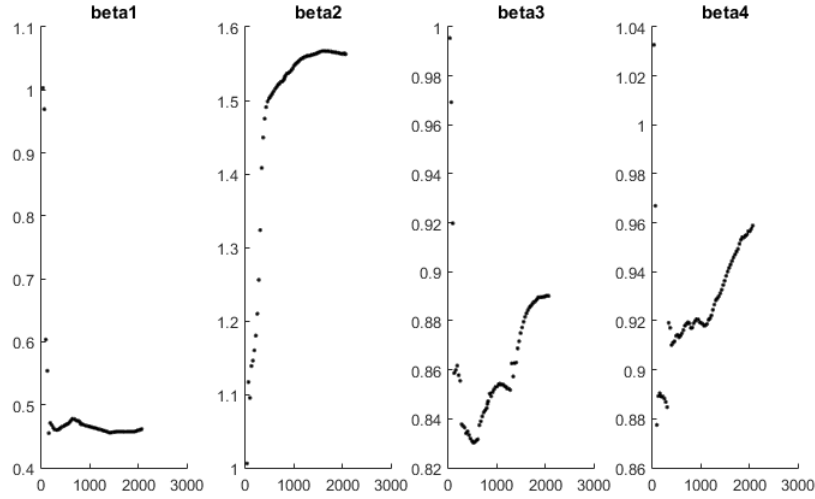


Figure 5.4: The average β_j plotted against time. $N = 10000$

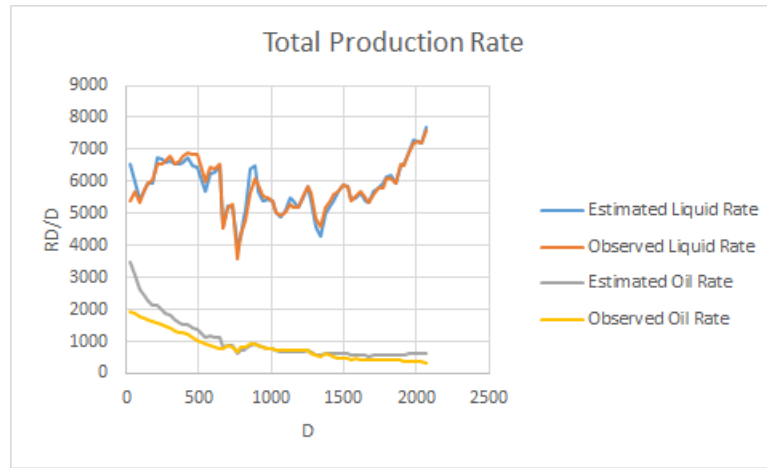


Figure 5.5: Total production rate after history matching plotted against time. The slight mismatch in oil rate at around 2000 days is caused by a rapid increase in injector 1.

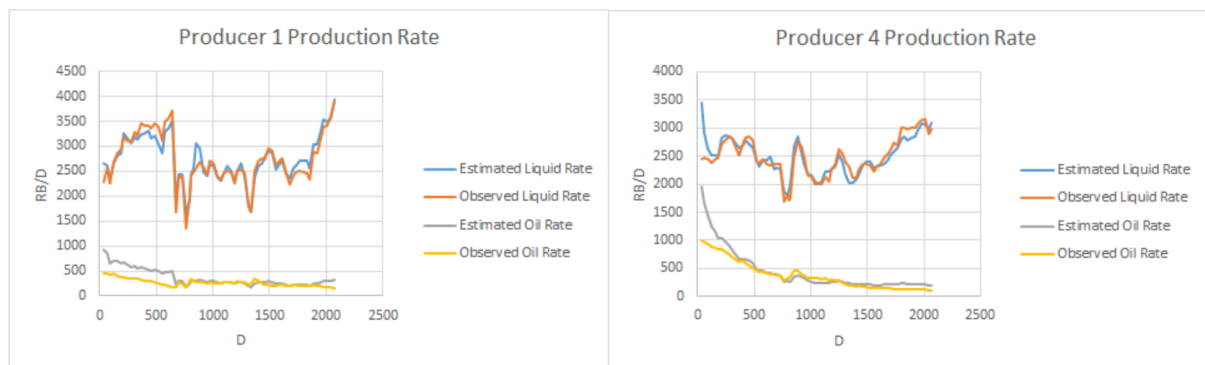


Figure 5.6: Production rate at producers P1 and P4 after history matching plotted against time.

reservoir flow are the high permeability streaks between injector 1 and producer 1, and between injector 3 and producer 4, which the EnKF results clearly reflect. Early in the production there

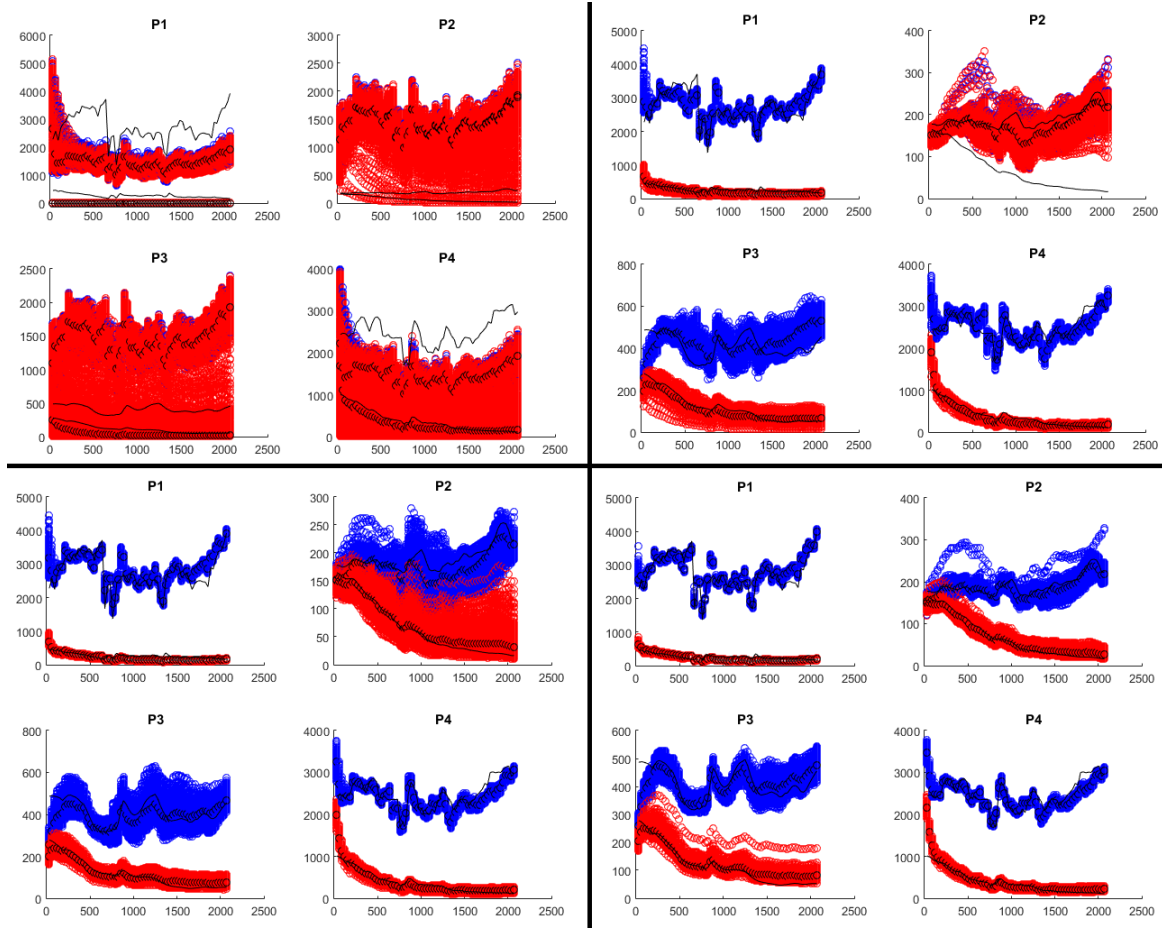


Figure 5.7: Liquid and oil rate after history matching with the EnKF. The black lines are the true observations of liquid rate and oil rate, while the dots are the rates when using the average model parameter values. The top left plots shows the initial rates without any history matching. The top right, the lower left and the lower right shows the rates after 10, 20 and 30 time steps of history matching, respectively. $N = 1000$

is a mismatch of the total rate and the rate at P4. This is caused by too high initial oil rate, which are the values solved in the VBA implementation. The resulting production shows that the EnKF can be used to identify the model parameters in the CRMP. However, it was necessary to have more prior knowledge of the possible ranges of the mean and standard deviation for the α .

5.6.2 CRMP and CRMIP

In this example the EnKF history matches both the CRMP and the CRMIP for the same reservoir and injection as in section 5.6.1. The resulting connectivities from the CRMP and CRMIP are compared to the 'true' connectivities. The 'true' connectivities are solved using Excel's GRG nonlinear solver in CRMP- and CRMIP implementations in VBA in Excel.

Two EnKF implementations of the CRMP and the CRMIP, respectively, were coded in MATLAB. The difference between the resulting connectivities after history matching and the 'true' connectivity was then plotted, with respect to both the number of realizations and the number of time steps history matched.

Note that the 'true' connectivities are not exactly the same for both the CRMP and the CRMIP. The 'true' values used can be found in the appendix on page 78.

For $N = 10$, $N = 100$ and $N = 1000$, the average difference in connectivity is plotted in figures 5.8, 5.9 and 5.10.

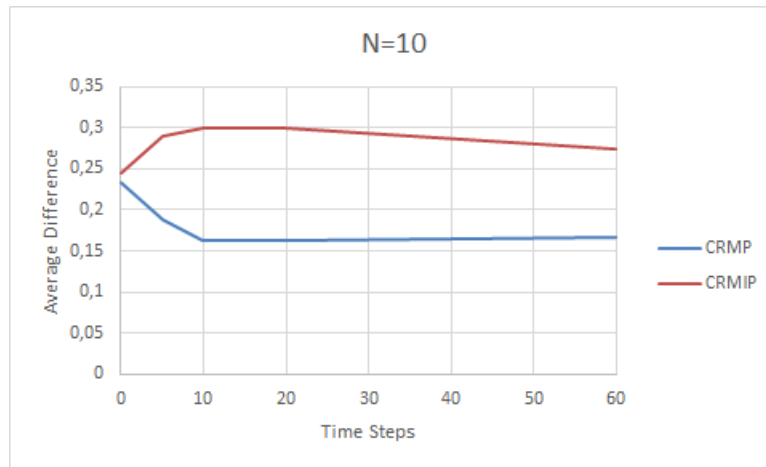


Figure 5.8: The average difference (error) in connectivity plotted against the number of time steps used in the EnKF history matching. $N = 10$

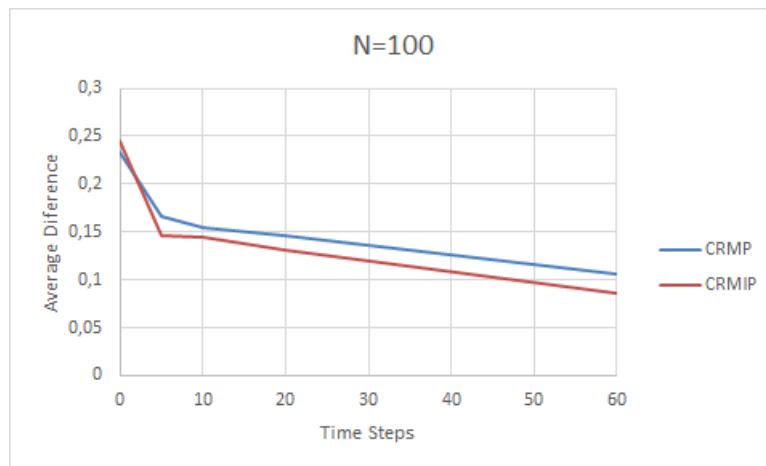


Figure 5.9: The average difference (error) in connectivity plotted against the number of time steps used in the EnKF history matching. $N = 100$

The resulting production rates can be seen in the appendix on page 73. The total estimated and the total observed rate will match well, even before any history matching has been done (can be seen in the appendix, figure 8.1 on page 73). This is due to the total connectivity f is

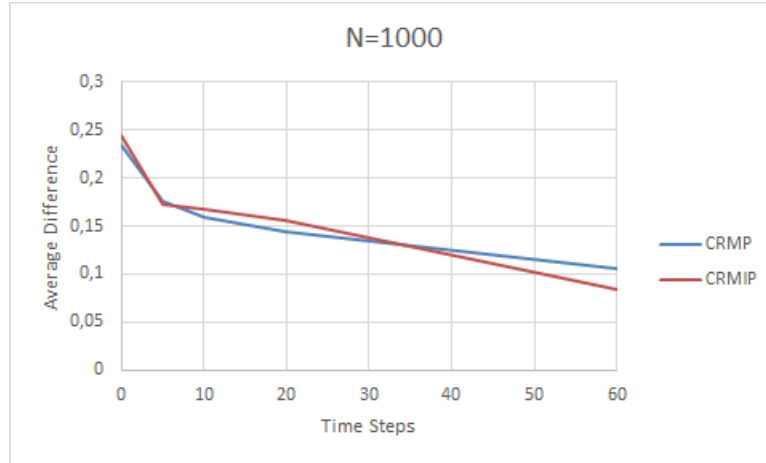


Figure 5.10: The average difference (error) in connectivity plotted against the number of time steps used in the EnKF history matching. $N = 1000$

equal to 1 for all injectors, meaning the total production will be the same despite changes in the individual f_{ij} values.

Figure 5.8 on the facing page shows that for $N = 10$, the EnKF provides no help in the history matching, while for $N = 100$ and $N = 1000$, the results improves for each time step towards the end.

The randomness of the EnKF prediction means the accuracy will vary for each run, so the results may not reflect the actual preciseness of the EnKF.

The plots in figure 5.8, 5.9 and 5.10 must not be confused with accuracy, as the increased number of variables used in the CRMIP will result in better matches.

5.7 Discussion

Overall the EnKF successfully identified all model parameters for the capacitance resistive models used for estimating liquid production rate. Although the liquid rates matched well with the identified model parameters, the match was not as good when estimating the oil rate using Gentil's fractional flow model. It required more prior knowledge on possible ranges of the mean and standard deviation for the α parameters.

The most promising quality of the EnKF, is its usability and the low computational requirements. It can solve systems consisting of large amount of model parameters, as long as they are not too nonlinear.

5.8 Conclusion

The ensemble Kalman filter is a data assimilation technique that is applicable on large-scale nonlinear systems. In this chapter it is shown that the EnKF provides good estimates for the model parameters in the CRMs.

The EnKF has low computational costs when handling large amount of variables compared to other techniques for nonlinear systems, such as the particle filter and the extended Kalman filter.

The EnKF estimation of model parameters was validated by comparing to the estimates from the GRG nonlinear solver in Excel literature work on the CRM.

