T H E U N I V E R S I T Y O F T U L S A

THE GRADUATE SCHOOL

ENSEMBLE-BASED OPTIMIZATION FOR HISTORY MATCHING, SURVEILLANCE OPTIMIZATION AND UNCERTAINTY QUANTIFICATION

by
Duc Huu Le

A thesis submitted in partial fulfillment of
the requirements for the degree of Doctor of Philosophy
in the Discipline of Petroleum Engineering

The Graduate School
The University of Tulsa

2015

T H E U N I V E R S I T Y O F T U L S A

THE GRADUATE SCHOOL

ENSEMBLE-BASED OPTIMIZATION FOR HISTORY MATCHING, SURVEILLANCE OPTIMIZATION AND UNCERTAINTY QUANTIFICATION

by
Duc Huu Le

A THESIS

APPROVED FOR THE DISCIPLINE OF

PETROLEUM ENGINEERING

By Thesis Committee

_____, Chair
Albert C. Reynolds

_____
Rami Younis

_____
Richard Redner

_____
Fahim Forouzanfar

ABSTRACT

Duc Huu Le (Doctor of Philosophy in Petroleum Engineering)

Ensemble-based Optimization for History Matching, Surveillance Optimization and Uncertainty Quantification

Directed by Albert C. Reynolds

215 pp., Chapter 5: Conclusions

(586 words)

In this work, we develop ensemble-based methods to improve two related processes in the closed-loop reservoir management framework, history matching and surveillance optimization. On the history matching side, Emerick and Reynolds recently introduced the ensemble smoother with multiple data assimilations (ES-MDA) method. Via computational examples, they demonstrated that ES-MDA provides both a better data match and a better quantification of uncertainty than is obtained with the ensemble Kalman filter (EnKF). However, like EnKF, ES-MDA can experience near ensemble collapse and can generate too many extreme values of rock property fields for complex problems. These negative effects can be avoided by a judicious choice of the ES-MDA inflation factors but, prior to this work, the optimal inflation factors could only be determined by trial and error. Here, we provide two automatic procedures for choosing the inflation factor for the next data assimilation step adaptively as the history match proceeds. We demonstrate that the adaptive ES-MDA algorithms can be superior to the original ES-MDA algorithm in an extreme, difficult synthetic problem. In more reasonable problems, the adaptive algorithm may still perform better but performance gap is much smaller.

We propose a procedure based on ES-MDA to history match data from non-Gaussian reservoir models, with a focus on those generated using multi-point statistics (MPS). ES-

MDA is applied to update the permeability field in the normal way but during ES-MDA process, we periodically apply the Expectation-Maximization algorithm to classify the updated permeability fields into channel and non-channel regions. Then we take the average of the new facies distributions over the whole ensemble to obtain a facies probability map which is used in the MPS algorithm as the soft data to generate new facies realizations. Obtaining a reasonable approximation of the correct facies distribution is only half of the problem; we also wish to obtain a plausible distribution of the permeability within each facies. This is done by performing a second data assimilation stage where we nearly fix the facies distribution and only adjust the permeability within each facies. For the examples considered in this research, the procedure is able to provide good data matches as well as posterior facies and permeability fields that reflect the main geological features of the true model.

On the surveillance optimization side, we aim to find an efficient method that can determine, among a suite of potential surveillance operations, the most beneficial operation and whether its benefit justifies the cost of collecting the data. The usefulness or the value of information of the data is defined here as the uncertainty reduction in the reservoir variable of interest J (e.g. cumulative oil production or net present value) once the reservoir model is updated by assimilating the data. An exhaustive history matching procedure exists to provide the answer to this problem but the required computational costs make it unfeasible for anything other than simple synthetic reservoir models. We propose an alternate procedure based on information theory where the mutual information between J and the random observed data vector D_{obs} is estimated using an ensemble of prior reservoir models. This mutual information reflects the strength of the relationship between J and the potential observed data and provides a way to qualitatively rank potential surveillance operations in terms of their usefulness. The expected uncertainty reduction in J is estimated by calculating the conditional entropy of J and translating the obtained value to the expected P90 - P10 of J. The proposed method is applied to four different problems and the results are verified using the exhaustive history matching method.

ACKNOWLEDGEMENTS

I would like to express my sincere appreciation to my advisor, Dr. Albert Reynolds, for his guidance, assistance and encouragement during my Ph.D. study at the University of Tulsa. I specially thank Dr. Rami Younis, Dr. Richard Redner and Dr. Fahim Forouzanfar for serving as members of my dissertation committee. I would like to extend my thanks to all the other faculty members of the McDougall School of Petroleum Engineering for their guidance through my courses of study as a graduate student.

I gratefully acknowledge financial support from the member companies of the University of Tulsa Petroleum Reservoir Exploitation Projects (TUPREP). I also thank my parents, my sister and my friends for their support during my study.

This work is dedicated to my wife, Anh, for her unconditional love and support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Closed-loop reservoir management (Jansen et al., 2009) is a general reservoir management framework that consists of multiple inter-related tasks. The tasks are performed sequentially during the life of a reservoir with the aim of maximizing the reservoir's value. Fig. 1.1 presents the schematic diagram of the closed-loop reservoir management framework.

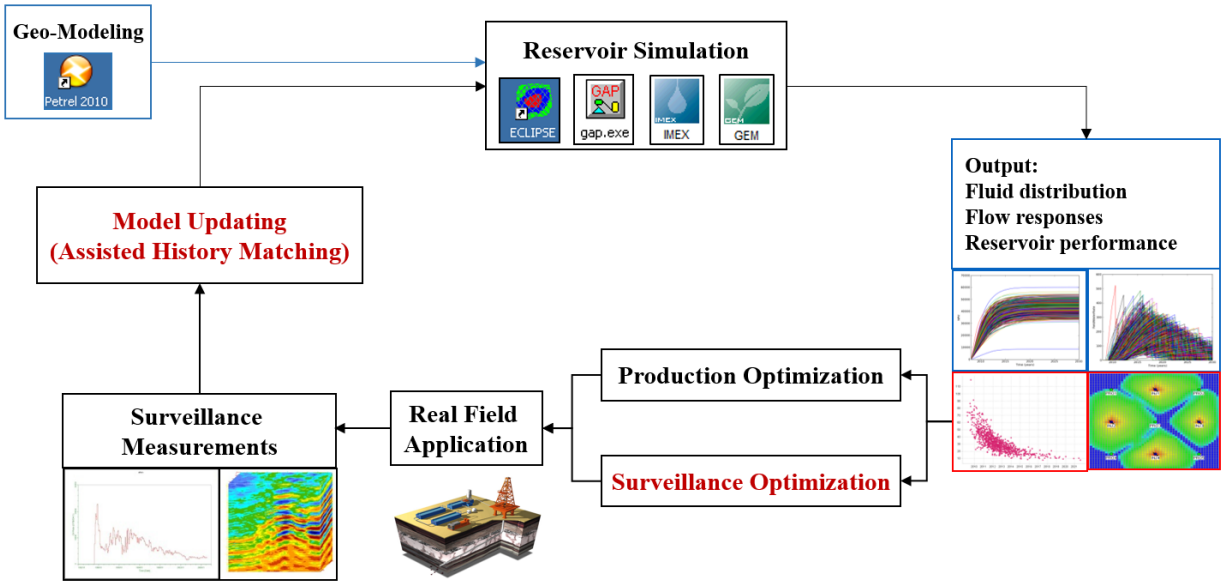


Figure 1.1: Schematic diagram of closed-loop reservoir management.

As shown in Fig. 1.1, the loop starts with with the geo-modeling process, where relevant data, e.g., geology, seismics, well logs and fluid properties, are gathered and combined to build a reservoir model that represents the current knowledge about the reservoir. This model is then run through a reservoir simulator to estimate the future performance of the reservoir such as flow rates, and the pressure and fluid distributions. In the next step, either production optimization or surveillance planning or both can be performed. Production optimization refers to the task of optimizing the future production of the reservoir by changing

the well control parameters such as bottom hole pressures, production rates and valve opening. This is done based on the current best model, so if the current model is significantly different from the true model, the optimized strategy based on the current model may not be optimal. Surveillance planning refers to the task of determining if there is a need to measure additional data to get more information about the reservoir. This is often determined on a benefit vs. cost basis, where the yet-to-be-measured data are evaluated based on their expected value in decision making (value of information) and then compared to the cost of measurement to see if the measurement of the data justify their cost. If they do, then the optimal surveillance operation will be carried out in the next step. After field applications of the optimal control strategy and/or the chosen surveillance operation are performed, new response data (measurements) are obtained from the reservoir. The new measurements are then integrated into the reservoir model by a process called history matching or data assimilation, whereby the model parameters are modified so that the simulation matches the observed data. The new model is then used to predict future reservoir performance and the loop repeats.

This research deals with two related processes in the closed-loop reservoir management, history matching and surveillance optimization, both using ensemble-based methods.

1.1 Literature Review on History Matching

The first focus of this research is on the history matching step, where the observed data are assimilated into the reservoir model to obtain a posterior model that gives an acceptable match of the historical data. The importance of this step is that the new model is expected to have more accurate predictive capability as well as reduce the uncertainty in the reservoir model parameters, which is vital to devise the most beneficial operating strategy for the field. History matching is an inverse problem where the number of observed data is generally much smaller than the number of variables. Hence, an infinite number of combinations of the model parameters are able to give an acceptable match of the observed data. The observed data is corrupted by measurement error which also introduces uncertainty into the history-matched

reservoir model and its prediction of future reservoir performance. Over the last decade, the main focus of history matching has shifted from finding a model that best matches the observed data to a set of models that are consistent with the observed data, which allows the quantification of uncertainty in the posterior model and performance prediction in order to manage risk. This is usually done based on the theory of Bayesian statistics, which provides a framework to represent the uncertainty in the model parameters conditioned to inaccurate observed data. According to Bayes's theorem, the conditional probability distribution (pdf) of the model parameters,$f(m|d_{obs})$, can be written as,

$$f(m|d_{obs}) = \frac{f(d_{obs}|m)f(m)}{f(d_{obs})} = aL(m|d_{obs})f(m), \tag{1.1}$$

where m is the N_m-dimensional vector of model parameters, d_{obs} is the N_d-dimensional vector of observed data, $f(m)$ is the prior probability density function (pdf) of the model parameters, $f(d_{obs})$ is the pdf of the observed data, $f(d_{obs}|m)$ is the conditional pdf of d_{obs} given m, which corresponds to the likelihood function, $L(m|d_{obs})$, and a is a normalizing constant. Assuming the prior pdf is Gaussian with $m \sim \mathcal{N}(m_{pr}, C_M)$ and the vector of measurement errors, ϵ, is Gaussian with $\epsilon \sim \mathcal{N}(0, C_D)$, we can write $f(m|d_{obs})$ as

$$\begin{aligned} f(m|d_{obs}) =& a \exp\left\{-\frac{1}{2}(m-m_{pr})^T C_M^{-1}(m-m_{pr})\right\} \times \\ & \left\{-\frac{1}{2}(g(m)-d_{obs})^T C_D^{-1}(g(m)-d_{obs})\right\} \\ =& a \exp\left\{-\frac{1}{2}(m-m_{pr})^T C_M^{-1}(m-m_{pr}) - \frac{1}{2}(g(m)-d_{obs})^T C_D^{-1}(g(m)-d_{obs})\right\} \\ =& a \exp\{-O(m)\}, \end{aligned} \tag{1.2}$$

where C_M is the $N_m \times N_m$ prior covariance matrix of the model parameters; C_D is the $N_d \times N_d$ covariance matrix of measurement errors, $g(m)$ is the vector of predicted data for a given vector m, generated from a forward model, e.g., a reservoir simulator, and

$$O(m) = \frac{1}{2}(m-m_{pr})^T C_M^{-1}(m-m_{pr}) + \frac{1}{2}(g(m)-d_{obs})^T C_D^{-1}(g(m)-d_{obs}) \tag{1.3}$$

is the objective function. We can write

$$O(m) = O_m(m) + O_d(m), \tag{1.4}$$

where

$$O_m(m) = \frac{1}{2}(m - m_{pr})^T C_M^{-1}(m - m_{pr}), \tag{1.5}$$

and

$$O_d(m) = \frac{1}{2}(g(m) - d_{obs})^T C_D^{-1}(g(m) - d_{obs}). \tag{1.6}$$

In history matching, we typically want to minimize this objective function because minimizing $O(m)$ is equivalent to maximizing the posterior pdf $f(m|d_{obs})$, which would result in the "most probable model." Note that the objective function consists of the model mismatch part, $O_m(m)$, and the data mismatch part, $O_d(m)$. If the posterior model gives a very high value of $O_m(m)$, it indicates a rough model, i.e., the model may contain unrealistically high or low values of the gridblock properties. If the posterior model has a high value of $O_d(m)$, this indicates a poor data match.

When the relationship between data and model parameters, i.e. $g(m)$, is linear, the posterior pdf $f(m|d_{obs})$ is also Gaussian with mean

$$m_{MAP} = m_{pr} - (C_M^{-1} + G^T C_D^{-1} G)^{-1} G^T C_D^{-1}(G m_{pr} - d_{obs}), \tag{1.7}$$

and covariance

$$C_{MAP} = C_M - C_M G^T (C_D + G^T C_M^{-1} G)^{-1} G C_M; \tag{1.8}$$

see Tarantola (2005).

When the function $g(m)$ is nonlinear, the mean and covariance given by Eqs. 1.7 and 1.8 cannot accurately describe the posterior pdf, especially for ones with multiple nodes. To solve this problem, Kitanidis (1995) and Oliver et al. (1996) independently introduced the Randomized Maximize Likelihood (RML) scheme to sample from the posterior model. In

the application of RML, we replace m by a sample, $m_{uc,j}$, from $N(m_{pr}, C_M)$ and replace d_{obs} by a sample, $d_{uc,j}$, from $N(d_{obs}, C_D)$ in the objective function of Eq. 1.3 to obtain

$$O_j(m) = \frac{1}{2}(d_{uc,j} - g(m))^T C_D^{-1}(d_{uc,j} - g(m)) + \frac{1}{2}(m - m_{uc,j})^T C_M^{-1}(m - m_{uc,j}), \qquad (1.9)$$

and then minimize the modified objective function, $O_j(m)$, to obtain a sample of the posterior pdf. By repeating the process with a set of different samples of m_{uc} and d_{uc}, we generate an ensemble of approximate samples of the posterior pdf. RML provides a theoretically rigorous sampling of the posterior pdf only in the linear Gaussian case (Oliver et al., 2008) but often appears to provide a reasonable sampling of the posterior pdf obtained by integrating production data into a prior geostatistical model (Emerick and Reynolds, 2013c). A disadvantage of the RML method is that a computationally efficient implementation requires the adjoint gradient, which is not supported in most commercial reservoir simulators, so that an efficient algorithm like the Gauss-Newton method can be utilized. Even then, the computational cost associated with RML may be high if the minimization problem requires many iterations to reach convergence. These limitations of RML give rise to a different class of data assimilation methods where the adjoint gradient is not required and the model uncertainty is represented by an ensemble of realizations. These methods are collectively known as ensemble-based history matching methods. In the following subsections, we review three of the most popular ensemble-based methods, namely the ensemble Kalman filter, the ensemble smoother and the ensemble smoother with multiple data assimilation.

1.1.1 Ensemble Kalman filter

The Ensemble Kalman filter (EnKF) was first introduced by (Evensen, 1994). Since its introduction, the method has been applied in a large number of fields, including hydrology (Chen and Zhang, 2006; Liu et al., 2008; Reichle et al., 2002), oceanography (Bertino et al., 2003; Keppenne and Rienecker, 2003), weather prediction (Houtekamer and Mitchell, 2005; Szunyogh et al., 2005) and petroleum reservoir history matching (Aanonsen et al., 2009; Oliver and Chen, 2010). An extensive list of applications of EnKF is presented in Evensen

(2007). EnKF was first applied in the oil industry by Lorentzen et al. (2001) to improve predictions of pressure behavior in a two-phase fluid flow model, and applied to a reservoir problem by Nævdal et al. (2002) to update permeability fields in near-well reservoir models. Following these pioneer works, the interest and use of EnKF as a history matching technique increased significantly (Nævdal et al., 2005; Gu and Oliver, 2005; Liu and Oliver, 2005; Gao et al., 2006; Dong et al., 2006; Devegowda et al., 2007; Skjervheim et al., 2007; Zafari and Reynolds, 2007; Thulin et al., 2007; Dovera and Della Rossa, 2007; Lødøen and Omre, 2008; Agbalaka and Oliver, 2008; Sakov and Oke, 2008; Li and Reynolds, 2009; Oliver and Chen, 2009; Chen and Oliver, 2009; Wang et al., 2010; Emerick and Reynolds, 2012a,b). Most importantly, it has been successfully used in field applications (Skjervheim et al., 2007; Bianco et al., 2007; Evensen et al., 2007; Haugen et al., 2008; Emerick and Reynolds, 2011b). EnKF is a sequential parameter-state estimation problem. At each time step where observed data are available, the state or primary reservoir variables (e.g., gridblock pressures, fluid saturations) are updated along with the model parameters (e.g., permeability, porosity). The reason for updating the primary reservoir variables is to avoid rerunning the simulations from time zero after every data assimilation step. This is done based on the assumptions that the updated primary variables are statistically consistent with the ones that would be obtained if one rerun the simulations with the updated updated model parameters from time zero. However, this consistency has only been proven in the linear-Gaussian case, where the prior model parameters follows a Gaussian distribution and the relationship between model and data is linear. In real petroleum applications, the relationship between model and data is governed by complex partial differential equations, which are usually highly nonlinear (Thulin et al., 2007). The inconsistency between the updated primary variables and the updated model parameters sometimes leads to poor performance of the EnKF method; see, for example, Wang et al. (2010).

Despite the strong assumption about the consistency between the updated primary variables and model parameters, EnKF is used in petroleum reservoir history matching due to the fact that it does not require the availability of the adjoint gradient, which is generally

unavailable in commercial reservoir simulators. This makes EnKF easy to adapt to different types of reservoir model parameters, different types of data and different commercial reservoir simulators. Another advantage is that EnKF does not require the explicit formation of the prior covariance matrix or the cross-covariance matrix between the model parameter and the predicted data, making EnKF efficient in terms of memory requirement.

1.1.2 Ensemble smoother

The ensemble smoother (ES) was proposed by van Leeuwen and Evensen (1996). Its formulation is similar to that of EnKF but it simulates all available data simultaneously in one single data assimilation step instead of sequentially in time. Because of this, ES is a parameter estimation method, which means it does not update the primary reservoir variables during the data assimilation step. As a result, ES is faster and easier to implement than EnKF and does not suffer from the problem of inconsistency between the updated primary variables and model parameters. However, when applied to petroleum reservoir history matching problem, ES often leads to a poorer data match than does EnKF. This behavior is explained by Reynolds et al. (2006), where it is shown that ES and EnKF are similar to applying, at each data assimilation step, one Gauss-Newton iteration with the average sensitivity matrix estimated from ensemble. EnKF is therefore similar to several small correction steps while ES is similar to a big correction step. Intuitively, for a nonlinear problem, one single Gauss-Newton step is not adequate to obtain a good data much, which explains the poor performance of ES.

1.1.3 Ensemble smoother with multiple data assimilation

The ensemble smoother with multiple data assimilation (ES-MDA) was introduced by Emerick and Reynolds (2012b, 2013a) to improve the ensemble Kalman filter (EnKF) (Evensen, 1994) for problems where the relation between the model (vector of reservoir model parameters) and data predicted from the forward model (i.e., the reservoir simulator) is highly nonlinear. As discussed above, the EnKF has two major shortcomings: (1) it can introduce inconsistencies between updated reservoir model parameters and updated states

(primary variables solved for by the simulator) because EnKF updates both model parameter and states (Thulin et al., 2007; Wang et al., 2010) and (2) it can produce an unacceptable data match. The ensemble smoother (ES) attempts to fix EnKF's first issue but exacerbates the second one, i.e., ES can provide an even worse data match than is obtained with EnKF (Emerick and Reynolds, 2013c).

ES-MDA was motivated by the analogy between ES and one Gauss-Newton iteration (Reynolds et al., 2006). Simply put, ES-MDA assimilates the same data set N_a times ($N_a \geq 1$) using the same form of the updating (analysis) formula used for the ensemble smoother. However, at the ℓth data assimilation step, the measurement error covariance matrix, C_D, is replaced by $\alpha_\ell C_D$ where $\alpha_\ell > 1$. The α_ℓ's are chosen to harmonically sum to one to ensure that, for the linear Gaussian case, ES-MDA produces a theoretically rigorous sampling of the posterior probability density function as the ensemble size goes to infinity. Iterative ensemble-based methods have been used previously to obtain a better data match than is obtained with EnKF or ES, but at least for the simple reservoir example considered in Emerick and Reynolds (2013c), the performance of ES-MDA was far superior to the other iterative ensemble-based methods tested, not only in the data match obtained, but more importantly, in terms of uncertainty quantification. Specifically, ES-MDA produced the best approximation of the posterior marginal probability density functions (pdf's) for the reservoir model parameters and the best approximation for the posterior probability distribution for predicted future rates where the correct posterior pdf's were generated using Markov chain Monte Carlo (MCMC), which theoretically produces a correct sampling of the posterior pdf for the vector of model parameters.

Even though ES-MDA works well for the problems considered (Emerick and Reynolds, 2012b, 2013a,c), as well as for field cases (Emerick and Reynolds, 2013b), there still exist two drawbacks in using ES-MDA. The first drawback is that the users have to determine the number of data assimilations a priori. In general, four to ten iterations appear to be sufficient, but there has been no clear guidance on how to choose the optimal number of iterations (data assimilations). The problem with this approach is that, if the data

assimilation does not lead to satisfactory results, the whole process has to be repeated with a different number of iterations. The second drawback is that it is not clear how to choose the optimal inflation factors. Equal inflation factors were used for most cases (Emerick and Reynolds, 2012b, 2013a) because this choice worked well. The two drawbacks of ES-MDA may not be apparent when dealing with relatively easy history-matching problems, including field cases, where the choice of four equal inflation factors often works well. For certain "difficult" problems, however, a simple implementation of ES-MDA may result in severe overshooting, or roughness, in the final permeability and porosity fields. This roughness arises from the ill conditioning of the matrix that must be inverted at each ES-MDA iteration and the relatively large step size used at early iterations which leads to a large change (over correction) in the rock property fields at early iterations. This over correction occurs because ES is similar to one Gauss-Newton iteration with a full step (unity stepsize) (Reynolds et al., 2006) whereas a smaller step size should be used at early iterations.

1.2 Literature Review on History Matching Non-Gaussian Facies

Traditionally, the reservoir model is assumed to contain only continuous variables such as permeability and porosity. In recent years, there has been a growing interest in history-matching problems where the model contains categorical facies variables. Unlike traditional reservoir models where petrophysical properties tend to be spatially smooth, multi-facies model are characterized by an abrupt change in petrophysical properties at facies boundaries. Multi-facies models are necessary to model channelized or fluvial reservoirs.

There are two main ways in which a multi-facies reservoir model can be generated: (1) by using object modeling and (2) by using multi-point statistics (MPS). In object modeling, the user defines a background facies (usually shale) and the size, shape and chance of appearance of the foreground geological bodies to include in the model. The algorithm would then stochastically generate and populate the reservoir grid with these objects until the user-specified percentages of all bodies have been reached. The advantage of this method is that the quality of the final model is often high, i.e. the objects appear exactly as defined

by the user. The disadvantage is that it is difficult to honor hard data if the number of hard data is large. In MPS, on the other hand, the user first needs to prepare a training image that describes the reservoir's geology. The training image can either be drawn by hand or be generated using object modeling. A pixel-based algorithm such as *snesim* (Strebelle, 2002) is then used to calculate the conditional distributions, or patterns, that describe the probability of different facies occurring at different positions relative to the position of a gridblock with known facies. The algorithm will then populate the reservoir model gridblock by gridblock using the pre-calculated conditional distributions until the facies that occupies each gridblock has been defined. The advantage of this approach is that, because the algorithm is pixel-based, it can be conditioned to hard data more easily than is the case for object modeling. The disadvantage is that the quality of the final model may not be as good as that obtained by object modeling due to the occasional appearance of stray pixels or stray groups of pixels that do not seem to belong to any particular geological body. There exists a third, albeit not so natural, way to generate multi-facies models. First, two Gaussian fields of the same size as the reservoir model, Y_1 and Y_2, are generated. Then, a two-dimensional truncation map, the axes of which represent the values of Y_1 and Y_2, is created. The map is divided into multiple regions, each of which corresponds to a particular facies. For each gridblock, using the values of Y_1 and Y_2 at that particular gridblock and the truncation map, the facies value can be determined. Because of the use of multiple Gaussian fields, this method is named pluri-Gaussian (Galli et al., 1994; Le Loc'h et al., 1994; Le Loc'h and Galli, 1997). The problem with this method is that there is no systematic way to determine the truncation map needed to reproduce a particular geological feature (Liu and Oliver, 2005). In fact, many geological features such as channel, which can be readily generated using the other two methods, cannot be easily obtained with this method (Zhao et al., 2008). In the pluri-Gaussian approach, the underling parameters are still Gaussian, so history matching models created with a pluri-Gaussian model is not much different from history matching regular Gaussian models.