5.9 Ensemble Kalman Filter Algorithm

EnKF steps	Formula	Description
% Input		
\mathbf{m}_o^u		% generate initial mod. para.
\mathbf{x}_o^u		% define/generate initial state
for k		% time step
% Forecast		
\mathbf{w}_k	$N(0, SD_w)$	% generate model noise
\mathbf{x}_k^f	$\mathbf{f}_k(\mathbf{x}_{k-1}^u, \mathbf{m}_{k-1}^u) + \mathbf{w}_k$	% forecast state
\mathbf{g}_k^f	$\mathbf{g}_k(\mathbf{x}_k^f, \mathbf{m}_{k-1}^u)$	% forecast measurement
\mathbf{Y}_k^f	$[\mathbf{m}_k^f; \mathbf{x}_k^f; \mathbf{g}_k^f]$	% big matrix \mathbf{Y}
\mathbf{P}_k^f	$\frac{1}{N-1}[\mathbf{Y}_k^f - E(\mathbf{Y}_k^f)][\mathbf{Y}_k^f - E(\mathbf{Y}_k^f)]^T$	% error covariance
% Filtering		
\mathbf{v}_k	$N(0, SD_v)$	% generate measurement noise
\mathbf{d}_k	$[\mathbf{d}_{obs} + \mathbf{v}_k^1 \quad \dots \quad \mathbf{d}_{obs} + \mathbf{v}_k^N]$	% perturb observation data
\mathbf{C}_{dk}	$\frac{1}{N-1}\mathbf{v}_k\mathbf{v}_k^T$	% error covariance
\mathbf{K}_k	$\mathbf{P}_k^f \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^f \mathbf{H}_k^T + \mathbf{C}_{dk})^{-1}$	% Kalman gain
\mathbf{Y}_k^u	$\mathbf{Y}_k^f + \mathbf{K}_k(\mathbf{d}_k - \mathbf{H}_k \mathbf{Y}_k^f)$	% updating \mathbf{Y}
\mathbf{P}_k^u	$\frac{1}{N-1}[\mathbf{Y}_k^u - E(\mathbf{Y}_k^u)][\mathbf{Y}_k^u - E(\mathbf{Y}_k^u)]^T$	% error covariance
end		

Ensemble Optimization

6.1 Introduction

The goal of optimization is to maximize a predefined objective by adjusting control variables. The objective in reservoir simulation could be to maximize oil production, or the net present value. The control variables could be injection rates, producing bottom hole pressure or well location.

Optimization plays a fundamental role in reservoir management as small adjustments can result in large gains of NPV. For the following examples, the NPV is given by the cost of injection, water production and oil production and a discount rate.

In this master thesis, the ensemble optimization (EnOpt) is used to maximize the NPV by changing the injection rates. Section 6.2 will provide some historical background of the use of optimization in reservoir simulation. Section 6.3 presents the optimization method of EnOpt. EnOpt will then be demonstrated using the CRMP and Gentil's fractional flow model in section 6.4.

6.2 Background

Several methods for determining optimal injection in reservoir simulation have been suggested in the past decades.

As an early example, Asheim (1986)[42] presented a method where he optimized water injection in a two phase flow reservoir with a high on a low permeability zone. The purpose of the numerical search algorithm was to maximize the net present value. The reservoir behavior was calculated using an IMPES (implicit pressure, explicit saturations) formulation. The results showed the importance of changing the injection rates over time for optimal rate profile. However, the lack of computational reservoir models in 1986 limited the usability, since each grid block contained one saturation- and one pressure variable, which must be calculated for each

time step, meaning a huge number of variables needed to be considered.

The computational development since then has made it possible to use optimization techniques on more complex reservoir models.

Nævdal et al. (2006)[43] described a continuous closed loop control of waterflooding, where the optimized control was generated based on the information available using a gradient-based optimization method. A reservoir model containing one production well and one injection well was used. The control was in this case individually inflow control valves in both producer and injector.

Nwaozo (2006)[44] used an ensemble of control variables to generate the control gradient, which can be considered preliminary work on the EnOpt[45]. Chen et al. (2009)[45] first proposed the ensemble-based optimization that is commonly described as the EnOpt[46].

Fonseca et al. (2015)[46] suggested an improvement to the EnOpt, which earlier had used constant covariance matrices for the control ensemble perturbations that resulted in a constant perturbation size. They suggested a covariance matrix that was to be updated at every time step. Their results showed this method to be efficient on challenging, small dimensional optimization problems.

Fonseca et al. (2015)[47] used the EnOpt to decide the control strategy of injection wells with inflow control devices (ICD) under geological uncertainty, and showed that the EnOpt resulted in an improved robust recovery strategy for realistic reservoir models.

As far as optimizing using CRMs, Sayarpour et al. (2009)[21] optimized the injection rate by reallocation and then used the injection rates in Eclipse, which resulted in increased oil production. They did not mention the specific optimization technique although it was most likely a solver in Excel. Weber (2009)[1] used the CRM to both optimize injection allocation and well location, by using optimization software from GAMS Development Corporation.

A complete close loop structure will be introduced in chapter 7.

6.3 Method

When optimizing injection for a reservoir model, one must deal with uncertainty. If we had complete knowledge of the reservoir (clairvoyance), the optimization would be quite easy and straight forward, since there would only be one defined realization of the model parameters. The reality is however, not straight forward, since our prior knowledge and the observed data contains uncertainty. The model parameters are therefore represented by an ensemble of possible parameter sets as a means of characterizing the individual parameter uncertainties. Subsequently, the optimal injection is the one that provides the highest expected NPV based on all possible parameter sets. It is therefore necessary to create an ensemble of the possible model parameters.

The structure used follows the structure presented in Fonseca et al. (2015)[46], and is also

known as the steepest ascent method.

The first step in the optimization is to produce N_r different random sets of control variables given by the prior control vector $\hat{\mathbf{u}}$ and a standard deviation C_u :

$$\mathbf{u}_r = N(\hat{\mathbf{u}}, C_u) \quad (6.1)$$

The second step is to estimate the objectives $J_r(\mathbf{u}_r)$ for each combination of model parameter ensemble and the control ensemble. Then, the mean-shifted ensemble matrix is computed:

$$\bar{\mathbf{U}} = [\mathbf{u}_1 - \hat{\mathbf{u}} \quad \mathbf{u}_2 - \hat{\mathbf{u}} \quad \dots \quad \mathbf{u}_{N_r} - \hat{\mathbf{u}}]^T \quad (6.2)$$

where \mathbf{u}_i are the new control vectors generated. The mean-shifted objective function vector becomes:

$$\mathbf{j} = [J_1(\mathbf{u}_1) - J_1(\hat{\mathbf{u}}) \quad J_2(\mathbf{u}_2) - J_2(\hat{\mathbf{u}}) \quad \dots \quad J_{N_r}(\mathbf{u}_{N_r}) - J_{N_r}(\hat{\mathbf{u}})]^T \quad (6.3)$$

The gradient \mathbf{G} [46] is approximated:

$$\mathbf{G} = (\bar{\mathbf{U}}^T \bar{\mathbf{U}})^{-1} \bar{\mathbf{U}}^T \mathbf{j} \quad (6.4)$$

\mathbf{G} may also be written[45] as:

$$\mathbf{G} = \mathbf{C}_{uu}^{-1} \mathbf{c}_{uJ} \quad (6.5)$$

where

$$\mathbf{C}_{uu} = \frac{1}{N_r - 1} (\bar{\mathbf{U}}^T \bar{\mathbf{U}}) \quad (6.6)$$

and

$$\mathbf{c}_{uJ} = \frac{1}{N_r - 1} (\bar{\mathbf{U}}^T \mathbf{j}) \quad (6.7)$$

This description of \mathbf{G} only applies for overdetermined cases, which is true for the example in this thesis, where N_r is greater than the length of \mathbf{u} . For an underdetermined case, where the length of \mathbf{u} is greater than N_r , $\bar{\mathbf{U}}^T \bar{\mathbf{U}}$ is rank-deficient and its inverse cannot be directly computed[46]. If that is the case, \mathbf{G} becomes:

$$\mathbf{G} = \mathbf{C}_{uu} \mathbf{c}_{uJ} \quad (6.8)$$

The new control vector (6.9) is then calculated:

$$\hat{\mathbf{u}}_{l+1} = \hat{\mathbf{u}}_l + \alpha \frac{\mathbf{G}}{\max|\mathbf{G}|} \quad (6.9)$$

where α is the step length when updating and l is the iteration number. If the new control vector does not generate a high enough objective value compared to the old control vector, the step length α is reduced. This process continuous until a stopping criteria is reached.

6.4 Example Using the CRMP and Gentils Fractional Flow Model

The goal of the following example is to maximize the objective value by optimizing the injection control in the reservoir model presented in section 3.1 on page 9. The GRG nonlinear solver in the VBA implementation will be used to solve the model parameters. The resulting realization will be used as the true model parameters.

In order to achieve the highest possible objective value by controlling the injection, it is necessary to know the exact model parameter, which in most cases would not be possible to achieve. This problem will be examined in more detail in chapter 7, where a possible solution for this issue will be tested.

The optimized injection will be used for both an Eclipse simulation of the reservoir, and in a CRMP implementation in VBA. The resulting production will then be interpreted.

6.4.1 Method

As in the EnKF example in section 5.6.1, the last 2000 days of production was used. As seen in figures 5.5 and 5.6, the oil rate tends to not match well at the start of production. The injection rates are therefore optimized from 750 days until the end of production.

The model parameters were set by using the GRG nonlinear solver in a VBA implementation of the CRMP and the Gentil fractional flow model.

The control vector consist of 10 control periods of around 100 – 120 days each, with exception to the last period of 243 days. For 5 injectors the control vector will have 50 values for a single realization. The prior control is a constant injection rate of 1500 RB/D for all injectors.

The standard deviation of the control vector was predefined to 100. There is no well defined method to decide the standard deviaton[46], and in this case a standard deviation of a little less than 10% (6.7%) of the initial injection rate, gave a consistent optimal control. $N_r = 100$ new control vectors were generated for each iteration. As N_r becomes larger, the accuracy increases. However, since there was only 50 control values for each realization, it was not necessary to increase the number of realizations.

The procedure then followed the structure in section 6.6 for the overdetermined case, with a $\alpha_{init} = 1000$.

The for loop will stop if the following criterion is reached: If the average NPV of the updated control vector \mathbf{u}_r ($r = 1$ to N_r), is less than for the old control vector $\hat{\mathbf{u}}$, α is divided by two, and the old control vector is updated once more. This process continues until the alpha has been divided 5 times without improving the NPV. The resulting control vector variables are the optimized injection rate values. This method follows the algorithm in section 6.6 for the overdetermined case.

The objective used is defined as:

$$J = \int_{t=0}^T \frac{q_o(t)r_o - q_{wp}(t)r_{wp} - q_{wi}(t)r_{wi}}{(1+b)^{t/\tau}} dt \quad (6.10)$$

where q_o , q_{wp} and q_{wi} are the oil rate, water production rate and water injection rate, respectively. The $r_o = 50$ USD/RB, $r_{wp} = 10$ USD/RB and $r_{wi} = 10$ USD/RB are the costs. The discount rate $b = 0.1$, while $\tau = 365$ days is the referencing time for discounting.

6.4.2 Results

The GRG nonlinear solver in Excel produced the model parameters shown in figure 6.1.

	P 1	P 2	P 3	P 4
f 1,j	0,76	0,00	0,01	0,22
f 2,j	0,43	0,02	0,16	0,38
f 3,j	0,27	0,05	0,03	0,65
f 4,j	0,31	0,13	0,03	0,53
f 5,j	0,32	0,01	0,13	0,53
τ	10	181	71	37
α	0,0078	6,9E-09	3,0E-05	6,9E-06
β	0,47	1,59	0,88	0,93

Figure 6.1: Model parameters generated by a VBA implementation of CRMP and Gentil's fractional flow model.

The ensemble optimization suggested that injection wells 3 and 5 should be closed, whilst the rest of the injection should be set as shown in figure 6.2 on the following page.

Figure 6.3 shows the improvement in the production by comparing the prior control vector to the new optimized control.

By running the new injection rates in Eclipse, the the objective became worse after the optimization. Both the water and oil production decreased slightly.

Although it is not easy to see on the plot, the objective value for the CRMP and Gentils fractional flow model increased after the optimization. The oil production increased by around

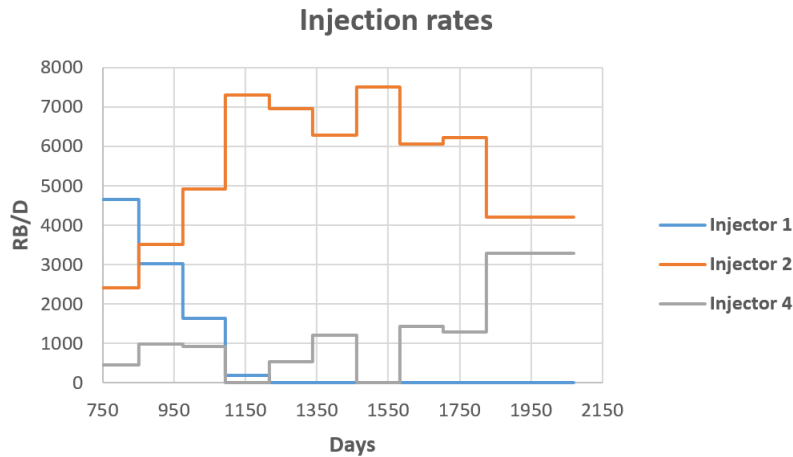


Figure 6.2: Optimized injection rates.

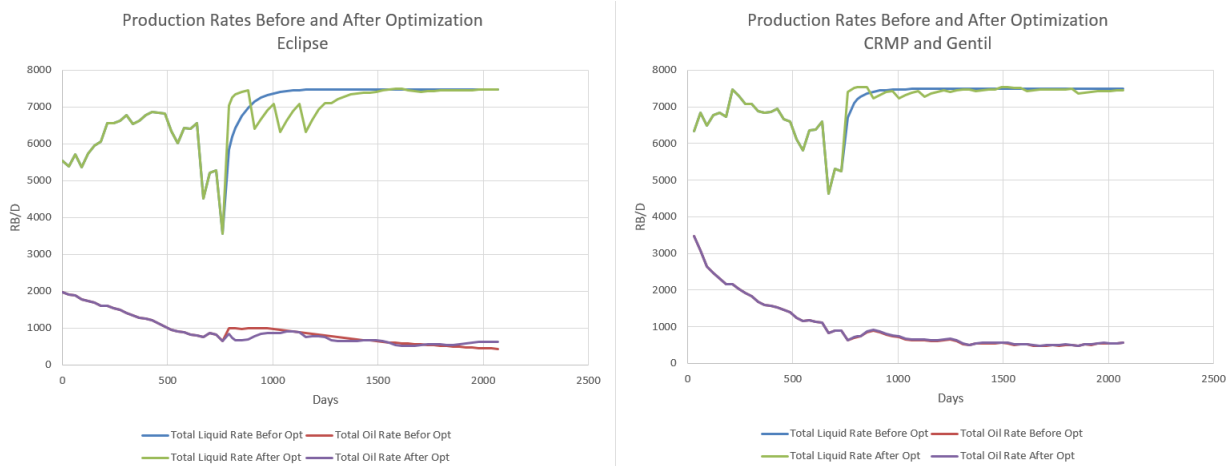


Figure 6.3: Total production rate and total oil rate before and after optimization.

19000 RB and the water production decreased by around 30000 RB, leading to a gain of approximately \$1 million by the definition of the objective.

Note that the large difference in early oil production in figure 6.3 for the Eclipse simulation compared to CRMP and Gentil's fractional flow, is due to Gentil's fractional flow model not being able to take into account primary recovery effects.

6.4.3 Discussion

The reason for the lower objective value in the Eclipse simulation, is not related to the ensemble optimization, but rather to the fact that the CRMP and Gentil's fractional flow model will not completely match the Eclipse production.

The fact that the reservoir is quite simple, and the criterion for injection control is total constant injection, means that there is not going to be a large difference in total production rate for different injection controls. This means it is not necessarily the CRMs lack of ability to

represent a real reservoir that causes the Eclipse simulation not to improve, but it could be a lack of complexity in the reservoir model and control criterion.