History matching a reservoir with multiple facies has always posed a great challenge

to researchers. Most traditional history-matching techniques were designed to work with continuous variables instead of categorical variables like facies. Even though they work reasonably well for problems where the relationship between data and the model is highly nonlinear, rarely can they be directly applied to problems that involve more than one facies. These problems are usually characterized by the abrupt change of petrophysical properties across facies boundary. The direct application of traditional methods to multi-facies problems usually results in posterior models that have no clear boundaries between facies so that the posterior models do not honor the geological description. In an attempt to preserve facies boundaries while using traditional methods, many researchers use reparameterization to transform the categorical facies variable into a continuous variable. After history matching, the variables in the transformed space are transformed back to the original space. Among many transformation algorithms available, the principle component analysis (PCA) is probably the most commonly used. Sarma et al. (2008) use a nonlinear extension of PCA, namely kernel PCA, to account for the non-Gaussian relationship between gridblock facies. Chen et al. (2015) use the regular PCA transformation but propose a method to determine the optimal truncation threshold used in back-transforming the variables from the continuous space to the discrete space. Vo and Durlofsky (2014) propose the Optimization-PCA algorithm where the back-transformation process is framed as an optimization problem with a penalty term to drive the continuous variable in the transformed space to either 0 or 1 when transforming back. Other transformation algorithms are also used. Jafarpour and McLaughlin (2009) apply the discrete cosine transformation, and then in a later paper (Li and Jafarpour, 2010), combine the discrete cosine transformation with a Bayesian sparse reconstruction algorithm. Hu et al. (2013) use the uniform score transformation. Sebacher et al. (2015) use a variation of the Gaussian score transformation. While these methods seem to work well for the problems presented, the transformation and back-transformation algorithms can be complex, and the determination of the optimal transformation parameters can be difficult. Note that, for these algorithms, the geological simulation process is only used to generate prior realizations and never used again during the history-matching process.

Another approach is to make use of the underlying geological simulation process during history matching to generate new, better-matched models, which ensure consistency between the current model and the geological description. For models generated using object modeling, Bi et al. (2000) and Zhang et al. (2002) explore the idea of modifying the size and position of the sand channel in a single-channel reservoir along with other petrophysical properties such as permeability, porosity to match the observed data. It is, however, not clear how to extend this idea to a model with multiple channels, where the channels are stochastically positioned. For models generated using MPS, there exists a mechanism to incorporate soft data (probabilities) into the simulation process in order to modify the chance of the appearance of facies in the model (Journel, 2002; Zhang and Journel, 2003). Caers (2003) uses this mechanism to draw new realizations from the training image with the soft data represented by different perturbations of the current model and then evaluates the new realizations to find one that results in smaller data mismatch than the current model. Jafapour and Khodabakhsi (2011) combine this mechanism with the EnKF algorithm, where the EnKF is used to modify the permeability field as in a regular history-matching problem. After each EnKF iteration, the facies distributions in the updated models are determined by a mapping between the permeability variable and the facies variable. A facies probability map is then calculated by averaging all the models in the ensemble. This is used as soft data in the MPS algorithm to generate new reservoir models that match the data better while being consistent with the geological description.

One thing that most papers in the literature do not consider is the variability of the permeability within each facies; they simply try to get the facies distribution correct while assuming each facies has a constant, known permeability value. Among the paper mentioned above, only Chen et al. (2015) and Vo and Durlofsky (2014) attempt to deal with the permeability variation within facies. Even then, both papers assume the correct permeability distribution within each facies is known a priori. In Chen et al. (2015), after the gridblock facies have been determined, the permeability distribution within each facies is mapped back to the prior distribution which is presumably known. In Vo and Durlofsky (2014), the

penalty term is designed so that the back-transformed variable has the tendency to get close to the log permeability mean of each facies. This requires an additional optimization step to determine the optimal parameters for the penalty term.

1.3 Literature Review on Surveillance Optimization

A different but related focus of this research is on the surveillance optimization step. The importance of this step surveillance optimization is that, due to the high cost associated with performing any measurement in the reservoir, the surveillance operations have to be chosen carefully to maximize the potential benefits as well as avoid spending money on redundant operations. In other words, we need to evaluate the potential surveillance operations to determine their potential usefulness. The usefulness of the measurement data can be related to the strength of the relationship between the data to be collected and the reservoir variable of interest J; here, J represents either the net present value or the cumulative oil production of the reservoir. The usefulness or the value of information of the data is defined here as the uncertainty reduction in J once the reservoir model is updated by assimilating the data. In the oil industry, it is customary for the uncertainty of a quantity to be reported as the difference between the P90 case and the P10 case (see, for example, Haskett (2003)). Therefore, the change in the P90-P10 range of J after data assimilation represents the uncertainty reduction provided by the surveillance data. The two questions that we try to answer in surveillance optimization are:

1. What are the best data to collect, i.e., which surveillance operation among those that are viable, will yield the greatest reduction in uncertainty?

2. What is the expected value of the uncertainty reduction? The answer to this question will determine if the value of information of data from the surveillance operation is good enough to justify the cost of collecting the data.

The first question is a qualitative one and there exists methods based on a linear-Gaussian assumption that can provide reasonable answers in certain cases. If the inverse

problem, typically "solved" by history matching, is formulated in the framework of Bayesian statistics, with the prior model Gaussian and there is a linear relation between predicted data and the vector of model parameters, m, (linear-Gaussian assumption) then the probability density function (pdf) conditional to data is Gaussian with covariance matrix denoted by C_{MAP}; see, for example, Tarantola (2005). Moreover, the posterior pdf of model parameters in this linear-Gaussian case is itself Gaussian and the posterior covariance matrix, C_{MAP}, and the model resolution matrix, R, depend on the covariance matrix for measurement errors and the sensitivity of observed data to model parameters, but are independent of the actual measurement outcome. This allows us to calculate the uncertainty reduction using only the prior reservoir model without the need to perform any history matching. For weakly non-linear problems, and when comparing significantly different surveillance scenarios, this method may provide an acceptable answer. For other cases, however, C_{MAP} and R can give only a rough characterization of the relative utility of the surveillance operations under consideration.

History matching is often used to quantify uncertainty reduction after a particular set of data has been measured. This is different from surveillance planning, where we do not know the outcome of the measurement until the actual measurement takes place, i.e., until a particular surveillance operation has been chosen and implemented to collect data. It is expected that for a particular surveillance scenario, the uncertainty reduction varies with the outcome of the measurements. However, history matching can still be adapted to the surveillance optimization problem. For each surveillance scenario, we need to generate different plausible reservoir models with different plausible outcomes, then history match an ensemble of models with each plausible outcome and calculate the average uncertainty reduction of that surveillance scenario. The procedure has to be repeated for all potential surveillance scenarios to conclude which scenario is the best to carry out. Although this is a reasonably rigorous way to solve the problem, the computational cost associated with history matching is simply too high to make it feasible. In the field, engineers often adopt a less rigorous way, using a simple risk analysis method. Typically, for each measurement

scenario, the engineers will define several plausible outcomes and assign a probability to each plausible outcome. Then the engineers will determine the uncertainty reduction associated with each plausible outcome, which does not necessarily come from the reservoir model, but more often from the experience and the intuition of the engineers. The average uncertainty reduction for a particular measurement scenario is now simply calculated as the weighted average of the uncertainty reduction associated with all plausible outcomes. This method, in principle, is based on the same idea as the rigorous history matching method mentioned above, but the difference lies in the fact that it involves little use of the reservoir model. The engineers simply tries to assign probabilities so that the uncertainty reduction seems reasonable and makes sense.

Attempts have been made to develop methods that can quantify the uncertainty reduction for a general case, without performing history matching. Abellan and Noetinger (2010), for example, use a concept in information theory called Kullback-Leibler divergence to calculate the information content provided by a set of measurement data. The information content is a direct indicator of the uncertainty reduction as more information content in the measurement data will lead to more uncertainty reduction in the reservoir model. Note that the particular type of Kullback-Leibler divergence that Abellan and Noetinger (2010) use is equivalent to the mutual information concept. There is no history matching involved in using this method, as one simply needs to generate an ensemble of realizations using the prior reservoir model and collect the vectors of observed data and reservoir variable J associated with all realizations. The Kullback-Leibler divergence can then be calculated from the cross-plot of the observed data and reservoir variable J. It reflects the strength of the general relationship, both linear and nonlinear, between observed data and J. However, the non-linear advantage of the Kullback-Leibler divergence comes with a big trade-off, that is, there exists no explicit formula to calculate the Kullback-Leibler divergence for a general case. Using the linear-Gaussian assumption, Abellan and Noetinger (2010) develop an equation to calculate the Kullback-Leibler divergence that requires only the covariance and sensitivity matrices. It is our understanding that, by using this equation, all the non-linear generality

of the Kullback-Leibler divergence formulation is lost, and we come down to an equation that can only be proved to be correct for linear cases. The authors present two examples to demonstrate the effectiveness of this method in choosing the best surveillance scenario. There is, however, no discussion of the authors on the accuracy of this method since their equation is developed for linear cases and the examples are both non-linear cases. They provide also no comparison between their method and other simple, linear-based methods such as the model resolution matrix mentioned above.

1.4 Research Objectives and Dissertation Outline

1.4.1 Research Objectives

The primary objective of our research is to develop new practical methods that improve several aspects of the closed-loop reservoir management workflow with particular focus on history matching and surveillance optimization. Specific elements in our research are as follows:

1. To develop adaptive ES-MDA algorithms for history matching that do not require a priori specification of the number of data assimilations, N_a, or a priori specification of the damping factors, α_i's.

2. To develop an ensemble-based history matching procedure that can be used with categorical variables, so that, for examples, the history matching preserves a distinct boundary between facies.

3. To develop a surveillance optimization method that can reliably estimate the expected uncertainty reduction of yet-to-be-measured data and at the same time, is computationally feasible, i.e. avoids history matching or at least minimizes the number of history matches required.

1.4.2 Dissertation Outline

There are five chapters in this dissertation. In Chapter 2, we present the derivation of

the adaptive ES-MDA algorithms, followed by the comparison with other popular methods when applied to a synthetic 3-D reservoir problem. In Chapter 3, we introduce a history matching procedure that is based on ES-MDA but adapted to work with reservoir models with multiple facies. The procedure is applied to three different cases with increasing degree of difficulty. In Chapter 4, we present our procedure based on information theory to estimate the uncertainty reduction expected from different surveillance scenarios. The estimation using our proposed procedure is verified with the results obtained using a rigorous history matching procedure. Chapter 5 presents our conclusions and summarizes the research contributions of this research.

CHAPTER 2

ADAPTIVE ES-MDA

2.1 Methodology

Based on the analogy between one iteration of ES and one Gauss-Newton iteration (Reynolds et al., 2006), Emerick and Reynolds (2012b, 2013a) proposed the ES-MDA algorithm, which repeats the ES procedure several times on the same observed data. The ES-MDA method can be summarized as follows:

1. Choose the number of data assimilations, N_a, and the multiplication coefficients of the data covariance matrix, α_ℓ, for $\ell = 1, 2, \ldots, N_a$. The coefficients α_ℓ's must be selected such that $\sum_{\ell=1}^{N_a} \frac{1}{\alpha_\ell} = 1$.

2. FOR $\ell = 1$ to N_a:

   (a) Run the ensemble from time zero.

   (b) For each ensemble member, perturb the observation vector using

   $$d_{\mathrm{uc},j} = d_{\mathrm{obs}} + \sqrt{\alpha_\ell} C_{\mathrm{D}}^{1/2} Z_j, \tag{2.1}$$

   where $d_{\mathrm{uc},j}$ is the vector of perturbed observations, d_{obs} is the vector of observed data, and $Z_j \sim \mathcal{N}(0, I_{N_d})$.

   (c) Update the ensemble using

   $$m_j^a = m_j^f + \widetilde{C}_{\mathrm{MD}}^f \left(\widetilde{C}_{\mathrm{DD}}^f + \alpha_\ell C_{\mathrm{D}} \right)^{-1} \left(d_{\mathrm{uc},j} - d_j^f \right), \tag{2.2}$$

where j is the ensemble index, $j = 1, ..., N_e$; m_j^a is the updated vector of model parameters; $\widetilde{C}_{\mathrm{MD}}^f$ is the cross-covariance matrix between the vector of model parameters, m^f, and the vector of predicted data, d^f; $\widetilde{C}_{\mathrm{DD}}^f$ is the auto-covariance matrix of predicted data.

END FOR.

Since the ES-MDA algorithm is motivated by the Gauss-Newton algorithm, it is natural to adopt ideas from previous works on how to control over correction with Gauss-Newton (Hanke, 1997, 2010; Gao and Reynolds, 2006; Tavakoli and Reynolds, 2010, 2011; Iglesias and Dawson, 2013) to improve the ES-MDA algorithm. The following subsections discuss two adaptive ES-MDA algorithms developed using this past knowledge as motivation.

2.1.1 ES-MDA-RS

One important lesson from previous works on Gauss-Newton is that excessive modification of the model parameters at each iteration is not ideal and often makes the model susceptible to overshooting (Li et al., 2003; Gao and Reynolds, 2006; Tavakoli and Reynolds, 2010). One way to alleviate this issue is to increase the inflation factor so that the change in model parameters at each iteration is below a reasonable threshold. We propose to use two standard deviations of the prior model as the limit of how much the model parameters can change in one iteration. This choice is motivated by the fact that 95% of the samples from a Gaussian distribution is within two standard deviations of the mean. In addition, we believe that the inflation factor should be an increasing function of the data mismatch because, intuitively, a smaller correction (greater inflation factor) should be used when the data mismatch is bigger and vice versa. This idea is similar to Gao and Reynolds (2006) where the authors use the magnitude of the objective function to determine the damping factor. Combining the two ideas, we propose an adaptive ensemble smoother algorithm called ES-MDA-RS where RS stands for restricted step. The algorithm is given below.

FOR each iteration:

1. Run the ensemble from time zero.

2. Calculate the average normalized objective function

$$\overline{O}_{Nd} = \frac{1}{N_e} \sum_{j=1}^{N_e} O_{Nd,j}, \tag{2.3}$$

where the normalized objective function $O_{Nd,j}$ is defined by

$$O_{Nd,j} = \frac{1}{2N_d} \sum_{j=1}^{N_e} (d_j^f - d_{obs})^T C_D^{-1} (d_j^f - d_{obs}). \tag{2.4}$$

3. Set $\alpha_\ell = 0.25 * \overline{O}_{Nd}$ as the initial guess for the inflation factor, where ℓ is the index of the current iteration. As discussed above, this choice is motivated by the proposal that the inflation factor should be based on the value of the objective function. This serves as the minimum value of the inflation factor for the current iteration.

4. For each ensemble member, perturb the observation vector using

$$d_{\text{uc, j}} = d_{\text{obs}} + \sqrt{\alpha_\ell} C_{\text{D}}^{1/2} Z_j, \tag{2.5}$$

where $Z_j \sim \mathcal{N}(0, I_{N_d})$.

5. Update the ensemble using

$$m_j^a = m_j^f + \widetilde{C}_{\text{MD}}^f \left(\widetilde{C}_{\text{DD}}^f + \alpha_\ell C_{\text{D}} \right)^{-1} \left(d_{\text{uc},j} - d_j^f \right), \tag{2.6}$$

where $\widetilde{C}_{\text{MD}}^f$ is the cross-covariance matrix between the vector of model parameters, m^f, and the vector of predicted data, d^f; $\widetilde{C}_{\text{DD}}^f$ is the auto-covariance matrix of predicted data.

6. For each ensemble member j and for each component i of the jth vector of model parameters:

- Calculate the change from the previous iteration to the current iteration

$$\delta m_{j,i} = |m^a_{j,i} - m^f_{j,i}|, \tag{2.7}$$

  where the double subscript j, i denotes the ith component of the jth realization.

- Compare $\delta m_{j,i}$ with the prior standard deviation of the ith model parameter, σ_i.

- If $\delta m_{j,i}$ is greater than $2\sigma_i$ for any model component of any ensemble member, discard the updated ensemble calculated in step 5, double α_ℓ and redo steps 4, 5 and 6 until no more violation occurs.

7. Calculate the sum of the inverse inflation factors up to the current iteration,

$$\beta_\ell = \sum_{k=1}^{\ell} \frac{1}{\alpha_k}. \tag{2.8}$$

- If $\beta_\ell = 1$, terminate the algorithm.

- If $\beta_\ell > 1$, increase α_ℓ to the value that makes β_ℓ exactly equal to 1, discard the updated ensemble calculated in step 5 and redo steps 4 and 5 to calculate the final model parameters, then terminate the algorithm.

- If $\beta_\ell < 1$, the algorithm continues to the next iteration, i.e., we return to step 1.

END FOR

In the standard ES-MDA (Emerick and Reynolds, 2012b, 2013a), the number of iterations (N_a) and the inflation factors (α_ℓ's) are set before doing any computations and steps 2 and 3 of the algorithm are deleted.

2.1.2 ES-MDA-RLM

Insight on how to improve ES-MDA can also be drawn from the work of Iglesias and Dawson (2013). Although their work focuses on generating a regularized solution of a pure least-squares problem, it provides insight of value here. Specifically, Iglesias and Dawson

(2013) seek to minimize the objective function given by

$$O_d(m) = \frac{1}{2}(d_{obs} - g(m))^T C_D^{-1}(d_{obs} - g(m)). \tag{2.9}$$

Note that this objective function is similar to that in Eq. 1.3 except the regularization term provided by the prior model is missing. Iglesias and Dawson replaced $g(m)$ in Eq. 2.9 by the approximation

$$g(m) \approx g(m^\ell) + G_\ell(m - m^\ell) = g(m^\ell) + G_\ell \delta m^{\ell+1}, \tag{2.10}$$

where G_ℓ is the sensitivity matrix defined above and $\delta m^{\ell+1} = m - m^\ell$ and then replace the minimization of $O_d(m)$ by the following linear inverse problem: find $w = m - m^\ell$ which minimizes

$$O_{dl}(m) \equiv O_{dl}(w) = \frac{1}{2}(d_{obs} - g(m_\ell) - G_\ell w)^T C_D^{-1}(d_{obs} - g(m_\ell) - G_\ell w). \tag{2.11}$$

Minimizing the objective function in Eq. 2.9 is generally ill-posed, in which case, we require regularization to define a unique solution. Iglesias and Dawson (2013) used a regularizing Levenberg-Marquardt scheme, or more specifically, the regularizing scheme of Hanke (1997, 2010), where Tikonov regularization is applied to the linear inverse problem of Eq. 2.11. In this case, $\delta m^{\ell+1}$ is obtained as

$$\delta m_{\ell+1}(\alpha) = \arg\min_w \left(\frac{1}{2}(d_{obs} - g(m_\ell) - G_\ell w)^T C_D^{-1}(d_{obs} - g(m_\ell) - G_\ell w) + \frac{1}{2}\alpha w^T C_M^{-1} w \right), \tag{2.12}$$

where, here, α is the regularization parameter. In essence, the scheme in Eq. 2.12 regularizes the linear inverse problem of Eq. 2.11 by penalizing the change, $\delta m^{\ell+1}$, in the model parameters over the iteration. The solution to the regularized minimization problem of Eq. 2.12 is

$$m^{\ell+1} = m^\ell + C_M G_\ell^T (G_\ell C_M G_\ell^T + \alpha C_D)^{-1}(d_{obs} - g(m^\ell)). \tag{2.13}$$

Eq. 2.13 has the same structure as the ES-MDA update equation (Eq. 2.6). In particu-

lar, the regularization parameter α in ES-MDA has the same effect as the regularization parameter α in Eq. 2.13. Therefore, we can adopt Hanke's recommendation for choosing the regularizing parameter α to our choice of the ES-MDA inflation factor (also denoted by α). According to the ideas of Hanke (also see Iglesias and Dawson (2013) on regularizing Levenberg-Marquardt), α should satisfy

$$\rho^2||C_D^{-1/2}(d_{\text{obs}} - g(m^\ell))||^2 \leq \alpha^2||C_D^{1/2}(G_\ell C_M G_\ell^T + \alpha C_D)^{-1}(d_{\text{obs}} - g(m^\ell))||^2, \tag{2.14}$$

for some ρ with $0 < \rho < 1$. As discussed later, finding α according to this criterion requires that we specify a value of ρ and there is no firm theoretical guidance for how to choose ρ. However, with this choice of α, Iglesias and Dawson (2013) prove that if the noise level approaches zero, then the regularized least squares solution converges to an m^* such that $g(m_{\text{true}}) = g(m^*)$, where m_{true} is the true model which results in d_{obs}. Given the aforementioned equivalent structure of Eq. 2.13 and the ES-MDA update equation (Eq. 2.6), we can by simple analogy extend Eq. 2.14 to ES-MDA to obtain a criterion for choosing the inflation factor at each iteration ℓ. In particular, we propose choosing α_ℓ so that

$$\rho^2||C_D^{-1/2}(d_{uc,j} - d_j^f)||^2 \leq \alpha_\ell^2||C_D^{1/2}(C_{DD}^f + \alpha_\ell C_D)^{-1}(d_{uc,j} - d_j^f)||^2. \tag{2.15}$$

From Eq. 2.15, we can see that larger ρ requires larger α_ℓ, which will lead to more iterations, but larger α_ℓ also invokes more damping, which should reduce overshooting (over correction) with ES-MDA and result in smoother and more geologically reasonable property fields. Iglesias and Dawson (2013) performed experiments on a 2D reservoir case with values of ρ ranging from 0.65 to 0.95 and showed that the final data misfit is smaller for larger values of ρ but at the cost of more iterations. The authors then suggested using $\rho \in [0.7, 0.8]$ for the best compromise between efficiency and accuracy. However, for our particular example case, due to the very large difference between the initial simulated data and the observed data, the value of α required to satisfy the criterion would be unreasonably large (greater than 10^{100}) if we used ρ in the range recommended by Iglesias and Dawson (2013). Because of this

issue, for the example shown here, we use $\rho = 0.2$ in our implementation of the new adaptive ES-MDA method. Here, this adaptive ES-MDA algorithm is denoted by ES-MDA-RLM where RLM stands for regularizing Levenberg-Marquardt. The algorithm is given below.

FOR each iteration:

1. Run the ensemble from time zero.

2. Calculate the average normalized objective function

$$\overline{O}_{Nd} = \frac{1}{N_e} \sum_{j=1}^{N_e} O_{Nd,j}, \tag{2.16}$$

   where the normalized objective function $O_{Nd,j}$ is defined by

$$O_{Nd,j} = \frac{1}{2N_d} \sum_{j=1}^{N_e} (d_j^f - d_{obs})^T C_D^{-1} (d_j^f - d_{obs}). \tag{2.17}$$

3. Set $\alpha_\ell = 0.25 * \overline{O}_{Nd}$ as the initial guess for the inflation factor, where ℓ is the index of the current iteration.

4. For each ensemble member, perturb the observation vector using

$$d_{\text{uc, j}} = d_{\text{obs}} + \sqrt{\alpha_\ell} C_{\text{D}}^{1/2} Z_j, \tag{2.18}$$

   where $Z_j \sim \mathcal{N}(0, I_{N_d})$.

5. For each ensemble member j:

   - Check the condition in Eq. 2.15, which is repeated below for continuity in presentation:

$$\rho^2 ||C_D^{-1/2}(d_{uc,j} - d_j^f)||^2 \leq \alpha_\ell^2 ||C_D^{1/2}(C_{DD}^f + \alpha_\ell C_D)^{-1}(d_{uc,j} - d_j^f)||^2. \tag{2.19}$$

- If the condition is violated for any ensemble member, double α_ℓ and redo steps 4 and 5 until no violation occurs.

6. Calculate the sum of the inverse inflation factors up to the current iteration,

$$\beta_\ell = \sum_{k=1}^{\ell} \frac{1}{\alpha_k}. \tag{2.20}$$

   - If $\beta_\ell = 1$, the algorithm ends after the completion of step 7 of this iteration.
   - If $\beta_\ell > 1$, increase α_ℓ to the value that makes β_ℓ exactly equal to 1. The algorithm ends after the completion of step 7 of this iteration.
   - If $\beta_\ell < 1$, the algorithm continues to the next iteration after the completion of step 7 of this iteration.

7. Update the ensemble using

$$m_j^a = m_j^f + \widetilde{C}_{\mathrm{MD}}^f \left(\widetilde{C}_{\mathrm{DD}}^f + \alpha_\ell C_{\mathrm{D}} \right)^{-1} \left(d_{\mathrm{uc},j} - d_j^f \right), \tag{2.21}$$

   where $\widetilde{C}_{\mathrm{MD}}^f$ is the cross-covariance matrix between the vector of model parameters, m^f, and the vector of predicted data, d^f; $\widetilde{C}_{\mathrm{DD}}^f$ is the auto-covariance matrix of predicted data.

END FOR

One advantage of the ES-MDA-RLM method compared to the ES-MDA-RS method is that we do not have to recompute the analysis equation (Eqs. 2.6 and 2.21) for every new guess of α. Instead, we only have to recompute the right-hand side of Eq. 2.15, which is much faster because $N_d \ll N_m$ in all cases of interest to us. This advantage becomes more apparent when the number of model parameters is on the order of hundreds of thousands or more.

2.1.3 ES-MDA-RLM with an Alternate Stopping Rule

Iglesias and Dawson (2013) provide a stopping rule for the regularized Levenberg-

Marquardt algorithm. The stopping rule is as follows:

$$||C_D^{-1/2}(d_{\text{obs}} - g(m^\ell))|| \leq \tau\eta, \tag{2.22}$$

where the theory dictates that τ must be chosen so that $\tau > \dfrac{1}{\rho}$ and $\eta \approx \sqrt{N_d}$, where N_d is the number of observed data. Note introducing τ requires that we have two parameters to choose a priori, τ and ρ. Iglesias and Dawson (2013) suggest using $\tau = \dfrac{1}{\rho - 10^{-3}}$ and we do so for the results shown in this research. To adapt the stopping rule to our ensemble-based methods, we replace $g(m^\ell)$ by $\bar{d}^f = \dfrac{1}{N_e}\sum_{j=1}^{N_e} d_j^f$, which is the average of the predicted data, so that our choice of the stoping rule is

$$||C_D^{-1/2}(d_{\text{obs}} - \bar{d}^f)|| \leq \tau\eta. \tag{2.23}$$

This stopping rule is hereafter referred to as the RLM stopping rule (RLM-SR) to distinguish it from the traditional ES-MDA termination criterion which requires we iterate until the sum of the inverse inflation factors is unity. If we use the RLM stopping rule, the adaptive ES-MDA procedure can no longer be proven to sample correctly in the linear-Gaussian case. One test we did on a simple linear Gaussian history matching problem indeed confirms that the algorithm does not sample correctly and the uncertainty in the posterior model depends on the value of ρ. With the RLM stopping rule, we introduce an alternate version of the ES-MDA-RLM algorithm, thereafter referred to as ES-MDA-RLM-SR. The algorithm is as follows:

FOR each iteration:

1. Run the ensemble from time zero.

2. Check the RLM stopping rule in Eq. 2.23, which is repeated below for continuity in presentation,

$$||C_D^{-1/2}(d_{\text{obs}} - \bar{d}_j^f)|| \leq \tau\eta. \tag{2.24}$$

If the stopping rule is satisfied, terminate the algorithm without proceeding to subsequent steps.

3. Calculate the average normalized objective function,

$$\overline{O}_{Nd} = \frac{1}{N_e} \sum_{j=1}^{N_e} O_{Nd,j}, \tag{2.25}$$

where the normalized objective function $O_{Nd,j}$ is defined by

$$O_{Nd,j} = \frac{1}{2N_d} \sum_{j=1}^{N_e} (d_j^f - d_{obs})^T C_D^{-1} (d_j^f - d_{obs}). \tag{2.26}$$

4. Set $\alpha_\ell = 0.25 * \overline{O}_{Nd}$, where ℓ is the index of the current iteration, as the initial guess for the inflation factor.

5. For each ensemble member, perturb the observation vector using

$$d_{\text{uc, j}} = d_{\text{obs}} + \sqrt{\alpha_\ell} C_{\text{D}}^{1/2} Z_j, \tag{2.27}$$

where $Z_j \sim \mathcal{N}(0, I_{N_d})$.

6. For each ensemble member j:

   - Check the condition in Eq. 2.15, repeated below for continuity in presentation,

$$\rho^2 ||C_D^{-1/2}(d_{uc,j} - d_j^f)||^2 \leq \alpha_\ell^2 ||C_D^{1/2}(C_{DD}^f + \alpha_\ell C_D)^{-1}(d_{uc,j} - d_j^f)||^2. \tag{2.28}$$

   - If the condition is violated for any ensemble member, double α_ℓ and redo steps 5 and 6 until no more violation occurs.

7. Update the ensemble using

$$m_j^a = m_j^f + \widetilde{C}_{\text{MD}}^f \left(\widetilde{C}_{\text{DD}}^f + \alpha_\ell C_{\text{D}} \right)^{-1} \left(d_{\text{uc},j} - d_j^f \right), \tag{2.29}$$

where $\widetilde{C}^f_{\mathrm{MD}}$ is the cross-covariance matrix between the vector of model parameters, m^f, and the vector of predicted data, d^f; $\widetilde{C}^f_{\mathrm{DD}}$ is the auto-covariance matrix of predicted data.

END FOR

At this point, it is worth mentioning that, independently of our work, Iglesias pursues his own adaptation of the regularizing Levenberg-Marquardt scheme to ensemble-based methods (Iglesias, 2014, 2015). There are four major differences between our proposed adaptive ES-MDA-RLM algorithm and Iglesias' algorithm:

1. We try two different termination criteria: one that comes directly from ES-MDA and guarantees that we sample correctly in the linear Gaussian case and one adapted from the stopping rule in Iglesias and Dawson (2013). (In the latest version of his paper, Iglesias (2015) also utilizes the stopping rule of Iglesias and Dawson (2013).)

2. We check the regularization criteria based on the worst-case ensemble member (Eq. 2.19) instead of based on the ensemble average as is done by Iglesias (2014, 2015).

3. At each ℓth ES-MDA data assimilation, we generate new perturbed data, $d_{uc,j}$'s, by sampling $\mathcal{N}(d_{obs}, \alpha_\ell C_D)$ whereas Iglesias (2015) generates the $d_{uc,j}$'s only at the first iteration. With this perturbation scheme and the requirement that the sum of inverse inflation factors is equal to unity, our method is guaranteed to sample correctly in the linear-Gaussian case. Proposition 2 of Iglesias (2015) states conditions under which it is guaranteed that his IR-ES algorithm samples correctly. However, the conditions require that ρ is small enough and τ is large enough so that IR-ES terminates after one iteration with an initial damping factor chosen equal to unity whereas these conditions are not part of the stated IR-ES (Algorithm II of Iglesias (2015)) which only requires $0 \leq \rho < 1$. By choosing an initial factor equal to unity, IR-ES reduces to the ES which is known to sample correctly as the ensemble size goes to infinity. Enforcing the conditions of Proposition 2 of Iglesias (2015) requires that it is known a priori that

we have a linear-Gaussian problem and that the sensitivity matrix in the data-model relationship is known.

4. Iglesias (2015) uses ρ between 0.7 and 0.9, consistent with the suggestion in Iglesias and Dawson (2013) but violates the theoretical condition that $\tau > 1/\rho$ by using $\tau = 1$ regardless of the choice of ρ. Note that Iglesias (2015) presents arguments for not enforcing $\tau > 1/\rho$ in the context of the Bayesian inverse problem and compares the sampling properties of his algorithm for an example where the correct posterior pdf is estimated by a Markov chain Monte Carlo simulation.

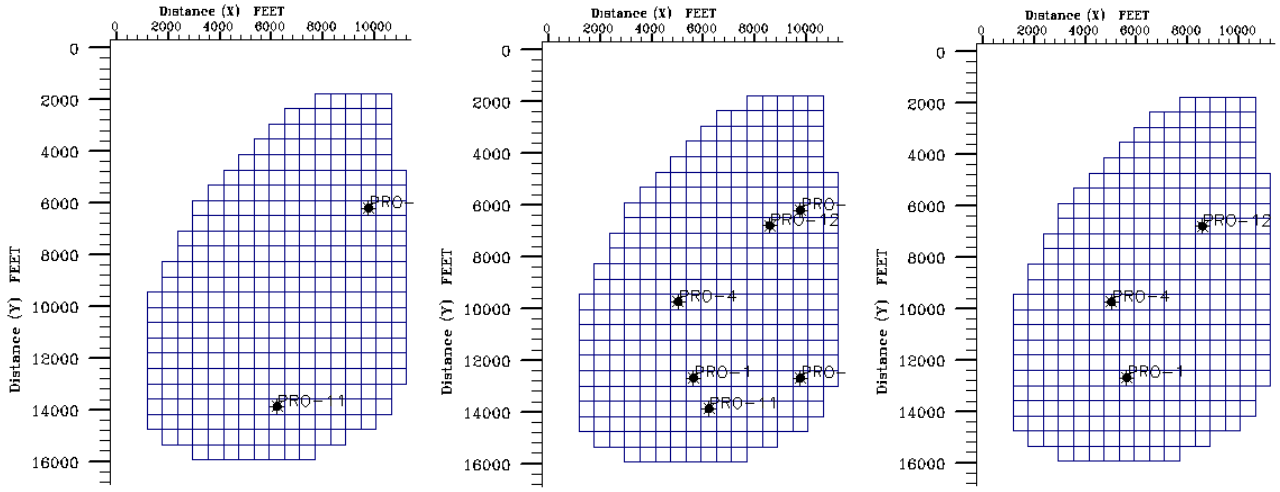
2.2 Case Study 1

2.2.1 Case description

We consider the three-phase (oil, gas and water), three-dimensional PUNQS3 reservoir model (Floris et al., 2001; Barker et al., 2001; Gao et al., 2006). The reservoir grid is $19 \times 28 \times 5$; the size of each gridblock is 590.55 ft $\times$ 590.55 ft in the x and y directions but the thicknesses of the gridblocks vary. There are 1761 active gridblocks in the model. There are six producers with the locations and perforated intervals the same as in the original model (Floris et al., 2001; Barker et al., 2001); see Fig. 2.1. All production wells are under oil rate control. The reservoir is supported by a strong analytical aquifer on the west and south sides. The production consists of a first year of extended well testing, followed by a 3-year shut-in period and then 12.5 years of production. The oil production rates for each well during the first year of well testing for each of the four 3-month periods are specified as 628.98, 1257.96, 628.98 and 314.49 STB/D. Subsequent to the 3-year shut-in period, the oil production rate at each well is set equal to 943.47 STB/D. The history-matching period consists of the first 11.5 years. The forecast period makes up the remaining 5.5 years. The data to be matched include bottomhole pressure (BHP), gas/oil ratio (GOR) and water cut (WCT). To generate the observed data, we add Gaussian noise to the true production data obtained from the reservoir simulator. The standard deviations for the measurement error

(noise) are specified as 10 psi for BHP and 3% of the true values for GOR and WCT data, where C_D, the data or measurement error covariance matrix, is assumed to be diagonal.

The model parameters to be matched are the horizontal and vertical permeability, the porosity, the fluid contact depths and the parameters of the relative permeability curves. The true horizontal log-permeability field is shown in Fig. 2.2. The unconditional log permeability and porosity fields are generated using sequential Gaussian co-simulation based on the geostatistical parameters given in Gao et al. (2006), where no hard data are used. The lack of hard data at wells makes this a challenging problem because almost none of the prior realizations have the high permeability streaks passing through the production wells, making the predicted well data dramatically different from the observed data.

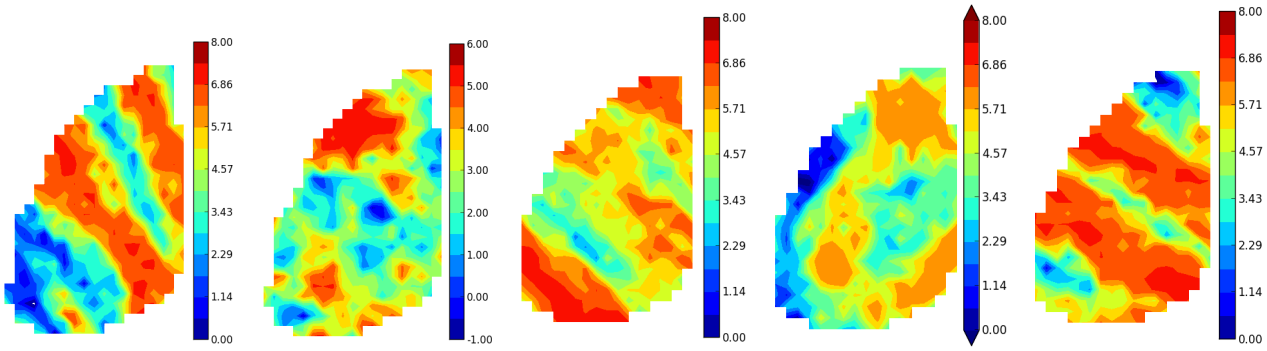


(a) Layer 3 (b) Layer 4 (c) Layer 5

Figure 2.1: Locations of perforations in layers 3, 4 and 5. There is no perforation in layers 1 and 2.

The true GOC and OWC depths are 7726.4 ft and 7857.6 ft, respectively. In the prior realizations of GOC and OWC depths, we deliberately introduce a bias in the prior means, setting these means equal to 7746.4 ft and 7877.6 ft, respectively. The standard deviation of the GOC and OWC depths in the prior realizations are both 20 ft.

To generate realizations of the relative permeability curves, we slightly modify the power law model used in Wang et al. (2010) to better fit the requirements of the simulator. For example, Eclipse 100 requires that the relative permeability data for water, k_{rw}, are



(a) Layer 1 (b) Layer 2 (c) Layer 3 (d) Layer 4 (e) Layer 5

Figure 2.2: True horizontal permeability field ($\ln k$)

available from $S_w = S_{iw}$ to S_w= 1, but the k_{rw} model used in Wang et al. (2010),

$$k_{rw} = k^*_{rw} \left(\frac{S_w - S_{iw}}{1.0 - S_{iw} - S_{orw}} \right)^{e_w}, \tag{2.30}$$

can only generate k_{rw} from $S_w = S_{iw}$ to $S_w = 1 - S_{orw}$. Eclipse will accept this incomplete input data and will then extrapolate to obtain the missing data. The power law relative permeability models that we use are given by

$$k_{rw} = k^*_{rw} \left(\frac{S_w - S_{iw}}{1.0 - S_{iw}} \right)^{e_w}, \tag{2.31}$$

$$k_{row} = k^*_{ro} \left(\frac{1.0 - S_w - S_{orw}}{1.0 - S_{iw} - S_{orw}} \right)^{e_{ow}}, \tag{2.32}$$

$$k_{rg} = k^*_{rg} \left(\frac{S_g - S_{gc}}{1.0 - S_{iw} - S_{gc}} \right)^{e_g}, \tag{2.33}$$

$$k_{rog} = k^*_{ro} \left(\frac{1.0 - S_g - S_{iw} - S_{org}}{1.0 - S_{iw} - S_{org}} \right)^{e_{og}}, \tag{2.34}$$

where the Stone Model I (Aziz and Settari, 1979) is used to calculate the three-phase relative permeability from the sets of two-phase relative permeability functions.

The parameters used to generate the prior realizations of the relative permeability

curves are shown in Table 2.1. Note that the values of the end-point saturations are fixed; we only try to match the exponents and the coefficients.

Table 2.1: Specifications of relative permeability paramaters

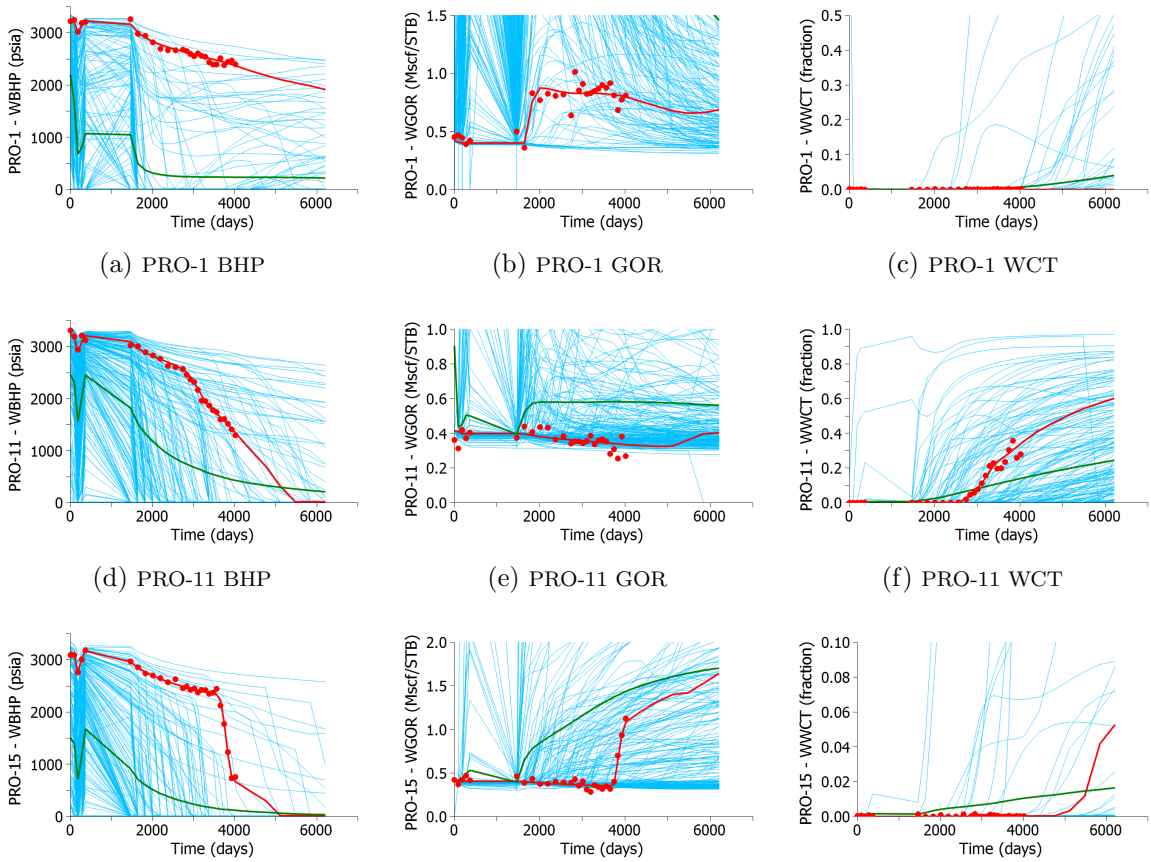
	Mean	Stdev	Min	Max
k^*_{rw}	1.0	0.0	1.0	1.0
k^*_{ro}	0.85	0.05	0.7	1.0
k^*_{rg}	0.25	0.05	0.1	0.4
e_w	4.75	0.9	2.5	7.0
e_{ow}	2.25	0.3	1.5	3.0
e_g	7.0	1.2	4.0	10.0
e_{og}	2.35	0.46	1.2	3.5

The prior predicted data are shown in Fig. 2.3 for 3 wells, PRO-1, PRO-11 and PRO-15. These three wells are chosen because they represents different speeds of pressure drops and water break through. As can be seen from the figure, the initial ensemble has an unreasonably large spread with the observed data barely within the spread of ensemble predictions. This case can therefore be considered an extreme case to illustrate the advantage of our proposed method. A more reasonable case will be considered in example 2.

2.2.2 Results

We perform history matching on 10 different ensembles where each ensemble has 100 ensemble members. Note that we repeat the history matching with multiple ensembles in order to investigate the consistency of the proposed methods, all of which are stochastic. We do not intend to imply that doing history matching with 10 ensembles of size 100 yields a better characterization of the posterior pdf than doing history matching with one ensemble of size 1000. In fact, using one ensemble of size 1000 seems to be give better results for this particular problem (Le and Reynolds, 2013). We test and compare the results of five methods:

- ES-MDA 8x: equal inflation factors of 8.
- ES-MDA 16x: equal inflation factors of 16.



(a) PRO-1 BHP (b) PRO-1 GOR (c) PRO-1 WCT

(d) PRO-11 BHP (e) PRO-11 GOR (f) PRO-11 WCT

(g) PRO-15 BHP (h) PRO-15 GOR (i) PRO-15 WCT

Figure 2.3: Prior ensemble of predicted data

- ES-MDA-RS: model change restriction of 2 standard deviations. On average, this algorithm finishes in 20 iterations.

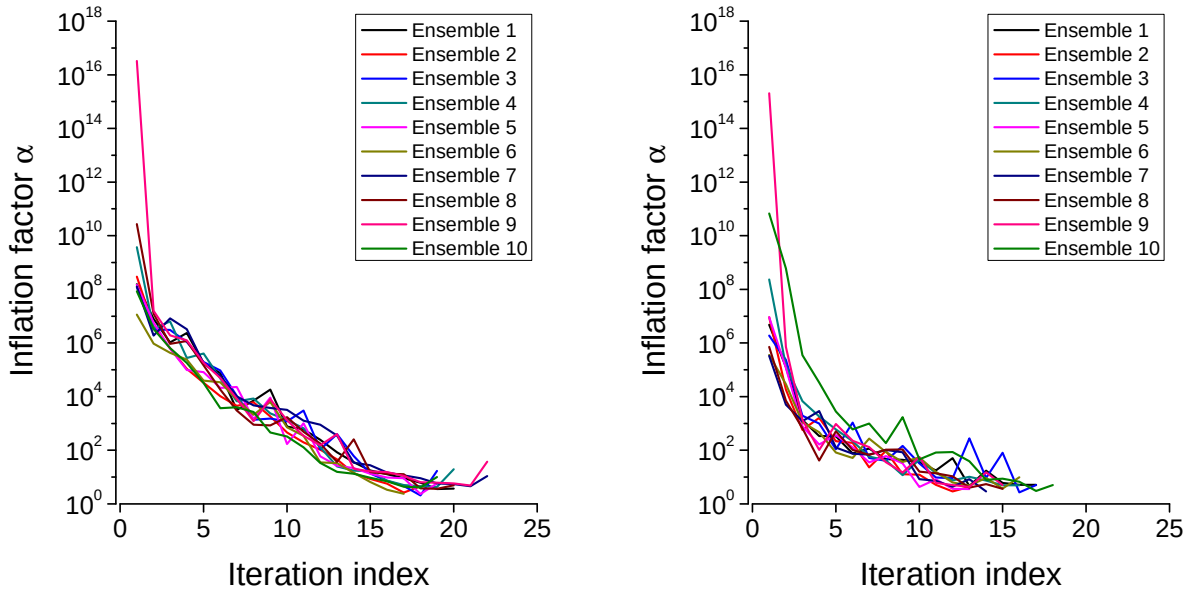
- ES-MDA-RLM: $\rho = 0.2$. On average, ES-MDA-RLM finishes in 16 iterations using traditional ES-MDA stopping rule and finishes in 12 iterations using the RLM stopping rule.

- LM-EnRML (Chen and Oliver, 2013). On average, this algorithm finishes in 25 iterations. The LM-EnRML algorithm is given in Appendix A.1.

In the LM-EnRML method, the initial value of the multiplier λ, λ_{ini}, is chosen to be 100,000, which, by the recommendation of Chen and Oliver (2013), is on the same order of magnitude as the average data mismatch calculated using Eq. 2.4. The parameter γ, which dictates

how much λ is changed from one iteration to the next, is 10 as recommended by Chen and Oliver (2013). Later, we see that the $\lambda_{\text{ini}} = 100,000$ does not work well, so we repeat the LM-EnRML runs with $\lambda_{\text{ini}} = 10,000$ and $\lambda_{\text{ini}} = 1,000$.

Fig. 2.4 shows the evolution of the inflation factor for the two adaptive ES-MDA algorithms, ES-MDA-RS and ES-MDA-RLM, with the traditional ES-MDA stopping rule. It can be seen that the initial inflation factor needs to be very high in order to satisfy the regularization conditions but it decreases to a much smaller value after a few iterations. This result is consistent with previous works of Gao and Reynolds (2006) and Tavakoli and Reynolds (2010) which show that controlling the change in parameters during the first few iterations is critical in order to avoid over correction of model parameters and unrealistically rough property fields.



(a) ES-MDA-RS (b) ES-MDA-RLM

Figure 2.4: Evolution of the inflation factors in the adaptive ES-MDA methods.

Figs. 2.5, 2.6 and 2.7, respectively, show the third, fourth and fifth layers' log horizontal permeability for the first realization of each posterior ensemble. The true log horizontal permeability is also plotted in the same figure for the convenience of the reader. It can be seen that the problem of overshooting due to ill conditioning is quite severe in the result

obtained with ES-MDA 8x, i.e. we obtain a very rough permeability field which contains mostly very high and very low permeability values. The problem is alleviated somewhat with ES-MDA 16x but is still serious. The two adaptive ES-MDA methods show much less overshooting than is obtained with the two standard ES-MDA methods. Using the RLM stopping rule, the ES-MDA-RLM algorithm completes earlier, which results in slightly less overshooting than the other two adaptive ES-MDA methods, even though we can no longer prove that this modified ES-MDA-RLM algorithm samples correctly for the linear Gaussian case. The LM-EnRML method with $\lambda_{\text{ini}} = 1,000$ shows much more overshooting and undershooting than with $\lambda_{\text{ini}} = 10,000$ and $100,000$. As we will show later, the reduced overshooting in LM-EnRML with $\lambda_{\text{ini}} = 100,000$ is partly because the algorithm does not significantly modify the initial values of the model so that the posterior models are similar to the prior models, and consequently we have virtually no overshooting. On the other hand, $\lambda_{\text{ini}} = 10,000$ seems to be the optimal choice among the three initial inflation factors, where the overshooting and undershooting is as light as $\lambda_{\text{ini}} = 100,000$ and, as will be shown later, the data match is as good as $\lambda_{\text{ini}} = 1,000$.