If the EnOpt is to be used as part of a larger implementation, the computational costs will be an important factor. It can be necessary to spend time finding the lowest number of realizations N_r that provide acceptable results.

The value of information, which is going to be calculated in the next chapter, could give a better indication of positive usability of EnOpt, as it will calculate the NPV using the prior and the updated control.

6.5 Conclusion

This chapter tested the EnOpt ability to optimize injection in a reservoir model using CRMP and Gentil's fractional flow model. The EnOpt combined well with the CRMP and Gentil's model, and produced consistent results, increasing the objective value compared to an initial base case. Based on the results, it cannot be concluded if EnOpt and CRM can be used to control a real (Eclipse) reservoir model, due to the simplistic reservoir model used and the constant total injection criterion.

The EnOpt formulation means it is applicable for more complex systems than has been tested in this thesis.

6.6 Ensemble Optimization Algorithm

EnOpt steps	Formula	Description
\mathbf{u}		% prior control
α_{init}		% initial step length
for l		% iteration step
\mathbf{u}_r	$N(\hat{\mathbf{u}}, C_u)$	% generate ensemble control
\mathbf{J}_r	$J(\mathbf{u}_r)$	% estimate objective
$\bar{\mathbf{U}}_l$	$[\mathbf{u}_{1l} - \hat{\mathbf{u}}_l \quad \dots \quad \mathbf{u}_{N_r l} - \hat{\mathbf{u}}_l]^T$	% meanshifted ensemble matrix
\mathbf{j}_l	$[J_1(\mathbf{u}_{1l}) - J_1(\hat{\mathbf{u}}_l) \dots J_{N_r}(\mathbf{u}_{N_r l}) - J_{N_r}(\hat{\mathbf{u}}_l)]^T$	% meanshifted objective vector
\mathbf{C}_{uu}	$\frac{1}{N_r - 1} (\bar{\mathbf{U}}^T \bar{\mathbf{U}})$	% ensemble covariance matrix
\mathbf{c}_{uJ}	$\frac{1}{N_r - 1} (\bar{\mathbf{U}}^T \mathbf{j})$	% cross-covariance vector
\mathbf{G}	$\mathbf{C}_{uu}^{-1} \mathbf{c}_{uJ}$ (if overdetermined) $\mathbf{C}_{uu} \mathbf{c}_{uJ}$ (if underdetermined)	% gradient vector
if	$\bar{J}_{new} - \bar{J}_{old} < \text{allowable error}$	% objective gain
	$\alpha = \frac{\alpha}{2}$	% reduce step length
	$\alpha \text{ counter} + = 1$	% count reductions
	(check stopping criterion)	
else		
	$\alpha = \alpha_{init}$	reset step length
	$\alpha \text{ counter} = 0$	reset counter
end		
$\hat{\mathbf{u}}_{l+1}$	$\hat{\mathbf{u}}_l + \alpha \frac{\mathbf{G}}{\max \mathbf{G} }$	% update control vector
end		

Closed Loop Reservoir Management

7.1 Introduction

One of the two main goals of this thesis is to implement and evaluate the usability of closed loop reservoir management with capacitance resistive models. This chapter aims to use the closed loop structure from Barros et al. (2015)[3] with CRMs to estimate the value of information (VOI).

Section 7.2 contains some background information of closed loop reservoir management using EnKF and optimization.

Section 7.2.1 describes an important concept for the CLRM used, which is value of information (VOI).

7.2 Background

Chen (2009)[45] presented a closed loop production optimization approach using ensemble optimization (EnOpt) with the ensemble Kalman filter (EnKF). The results showed that this framework was able to history match a reservoirs geological features and increase the NPV for two different reservoir model with varying permeability and number of wells.

Brouwer (2004)[48] developed a closed loop structure where the control was optimized as new information became available. His motivation was to make smart wells that can adapt to unexpected changes in a reservoir.

As mentioned in section 6.2, Nævdal et al. (2006)[43] used the EnKF in a closed loop control in a reservoir model with one producer and one injector. In this case, the EnKF was used to update a reservoir model based on the information available. The permeabilities were model parameters, while the states contained pressures and saturations. After the update, the optimal control strategy was identified. The control determined the injection from different inflow control valves in both the injector and producer. This resulted in a higher total oil production than

using conventional wells.

The basic structure of a closed loop reservoir management can be seen in figure 7.1. When working on the closed loop implementation it is useful to have an understanding of this figure and figure 7.2, which has been used to explain CLRM in several articles[2, , 3].

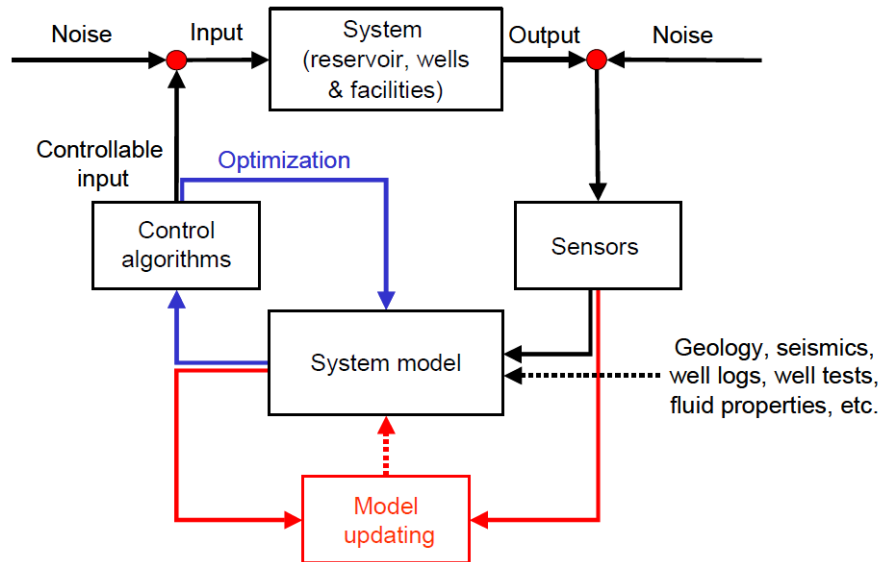


Figure 7.1: Closed loop reservoir control management. This figure was originally used in Brouwer et al. (2004)[2].

7.2.1 Value of Information

Value-of-analysis is used to determine whether a reduction in uncertainty is worth the cost of gathering information[19]. For a better description of the concept of value of information, as well as its use in the oil gas industry, see Bratvold and Begg (2010)[19] and Bratvold et al. (2009)[8].

In this master thesis, the VOI is defined as:

$$\text{Value of new information} = \text{Value with new information} - \text{Value with old information} \quad (7.1)$$

The simplicity of the formula does not reflect the complexity of the actual calculation. In real life, the calculation relies on numerous factors. One must first define what information should be taken into account. In the petroleum industry this information may come from acquiring data, technical studies, hiring consultants, performing diagnostic tests, etc[19].

For each information there is an underlying uncertainty, both for current information (are the studies reliable? In what degree is the historic data relevant for data decision context? Does the tests include model uncertainty?), and for new information (observation and model

uncertainty). It is also useful to perform sensitivity analysis in order to determine the ranges that new information must have in order to produce value.

Value of clairvoyance (VOC), or value of perfect information, is a useful term when using VOI in a decision context. A clairvoyant is defined as someone with complete knowledge of information, both current and future.

Although, it is not possible to make a decision based on VOC (unless it eliminates all other alternatives), it has some helpful properties. The value of clairvoyance may be used as an upper boundary for what a person or a company should be willing to spend for the information. If for example a seismic survey is quoted to cost more than the value of clairvoyance, it should automatically be rejected. When writing a implementation of a VOI structure, the VOC is useful to identify errors, as the calculated VOI should not be higher than the VOC ceiling.

7.3 CLRM for VOI

Barros et al. (2015)[3] presented a new methodology for estimating VOI using closed loop reservoir management. They concluded that their method was more complete compared to previous work which aimed to estimate VOI, but also slower computationally.

The procedure for estimating value of information in this master thesis is based Barros et al. (2015)[3]. The premise of said article, was to provide a way for estimating VOI (in reservoir simulation) of a test (ex. doing a seismic survey) using history matching, and prior knowledge of the mean and standard deviation of the individual model parameters (ex. Saturations, porosities, etc.) that follows a normal distribution. It is also unknown what a test would show given the actual reservoir parameters. A test will contain uncertainty and probably show results that differ from the actual true values[29].

Without any information of the model parameters, one must assume a uniform distribution of equal opportunity in the entire sample space. This would not be computationally feasible. It is therefore necessary to have prior knowledge that limits the possible parameter values. Additionally, for saturations and porosities, we know the values must lie between 0 and 1. In a oil-water system, the saturations must be $S_o + S_w \leq 1$. We can also assume that the porosity will not exceed 50%. Additional constraints results in less computational costs for the VOI implementation.

To calculate the VOI, all relevant model parameters must be taken into account. This would most likely include sensitivity analysis regarding the importance of the different parameters, as well as to what degree it is possible to simplify the problem and reduce the parameters space in order to reduce computational costs.

7.3.1 CLRM for VOI Structure

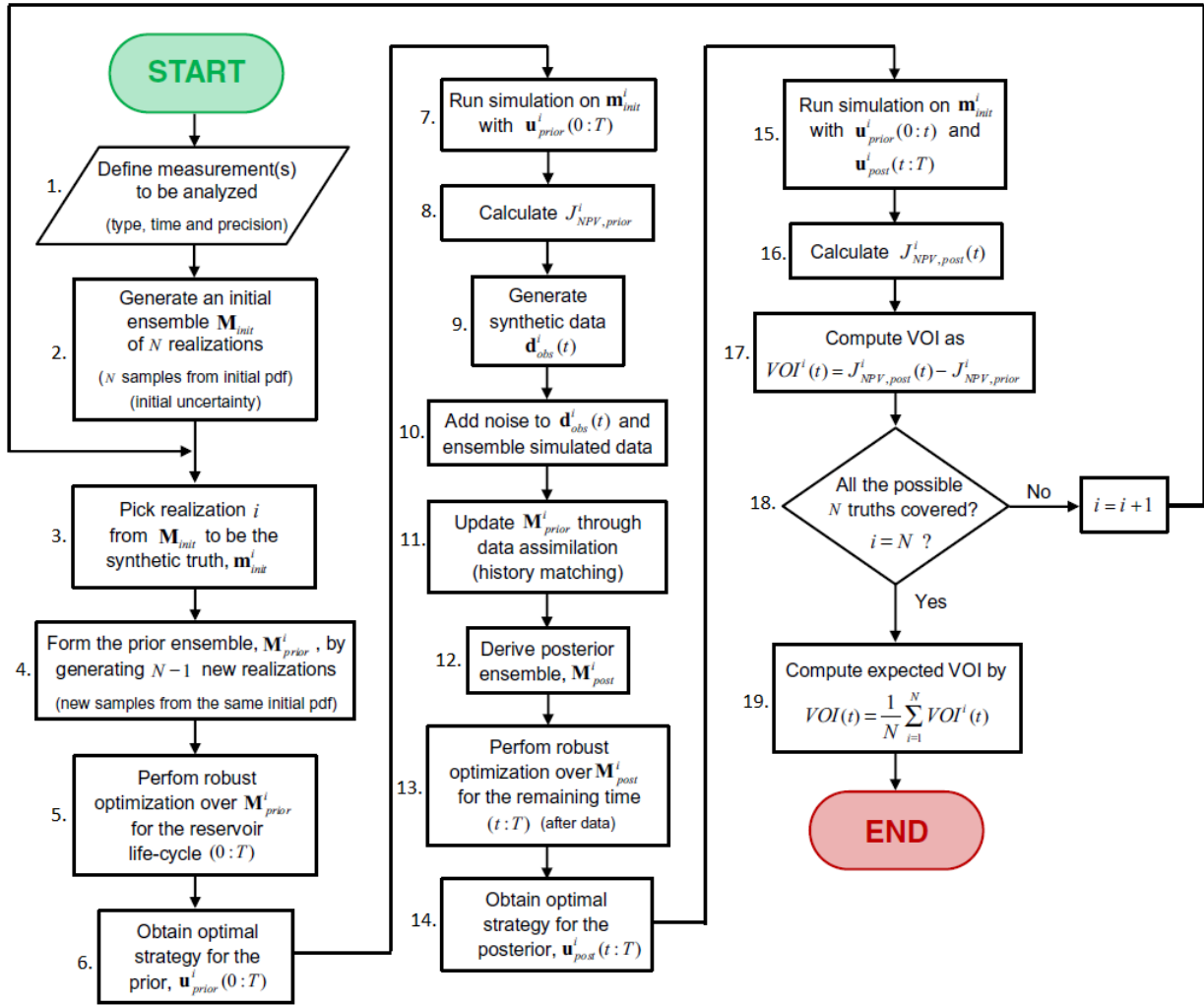


Figure 7.2: Closed loop reservoir management structure used to estimate expected value of information. The figure was originally used in Barros et al. (2015)[3]. The step numbers were not included in the original figure and was edited in later to be used in this master thesis.

1st step. One first has to define what the measurements are going to be. For a reservoir model, the rates would be a logical choice, as it is in this project. However, the measurements could in reality be anything, as long as it is possible to calculate an objective from the measurements.

2. Generate an initial ensemble \mathbf{M}_{init} . These are N different sets of model parameter realizations, which has to be generated. The reason being that value of information in this structure, actually is the average value of information created from each realization. This means it is necessary to have some prior knowledge of what values the parameter can have. Therefore, the model parameters are generated randomly from a known mean, as well as a predefined standard deviation.

3. Pick a realization to be the "truth". When writing a code of the structure, a for loop picks one of the realizations in M_{init} to be the synthetic truth m_{init} .

4. The prior ensemble M_{prior} is then generated. Depending on computational speed requirements this could either mean generating $N - 1$ new realizations or using the $N - 1$ realizations left from step 2.

5 and 6. The ensemble optimization, as presented in chapter 6, is used to generate the control vector u_{prior} that creates the highest distribution of the objectives for the total production time from the different realizations in M_{prior} .

7. The optimal control vector created in step 5 is then used with the synthetic true model parameters over the total production time.

8. Calculate objective for the synthetic truth. Step 7 will result in the prior objective, i.e. the objective without any information except the prior knowledge in step 2.

9. Generate synthetic data $d_{obs}(t)$. Using the prior control vector u_{prior} and the synthetic true model parameters m_{init} , one calculates the synthetic data at time t . This means that if the structure is to be used to evaluate the best time to gain information, it is necessary to evaluate synthetic data at every relevant time step.

10. Add noise to d_{obs} . If the true synthetic data is known, the optimization of the control would be trivial. The data observed from a test will contain observation error. The error is randomly generated with a mean = 0 and a predefined standard deviation. This results in the observation as estimated in equation 5.24 on page 34.

11. Update the prior ensemble. The prior ensemble M_{prior} is updated through history matching using the data gained in step 10. The history matching is done through data assimilation using the ensemble Kalman filter.

12. The updated $M_{posterior}$ is gained from the data assimilation in step 11.

13 and 14. Optimize the posterior control $u_{posterior}$. Using the updated $M_{posterior}$, one generates the optimal control vector similar to step 5. The difference is that the control vector can only change for the time after t in $d_{obs}(t)$. This means one generates the optimal control vector from time t to the end of the production time T .

15. The synthetic truth m_{init} is run with the prior control u_{prior} from the start of the production to time t , and the posterior control vector $u_{posterior}$ for the remaining time.

16. The posterior objective is then calculated from step 15.

17. Compute VOI. The value of information is calculated by subtracting the prior objective from the posterior objective. The result is interpreted as how much objective (NPV) one would gain from information at time t .