Fig. 2.8 shows the ratio of the posterior standard deviation to the prior standard deviation for the gridblock log-horizontal permeabilities of layer 3. Note that the standard deviation values are calculated for each individual ensemble. Without a rigorous sampling method like MCMC, it is not possible to know the correct values of the standard deviation. All the ensemble smoother methods, including ES-MDA 8x, ES-MDA 16x, ES-MDA-RS and ES-MDA-RLM, give standard deviations that do not vary significantly across ensembles, which shows that these methods are quite consistent. For ES-MDA-RLM with the RLM stopping rule, the values of posterior standard deviations are generally much greater than the values obtained with other ensemble smoother methods, which is expected because the data assimilation is terminated earlier. It is interesting to observe that the standard deviations obtained by LM-EnRML with $\lambda_{\text{ini}} = 100,000$ vary significantly from one ensemble to another. For ensembles #3, 7, 8 and 10, the posterior standard deviations obtained with LM-EnRML are not much smaller than the prior standard deviations. As shown later, this is because the

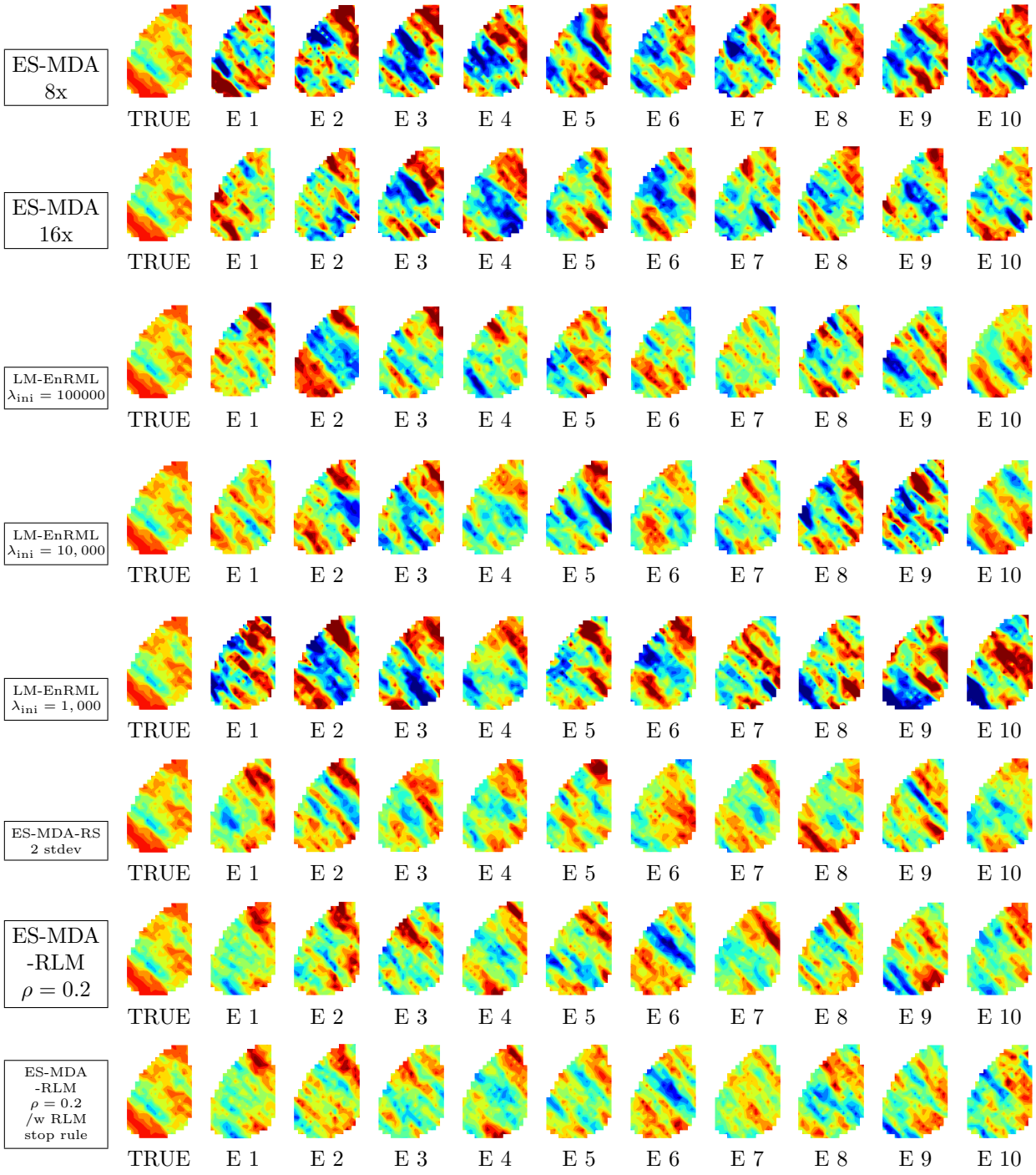


Figure 2.5: Layer 3 log horizontal permeability of the first realization of each ensemble.

algorithm is unable to obtain a good data match for these ensembles and is thus unable to reduce the uncertainty.

We propose the following metric to measure how close the posterior model is to the

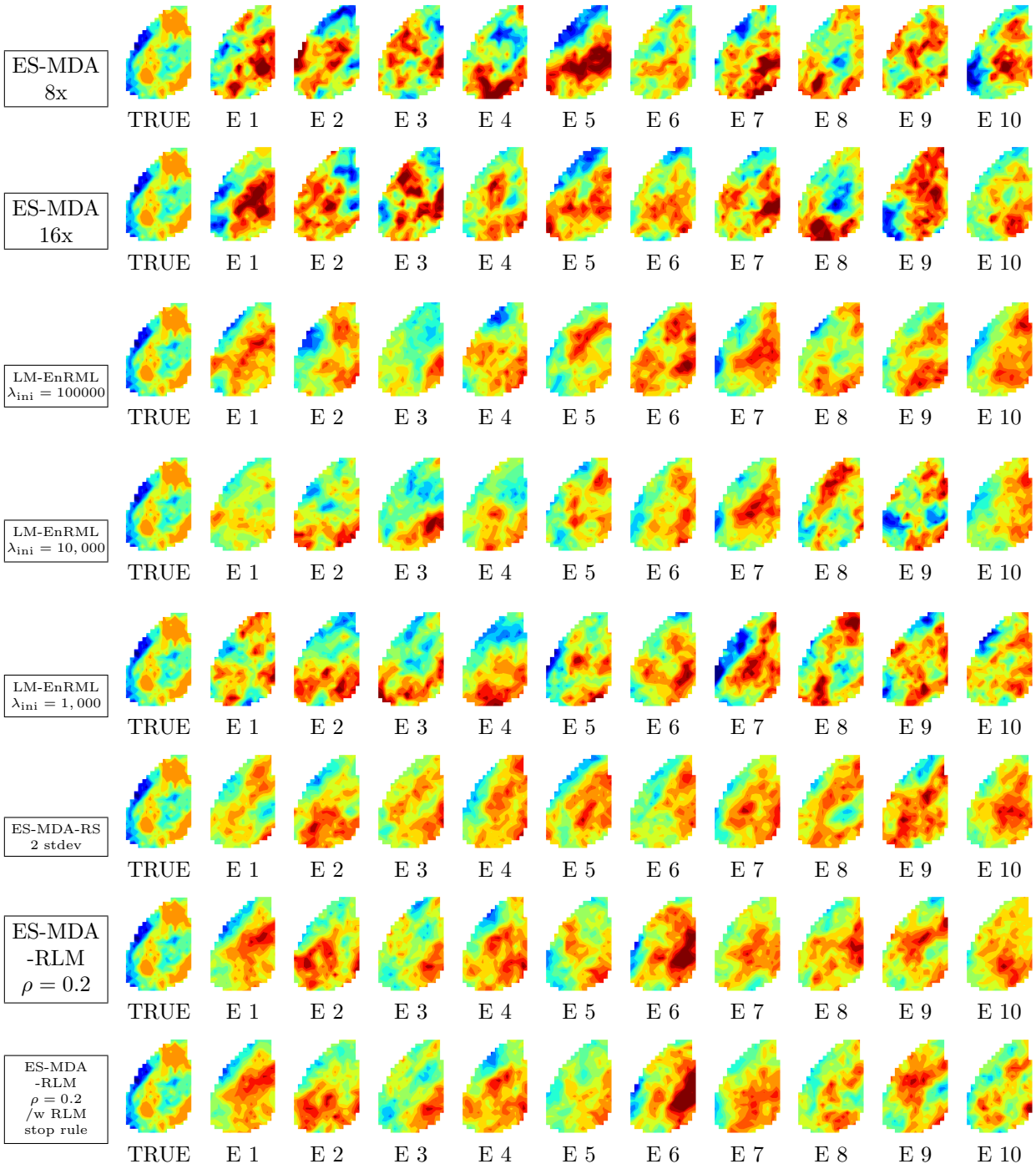


Figure 2.6: Layer 4 log horizontal permeability of the first realization of each ensemble.

true model:

$$R = \frac{1}{N_m} ||C_M^{-1/2}(m - m_{\text{true}})||_1, \tag{2.35}$$

where C_M is the prior covariance matrix of model parameters and $|| * ||_1$ denotes the L_1

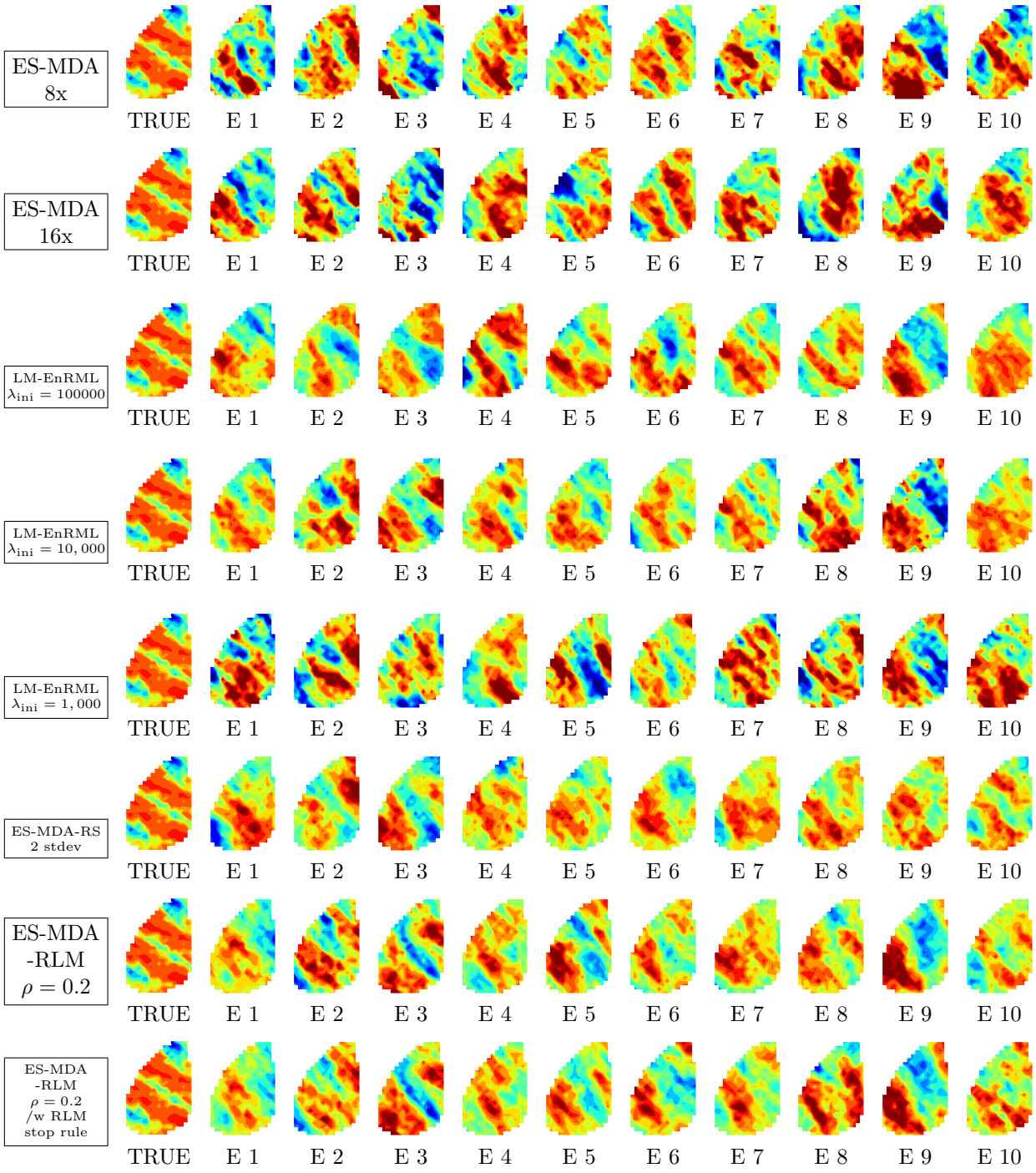


Figure 2.7: Layer 5 log horizontal permeability of the first realization of each ensemble.

norm. R can be interpreted as the average difference between the value of a posterior model parameter and its true value in terms of the prior standard deviations. If m_{true} is generated from the same distribution as the prior models it can be shown that the expected difference between a prior realization and the true model is $2/\sqrt{\pi} \approx 1.13$ (see Appendix A.2). In

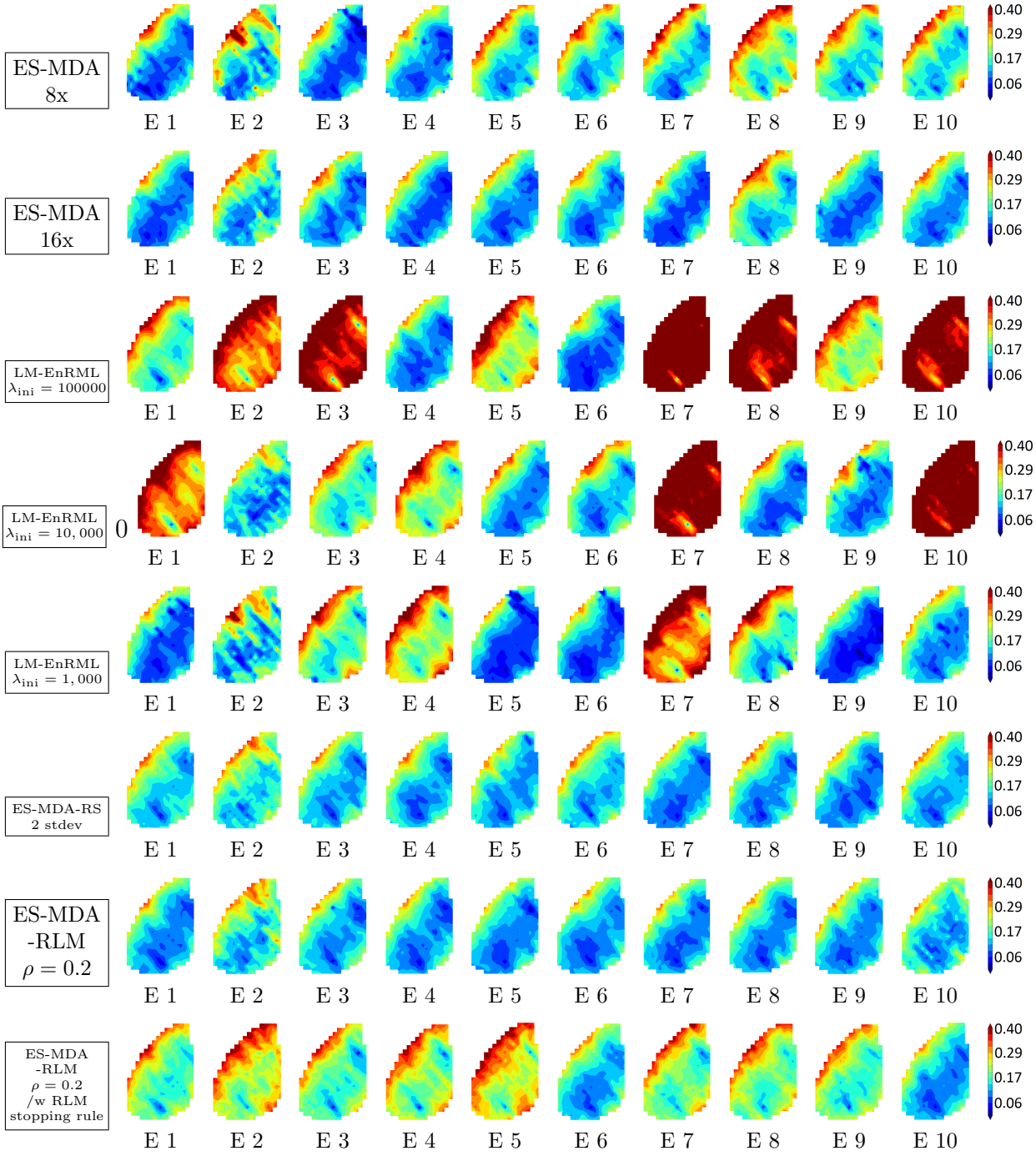


Figure 2.8: Standard deviation of the third layer's log horizontal permeability.

practice, because the prior covariance matrix of model parameters can be excessively large, we replace $C_M^{1/2}$ by D_M, which is a diagonal matrix whose diagonal elements are the prior

standard deviations of the model parameters. Eq. 2.35 then becomes

$$R = \frac{1}{N_m} ||D_M^{-1}(m - m_{true})||_1, \tag{2.36}$$

The plot of R calculated using Eq. 2.36 for different history-matching methods is shown in Fig. 2.9. The difference between the prior realizations and the true model has the mean (and median) of 1.10, which is very close to expected value of 1.13. It is interesting to note that the two regular ES-MDA methods lead to higher values of R than the prior model. This is most likely due to the excessive overshooting exhibited by these two methods. For the same reason, LM-EnRML with $\lambda_{\text{ini}} = 1,000$ results in higher value of R compared to LM-EnRML with $\lambda_{\text{ini}} = 10,000$ and $100,000$. The adaptive ES-MDA methods are able to alleviate this problem, resulting in smaller values of R than are obtained with the prior model.

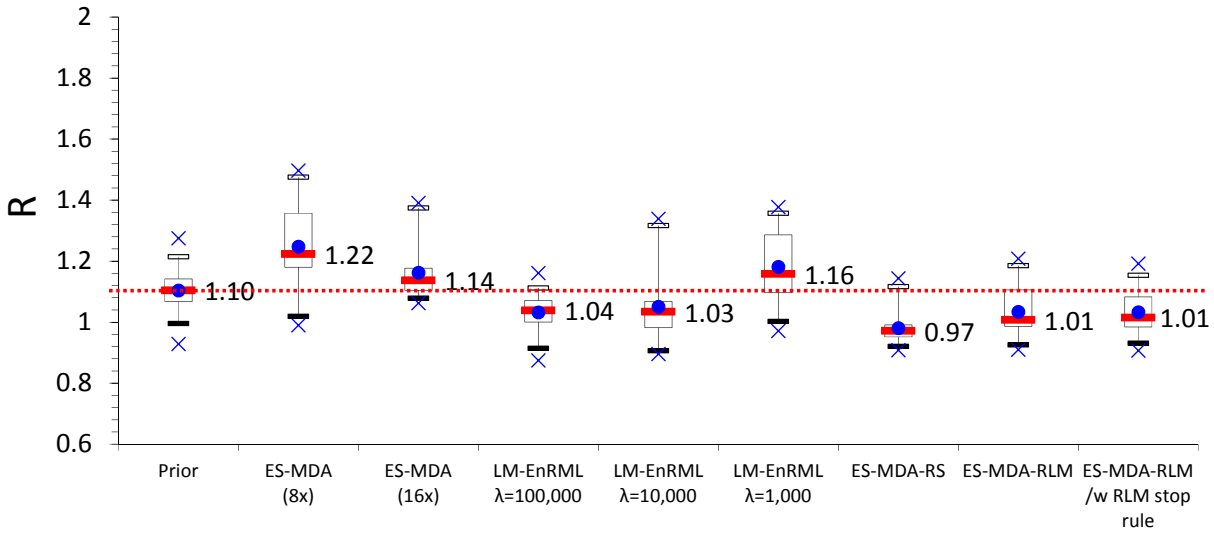


Figure 2.9: Difference between posterior models and the true model based on Eq. 2.36

Fig. 2.10 shows the box plot of the normalized objective function, which is created by combining all 10 ensembles. Overall, the two adaptive ES-MDA methods, ES-MDA-RS and ES-MDA-RLM with the traditional ES-MDA stopping rule, result in the best data match although this does not mean that these methods necessarily give a good approximation of the posterior pdf. On the other hand, a large value of the objective function suggests that

the corresponding model pertains to a high value of the posterior pdf which often means that this model is from a low probability region. Note that the range of the objective function values obtained from LM-EnRML with $\lambda_{\text{ini}} = 100,000$ is relatively large. A careful investigation of the objective function for each ensemble obtained with LM-EnRML reveals that the method is not consistent for this example problem; it performs very well for some ensembles but performs badly for others. Reducing the initial value of λ to 10,000 gives a better overall data match but further reduction to 1,000 leads to more severe overshooting. The ES-MDA-RLM algorithm with the RLM stopping rule results in a bigger data mismatch than is obtained without using this rule; this is the expected result because this stopping rule trades a tight data match for more variation in the posterior ensemble. In fact, the stopping rule is designed to avoid over matching the data, based on a discrepancy rule (Hanke, 1997, 2010; Iglesias and Dawson, 2013; Iglesias, 2015).

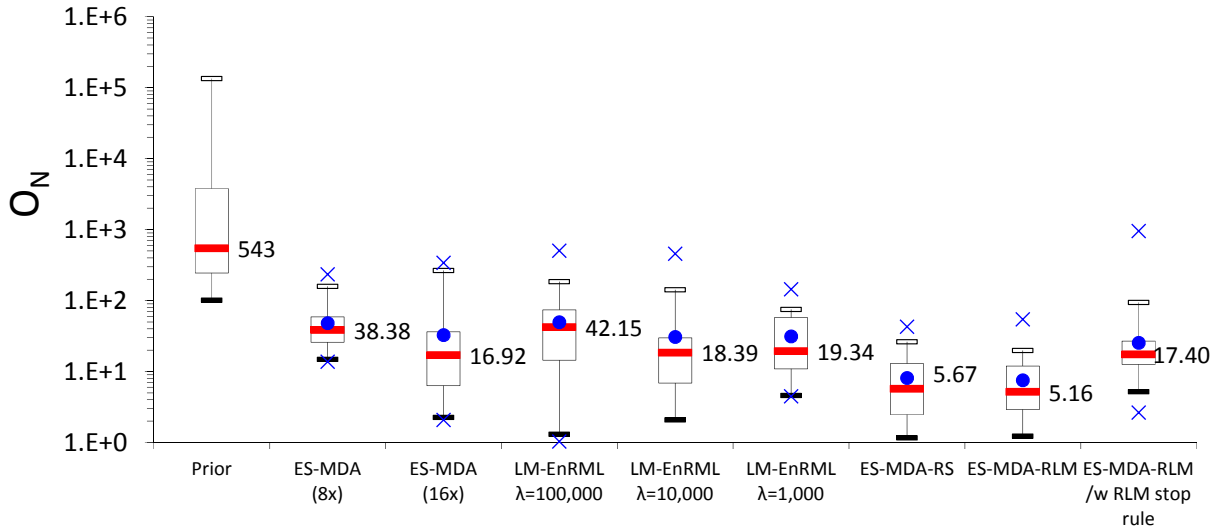


Figure 2.10: Normalized objective function.

The predicted well data from 10 ensembles for three wells PRO-1, PRO-11 and PRO-15 are shown in Fig. 2.11, 2.12 and 2.13. The results for the other wells are similar. Note that the history period consists of the first 4,000 days while the forecast period consists of the remaining 2,000 days. For the history period, the two adaptive ES-MDA methods clearly give better data matches than are obtained with the other methods. For the forecast period,

it is interesting to see that the history-matched ensembles can sometimes lead to poor and biased predictions even though the corresponding history-match is good.

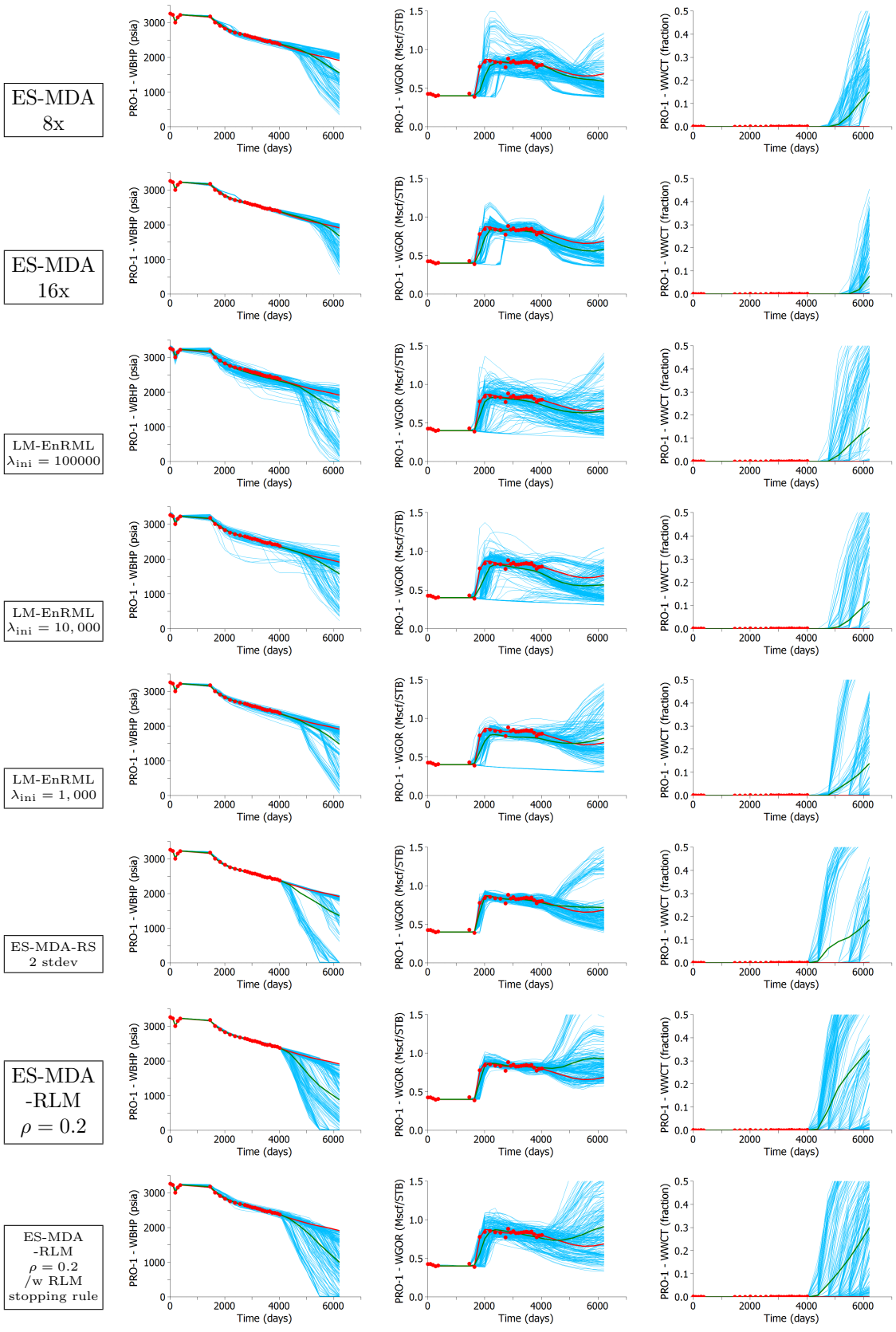


Figure 2.11: Posterior production data at PRO-1 obtained with different history-matching methods.

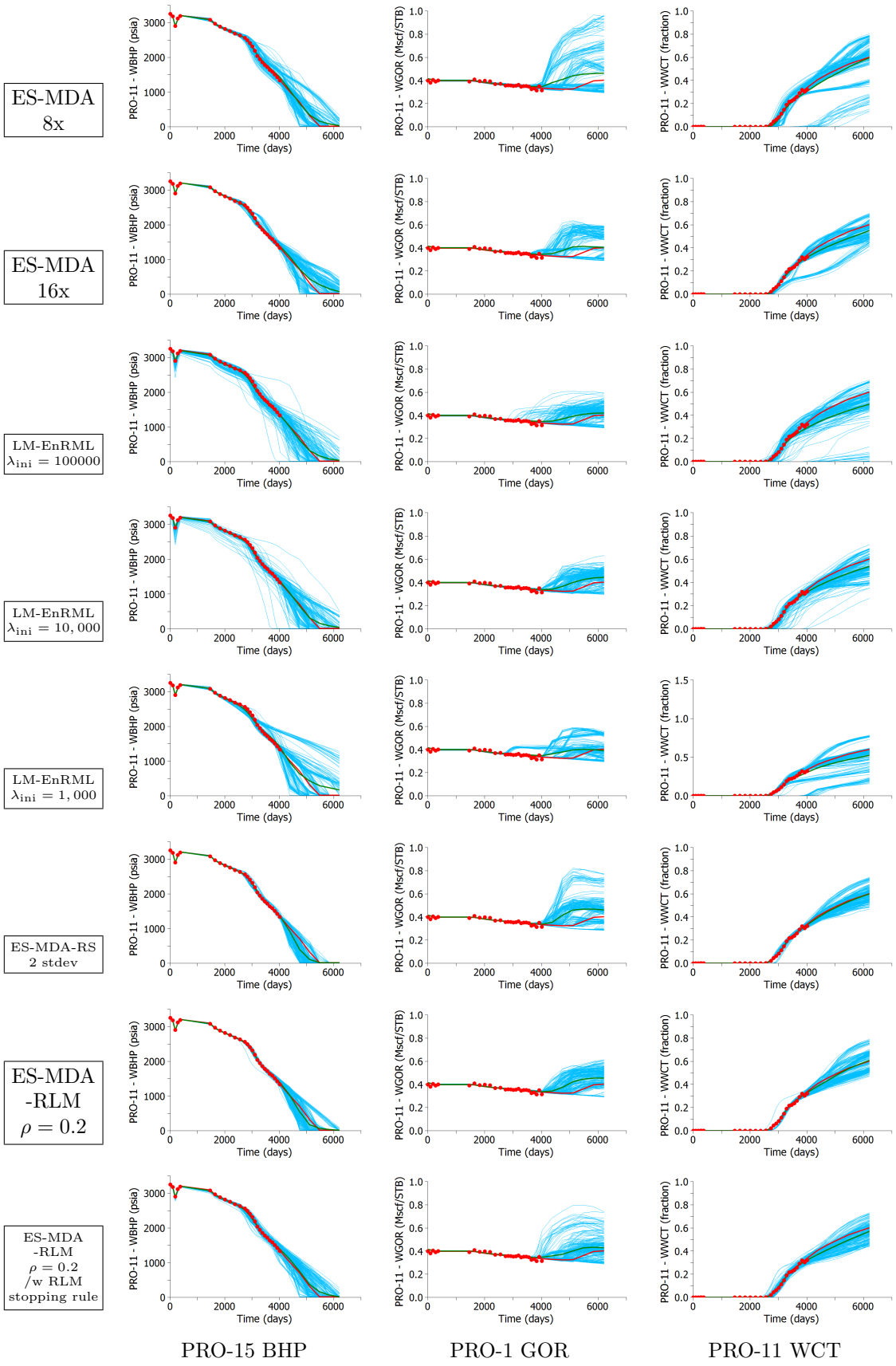


Figure 2.12: Posterior production data at PRO-11 obtained with different history-matching methods.

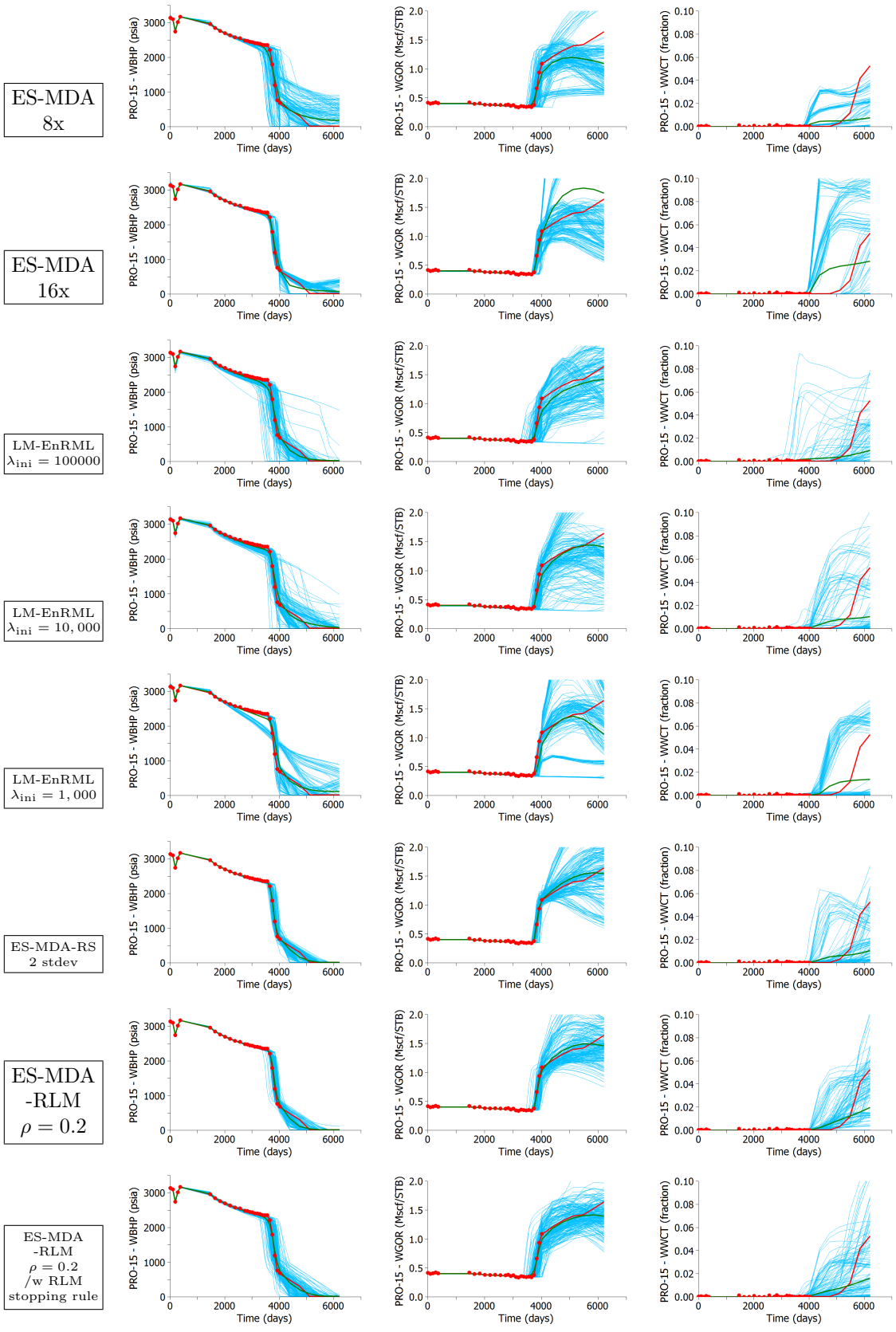


Figure 2.13: Posterior production data at PRO-15 obtained with different history-matching methods.

2.2.3 Further investigation

In this section, we consider in more detail one of the cases where the history match is good but the future prediction is poor and biased. It is important to note that the reservoir is a dome-shaped reservoir so the layers are not horizontal; a reservoir cross section along the East-West direction is shown in Fig. 2.14. We investigate two ensembles obtained with the ES-MDA-RS method, namely ensemble 1 and ensemble 3. First, we investigate ensemble 3, which displays the most discrepancy between the ensemble prediction and the prediction of the true reservoir model. Fig. 2.15 shows the GOR of well PRO-1 for both the history and forecast perior. The rapid increase in GOR which begins at around day 1600 is due to the free gas from the gas cap starting to enter the well.

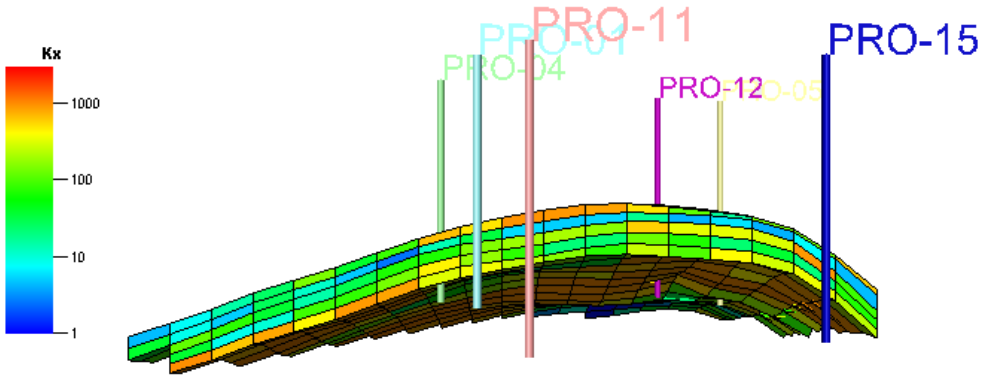


Figure 2.14: A reservoir cross section.

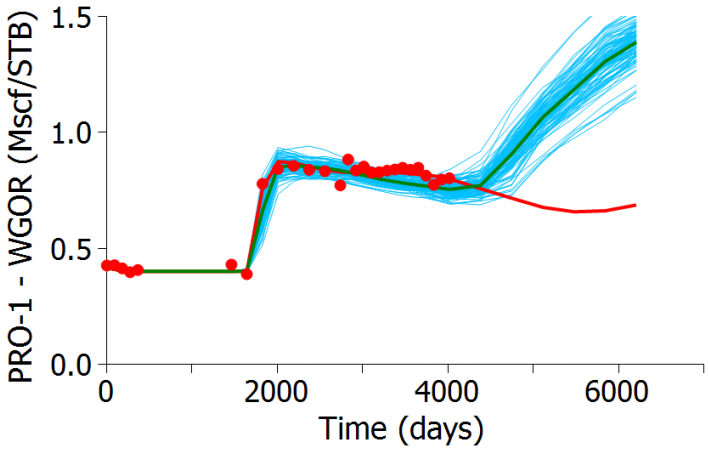


Figure 2.15: PRO-1 GOR in ensemble 3.

The posterior GOC depths in this ensemble have a mean of 7725.1 ft which is very close to the true value of 7726.4 ft, so posterior realizations of the GOC are unlikely to be the reason that the GOR prediction is biased. To investigate this bahavior, we perform the following modifications:

1. Replace the wrong GOC in the posterior models by the true GOC value to obtain the results of Fig. 2.16. As expected, this has a minor effect on the prediction since the posterior GOC values are already very close to the true GOC value.

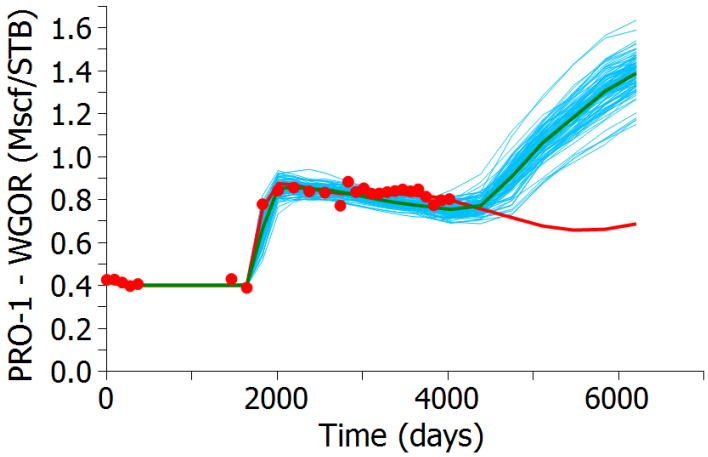


Figure 2.16: PRO-1 GOR in ensemble 3 when posterior GOC is replaced by true GOC.

2. Replace the posterior horizontal permeability by the true horizontal permeability to generate the results of Fig. 2.17. This has a major impact on the prediction, making the prediction much closer to the true data.

3. Replace the posterior vertical permeability by the true vertical permeability to obtain the results of Fig. 2.18. This has a small impact on the prediction, making the GOR prediction go down slightly at the end of the forecast period compared to the results of Fig. 2.15.

4. Replace the posterior porosity by the true porosity to obtain the results of Fig. 2.19. This changes the shape of the prediction significantly, making it higher in the middle but lower in the end.

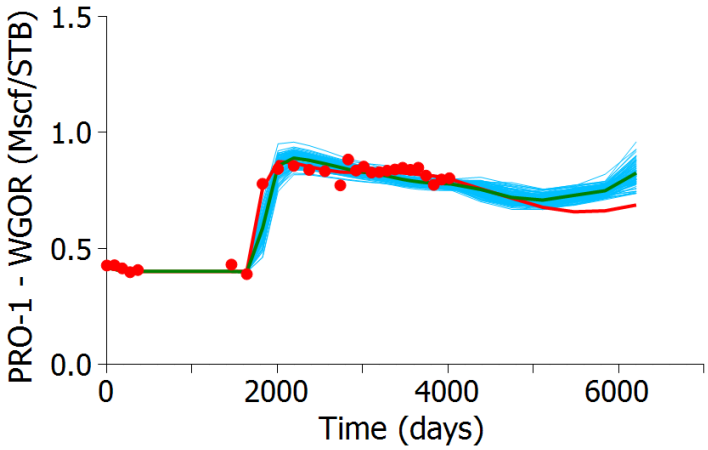


Figure 2.17: PRO-1 GOR in ensemble 3 when posterior horizontal permeability is replaced by true horizontal permeability.

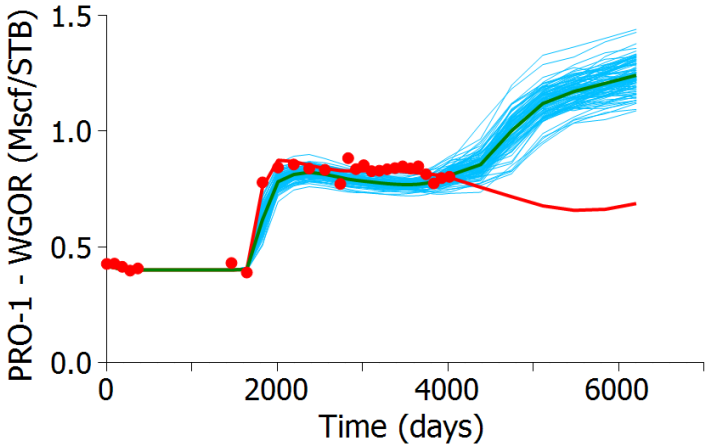


Figure 2.18: PRO-1 GOR in ensemble 3 when posterior vertical permeability is replaced by true vertical permeability.

5. Replace the posterior porosity, horizontal and vertical permeability by the true values to obtain the results of Fig. 2.20. This makes the prediction very close to the prediction from the true model.

From the above experiments, it seems that the horizontal permeability field has the highest impact on the PRO-1 GOR prediction. It is therefore useful to compare the posterior permeability and the true permeability to understand what causes the difference. Since well PRO-1 is perforated in layers 4 and 5, we compare the horizontal permeability in these two layers between one posterior realization and the true model (Fig. 2.21). It can be seen that

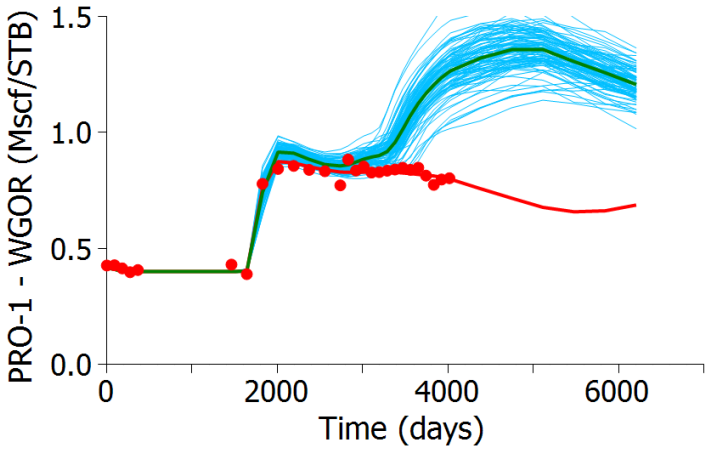


Figure 2.19: PRO-1 GOR in ensemble 3 when posterior porosity is replaced by true porosity.

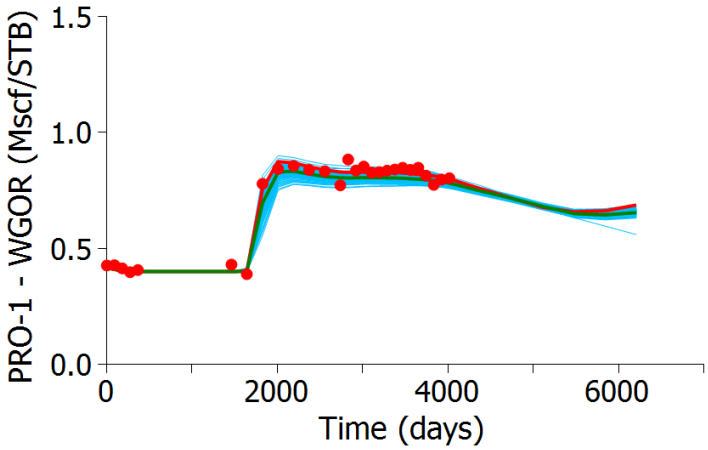
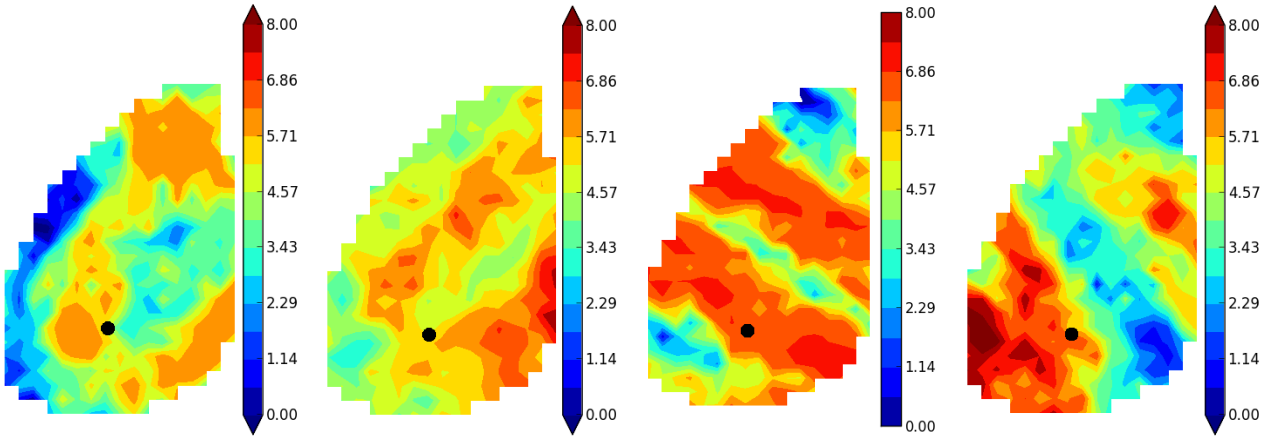


Figure 2.20: PRO-1 GOR in ensemble 3 when posterior porosity, horizontal and vertical permeability are replaced by true values.

the high permeability streak in layer 5 of the true model is not replicated in the posterior model. Because the well is under oil rate control, the pressure at well PRO-1 drops much faster in the posterior model than in the true model, causing more gas to come out of the solution, which leads to a higher GOR. This is further verified by noting that the solution GOR (dissolved gas-oil ratio, R_s) near well PRO-1 at late times is much lower in the posterior model than in the true model.

Another ensemble we investigate is ensemble 1. In this ensemble, the posterior GOC depths are vastly different from the true value. The mean posterior GOC depth is 7710 ft compared to the true value of 7726.4 ft. However, the GOR match is still fairly reasonable,



(a) Layer 4 - True (b) Layer 4 - Posterior (c) Layer 5 - True (d) Layer 5 - Posterior

Figure 2.21: Horizontal permeability field ($\ln k$). The black dot is the location of well PRO-1.

so there must be another "wrong" model parameter to compensate for the GOC. Fig. 2.22 shows the GOR of well PRO-1 after history matching.

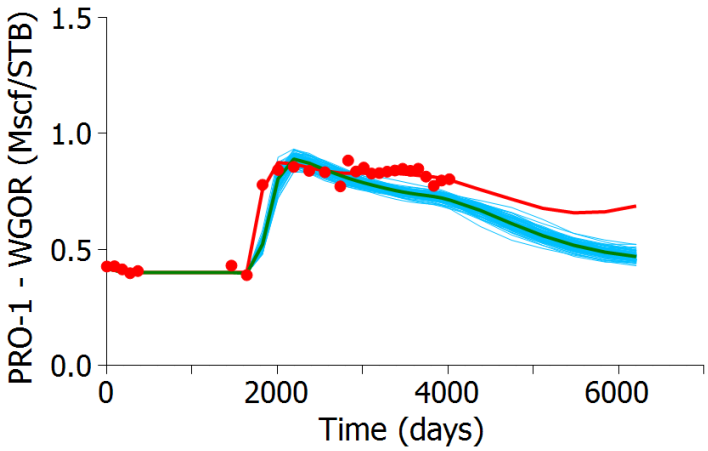


Figure 2.22: PRO-1 GOR in ensemble 1.

We perform the following experiments:

1. Replace the wrong GOC in the posterior models by the true GOC value to obtain the results of Fig. 2.23. As expected, because the true GOC is deeper, the GOR increases significantly.

2. Replace the posterior horizontal permeability by the true horizontal permeability to obtain the results of Fig. 2.24; note the reduction of GOR values and the delay of the

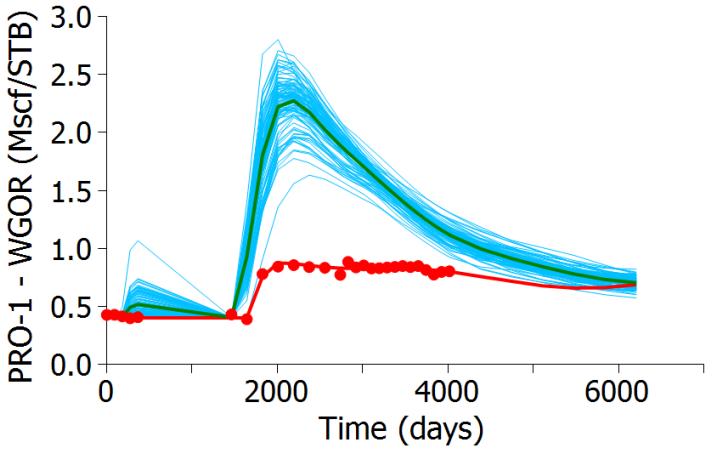


Figure 2.23: PRO-1 GOR in ensemble 1 when posterior GOC is replaced by true GOC.

breakthrough time.

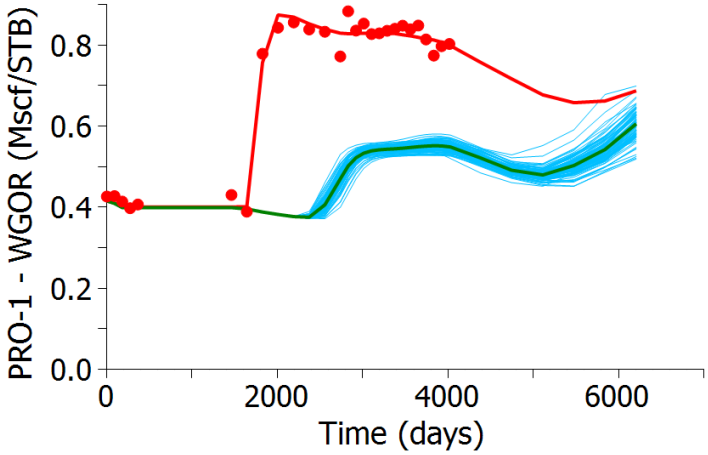


Figure 2.24: PRO-1 GOR in ensemble 1 when posterior horizontal permeability is replaced by true horizontal permeability.

3. Replace both the posterior GOC and the posterior horizontal permeability by the true values to obtain the results of Fig. 2.25. We can see that the two types of model parameters compensate for each other and lead to a GOR match that is almost as good as that in the unchanged posterior model. Also note how the GOR in the forecast period differs between Fig. 2.22 and Fig. 2.25.

4. Replace the posterior vertical permeability by the true vertical permeability. We don't show a figure here because this does not significantly alter the GOR data.