18. The process from step 3 to step 17 is repeated iteratively until all realizations in the initial ensemble M_{init} has been used as synthetic truth.

19. The expected value of information is computed by taking the average VOI estimated in step 17 for each iteration.

7.4 Case with CRMT and BLBFFM

The following case uses closed loop reservoir management control for the reservoir presented in section 3.2 with one injection well and one production well. The structure seen in figure 7.2 is used to estimate the value of information and the value of clairvoyance.

7.4.1 Methods

The information gained in this case comes from the model parameters that are actual reservoir properties, which includes the average porosity, irreducible water saturation, initial water saturation, residual oil saturation and the time constant τ (equation 4.5 on page 16), which is a function of total compressibility and pore volume. The other variables; initial liquid rate, m and n are assumed to be known and has the values 527 RB/D, 35.5 and 1.75, respectively. The values of $q(0)$, m and n was set using by solving Eclipse data with the CRMT and the BLBFFM with the nonlinear GRG solver in the VBA implementation.

The reservoir produces for 3000 days. The control vector u contains 10 different control values, each for a control period of 300 days.

$N = 30$ different realizations of the model parameters was used, as well as $N_r = 20$ different realizations of control vectors when optimizing.

The objective in this and the following example is:

$$J_{NPV} = \frac{1}{N} \sum_{i=1}^N J_i \quad (7.2)$$

where the individual objective realization J_i is:

$$J_i = \int_{t=0}^T \frac{q_o(t, \mathbf{m}_i)r_o - q_{wp}(t, \mathbf{m}_i)r_{wp} - q_{wi}(t, \mathbf{m}_i)r_{wi}}{(1+b)^{t/\tau}} dt \quad (7.3)$$

where q_o , q_{wp} and q_{wi} are the oil rate, water production rate and water injection rate, respectively. The costs $r_o = 50$ USD/RB, $r_{wp} = 10$ USD/RB, $r_{wi} = 20$ USD/RB, the reference time for discounting $\tau = 365$ days and the discount rate $b = 0.1$.

7.4.2 Results

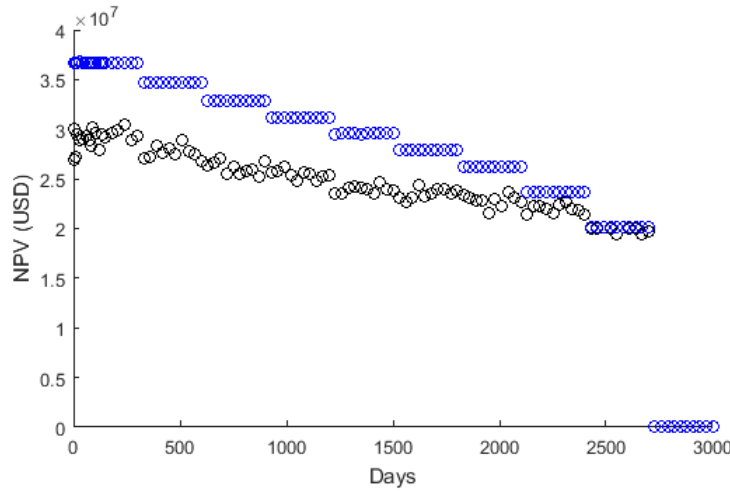


Figure 7.3: Expected Value of Information (black dots) and Expected Value of Clairvoyance (blue dots) plotted against the day in which the information was gathered.

The resulting expected VOI and expected VOC is plotted in figure 7.3. The net present value of VOI indicate the maximum amount of money one should be willing to spend on gathering information at that time. As an example, this means that, assuming risk neutrality, one should be willing to spend around \$30 million or less for the information at around 290 days, and no more than \$36.5 million. The reason for such a high VOI, is probably caused by a significant number of model parameter realizations resulting in very little oil production. The net present value of information gathered is highest at earlier time and then decreases towards the end of production. The VOI and VOC is zero in the last control period, since it is not possible to use information gathered in this period to change any control values.

7.4.3 Discussion

As expected, the VOC provides an upper boundary for the VOI.

The value of information tends to be higher at the end of control periods and lower at the start of control periods. This is more noticeable at earlier part of the production. The reason for this increase in VOI, is that the later time tends to give more accurate information, meaning that the information gives a better estimate for the "true" model parameters. Since the control value can only change before the control period starts, the information gathered at the end of a control period is more valuable.

If some of the possible model parameter values are not realistic, this will affect the VOC and the VOI. Lets say the porosity of a realisation was approximately zero. In that instance the VOI would have a unrealistic high value, since the oil production would be low, which means it will be valuable to reduce the injection to a minimum. It will therefore be a large difference in net present value using the prior control and the updated control. A unrealistically high porosity (for example 40 %), would also result in too high VOI and VOC.

It is important to acknowledge that every simulation will result in different VOI and VOC. To get a similar result for multiple simulations, the number of realizations, both for the model parameters and the control vectors, would have to increase. For the implementation in this project, $N = 30$ was used since $N = 20$ did not produce a good result, and $N > 30$ would have taken too long to run the simulations. $N_r = 11$ is a low number of realizations, but produced reasonable and consistent optimization results.

7.5 Example CRMP and Gentils Fractional Flow Model

The next example combines the structure in figure 7.2 with the CRMP and Gentil's fractional flow model. This enables information to be gathered from individual wells. In this example, information is gathered from only producer 1.

7.5.1 Methods

Since Gentil's fractional flow model only gives a good match at later time of production, the initial reservoir pressure was set to $p_{ri} = 250$. The model parameters and initial conditions were the same as the EnKF example in section 5.6.1 on page 35, with regards to the history matching.

For this case, only the liquid and oil rates based on the synthetic true model parameters m_{init} from Producer 1 was used in the EnKF update, meaning only prior information is available for the rest of the field.

7.5.2 Results

Figure 7.4 shows the expected VOI and VOC. The VOI increases until around 300 – 400 days, and then decreases towards the end. The figure indicates that the value of information is worth the most at 300 days, where one should be willing to pay around \$42 million or less, assuming risk neutrality.

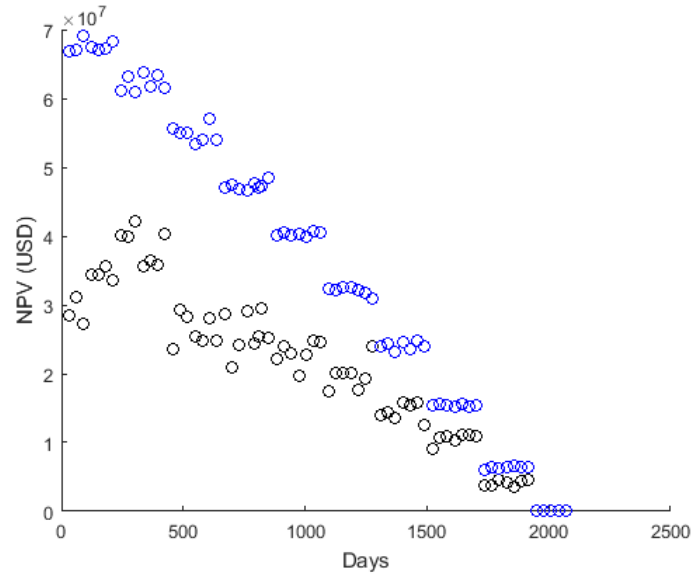


Figure 7.4: Expected Value of Information (black dots) and Expected Value of Clairvoyance (blue dots) for information of producer 1 plotted against the day in which the information was gathered. $N = 30$, $N_r = 11$

7.5.3 Discussion

As seen in figure 7.4, the VOC is not constant for each control period, which it should be by definition. The number of control variables that can be changed in a specific control period is constant, meaning that the NPV should stay the same. The probable reason is that due to the slow computer speed, the number of control vector realizations was lowered to $N_r = 11$, which resulted in less accurate calculation.

The Gentil's fractional flow uses fitting parameters α and β to estimate the oil production and does not include real reservoir parameters. The resulting VOI does therefore not reveal what kind of information that should be gathered, but may be helpful to give an indication on when it would be most profitable to gather the information.

For a significant number of model realizations, the β became too high and as a result produced a very small or negligible oil fraction. The β s was generated around a mean of 2, while in retrospect a mean for the β equal 1 would have been a better choice, as more of the realizations would produce significant amounts of oil.

7.6 Implementation Challenges

There was a problem when the upper boundary was set too low, and the number of control vector realization N_r was set too low. If for example, a control value at a control period before the updated control vector was equal to the upper boundary, and all the ensemble control values in the same control period was higher than the upper boundary, which means the values are lowered to the upper boundary, then the \bar{U} would have an empty row. This in turn would mean that the gradient G is calculated using the inverse of a matrix containing an empty column and an empty row, which is not possible to calculate. This effect can be avoided by increasing the number of control vector realizations. Another possible solution is to include a truncated normal distribution (a MATLAB version can be downloaded online[49]), where the randomly generated variables falls between a pre-set range. This was only an issue for CRMT and BLBFFM. This effect did not cause problems for the case with CRMP and Gentil fractional flow model, since the boundary was a constant total injection rate of 7500 RB/D, which was divided by the five injection wells, meaning the individual well had a more random nature to the injection rates and was much less likely to have a boundary issue.

There is also an inverse matrix stability issue when calculating the gradient G which may occur when the matrix contains very small numbers. This leads to the MATLAB warning “matrix is singular/badly scaled”. To reduce this issue, the matrix division operator $x=A \backslash b$ was used instead of $x=\text{inv}(A)*b$. The MATLAB[50] ‘help documentaion’ advices to use $A \backslash b$ and not $\text{inv}(A)*b$ when solving a system of linear equations $Ax=b$.

7.7 Conclusion

This chapter used a CLRM structure consisting of EnOpt and EnKF to estimate expected VOI for two different CRMs. The structure provided reasonable values for both VOI and VOC.

The capacitance resistive models have been shown to be suitable for all parts of the closed loop structure. The oil fractional flow model however, affect the usability. As seen in the example in section 4.6.1, figure 4.12 shows that Gentil’s fractional flow model results in a significant mismatch at the start of production. It is therefore important to be certain that primary recovery has ended. In the end it became necessary to lower the initial reservoir pressure to producing bottom hole pressure and skip the first 1000 days to be able to history match using the EnKF. The CRMT and BLBFFM may seem to be a better fit, but since only one example of a reservoir consisting of one producer and one injector was used, it cannot be concluded.

The premises for the VOI assessments discussed here should be transferable to real cases. In a decision context, the selection of the model parameters is an important part of the process.

Being able to identify the necessary parameters, is essential to derive a robust estimation of the VOI. It is also be important to have a clearer understanding of the various models' usability and limitations.

In the implementation in this thesis, there was an issue when deciding a standard deviation for the initial model parameter realizations, the lower and upper values of the estimated oil rate may be unrealistic. This will lead to a high VOI, which makes it necessary to increase the ensemble size.

A big weakness of this implementation is the computational speed. Running the simulation with CRMT and BLBFFM with $N = 30$ realizations took around 12 hours (using Intel(R)Xeon(R)CPU X5650@2.67GHz 2.66GHz (2 processors)), while the CRMP and Gentil's fractional flow model took 3-7 days, depending on the number control vector realizations and the number of measurement data. There is potential to reduce the computational costs by further vectorization and optimization of the EnOpt code, but this will not change the computational burden by orders of magnitude. The main reason for the high computational costs is the probabilistic optimization. A possible solution is to limit the possible injection values to a relatively small subset.

The master thesis may be summarized with a few concluding remarks:

- The ensemble Kalman filter successfully identified the model parameters of the capacitance resistive models and the oil fractional flow models, but had some difficulties identifying the α parameters.
- The closed loop reservoir management produced reasonable values of information and values of clairvoyance when combined with capacitance resistive models and oil fractional flow models.
- The use of EnKF with CRM equations shows promise in a closed loop reservoir management structure, but the ensemble optimization used in this thesis may be too slow to be used on more complex reservoir models.

Further Work

A critical time period for estimating VOI is around water breakthrough. At this period uncertainty connected to the WOR can be large, and therefore possible large VOI. Implementing an oil fractional flow, which takes into consideration water breakthrough uncertainty at individual wells, would provide better VOI estimations. In this case it would also be important to evaluate the usability of the CRMs liquid rate estimation during primary recovery.

One of the main issues with the closed loop reservoir management structure in this thesis, is the slow computational speed, mainly caused by the slow optimization. Being able to increase the optimization speed, would significantly increase the usability of the VOI structure.

To be used for more realistic cases, work needs to be done with evaluating how the prior model parameter ensemble is defined. It can be interesting to evaluate how the ensemble Kalman filter reacts when the model parameters follows a more nonlinear distribution. One can also evaluate how the result changes if model noise for the EnKF is included.

An important part of the EnKF that has not been addressed in detail in this thesis, is the covariance matrices. For a more complete work on the EnKF, one should spend more time explaining their physical meaning and how they affect the estimation.

The results in chapter 6, indicates that it can be interesting to evaluate the CRMs ability to be used in injection optimization for a real reservoir or a more complex reservoir model.

For any students trying to implement the same VOI structure, I suggest making new codes for the EnOpt and the EnKF using the formulas given in this thesis. There is significant potential for reducing the computational costs by vectorization that was not used in the implemented codes in the appendix.

Bibliography

- [1] D.B. Weber. The use of capacitance-resistance models to optimize injection allocation and well location in water floods. *PhD Dissertation. The University of Texas at Austin, Austin, Texas*, 2009.
- [2] D.R. Brouwer, G. Nævdal, J.D. Jansen, E.H. Vefring, and C.P.J.W. van Kruijsdijk. Improved reservoir management through optimal control and continuous updating. *paper prepared for presentation at SPE Annual Technical Conference and Exhibition in Houston, Texas, 26–29 Sep. 2004, Society of Petroleum Engineers*, 2004.
- [3] E.G.D. Barros, P.M.J. Van den Hof, and J.D. Jansen. Value of information in closed-loop reservoir management. *Comput Geosci*, 2015.
- [4] G. Nævdal, T. Mannseth, and E.H. Vefring. Near-well reservoir monitoring through ensemble kalman filter. *Paper presented at the SPE/DOE Improved Oil Recovery Symposium, Tulsa, Paper SPE 75235*, 2002.
- [5] L. Chu, A.C. Reynolds, and D.S. Oliver. Reservoir description from static and well-test data using efficient gradient methods. *Paper presented at the International Meeting on Petroleum Engineering, Beijing, China, Paper SPE 29999*, 1995.
- [6] G. Evensen. Sequential data assimilation with a nonlinear quasi-geostrophic model using monte carlo methods to forecast error statistics. *Journal of Geophysical Research*, 99(C5):10,143–10,162, 1994.
- [7] D.S. Oliver, L.B. Cunha, and A.C. Reynolds. Markov chain monte carlo methods for conditioning a permeability field to pressure data. *Mathematical Geology*, 29(1), 1997.
- [8] R.B. Bratvold, J.E. Bickel, and H.P. Lohne. Value of information in the oil and gas industry: Past, present, and future. *Society of Petroleum Engineers Reservoir Evaluation Engineering*, 12(4):630–638, 2009.