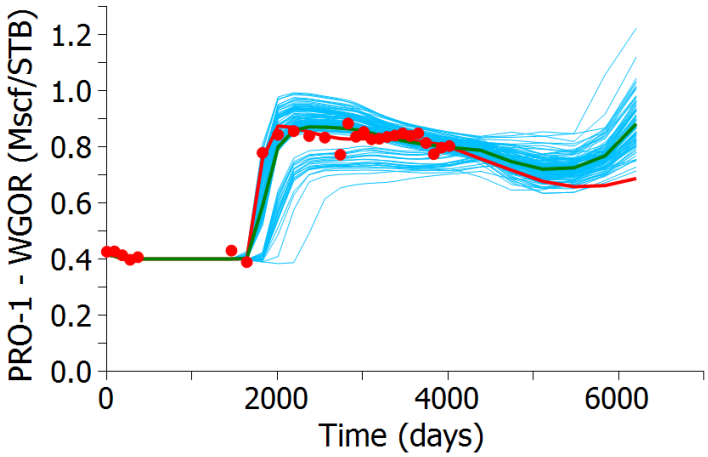
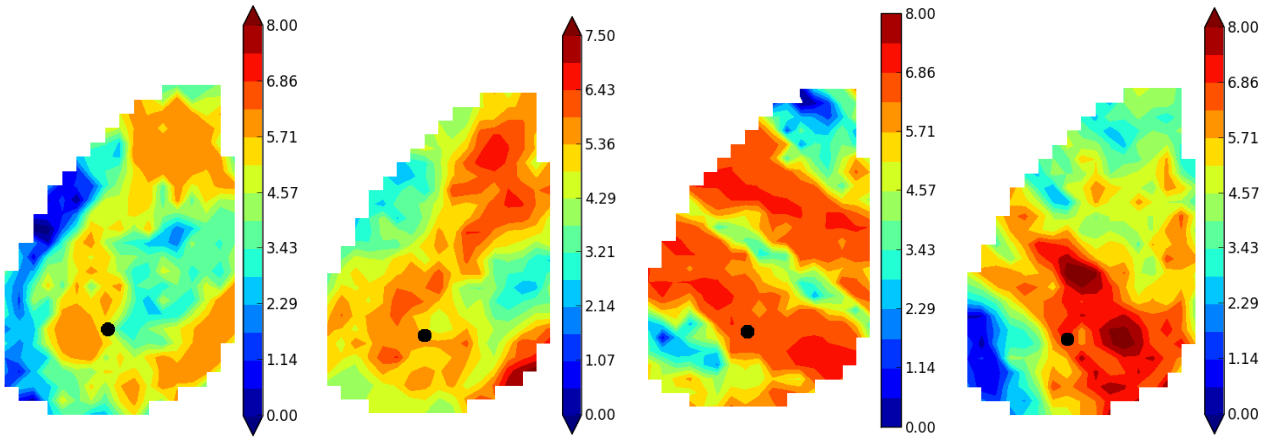


Figure 2.25: PRO-1 GOR in ensemble 1 when posterior GOC and horizontal permeability are replaced by true values.

Now, it is interesting to find out what difference between the posterior permeability and the true permeability leads to the difference in GOR values. Since well PRO-1 is perforated in layers 4 and 5, we compare the horizontal permeability in these two layers between one posterior realization and the true model (Fig. 2.26). It is most likely that the higher permeability in the posterior model allows gas breakthrough to happen faster at well PRO-1, leading to higher GOR prediction compared to the true model, provided that everything else is the same. This is compensated by the shallower GOC depth in the posterior model so that the data match is still good.

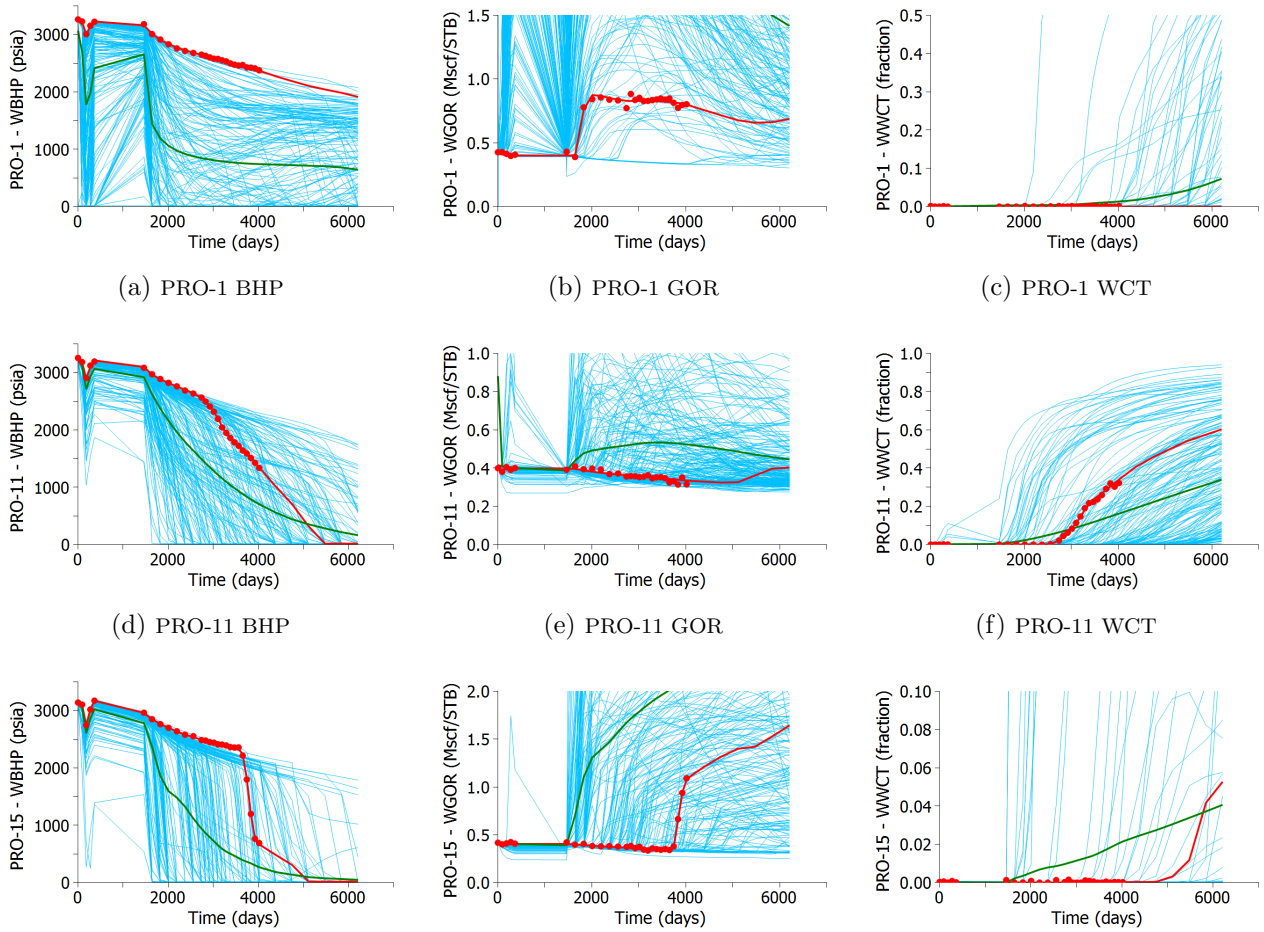


(a) Layer 4 - True (b) Layer 4 - Posterior (c) Layer 5 - True (d) Layer 5 - Posterior

Figure 2.26: Horizontal permeability field ($\ln k$). The black dot is well PRO-1 location.

2.3 Case Study 2

This case is similar to the first case except that hard data are used to generate the prior realizations. The prior ensembles of predicted data of three wells PRO-1, PRO-11 and PRO-15 are shown in Fig. 2.27. Compared to the first case, the prediction now shows better behavior as the ensemble of pressure predictions spread more evenly around the observed data.



(a) PRO-1 BHP (b) PRO-1 GOR (c) PRO-1 WCT

(d) PRO-11 BHP (e) PRO-11 GOR (f) PRO-11 WCT

(g) PRO-15 BHP (h) PRO-15 GOR (i) PRO-15 WCT

Figure 2.27: Predicted data of prior realizations for case study 2.

We apply ES-MDA 8x, ES-MDA 16x, LM-EnRML, ES-MDA-RS and ES-MDA-RLM to assimilate the data. For the LM-EnRML method, since the prior data mismatch is now smaller, we use smaller initial values of λ, namely 10,000, 1,000 and 100. Similar to the first case, each history matching method is performed on 10 ensembles, each of size 100.

The bottom three layers' permeability fields obtained after history matching are shown in Fig. 2.28, 2.29 and 2.30. For the regular ES-MDA methods, the overshooting and undershooting problems are now less severe compare to example 1, even though they still exist. Compared to the regular ES-MDA methods, the adaptive ensemble smoother methods result in less overshooting and undershooting, even though the difference is not as large as in the first example. The LM-EnRML method shows similarly degree of overshooting and undershooting as the adaptive ES-MDA methods when the initial λ value of 10,000 is used but exhibits more overshooting and undershooting when the initial λ is reduced to 1,000 and 100.

The degree of overshooting and undershooting associated with each method can be verified by calculating the difference between the posterior models and the true model using Eq. 2.35. The result is plotted in Fig. 2.31. The two regular ES-MDA methods and the LM-EnRML method with $\lambda_{\text{ini}} = 100$ are the three methods that exhibit the most severe overshooting and undershooting problems in Fig. 2.28 and expectedly, they are the three methods that increase the values of R after data assimilation.

The predicted data for wells PRO-1, PRO-11 and PRO-15, respectively, are shown in Figs. 2.32, 2.33 and 2.34 for different history matching methods. We can see that most methods except LM-EnRML with $\lambda_{\text{ini}} = 10{,}000$ perform quite well and result in good agreement with the observed data. This is different from the case where no hard data are available, where the regular ES-MDA methods do not result in good data matches. The performance of the methods are confirmed in the plot of objective function (Fig. 2.35); note how the posterior values of the objective function are generally smaller than the those in the first example.

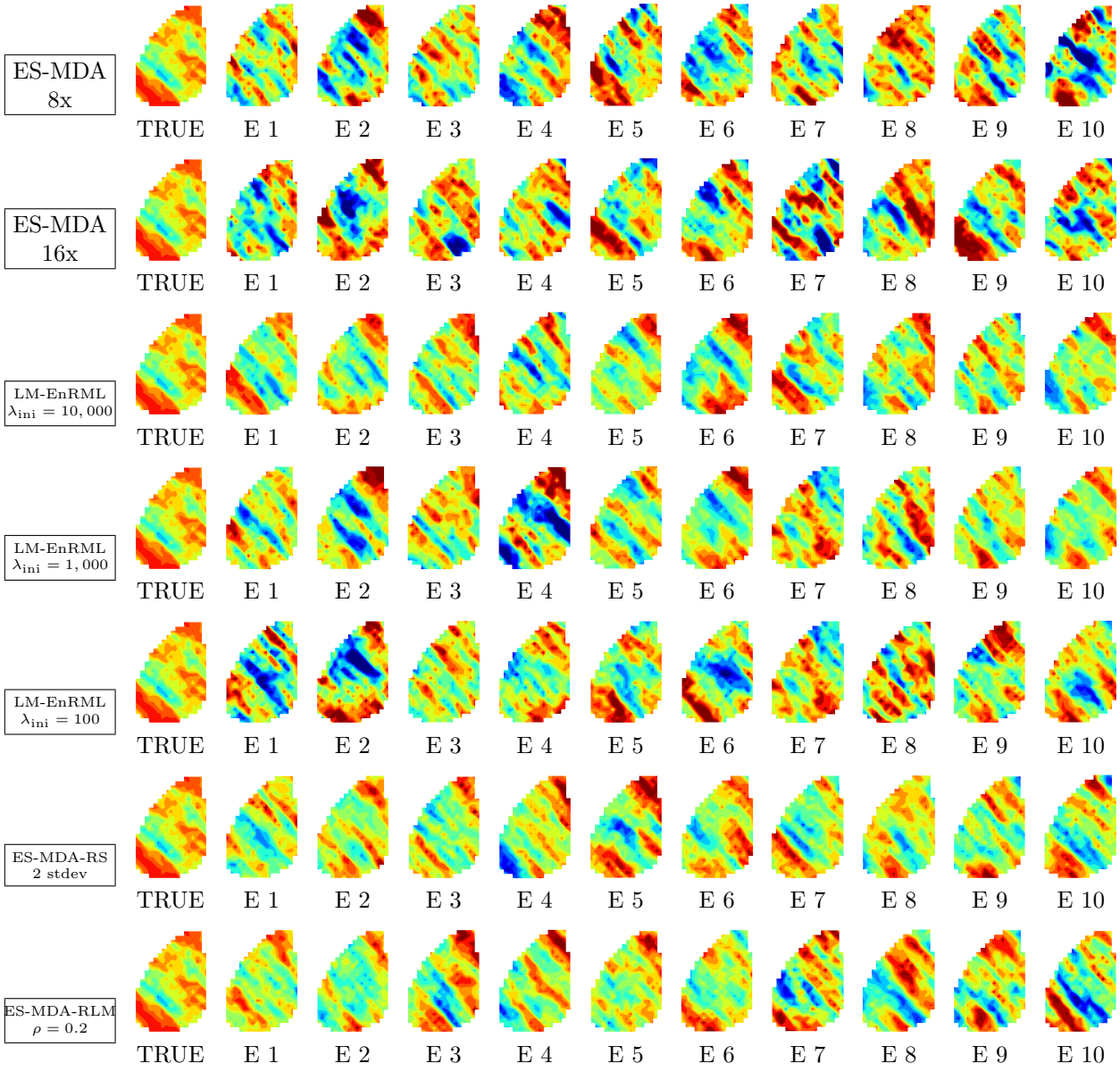


Figure 2.28: Layer 3 log horizontal permeability of the first realization of each ensemble.

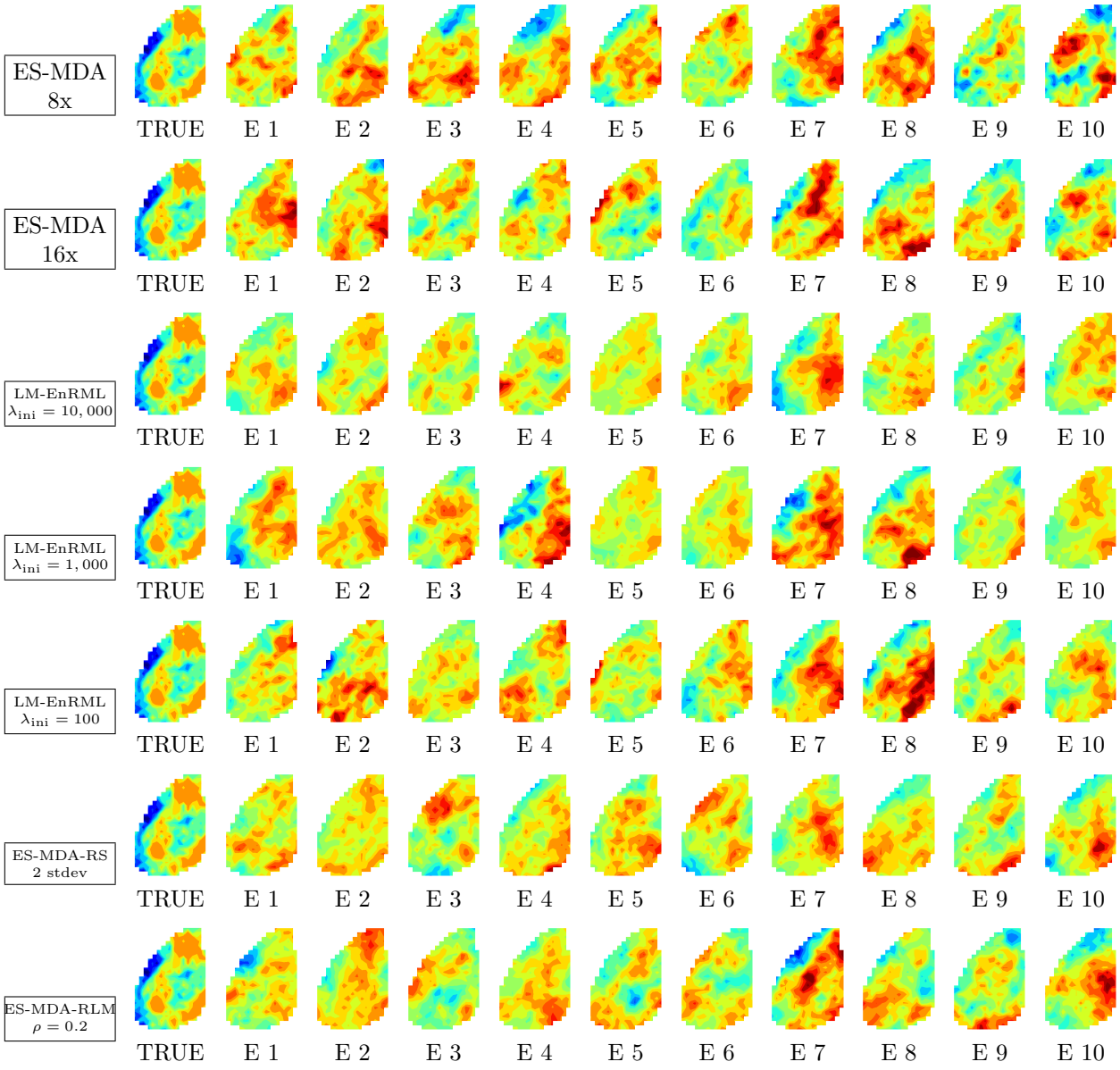


Figure 2.29: Layer 4 log horizontal permeability of the first realization of each ensemble.

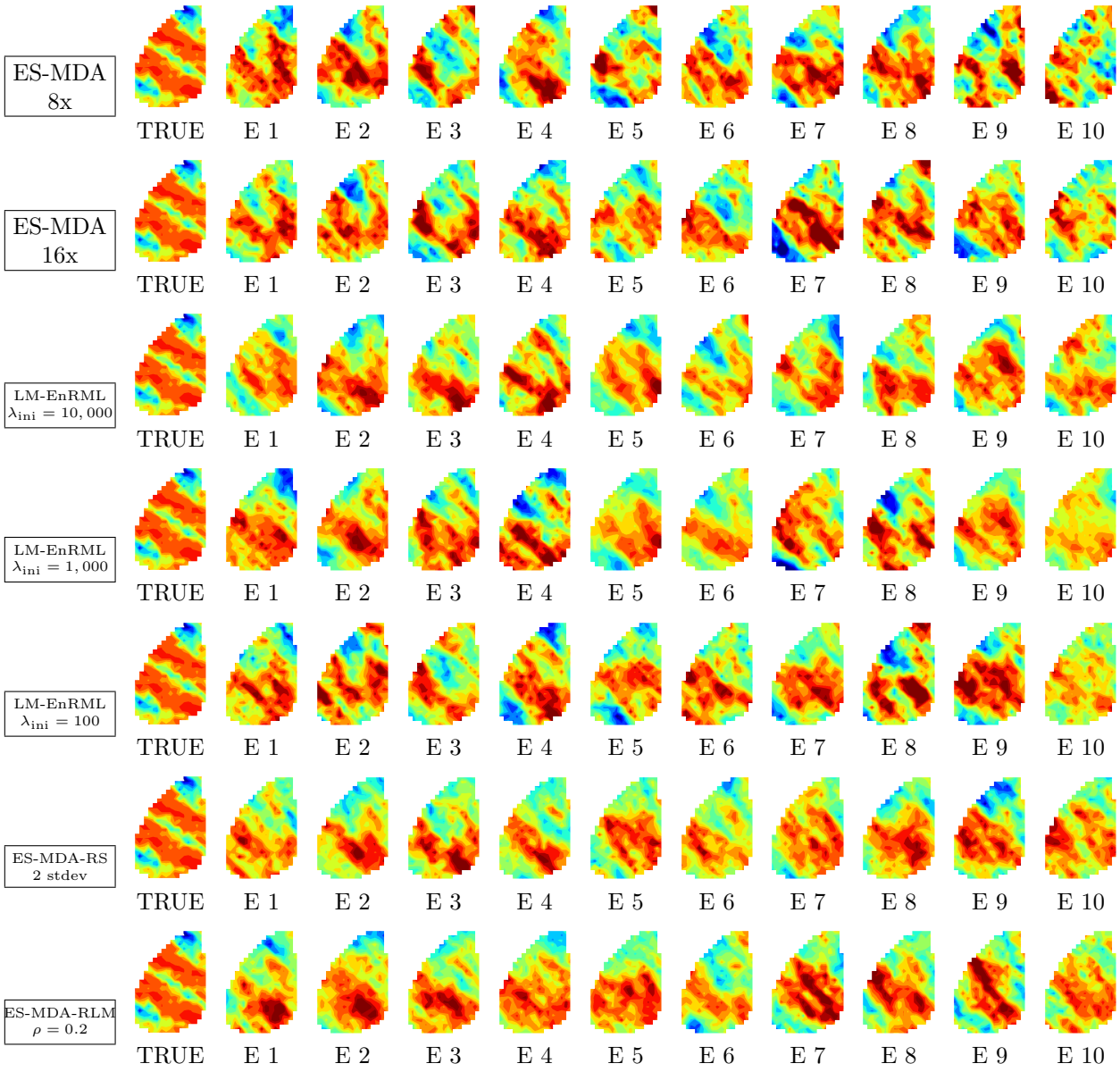


Figure 2.30: Layer 5 log horizontal permeability of the first realization of each ensemble.

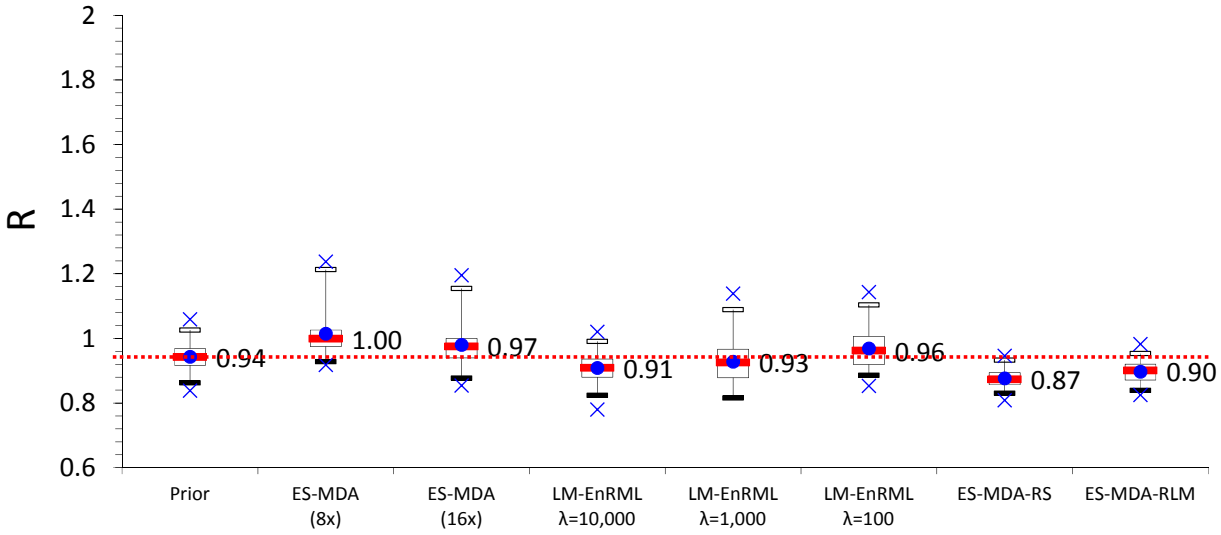


Figure 2.31: Difference between a posterior model and the true model based on Eq. 2.36

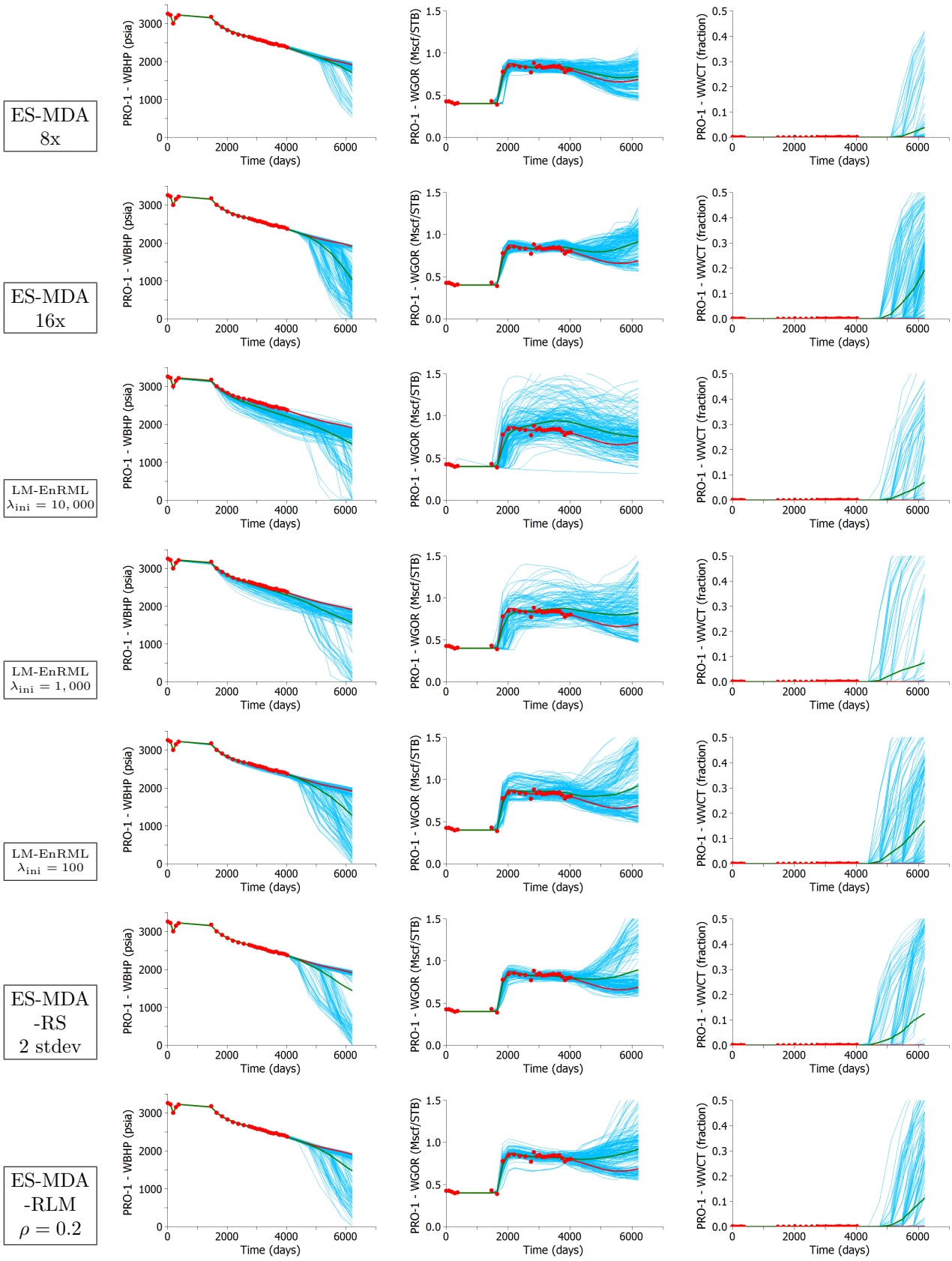


Figure 2.32: Posterior production data at PRO-1 obtained with different history-matching methods.