-
- [9] S. Begg, R. Bratvold, and J. Campbell. The value of flexibility in managing uncertainty in oil and gas investments. *SPE Annual Technical Conference and Exhibition, San Antonio, Texas.*, Paper SPE 77586, 2002.
- [10] J.E. Bickel. Discretization, simulation, and the value of information. *SPE Annual Technical Conference and Exhibition, Denver, 2011.*, Paper SPE 145690, 2012.
- [11] B.J.A. Willigers, S. Begg, and R.B. Bratvold. Combining geostatistics with bayesian updating to continually optimize drilling strategy in shale-gas plays. *EAGE Annual Conference Exhibition/SPE Europec, London, 2013*, Paper SPE 164816, 2014.
- [12] A.D.D. Stensgaard. Capacitance resistive modelling for waterflood estimation. *Specialization Project TPG4530, NTNU*, 2015.
- [13] A. Albertoni and L.W. Lake. Inferring interwell connectivity only from well-rate fluctuations in waterfloods. *SPEREE*, 6(1):6–16, 2003.
- [14] A.A. Yousef, P.H. Gentil, J.L. Jensen, and L.W. Lake. A capacitance model to infer interwell connectivity from production- and injection-rate fluctuations. *SPEREE*, 9(5):630–646, 2006.
- [15] X. Liang. A simple model to infer interwell connectivity only from well-rate fluctuations in waterfloods. *Journal of Petroleum Science and Engineering*, 70:35–43, 2010.
- [16] M. Sayarpour. Development and application of capacitance-resistive models to water/co₂ floods. *PhD Dissertation. The University of Texas at Austin, Austin, Texas*, 2008.
- [17] G. Evensen. The ensemble kalman filter: theoretical formulation and practical implementation. *Ocean Dynamics*, 53, 2003.
- [18] J. Eidsvik, T. Mukerji, and D. Bhattacharjya. Value of information in the earth sciences: Integrating spatial modeling and decision analysis. *Cambridge University Press*, 2015.
- [19] R.B. Bratvold and S.H. Begg. Making good decisions. *Society of Petroleum Engineers*, 2010.
- [20] A. Fahimuddin. 4d seismic history matching using the ensemble kalman filter (enkf): Possibilities and challenges. *PhD Dissertation. University of Bergen, Bergen, Norway*, 2010.
- [21] M. Sayarpour, E. Zuluaga, C.S. Kabir, and L.W. Lake. The use of capacitance-resistance models for rapid estimation of waterflood performance and optimization. *Journal of Petroleum Science and Engineering*, 69:227–238, 2009.
-

-
- [22] M. Sayarpour, C.S. Kabir, K. Sepehrnoori, and L.W. Lake. Probabilistic history matching with the capacitance-resistance model in waterfloods: A precursor to numerical modeling. *Journal of Petroleum Science and Engineering*, 78:96–108, 2011.
- [23] H.R. Warner Jr. Waterflooding. In L.W. Lake and E.D. Holstein, editors, *Petroleum engineering handbook*, volume V, pages 1037–1102. Society of Petroleum Engineers, 2006.
- [24] S.E. Buckley and M.C. Leverett. Mechanism of fluid displacement in sands. *AIME*, 146:107–116, 1942.
- [25] J. Kleppe. Hand-out note no. 6: Dykstra-Parson’s method for isolated layers. retrieved from: <http://www.ipt.ntnu.no/ kleppe/tpg4150/>. *TPG4150, NTNU*, 2015.
- [26] P.H. Gentil. The use of multilinear regression models in patterned waterfloods: Physical meaning of the regression coefficients. *Master Thesis. The University of Texas at Austin, Austin, Texas*, 2005.
- [27] M.C. Leverett. Capillary behavior in porous solids. *AIME*, 142:152–169, 1941.
- [28] R.E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME-Journal of Basic Engineering*, 82 (Series D):35–45, 1960.
- [29] S.I. Aanonsen, G. Nævdal, D.S. Oliver, A.C. Reynolds, and B. Vallès. The ensemble kalman filter in reservoir engineering-a review. *Society of Petroleum Engineers*, 14:393–412, 2009.
- [30] G. Welch and G. Bishop. An introduction to the kalman filter. *Department of Computer Science, University of North Carolina at Chapel Hill*, Tech. Rep. TR95041, 2006.
- [31] J.P. Jensen. Ensemble kalman filtering for state and parameter estimation on a reservoir model. *Master Thesis. Norwegian University of Science and Technology, Trondheim, Norway*, 2007.
- [32] R.J. Lorentzen, K.K. Fjelde, J. Frøyen, A.C.V.M Lage, G. Nævdal, and E.H Vefring. Underbalanced drilling: Real time data interpretation and decision support. In *Paper SPE 67693 presented at the SPE/IADC Drilling Conference*, Amsterdam, The Netherlands, 2001.
- [33] R.J. Lorentzen, G. Nævdal, and A.C.V.M Lage. Tuning of parameters in a two-phase flow model using an ensemble kalman filter. *International Journal of Multiphase Flow*, 29:1283–1309, 2003.
-

-
- [34] G. Evensen, J. Hove, H.C. Meisingset, E. Reiso, K.S. Seim, and Ø. Espelid. Using the enkf for assisted history matching of a north sea reservoir model. In *Paper 106184 presented at the SPE Reservoir Simulation Symposium*, Woodlands, Texas, 2007.
- [35] P. Sakov. Asynchronous data assimilation with the enkf, lecture retrieved from: <http://enkf.nersc.no/presentations/>. *Nansen Environmental and Remote Sensing Center, Norway*, EnKF home page of Geir Evensen, 2011.
- [36] G. Burgers, P.J. van Leeuwen, and G. Evensen. Analysis scheme in the ensemble kalman filter. *Monthly Weather Review, American Meteorological Society*, 126((6)):1719–1724, 1998.
- [37] Y. Chen, D.S. Oliver, and D. Zhang. Data assimilation for nonlinear problems by ensemble kalman filter with reparameterization. *Journal of Petroleum Science and Engineering*, 66:1–14, 2009.
- [38] A.C. Reynolds, M. Zafari, and G. Li. Iterative forms of the ensemble kalman filter. In *ECMOR X - 10th European Conference on the Mathematics of Oil Recovery*, Amsterdam, the Netherlands, 2006.
- [39] Y. Gu and D.S. Oliver. An iterative ensemble kalman filter for multiphase fluid flow data assimilation. *Society of Petroleum Engineers*, 12((4)):438–446, 2007.
- [40] G. Li and A.C. Reynolds. Iterative ensemble kalman filters for data assimilation. *Society of Petroleum Engineers*, 14((3)), 2009.
- [41] Y. Gu. History matching production data using the ensemble kalman filter. *PhD Dissertation. University of Oklahoma, Norman, Oklahoma*, 2006.
- [42] H. Asheim. Optimal control of water drive. *Society of Petroleum Engineers 15978*, provided to the Society of Petroleum Engineers for distribution and possible publication in an SPE journal., 1986.
- [43] G. Nævdal, D.R. Brouwe, and J.D. Jansen. Waterflooding using closed-loop control. *Computational Geosciences*, 10:37–60, 2006.
- [44] J. Nwaozo. Dynamic optimization of water flood reservoir. *Master thesis, University of Oklahoma, Norman, Oklahoma*, 2006.
- [45] Y. Chen, D.S. Oliver, and D. Zhang. Efficient ensemble-based closed-loop production optimization. *Society of Petroleum Engineers*, 14((4)):634–645, 2009.

-
- [46] R.M. Fonseca, O. Leeuwenburgh, and P.M.J Van den Hof. Improving the ensemble-optimization method through covariance-matrix adaptation. *Society of Petroleum Engineers*, 2015.
- [47] R.M. Fonseca, O. Leeuwenburgh, E. Della Rossa, P.M.J. Van den Hof, and Jansen J.D. Ensemble-based multiobjective optimization of on/off control devices under geological uncertainty. *Society of Petroleum Engineers*, accepted for presentation at the SPE Reservoir Simulation Symposium, Houston, 23–25 Feb. 2015, 2015.
- [48] D.R. Brouwer. Dynamic water flood optimization with smart wells using optimal control theory. *PhD Dissertation. Delft University of Technology, Delft, the Netherlands*, 9(4):391–402, 2004.
- [49] B. Luong. Truncated gaussian. retrieved from <http://www.mathworks.com/matlabcentral/fileexchange/23832-truncated-gaussian>, 2009 (updated 2010).
- [50] MATLAB. *version 8.5.0.197613 (R2015a)*. The MathWorks Inc., 2015.

Appendix A

Results EnKF for CRMP and Gentils fractional flow for the reservoir in section 3.1. Only the last 2000 days of production was used.

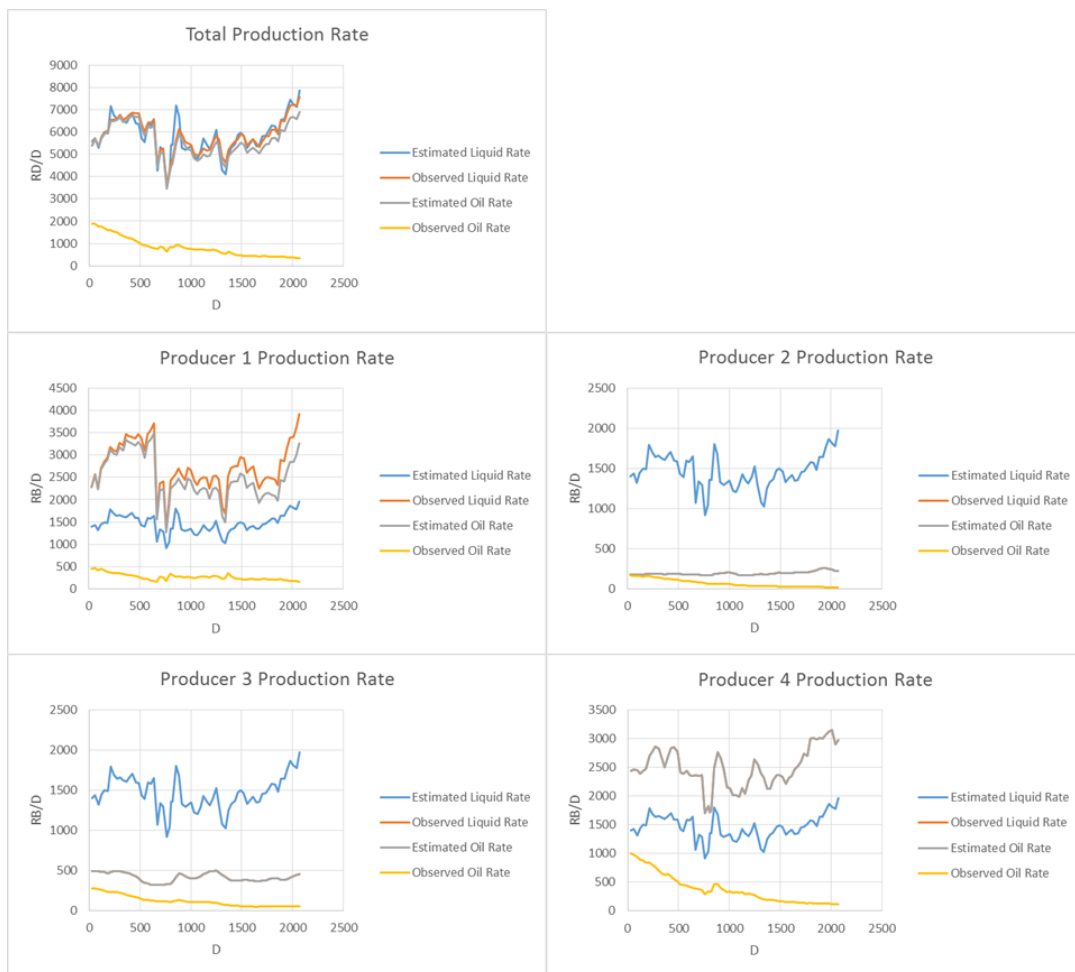


Figure 8.1: Production rates without any knowledge of the model parameters. For graphs where observed liquid rate are not visible, it is approximately equal to the estimated oil rate.

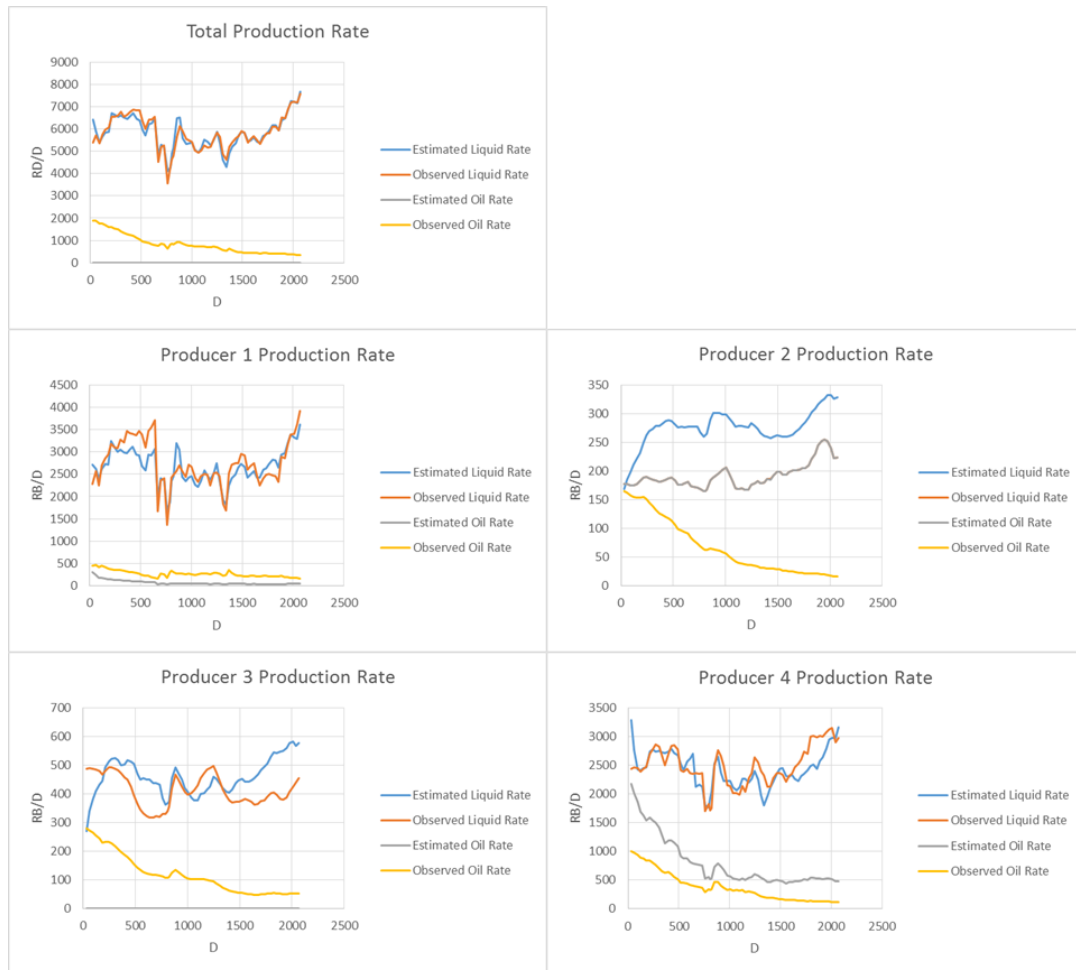


Figure 8.2: Production rates after history matching with the EnKF for the first 5 time steps. For graphs where observed liquid rate are not visible, it is approximately equal to the estimated oil rate.



Figure 8.3: Production rates after history matching with the EnKF for the first 10 time steps. For graphs where observed liquid rate are not visible, it is approximately equal to the estimated oil rate.



Figure 8.4: Production rates after history matching with the EnKF for the first 20 time steps. For graphs where observed liquid rate are not visible, it is approximately equal to the estimated oil rate.



Figure 8.5: Production rates after history matching with the EnKF for all 60 time steps.

The 'true' connectivity values used in section 5.6.2 on page 39.