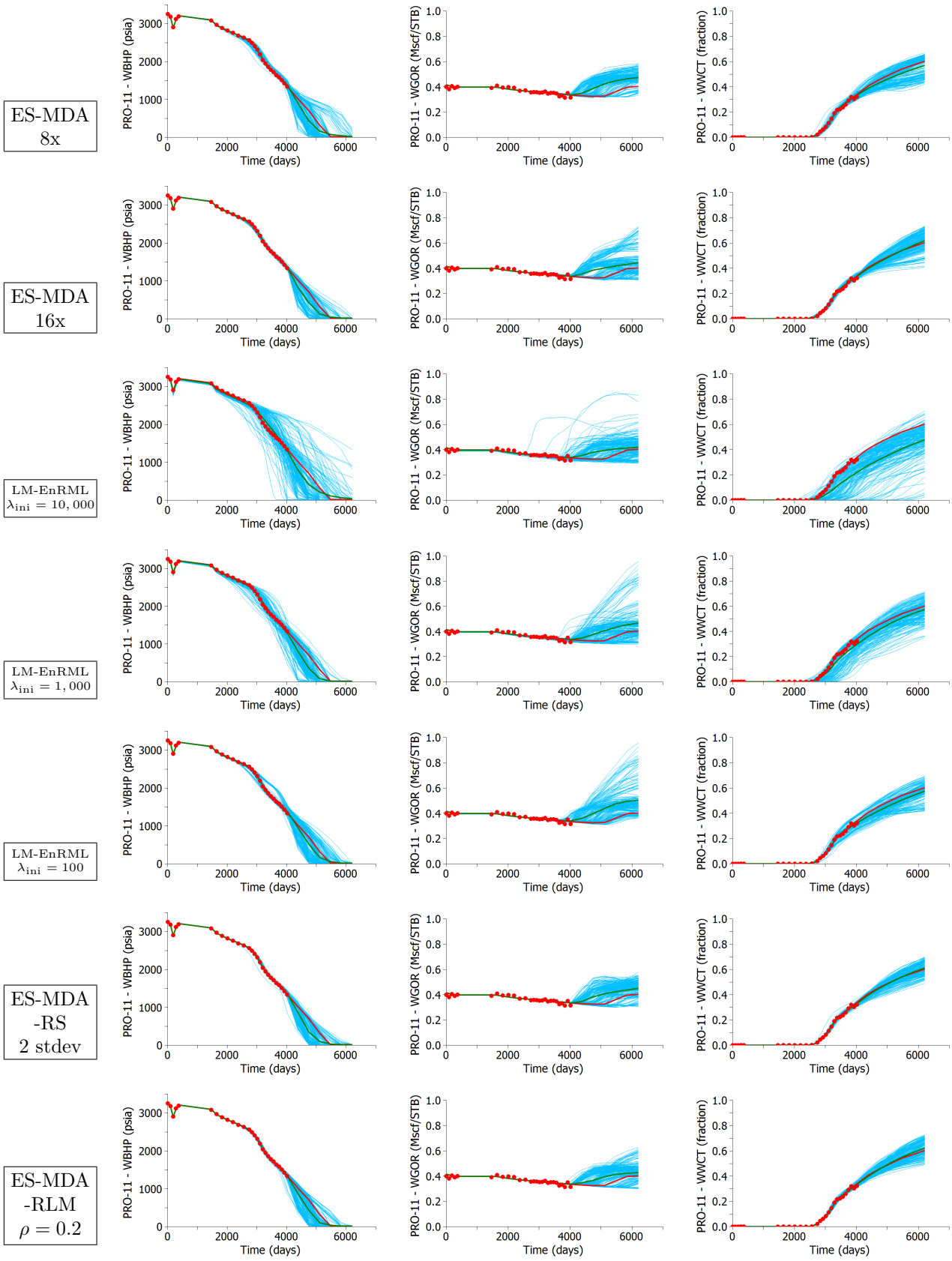


Figure 2.33: Posterior production data at PRO-11 obtained with different history-matching methods.

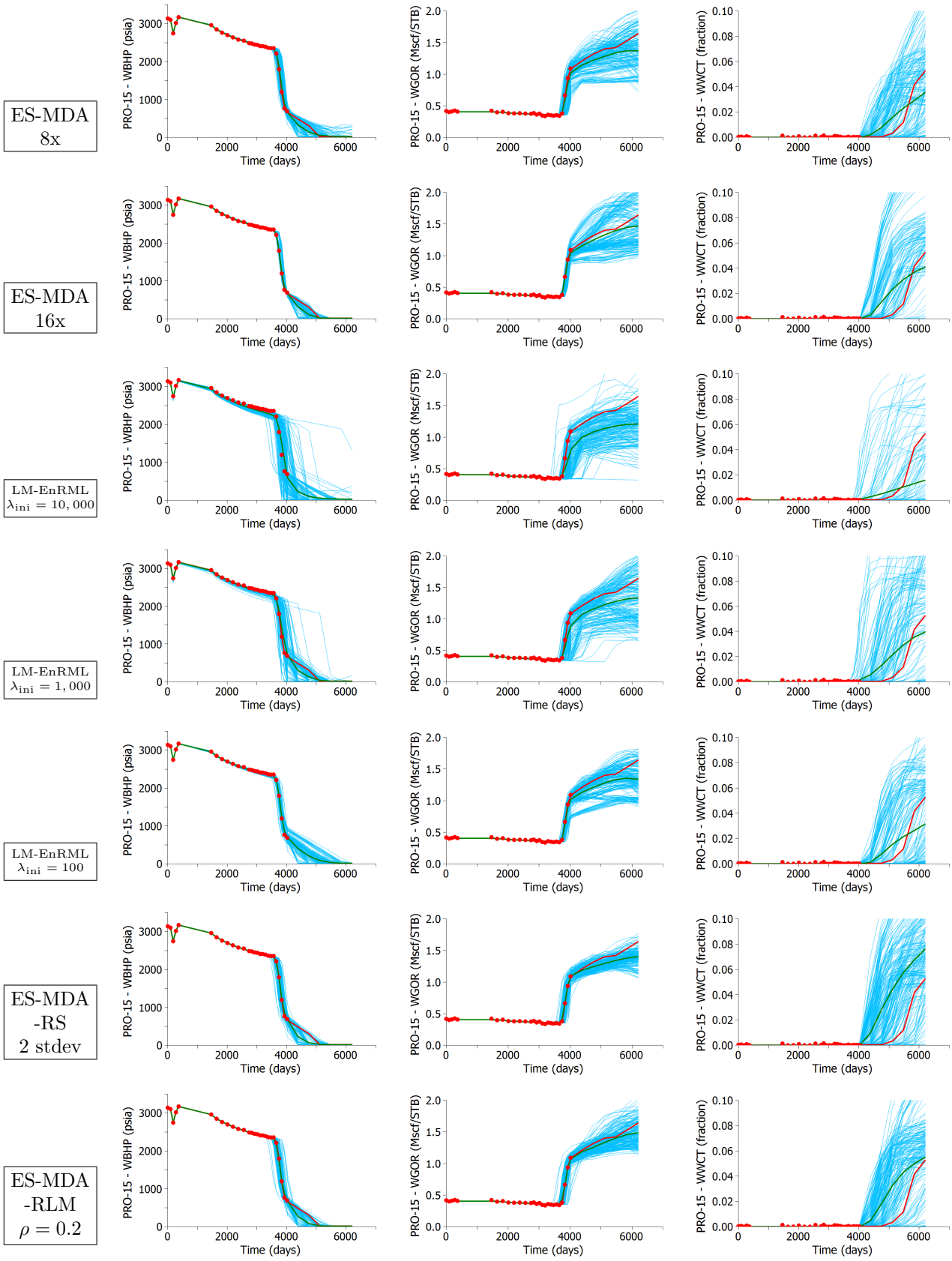


Figure 2.34: Posterior production data of well PRO-15 obtained with different history-matching methods.

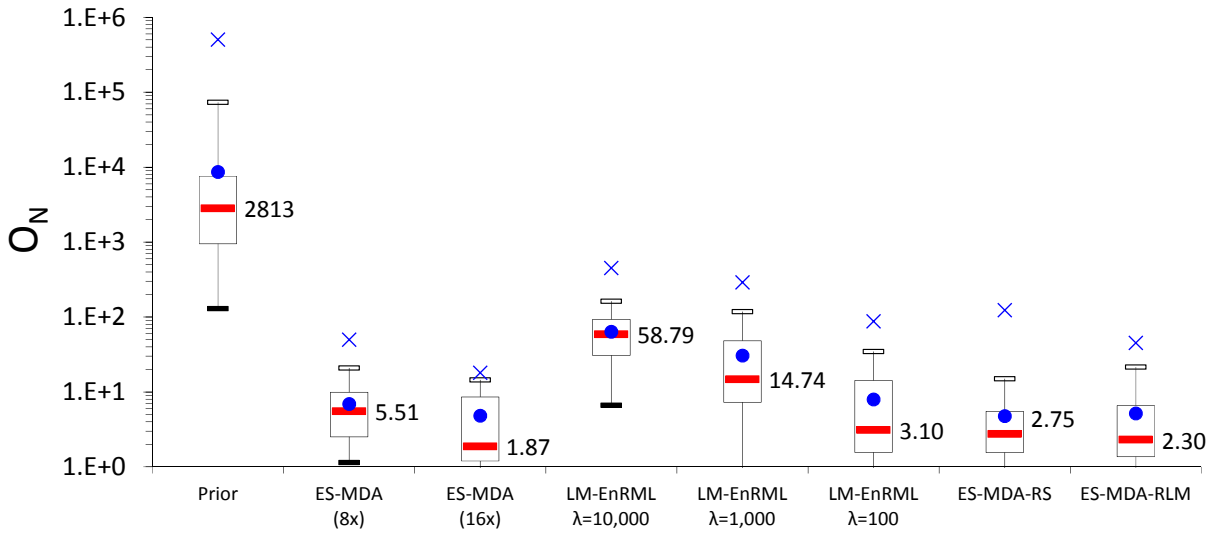


Figure 2.35: Normalized objective function.

2.4 Case Study 3

The third case is the Brugge field (Peters et al., 2010). This is a counter example where we demonstrate that sometimes the adaptive ES-MDA methods do not offer much improvement over the regular ES-MDA method. The Brugge field is a synthetic reservoir designed as a benchmark problem for evaluating the combination of different history matching and production optimization methods. The reservoir grid is $139 \times 48 \times 9$. There are 20 producers and 10 injectors in the model. The production wells are under total rate control while the injection wells are under water injection rate control. Approximately 10 years (3651 days) of production history are available, which include the oil and water rates at production wells and the bottom hole pressure at both production and injection wells.

In the original Brugge data set, there are 104 realizations of the unknown properties, which consists of porosity, horizontal and vertical permeabilities and net-to-gross ratio. The true model has not been made available. The first realization of the permeability fields are shown in Fig. 2.36. We use the first 2500 days of data for history matching, the remaining data are used to evaluate the forecast capability of our data assimilation methods. We assume that the noise levels for the oil rates and water rates, respectively, are 3% and 5% of the observed values and the noise level for the bottom hold pressure is 0.5 bars (7.25 psi).

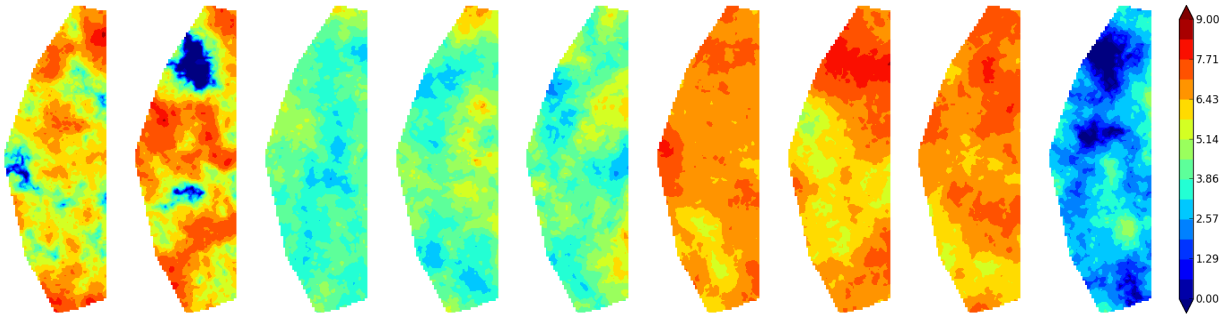


Figure 2.36: The first prior realization of the permeability field (case 3)

We assimilate data using ES-MDA 4x, ES-MDA 8x, ES-MDA-RS, ES-MDA-RLM. It takes the ES-MDA-RS 17 iterations to finish. For the ES-MDA-RLM algorithm, we use ρ equal to 0.2. For this choice of ρ, the condition of Eq. 2.19 is satisfied with the initial guess

of the inflation factor at all iterations and the data assimilation completes in 6 iterations. The first realizations of the posterior permeability fields are shown in Fig. 2.37. We can see that there seems to be some slight overshooting in the first two layers but without the true model, we can not be certain. However, the problem is not too severe and increasing the number of iterations has little effect on the problem.

The data matches obtained with four different methods are also quite similar. Fig.2.38 shows the plot of the objective function, where the values of final objective function corresponding to four methods are almost the same. Figs.2.39, 2.40, 2.41 and 2.42, respectively, show the predicted data obtained using ES-MDA 4x, ES-MDA 8x, ES-MDA-RS and ES-MDA-RLM. For each figure, the plots are organized into four rows. In the first three rows, we plot the bottom hole pressures, the oil rates and the water rates of some representative production wells. In the last row, we plot the bottom hole pressures of some representative injection wells. It can be seen that the data matches are very similar among different methods and are fairly reasonable for most wells. Only in well P-10 does the data match differs from the observed data, where the predicted oil rate is higher than the observed data and, accordingly, the predicted water rate is lower than the observed data.

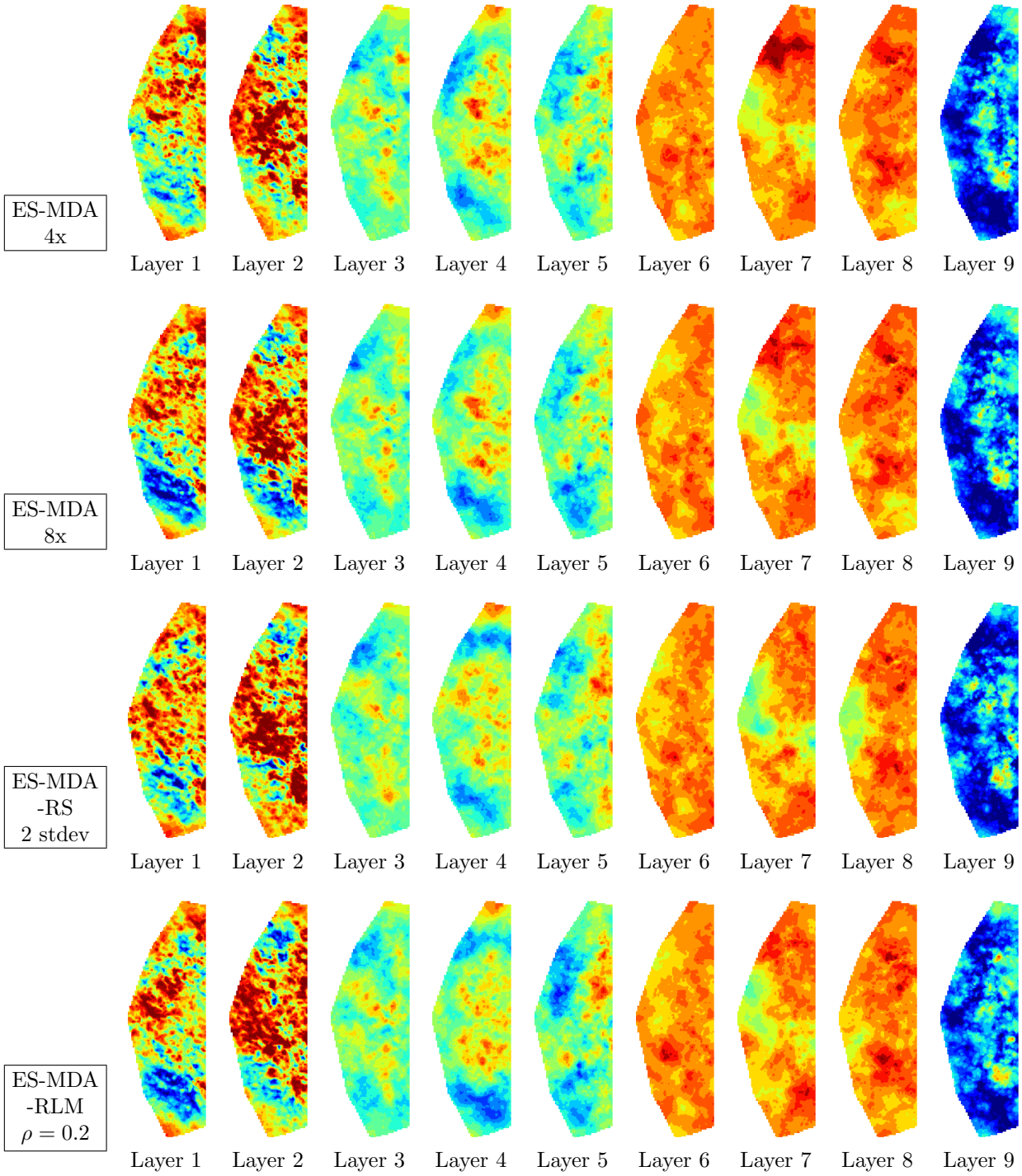


Figure 2.37: The first posterior realization of the permeability field (case 3)

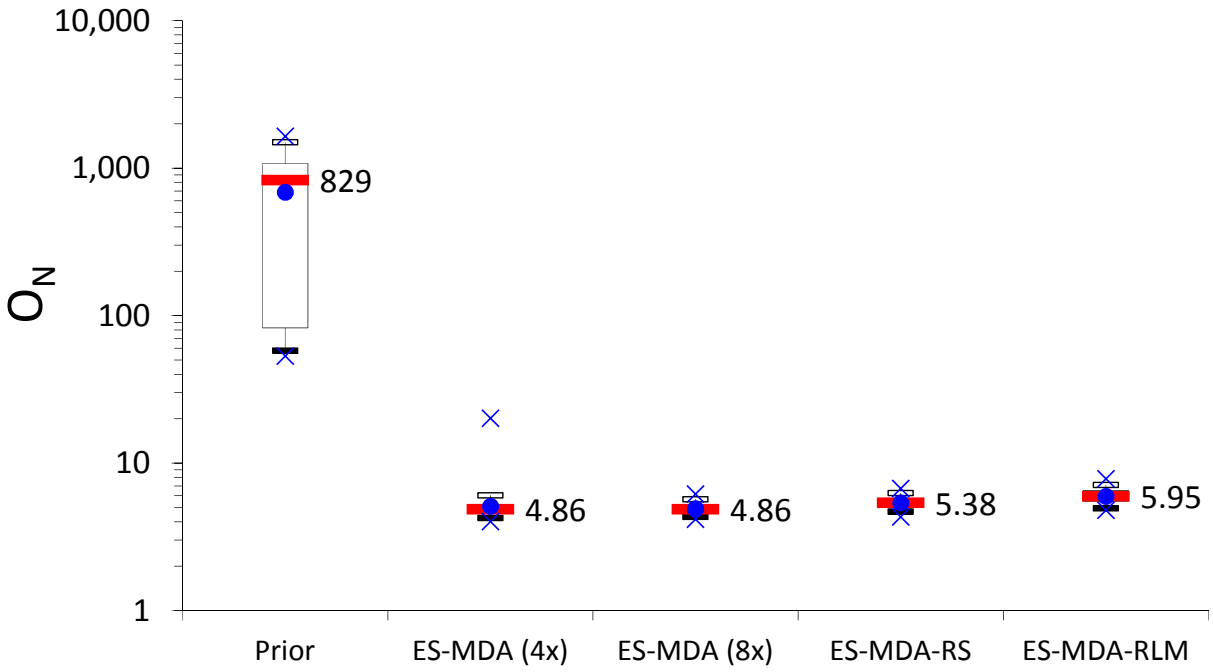


Figure 2.38: Normalized objective function (case 3).