	P 1	P 2	P 3	P 4
f 1,j	0,925	0,000	0,000	0,075
f 2,j	0,469	0,031	0,160	0,339
f 3,j	0,136	0,000	0,001	0,862
f 4,j	0,200	0,065	0,029	0,706
f 5,j	0,186	0,081	0,166	0,506

Figure 8.6: CRMP connectivity solved by Excels GRG nonlinear solver.

	P 1	P 2	P 3	P 4
f 1,j	0,949	0,000	0,008	0,042
f 2,j	0,499	0,022	0,173	0,306
f 3,j	0,092	0,000	0,000	0,908
f 4,j	0,222	0,167	0,000	0,611
f 5,j	0,147	0,014	0,171	0,668

Figure 8.7: CRMIP connectivity solved by Excels GRG nonlinear solver.

Results EnKF for CRMP and Gentils fractional flow for the reservoir in section 3.1. Only the last 2000 days of production was used.

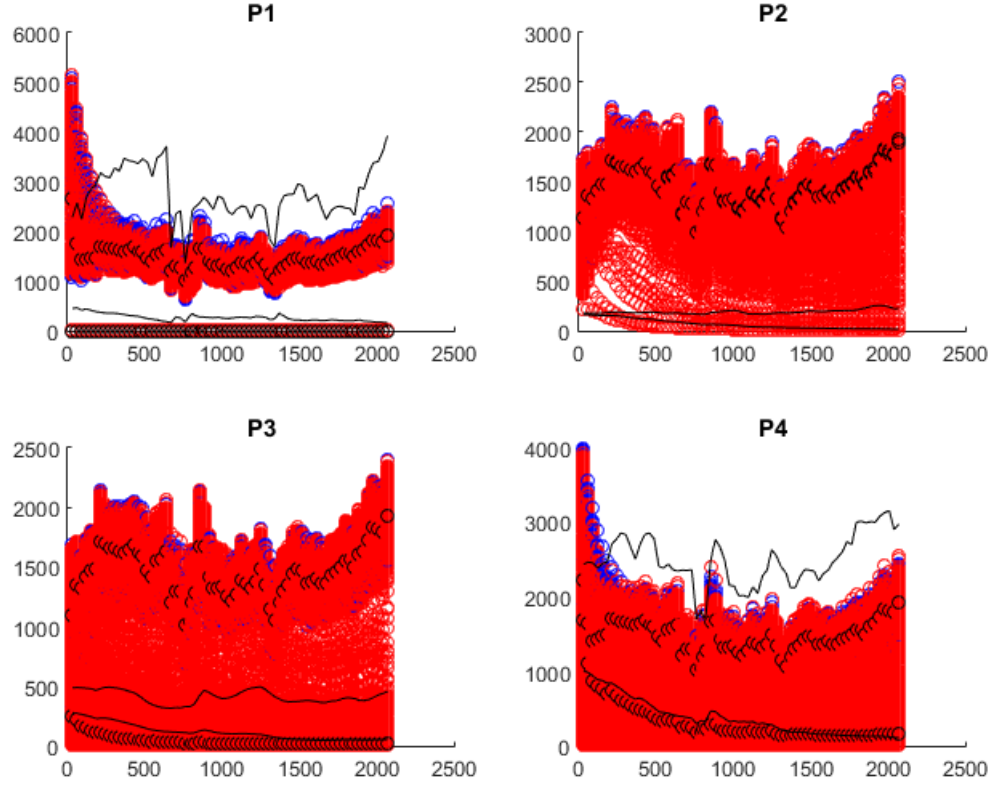


Figure 8.8: The initial range of liquid rate (blue) and oil rate (red) when using the EnKF. The black lines are the true observations of liquid rate and oil rate, while the dots are the rates when using the average model parameter values. $N = 1000$

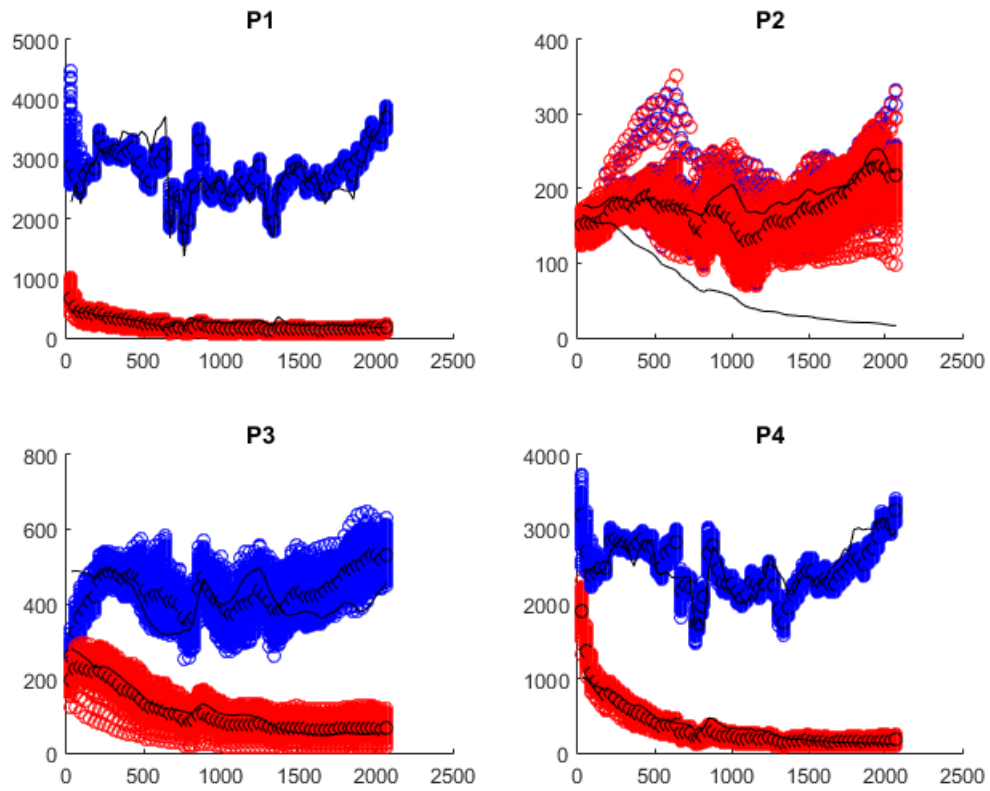


Figure 8.9: Liquid rate (blue) and oil rate (red) when using the EnKF after 10 time steps. The black lines are the true observations of liquid rate and oil rate, while the dots are the rates when using the average model parameter values. $N = 1000$

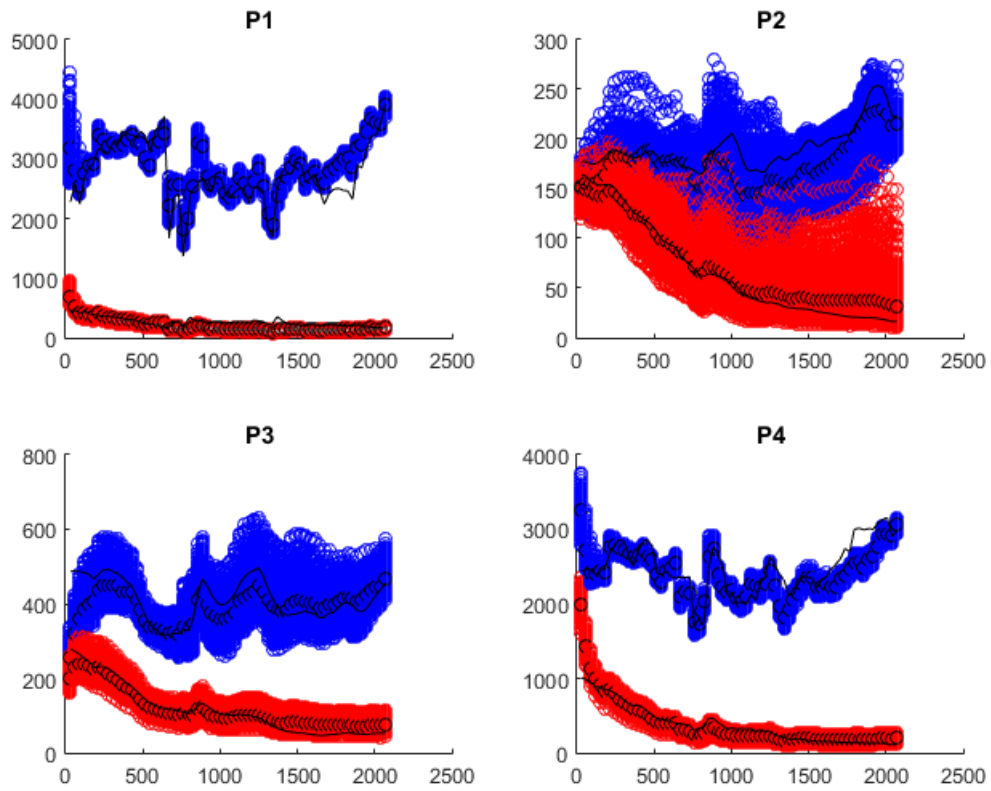


Figure 8.10: Liquid rate (blue) and oil rate (red) when using the EnKF after 20 time steps. The black lines are the true observations of liquid rate and oil rate, while the dots are the rates when using the average model parameter values. $N = 1000$

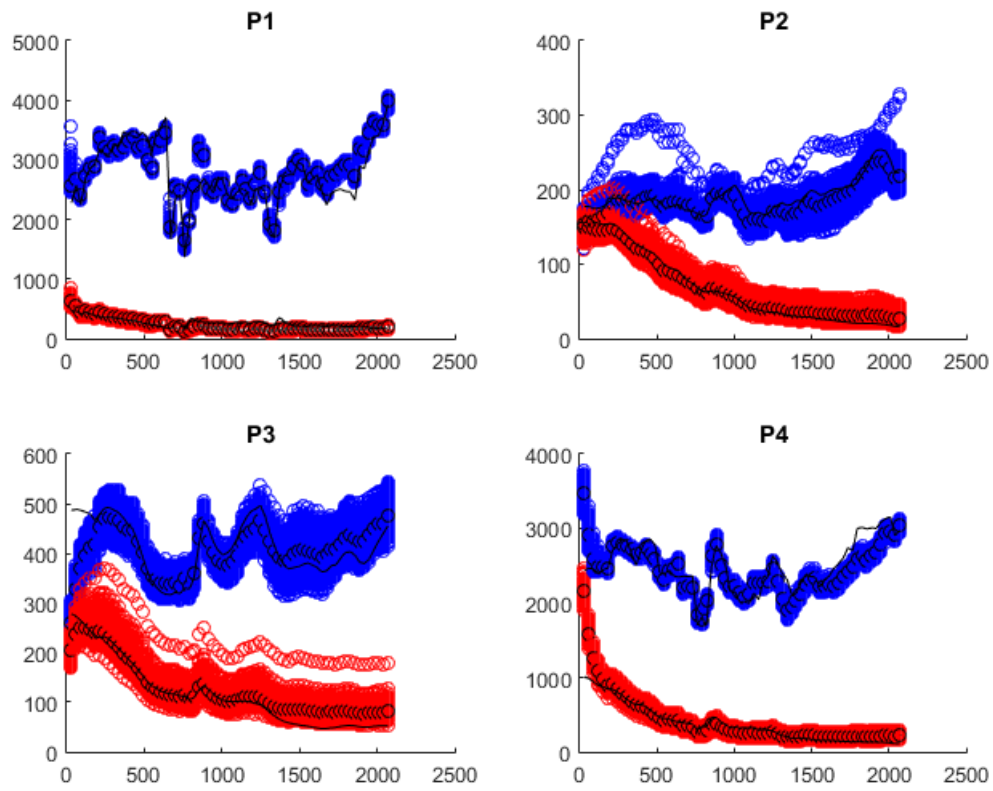


Figure 8.11: Liquid rate (blue) and oil rate (red) when using the EnKF after 30 time steps. The black lines are the true observations of liquid rate and oil rate, while the dots are the rates when using the average model parameter values. $N = 1000$

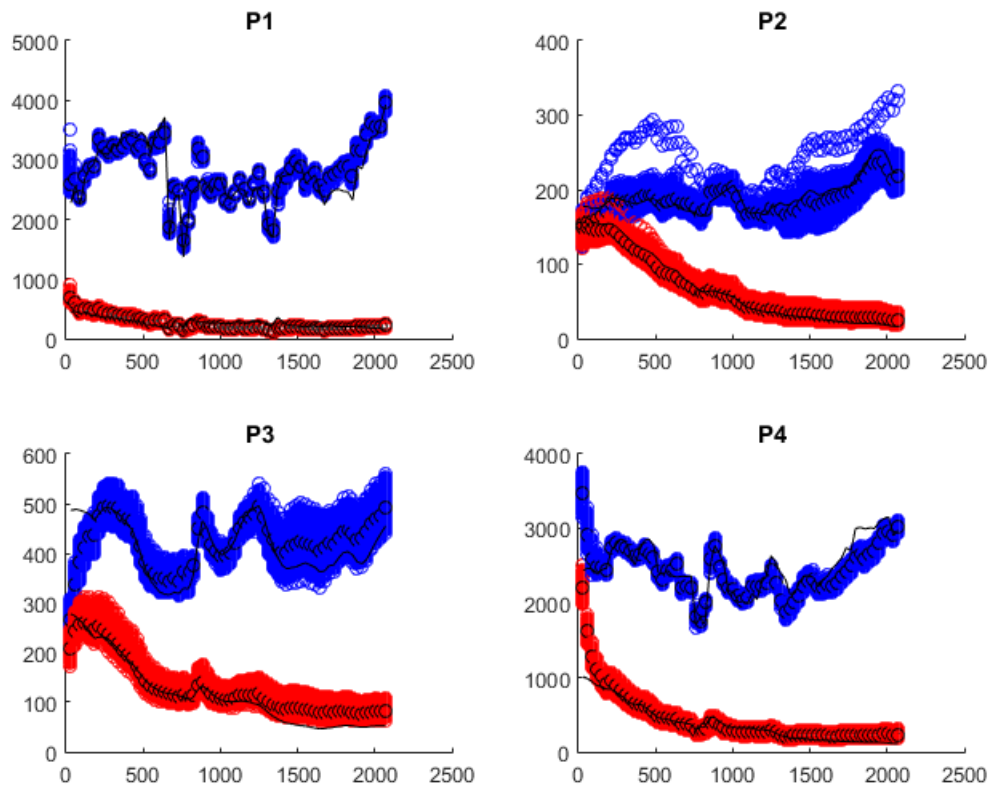


Figure 8.12: Liquid rate (blue) and oil rate (red) when using the EnKF after 40 time steps. The black lines are the true observations of liquid rate and oil rate, while the dots are the rates when using the average model parameter values. $N = 1000$

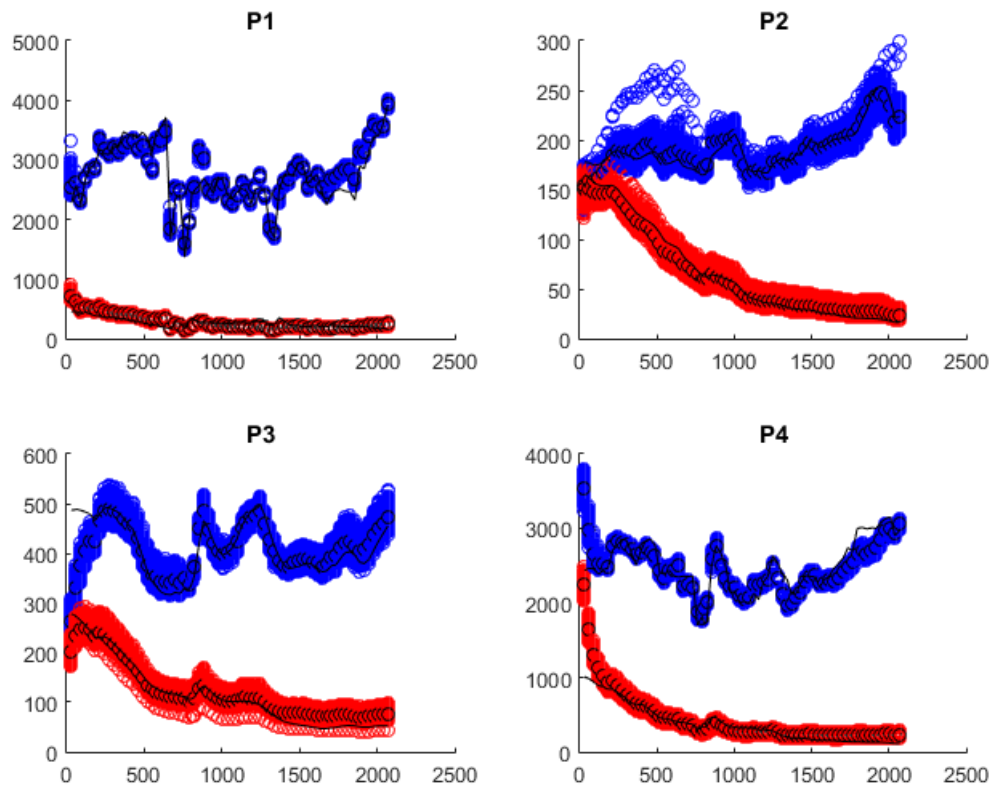


Figure 8.13: Liquid rate (blue) and oil rate (red) when using the EnKF after 50 time steps. The black lines are the true observations of liquid rate and oil rate, while the dots are the rates when using the average model parameter values. $N = 1000$

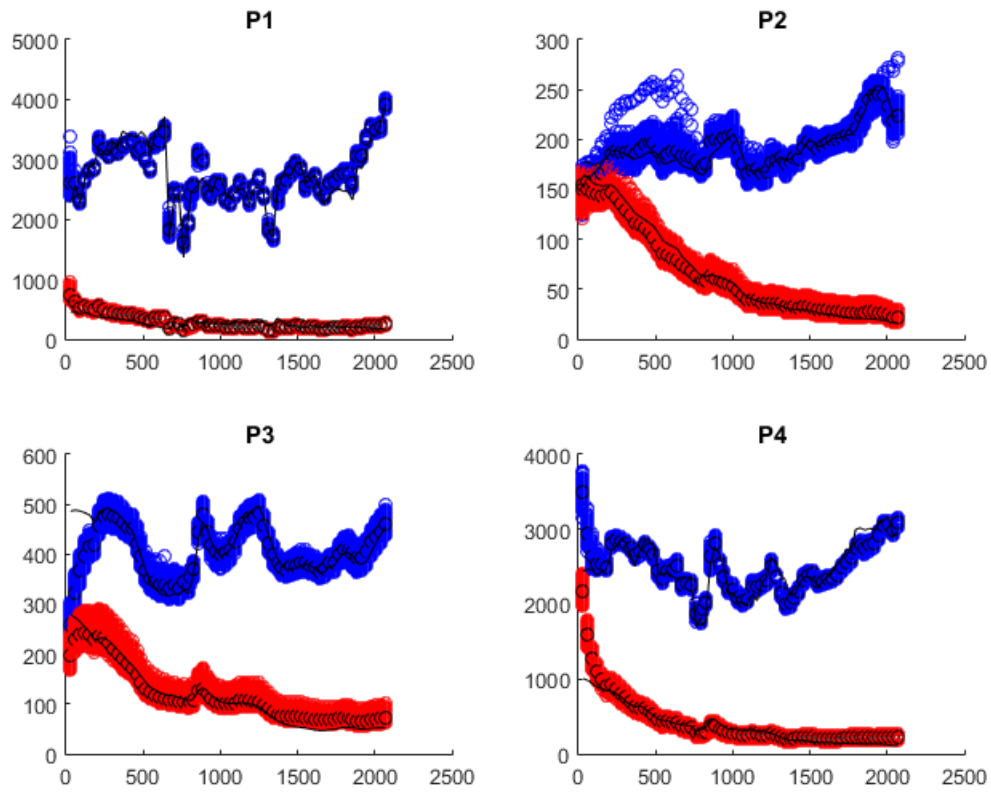


Figure 8.14: Liquid rate (blue) and oil rate (red) when using the EnKF after 60 time steps. The black lines are the true observations of liquid rate and oil rate, while the dots are the rates when using the average model parameter values. $N = 1000$

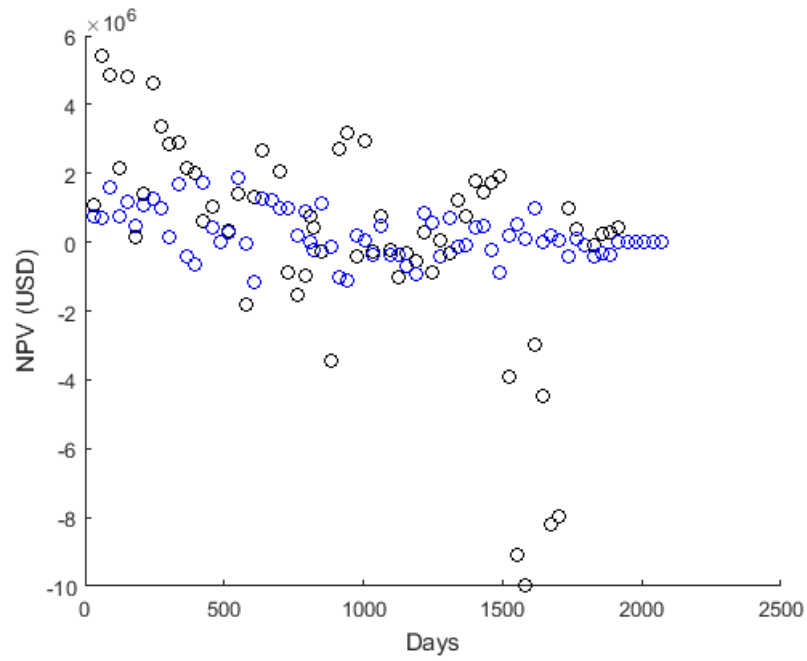


Figure 8.15: Expected VOI (black) and expected VOC (blue) estimation using CRMIP and Gentils fractional flow model for information from producer 1. The bad result shows the necessity of large enough model parameter sets. $N = 10$, $N_r = 11$

Appendix B

Example of EnOpt in MATLAB (CRMT and BLBFFM)

```
% input: model parameter realizations (tau,Swir,Sor,phi,Sw),
% output: optimized control
function [u_control] = OptimizationFunction(M)

time=[0 1 3.071611 9.286445 19.28654 30 45 60 75 82.5 90 105 120 135 150
180 210 240 270 300 330 360 390 420 450 480 510 540 570 600 630 660 690
720 750 780 810 840 870 900 930 960 990 1020 1050 1080 1110 1140 1170
1200 1230 1260 1290 1320 1350 1380 1410 1440 1470 1500 1530 1560 1590
1620 1650 1680 1710 1740 1770 1800 1830 1860 1890 1920 1950 1980 2010
2040 2070 2100 2130 2160 2190 2220 2250 2280 2310 2340 2370 2400 2430
2460 2490 2520 2550 2580 2610 2640 2670 2700 2730 2760 2790 2820 2850
2880 2910 2940 2970 3000];
FormationVolume=369024000; %formation volume used to calculate Vp

ro=50; % cost (positive) of oil per RB
rwp=10; % cost (negative) of water production per RB
rwi=20; % cost (negative) of water injection per RB

b=.1; % discount rate

N=size(M,2)+1; % no. of model parameter realizations + 1

Nr=20; % no. of control vector realizations

% parameters that are assumed to be known:
q0=526.93;
m=35.46;
n=1.7535;

timesteps=size(time,2)-1; % no. of time steps

u_UB=4000; % upper boundary control
u_LB=300; % lower boundary control
```

```

Cu=250; % SD used to generate new control vectors

ControlSteps=10; % no. of control steps

a_org=250; % initial step length for updating control
a=a_org;

% setting initial control
u=zeros(ControlSteps,1);
u(:,1) = (u_UB-u_LB)/2+u_LB;

u_r_ensemble=zeros(timesteps,Nr); % new control for each time step
u_ensemble=zeros(timesteps,Nr); % old control for each time step

J_comparison=zeros(1,2); % vector for objective comparison
u_comparison=zeros(ControlSteps,2); % matrix for control comparison

a_counter=0; % no. of times a has been divided by 2
acceptable_a_counter=5; % max. no. of times a can be reduced

loop_finished=0; % criteria for while loop

%the code should be changed to not count old objective multiple times
q_ur=zeros(timesteps+1,N-1,Nr); % new liq. rate
qo_ur=zeros(timesteps+1,N-1,Nr); % new oil rate

q_u=zeros(timesteps+1,N-1,Nr); % old liq. rate
qo_u=zeros(timesteps+1,N-1,Nr); % old oil rate

Sw_ur=zeros(timesteps+1,N-1,Nr); % water saturation
Sw_u=zeros(timesteps+1,N-1,Nr);

% initializing rates and saturation
for rr=1:(N-1)
    q_ur(1,rr,:)=q0;
    q_u(1,rr,:)=q0;
    Sw_ur(1,rr,:)=M(4,rr);
    Sw_u(1,rr,:)=M(4,rr);
end

while loop_finished==0;

    u_matrix = repmat(u,1,Nr); % expanding the old control

```

```

u_r_matrix = normrnd(u_matrix,Cu); % guessing new control
u_r_matrix = max(u_r_matrix,u_LB); % adding boundaries
u_r_matrix = min(u_r_matrix,u_UB); % adding boundaries

% The reason for the following procedure is
% the fact that the control periods does not
% have the same number of time steps. In this
% case the first control period contains 19
% time steps, while the rest contain 10
u_ensemble(1:19,:)=repmat(u_matrix(1,:),19,1);
u_ensemble(20:29,:)=repmat(u_matrix(2,:),10,1);
u_ensemble(30:39,:)=repmat(u_matrix(3,:),10,1);
u_ensemble(40:49,:)=repmat(u_matrix(4,:),10,1);
u_ensemble(50:59,:)=repmat(u_matrix(5,:),10,1);
u_ensemble(60:69,:)=repmat(u_matrix(6,:),10,1);
u_ensemble(70:79,:)=repmat(u_matrix(7,:),10,1);
u_ensemble(80:89,:)=repmat(u_matrix(8,:),10,1);
u_ensemble(90:99,:)=repmat(u_matrix(9,:),10,1);
u_ensemble(100:109,:)=repmat(u_matrix(10,:),10,1);

% new control for each time step
u_r_ensemble(1:19,:)=repmat(u_r_matrix(1,:),19,1);
u_r_ensemble(20:29,:)=repmat(u_r_matrix(2,:),10,1);
u_r_ensemble(30:39,:)=repmat(u_r_matrix(3,:),10,1);
u_r_ensemble(40:49,:)=repmat(u_r_matrix(4,:),10,1);
u_r_ensemble(50:59,:)=repmat(u_r_matrix(5,:),10,1);
u_r_ensemble(60:69,:)=repmat(u_r_matrix(6,:),10,1);
u_r_ensemble(70:79,:)=repmat(u_r_matrix(7,:),10,1);
u_r_ensemble(80:89,:)=repmat(u_r_matrix(8,:),10,1);
u_r_ensemble(90:99,:)=repmat(u_r_matrix(9,:),10,1);
u_r_ensemble(100:109,:)=repmat(u_r_matrix(10,:),10,1);

% matrices for objective
J_ur=zeros(N-1,Nr);
J_u=zeros(N-1,Nr);

% loop for going through each set of model parameters
for ii=1:(N-1)

    % retrieve model parameter
    tau=M(1,ii);
    Swir=M(2,ii);
    Sor=M(3,ii);
    phi=M(5,ii);

    Vp=phi*FormationVolume; %pore volume

```

```

for t=1:timesteps

    tprev=time(t);    % previous time step
    tnow=time(t+1);   % current time step

    for kk=1:Nr % goin throug each control realization

        q_ur(t+1,ii,kk) = q_ur(t,ii,kk)*exp(-(tnow-tprev)/tau)
        + (1-exp(-(tnow-tprev)/tau))*u_r_ensemble(t,kk);
        q_u(t+1,ii,kk) = q_u(t,ii,kk)*exp(-(tnow-tprev)/tau)
        + (1-exp(-(tnow-tprev)/tau))*u_ensemble(t,kk);

        qw_ur = q_ur(t,ii,kk) - qo_ur(t,ii,kk);
        qw_u = q_u(t,ii,kk) - qo_u(t,ii,kk);

        Sw_ur(t+1,ii,kk) = Sw_ur(t,ii,kk) + (u_r_ensemble(t,kk)
        - qw_ur)/Vp * (tnow-tprev);
        Sw_u(t+1,ii,kk) = Sw_u(t,ii,kk) + (u_ensemble(t,kk)
        - qw_u)/Vp * (tnow-tprev);

        S_ur = (Sw_ur(t+1,ii,kk) - Swir)/(1 - Sor - Swir);
        S_u = (Sw_u(t+1,ii,kk) - Swir)/(1 - Sor - Swir);
        S_ur=max(S_ur,0);
        S_u=max(S_u,0);

        qo_ur(t+1,ii,kk) = q_ur(t+1,ii,kk)
        * (1 - (1 + (1-S_ur)^m/(1*S_ur^n))^(-1));
        %assuming endpoint mob ratio=1
        qo_u(t+1,ii,kk) = q_u(t+1,ii,kk)
        * (1 - (1 + (1-S_u)^m/(1*S_u^n))^(-1));
        qo_ur(t+1,ii,kk)=min(qo_ur(t+1,ii,kk),q_ur(t+1,ii,kk));
        qo_u(t+1,ii,kk)=min(qo_u(t+1,ii,kk),q_u(t+1,ii,kk));

        J_ur(ii,kk) = J_ur(ii,kk) + (qo_ur(t+1,ii,kk)*ro
        - (q_ur(t+1,ii,kk)-qo_ur(t+1,ii,kk))*rwp
        - u_r_ensemble(t,kk)*rwi)*(tnow-tprev)/(1+b)^(tnow/365);
        J_u(ii,kk) = J_u(ii,kk) + (qo_u(t+1,ii,kk)*ro
        - (q_u(t+1,ii,kk)-qo_u(t+1,ii,kk))*rwp
        - u_ensemble(t,kk)*rwi)*(tnow-tprev)/(1+b)^(tnow/365);

    end
end
end

% calculating the objective mean for each control realization

```

```

J_ur_final=mean(J_ur,1);
J_u_final=mean(J_u,1);

J_comparison(1,2)=mean(J_u_final);
u_comparison(:,2)=u;

% calculating meanshifted matrices
U_meanshifted=(u_r_matrix-u_matrix).';
j_meanshifted=(J_ur_final-J_u_final).';

% control gradient
G1=(U_meanshifted.'*U_meanshifted);
G2=(U_meanshifted.'*j_meanshifted);
G=G1\G2;
G=G./max(abs(G)); %normalising G

% comparing previous and current objective (when simulating,
% the acceptable boundary should be set higher than 0)
% Since J_comparison is originally defined empty, the first
% mean(J_old_final) must be positive
if (J_comparison(1,2)-J_comparison(1,1)) < 0
    u=u_comparison(:,1);
    a=a/2;
    a_counter=a_counter+1
    if a_counter >= acceptable_a_counter
        loop_finished=1;
        u_control=u;
        return
    end
else
    a_counter=0; %resetting alpha_counter
    a=a_org; %resetting alpha
    J_comparison(1,1)=J_comparison(1,2);
    %setting the new objective as the old objective
    % for next time step
    u_comparison(:,1)=u; %setting the new control
    % vector as the old control vector for next time step
end

%updating u
u=u+a*G;
u=max(u,u_LB);
u=min(u,u_UB);

end
end

```

Example of EnKF in MATLAB (CRMT and BLBFFM)

In this example the EnKF history matches at only one time step

```
% input: M_prior, m_synthetic_truth, u_prior, timestep of the information
% output: M_posterior
function [M_posterior] = EnKFfunction(M_prior,m_synthetic_truth,u_prior
,timestep)

time=[0 1 3.071611 9.286445 19.28654 30 45 60 75 82.5 90 105 120 135 150
180 210 240 270 300 330 360 390 420 450 480 510 540 570 600 630 660 690
720 750 780 810 840 870 900 930 960 990 1020 1050 1080 1110 1140 1170
1200 1230 1260 1290 1320 1350 1380 1410 1440 1470 1500 1530 1560 1590
1620 1650 1680 1710 1740 1770 1800 1830 1860 1890 1920 1950 1980 2010
2040 2070 2100 2130 2160 2190 2220 2250 2280 2310 2340 2370 2400 2430
2460 2490 2520 2550 2580 2610 2640 2670 2700 2730 2760 2790 2820 2850
2880 2910 2940 2970 3000];
FormationVolume=369024000;

length_M_prior=size(M_prior,2);

u=zeros(109,1);

u(1:19)=u_prior(1);
u(20:29)=u_prior(2);
u(30:39)=u_prior(3);
u(40:49)=u_prior(4);
u(50:59)=u_prior(5);
u(60:69)=u_prior(6);
u(70:79)=u_prior(7);
u(80:89)=u_prior(8);
u(90:99)=u_prior(9);
u(100:109)=u_prior(10);

q0=526.93;
m=35.46;
n=1.7535;

q_true=q0;
qo_true=0;
Sw_true=m_synthetic_truth(4);

tau_true=m_synthetic_truth(1);
m_true=m;
n_true=n;
```

```

Swir_true=m_synthetic_truth(2);
Sor_true=m_synthetic_truth(3);
phi_true=m_synthetic_truth(5);

Vp_true=phi_true*FormationVolume;

for t=1:timestep
    tprev=time(t);
    tnow=time(t+1);

    Sw_true = Sw_true + (u(t) - (q_true-qo_true))*(tnow-tprev)/Vp_true;
    q_true = q_true*exp(-(tnow-tprev)/tau_true)
    + (1-exp(-(tnow-tprev)/tau_true))*u(t);
    S_true = (Sw_true-Swir_true)/(1-Sor_true-Swir_true);
    S_true=max(S_true,0);
    S_true=min(S_true,1);
    qo_true = q_true* (1 - (1 + (1-S_true)^m_true/
    (1*S_true^n_true)) ^(-1));
end

SD_q=100;
SD_qo=100;
error=zeros(2,length_M_prior);
error(1,:)=normrnd(0,SD_q,[1,length_M_prior]);
error(2,:)=normrnd(0,SD_qo,[1,length_M_prior]);
d=repmat([q_true;qo_true],1,length_M_prior)+error;
d=max(d,0);

x=zeros(2,length_M_prior);
z=zeros(2,length_M_prior);

for nn=1:length_M_prior

    q=q0;
    qo=0;
    Sw=M_prior(4,nn);
    tau=M_prior(1,nn);
    Swir=M_prior(2,nn);
    Sor=M_prior(3,nn);
    phi=M_prior(5,nn);

    Vp=phi*FormationVolume;

    for t=1:timestep
        tprev=time(t);
        tnow=time(t+1);

```

```

        Sw = Sw + (u(t) - (q-qo))*(tnow-tprev)/Vp;
        q = q*exp(-(tnow-tprev)/tau) + (1-exp(-(tnow-tprev)/tau))*u(t);
        S = (Sw-Swir)/(1-Sor-Swir);
        S = max(S,0);
        S = min(S,1);
        qo = q*(1 - (1 + (1-S)^m/(1*S^n))^(-1));
    end
    x(:,nn)=[Sw;S];
    z(:,nn)=[q;qo];
end

H=zeros(2,9);
H(1,8)=1;
H(2,9)=1;

Y=[M_prior;x;z];

Cd=1/(length_M_prior-1)*(error*error. ');

L = 1/sqrt(length_M_prior-1)*(Y-repmat(mean(Y,2),1,length_M_prior));
HPH = (H*L) * (L.'*H. ');

Y = Y + ((L)*(L.'*H. ')) * ((HPH + Cd)\(d - H*Y));
Y=max(Y,0);
M_posterior=Y(1:5,:);

end

```

Example of VOI structure in MATLAB (CRMT and BLBFFM)

```
time=[0 1 3.071611 9.286445 19.28654 30 45 60 75 82.5 90 105 120 135 150
180 210 240 270 300 330 360 390 420 450 480 510 540 570 600 630 660 690
720 750 780 810 840 870 900 930 960 990 1020 1050 1080 1110 1140 1170
1200 1230 1260 1290 1320 1350 1380 1410 1440 1470 1500 1530 1560 1590
1620 1650 1680 1710 1740 1770 1800 1830 1860 1890 1920 1950 1980 2010
2040 2070 2100 2130 2160 2190 2220 2250 2280 2310 2340 2370 2400 2430
2460 2490 2520 2550 2580 2610 2640 2670 2700 2730 2760 2790 2820 2850
2880 2910 2940 2970 3000];
FormationVolume=369024000;

ro=50;
rwp=10;
rwi=20;

b=.1;

q0=526.93;
m=35.46;
n=1.7535;

timesteps = size(time,2)-1;

N=30;

M_init = ensembleFunction(N);

J_prior=zeros(N,1); %one prior objective for each realisation
J_posterior=zeros(N,timesteps);
J_ValueOfClairvoyance=zeros(N,timesteps);

% for loop, in order to pick a synthetic truth
for ii=1:N

    m_synthetic_truth=M_init(:,ii);

    tau = m_synthetic_truth(1);
    Swir = m_synthetic_truth(2);
    Sor = m_synthetic_truth(3);
    Sw0 = m_synthetic_truth(4);
    phi = m_synthetic_truth(5);
    Vp = phi*FormationVolume;
```

```

M_prior = M_init(:, [1:ii-1 ii+1:N]);

u_prior = PriorOpt1(M_prior);
u_prior_time=zeros(timesteps,1);
u_posterior=repmat(u_prior,1,timesteps);
u_posterior_time=zeros(timesteps,timesteps);
u_VOC=u_posterior;
u_VOC_time=u_posterior_time;

M_posterior=zeros(5, (N-1),timesteps);    % 32 differen model parameters

for t=1:timesteps
    M_posterior(:, :,t) = EnKF1(M_prior,m_synthetic_truth,u_prior,t);

    if t<=19
        u_posterior(:,t) = OptFunc9controlStepsLeft(M_posterior(:, :,t)
            ,u_prior);
        u_VOC(:,t) = OptFunc9controlStepsLeft(m_synthetic_truth
            ,u_prior);
    elseif t>19 && t<=29
        u_posterior(:,t) = OptFunc8controlStepsLeft(M_posterior(:, :,t)
            ,u_prior);
        u_VOC(:,t) = OptFunc8controlStepsLeft(m_synthetic_truth
            ,u_prior);
    elseif t>29 && t<=39
        u_posterior(:,t) = OptFunc7controlStepsLeft(M_posterior(:, :,t)
            ,u_prior);
        u_VOC(:,t) = OptFunc7controlStepsLeft(m_synthetic_truth
            ,u_prior);
    elseif t>39 && t<=49
        u_posterior(:,t) = OptFunc6controlStepsLeft(M_posterior(:, :,t)
            ,u_prior);
        u_VOC(:,t) = OptFunc6controlStepsLeft(m_synthetic_truth
            ,u_prior);
    elseif t>49 && t<=59
        u_posterior(:,t) = OptFunc5controlStepsLeft(M_posterior(:, :,t)
            ,u_prior);
        u_VOC(:,t) = OptFunc5controlStepsLeft(m_synthetic_truth
            ,u_prior);
    elseif t>59 && t<=69
        u_posterior(:,t) = OptFunc4controlStepsLeft(M_posterior(:, :,t)
            ,u_prior);
        u_VOC(:,t) = OptFunc4controlStepsLeft(m_synthetic_truth
            ,u_prior);
    elseif t>69 && t<=79
        u_posterior(:,t) = OptFunc3controlStepsLeft(M_posterior(:, :,t)

```

```

        ,u_prior);
        u_VOC(:,t) = OptFunc3controlStepsLeft(m_synthetic_truth
        ,u_prior);
    elseif t>79 && t<=89
        u_posterior(:,t) = OptFunc2controlStepsLeft(M_posterior(:, :,t)
        ,u_prior);
        u_VOC(:,t) = OptFunc2controlStepsLeft(m_synthetic_truth
        ,u_prior);
    elseif t>89 && t<=99
        u_posterior(:,t) = OptFunc1controlStepsLeft(M_posterior(:, :,t)
        ,u_prior);
        u_VOC(:,t) = OptFunc1controlStepsLeft(m_synthetic_truth
        ,u_prior);
    end
end

u_prior_time(1:19)=u_prior(1);
u_prior_time(20:29)=u_prior(2);
u_prior_time(30:39)=u_prior(3);
u_prior_time(40:49)=u_prior(4);
u_prior_time(50:59)=u_prior(5);
u_prior_time(60:69)=u_prior(6);
u_prior_time(70:79)=u_prior(7);
u_prior_time(80:89)=u_prior(8);
u_prior_time(90:99)=u_prior(9);
u_prior_time(100:109)=u_prior(10);
u_posterior_time(1:19, :)=repmat(u_posterior(1, :),19,1);
u_posterior_time(20:29, :)=repmat(u_posterior(2, :),10,1);
u_posterior_time(30:39, :)=repmat(u_posterior(3, :),10,1);
u_posterior_time(40:49, :)=repmat(u_posterior(4, :),10,1);
u_posterior_time(50:59, :)=repmat(u_posterior(5, :),10,1);
u_posterior_time(60:69, :)=repmat(u_posterior(6, :),10,1);
u_posterior_time(70:79, :)=repmat(u_posterior(7, :),10,1);
u_posterior_time(80:89, :)=repmat(u_posterior(8, :),10,1);
u_posterior_time(90:99, :)=repmat(u_posterior(9, :),10,1);
u_posterior_time(100:109, :)=repmat(u_posterior(10, :),10,1);
u_VOC_time(1:19, :)=repmat(u_VOC(1, :),19,1);
u_VOC_time(20:29, :)=repmat(u_VOC(2, :),10,1);
u_VOC_time(30:39, :)=repmat(u_VOC(3, :),10,1);
u_VOC_time(40:49, :)=repmat(u_VOC(4, :),10,1);
u_VOC_time(50:59, :)=repmat(u_VOC(5, :),10,1);
u_VOC_time(60:69, :)=repmat(u_VOC(6, :),10,1);
u_VOC_time(70:79, :)=repmat(u_VOC(7, :),10,1);
u_VOC_time(80:89, :)=repmat(u_VOC(8, :),10,1);
u_VOC_time(90:99, :)=repmat(u_VOC(9, :),10,1);
u_VOC_time(100:109, :)=repmat(u_VOC(10, :),10,1);

```

```

q_prior=q0;
qo_prior=0;
NPV_prior=0;
Sw_prior=Sw0;
q_posterior=q0*ones(1,timesteps);
qo_posterior=zeros(1,timesteps);
NPV_posterior=zeros(1,timesteps);
Sw_posterior=Sw0*ones(1,timesteps);
q_VOC=q0*ones(1,timesteps);
qo_VOC=zeros(1,timesteps);
NPV_VOC=zeros(1,timesteps);
Sw_VOC=Sw0*ones(1,timesteps);

for tt=1:timesteps
    tprev=time(tt);
    tnow=time(tt+1);

    Sw_prior = Sw_prior + (u_prior_time(tt) - (q_prior-qo_prior))
    * (tnow-tprev)/Vp;
    q_prior = q_prior*exp(-(tnow-tprev)/tau)
    + (1-exp(-(tnow-tprev)/tau))*u_prior_time(tt);
    S_prior = (Sw_prior-Swir)/(1-Sor-Swir);
    S_prior = max(S_prior,0);
    qo_prior = q_prior* (1 - (1 + (1-S_prior)^m/(1*S_prior^n)) ^(-1));

    NPV_prior = NPV_prior + ( qo_prior*ro - (q_prior-qo_prior)*rwp
    - u_prior_time(tt)*rwi ) * (tnow-tprev) / (1+b)^(tnow/365);

    for nopos=1:timesteps
        Sw_posterior(1,nopos) = Sw_posterior(1,nopos)
        + (u_posterior_time(tt,nopos) - (q_posterior(1,nopos)
        -qo_posterior(1,nopos)))*(tnow-tprev)/Vp;
        q_posterior(1,nopos) = q_posterior(1,nopos)*exp(-(tnow-tprev)/tau)
        + (1-exp(-(tnow-tprev)/tau))*u_posterior_time(tt,nopos);
        S_posterior = (Sw_posterior(1,nopos)-Swir)/(1-Sor-Swir);
        S_posterior = max(S_posterior,0);
        qo_posterior(1,nopos) = q_posterior(1,nopos)* (1
        - (1 + (1-S_posterior)^m/(1*S_posterior^n)) ^(-1));

        NPV_posterior(1,nopos) = NPV_posterior(1,nopos)
        + ( qo_posterior(1,nopos)*ro - (q_posterior(1,nopos)
        -qo_posterior(1,nopos))*rwp - u_posterior_time(tt,nopos)*rwi )
        * (tnow-tprev) / (1+b)^(tnow/365);

        Sw_VOC(1,nopos) = Sw_VOC(1,nopos) + (u_VOC_time(tt,nopos)

```

```

- (q_VOC(1,nopos)-qo_VOC(1,nopos)))*(tnow-tprev)/Vp;
q_VOC(1,nopos) = q_VOC(1,nopos)*exp(-(tnow-tprev)/tau)
+ (1-exp(-(tnow-tprev)/tau))*u_VOC_time(tt,nopos);
S_VOC = (Sw_VOC(1,nopos)-Swir)/(1-Sor-Swir);
S_VOC = max(S_VOC,0);
qo_VOC(1,nopos) = q_VOC(1,nopos)* (1
- (1 + (1-S_VOC)^m/(1*S_VOC^n)) ^(-1));

NPV_VOC(1,nopos) = NPV_VOC(1,nopos) + ( qo_VOC(1,nopos)*ro
- (q_VOC(1,nopos)-qo_VOC(1,nopos))*rwp
- u_VOC_time(tt,nopos)*rwi ) * (tnow-tprev) / (1+b)^(tnow/365);
end
end
J_prior(ii,1) = NPV_prior;
for nopos=1:timesteps
    J_posterior(ii,nopos) = NPV_posterior(1,nopos);
    J_ValueOfClairvoyance(ii,nopos) = NPV_VOC(1,nopos);
end
end
VOI=zeros(1,timesteps);
VOC=zeros(1,timesteps);
for yy=1:timesteps
    for jj=1:N
        VOI(1,yy) = VOI(1,yy) + (J_posterior(jj,yy)-J_prior(jj,1))/N;
        VOC(1,yy) = VOC(1,yy) + (J_ValueOfClairvoyance(jj,yy)
-J_prior(jj,1))/N;
    end
end
end

```