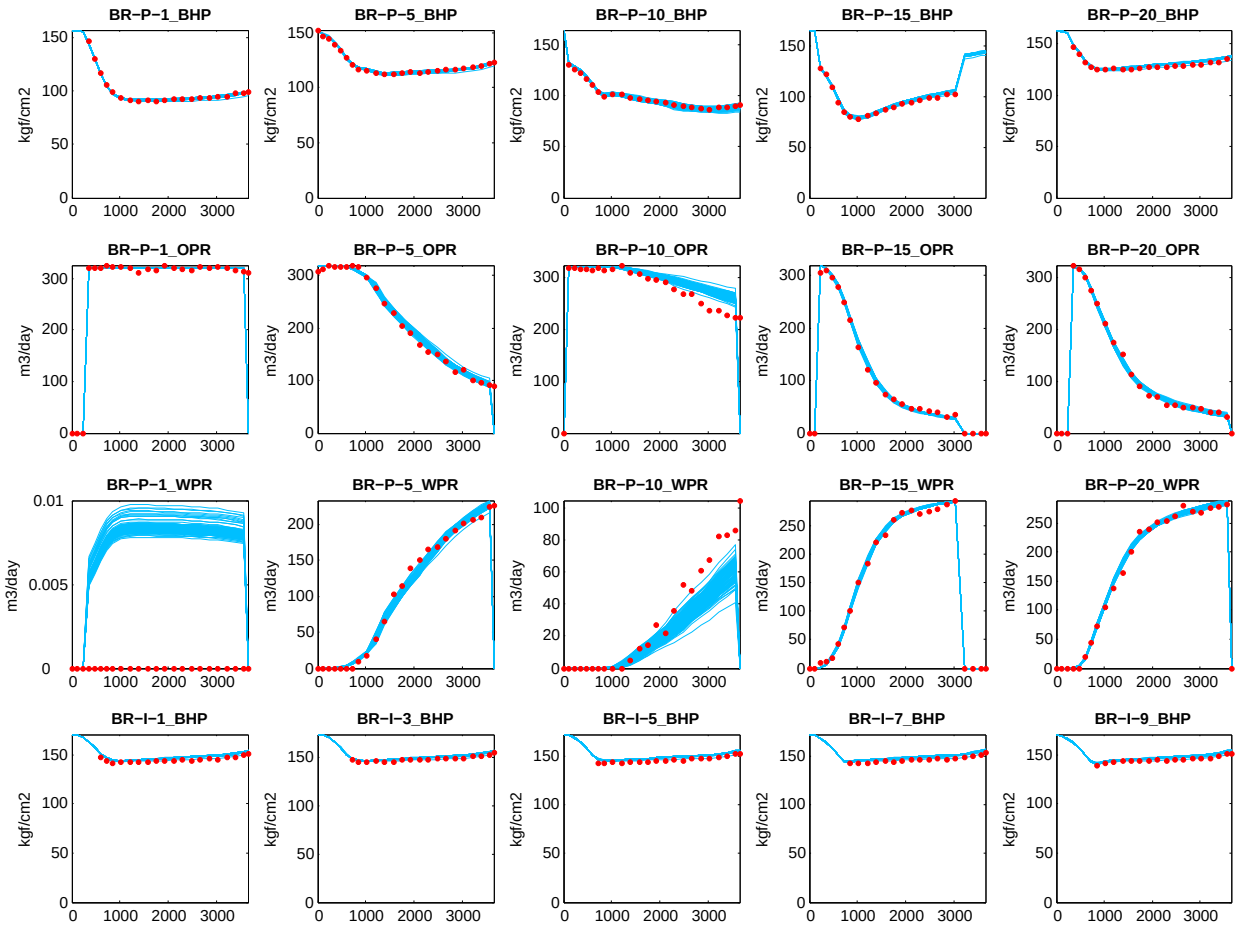


Figure 2.39: Ensembles of well data obtained with ES-MDA 4x

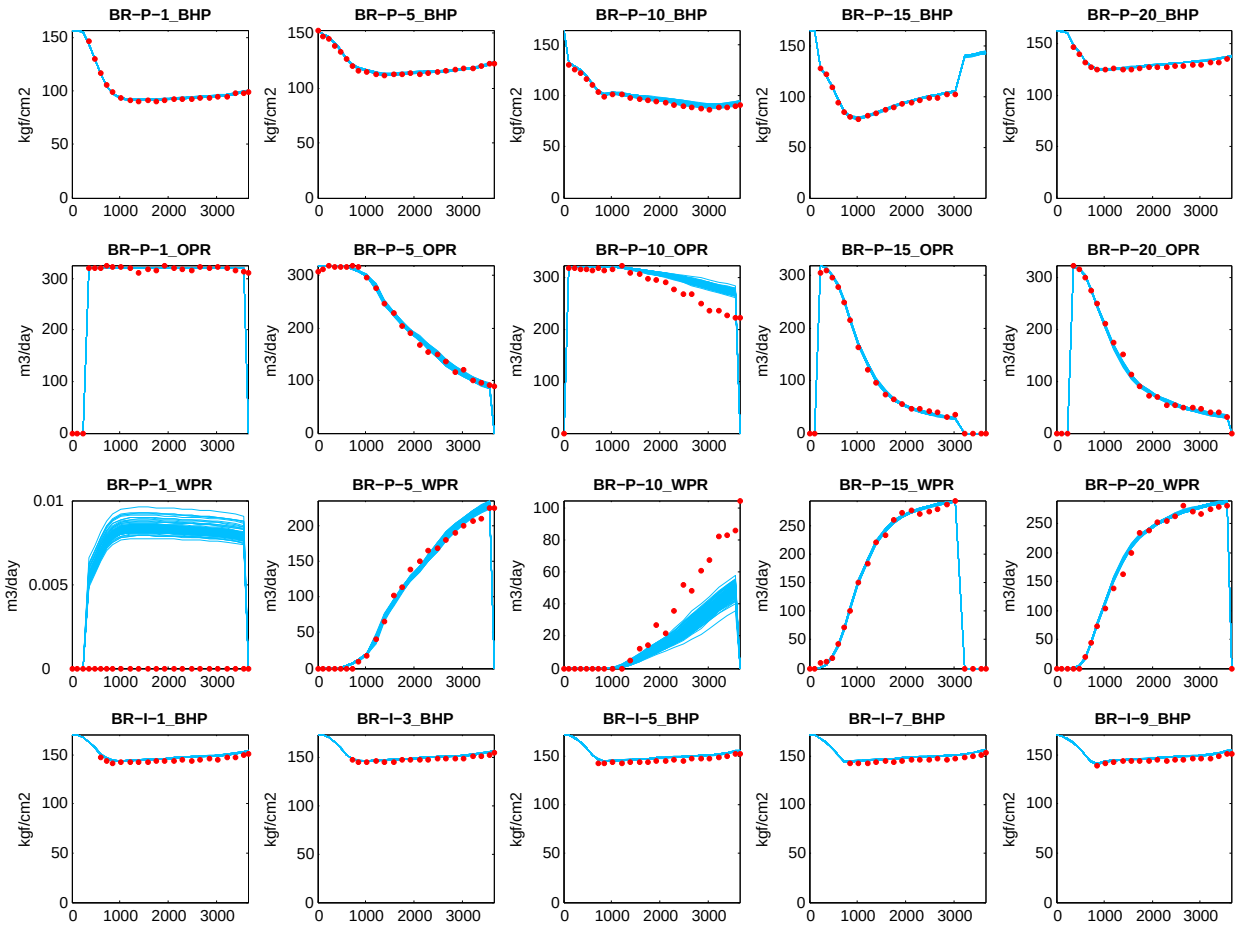


Figure 2.40: Ensembles of well data obtained with ES-MDA 8x

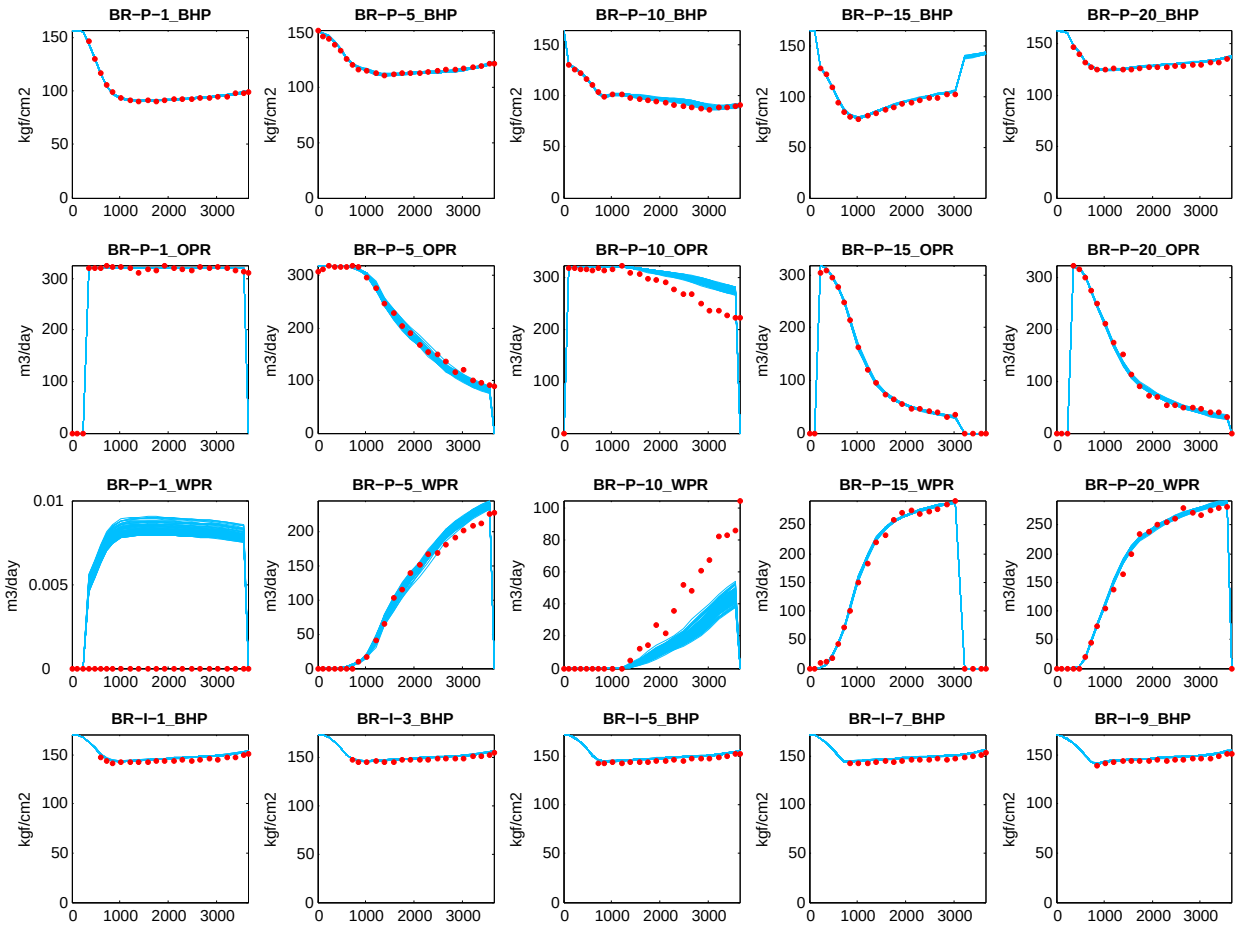


Figure 2.41: Ensembles of well data obtained with ES-MDA-RS

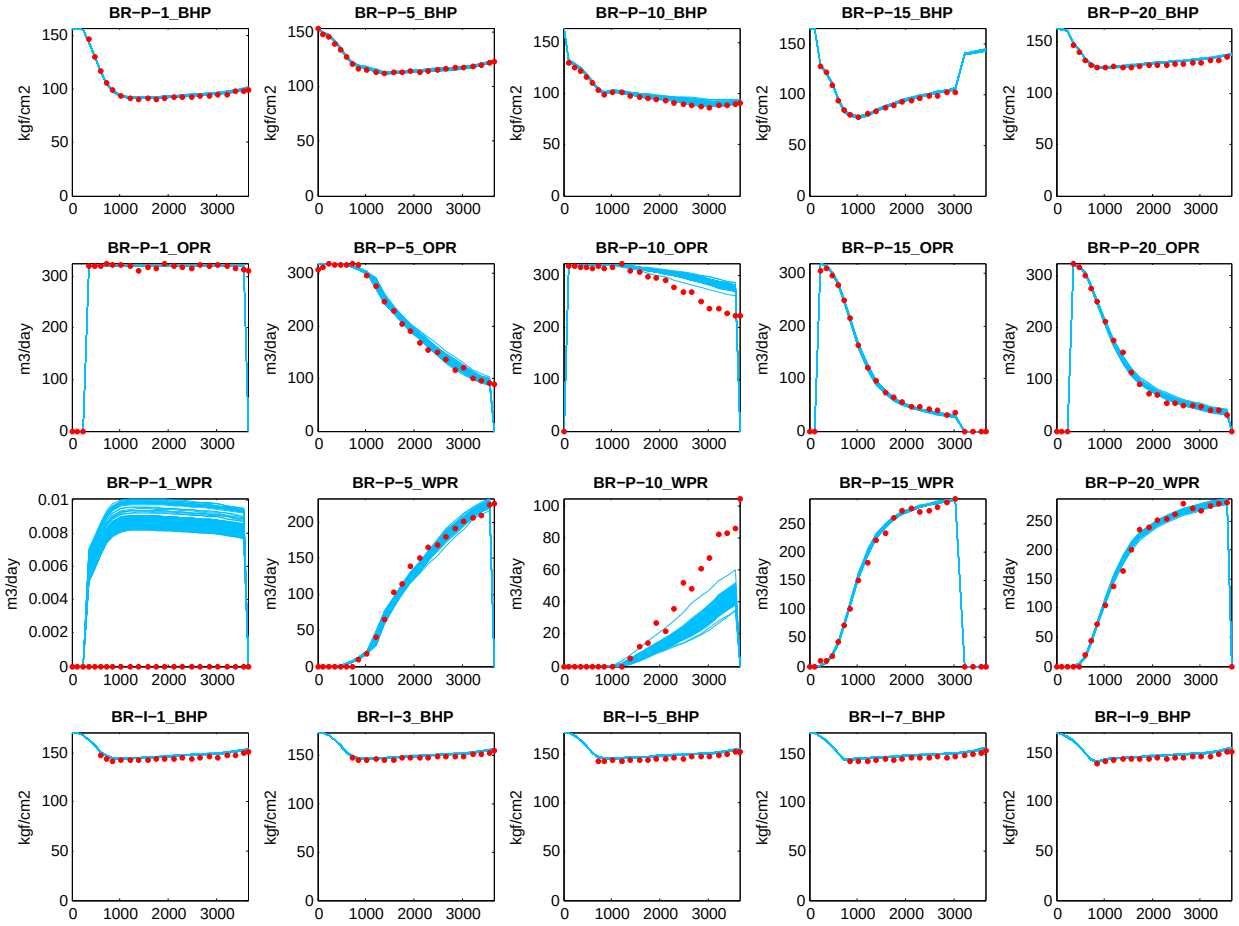


Figure 2.42: Ensembles of well data obtained with ES-MDA-RLM

CHAPTER 3

HISTORY MATCHING NON-GAUSSIAN FACIES

In this chapter, we develop a history matching procedure based on ES-MDA that can be used with multi-facies reservoir models, so that, for examples, the history matching preserves a distinct boundary between facies. Our adaptation of the ES-MDA algorithm is similar to the approach being used by Jafapour and Khodabakhsi (2011). The ES-MDA algorithm is used to modify the reservoir permeability, from which a facies probability map can be calculated and used in the MPS simulation algorithm. To illustrate our method, we design a channelized reservoir problem that is more difficult to resolve by history-matching than most cases that appear in the literature. First, the channel system is more complex with a relatively high number of channels while the number of wells remains modest at thirteen. Secondly, unlike most previous examples where the permeability within each facies is assumed to be a constant, known value, we assume significant variability for the permeability within a facies and show that obtaining a reasonable approximation of the true permeability distribution is necessary to obtain a good data match. We also try to address the problem where the prior means of the permeabilities are unknown. This chapter of the dissertation is organized as follows: in the next section, we explain the methodology. Then, we apply the proposed method to three test cases with increasing degrees of complexity.

3.1 Methodology

We consider the history-matching problem in which the reservoir model consists of M facies with π_j, $j = 1, ..., M$, being the probability of appearance of each facies; $\sum_{j=1}^{M} \pi_j = 1$. We assume that the petrophysical properties within each facies follow a Gaussian distribution. For the purpose of updating the probability for the distribution of facies, the reservoir

model for the petrophysical properties is represented by a Gaussian mixture model (GMM);

$$\Theta = (\pi_j, f_j),\ j = 1, ..., M, \tag{3.1}$$

where M is the number of facies, i.e., number of Gaussian models; f_j denotes the j-th Gaussian probability density function (pdf), and π_j denotes the probability of each facies or each Gaussian distribution, i.e. the probability that a gridblock is occupied by facies j. Note that the only petrophysical property we consider in this dissertation is permeability even though the method can be extended to include more properties.

The generation of the prior reservoir model, which consists of the facies F and the vector of gridblock permeabilities m, is done based on the following equation:

$$f(m, F) = f(m|F)f(F), \tag{3.2}$$

where throughout, f is used to denote a probability density function (pdf). We first sample the facies distribution, $f(F)$, using the MPS algorithm implemented in Petrel 2010 (Schlumberger, 2010), and then we sample the permeability distribution within each facies, $f(m|F)$, by applying the sequential Gaussian simulation algorithm. Note that this is different from sampling directly from the Gaussian mixture model.

In our examples, both the facies and gridblock permeabilities are unknown. Given the observed data D, we update the facies and permeabilities in two stages. In the first stage, we try to obtain a good approximation of the correct facies distribution. In the second stage, we focus on updating the permeabilities to obtain a good data match. Mathematically, this can be written as

$$f(m, F|D) = f(m|F, D)f(F|D), \tag{3.3}$$

where $f(F|D)$, the pdf for F conditional to the data, is sampled in stage I of the algorithm described below.

All examples in this research use the training image shown in Fig. 3.1. We first list the

steps of the procedure, then discuss in details the implications of each step. The procedure is as follows:

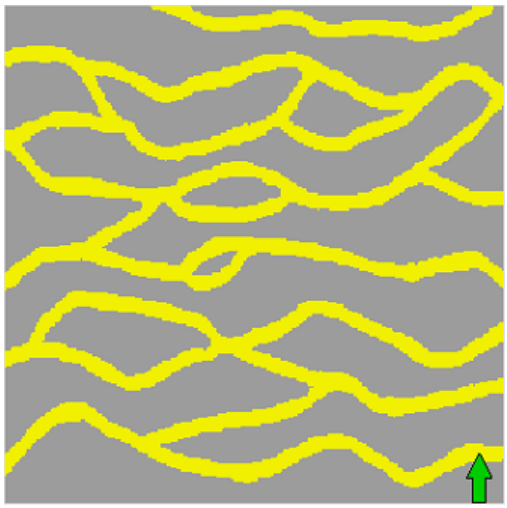


Figure 3.1: Channel training image (Caers and Zhang, 2004)

I. Stage 1

This stage consists of multiple iterations (data assimilation steps), the number of which is selected to be divisible by four. Each iteration has two steps. Step 1 is performed every iteration while step 2 is only performed every four iterations, hence the requirement that the total number of data assimilations be divisible by four.

FOR $\ell = 1, 2, ...N_a$:

1. *Step 1*: Apply the ES-MDA update equation (Eq. 2.2) to update the permeability field.
2. *Step 2* (only performed when ℓ is divisible by 4): regenerate the distribution of facies and permeability.
   (a) Apply the Neighborhood Expectation-Maximization (NEM) algorithm (Han et al., 2011) to assign each gridblock to the most likely facies based on the updated permeability from step 1.
   (b) Calculate the average probability that a facies appears at each gridblock by averaging the facies distributions obtained in step 2(a) across the whole ensemble.

(c) Use the probability map in combination with the MPS algorithm to generate new facies distributions. There are two possibilities:

- ℓ is not the last iteration. In this case, we directly incorporate the probability map into the MPS algorithm using the Tau model (Journel, 2002; Zhang and Journel, 2003).
- ℓ is the last iteration. In this case, we "enhance" the probability map before incorporating it into the MPS algorithm. See the discussion below for details.

(d) Populate the facies distribution from step 2(c) with random permeability values generated using sequential Gaussian simulation with the means and variances of the two prior Gaussian distributions for channel and non-channel permeability field.

At the end of stage I, we have generated an approximate sampling of $f(F|D)$.

II. Stage 2

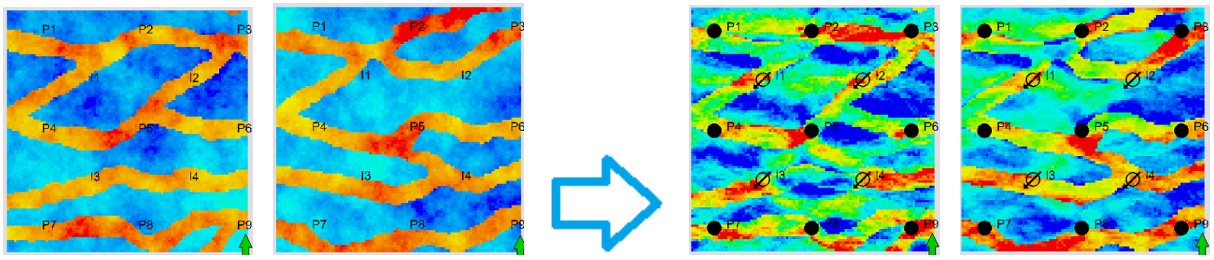
In this stage, we apply a regular ES-MDA data assimilation procedure to the ensemble of realizations obtained at the end of stage 1. As required by ES-MDA, the number of iterations and the inflation factors have to be defined before the data assimilation starts, and the sum of the inverse inflation factors has to be equal to unity. At the end of this stage, we have generated an approximate sampling of the joint distribution, $f(m, F|D) = f(m|F, D)f(F|D)$.

3.1.1 Stage 1

Step 1: In stage 1, any inflation factors can be used in the ES-MDA update equation; they can either be the same or vary between ES-MDA iterations and they do not have to follow any rule. In addition, the users do not need to specify the number of iterations at the beginning but can perform as many as needed, as long as it is divisible by four. As discussed later, this does not impact our sampling correctness if our procedure is applied to a linear-Gaussian problem. In our implementation, we simply make the inflation factors

equal to the number of iterations we intend to use. For example, if we anticipate stage 1 would take 24 iterations to complete, we would use the inflation factors equal to 24 for all iterations.

Step 2: After the application of ES-MDA in step 1, the facies boundaries become smeared, i.e., the boundary between facies will become less clear due to the introduction of permeability values that are highly improbable based on the prior Gaussian mixture model of Eq. 3.1. Fig. 3.2 illustrates how the facies boundaries may become smeared after one iteration (one data assimilation step) of ES-MDA.



(a) Realz 1 (prior) (b) Realz 2 (prior) (d) Realz 1 (after ES-MDA) (e) Realz 2 (after ES-MDA)

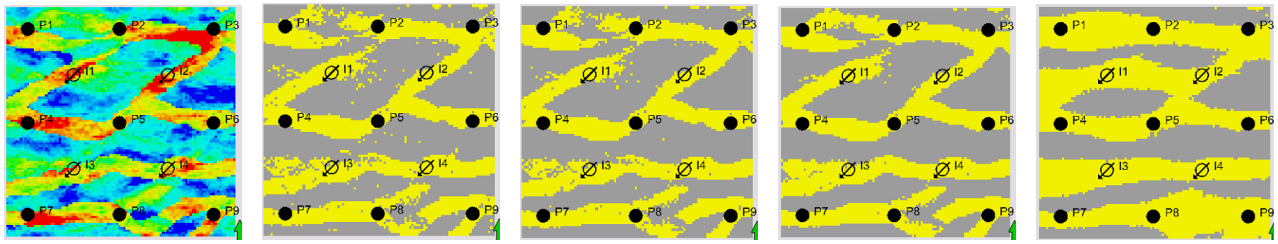
Figure 3.2: Facies boundary become smeared after one data assimilation step with ES-MDA.

After several iterations, the facies boundaries become so badly smeared that new realizations have to be generated to keep the realizations consistent with the geostatistical features prescribed by the training image. This step is similar to what is done in Jafapour and Khodabakhsi (2011) except that Jafapour and Khodabakhsi (2011) used EnKF instead of ES-MDA to update the permeability. We recognize from our experience that we do not have to regenerate the facies distribution at every iteration of ES-MDA. For the examples considered in this paper, regenerating the facies every four ES-MDA iterations seem to be the optimal choice. Performing step 2 too often does not allow ES-MDA enough time to remove the incorrect large scale continuities in the prior realizations and create new continuities that reflect the true model. On the other hand, waiting longer than four iterations to perform step 2 generally leads to a longer history matching process where more iterations are needed to obtain a similar result. It is possible that a different choice of the inflation factors in

step 1 can allow ES-MDA to update the model faster, thus allowing step 2 to be performed more regularly and reducing the number of iterations needed but at the moment, we have not investigated this issue. The detailed explanation of each sub-step is as follows:

- *Step 2(a)*: The Neighborhood Expectation-Maximization (NEM) algorithm Han et al. (2011) is an extension of the Expectation-Maximization (EM) algorithm (Hartley, 1958; Dempster et al., 1977; Meng, 1993, 1994; Kung et al., 2004; Zhao et al., 2007); see Appendix B.1 for a quick summary of how the NEM algorithm is implemented. The EM algorithm iteratively calculates the probability of each gridblock belonging to each facies while updating the mean, covariance and proportion of each Gaussian distribution. The output of the algorithm is the facies classification of each gridblock and the updated mean, covariance and proportion of each Gaussian distribution. We, however, only make use of the facies classification to calculate the probability map in step 2(b). The updated mean, variance and proportion are usually inconsistent with the known values or misleading (Le et al., 2015) so we choose to keep the prior values instead. This is appropriate because in stage 1, we concentrate on obtaining a reasonably correct sampling of the facies distribution, sampling the permeability will be the focus of stage 2. The improvement of the NEM algorithm over the standard EM algorithm is that the user can define a neighborhood around each gridblock to promote facies continuity in the vicinity of that gridblock. This is particularly useful because the training image in our research has its major continuity direction in the east-west direction, so it is better that we define a non-square neighborhood instead of a square one. Fig. 3.3 shows the facies classifications using different neighborhood sizes for the same permeability field. It can be seen that a neighborhood size of 9×3 seems to eliminate most of the stray pixels and promote more continuity in the East-West direction. This improvement will later translate to further improvement in the probability map in step 2(b).

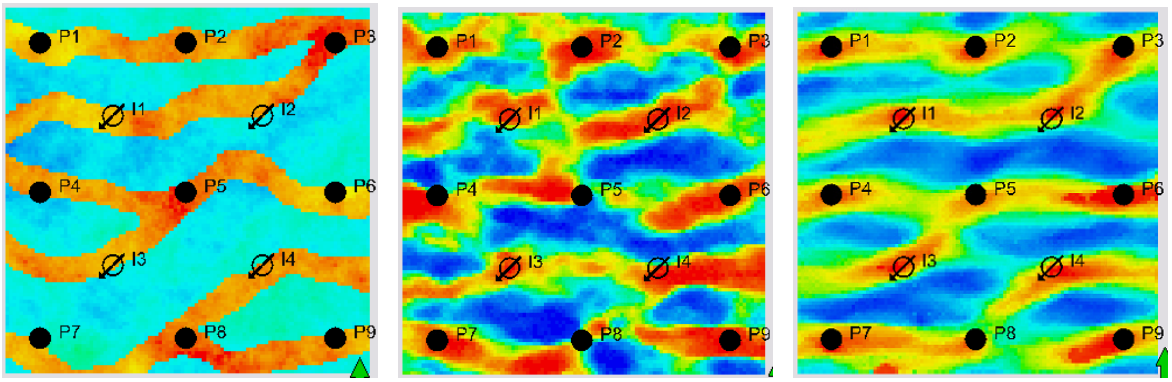
- *Step 2(b)*: From step 2(a), we have N_e facies fields, where N_e is the ensemble size. In this step, we assign a value of 0 to shale gridblock and a value of 1 to gridblocks occupied



(a) Permeability (b) Neighbor 3x3 (c) Neighbor 5x3 (d) Neighbor 9x3 (e) Neighbor 21x3

Figure 3.3: Facies classification using different neighborhood sizes.

by channel and calculate the average of the N_e facies fields. The result is a probability map that dictates the probability that a gridblock is occupied by the channel facies as well as the probability that a gridblock is filled with shale. Fig. 3.4 compares a probability map obtained with this procedure using two different neighborhood sizes. The true model is plotted in the same figure for the reader's convenience. It can be easily seen that the quality of the probability map obtained with a 9×3 neighborhood is superior to the one obtained with a 3×3 neighborhood.

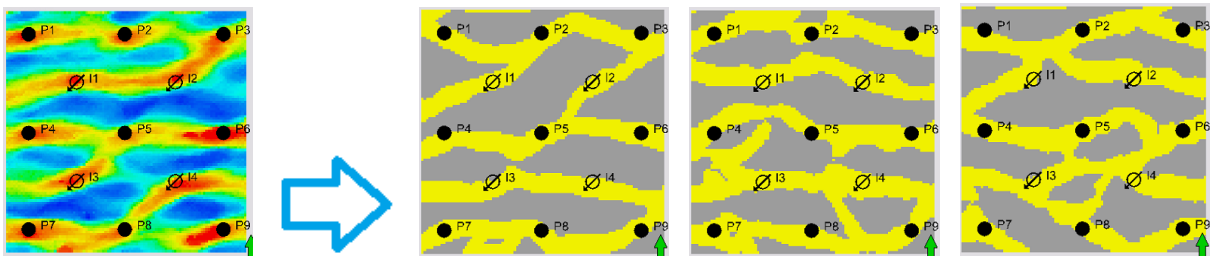


(a) True model (b) Neighbor 3x3 (c) Neighbor 9x3

Figure 3.4: Probability map obtained using different neighborhood sizes.

- *Step 2(c)*: For iterations that are not the last iteration, we directly incorporate the probability map (soft data) into the MPS simulation algorithm using the Tau model (Journel, 2002; Zhang and Journel, 2003) similar to Jafapour and Khodabakhsi (2011); see Appendix B.2 for theoretical details of the procedure. From our experience, the influence of the probability map on the simulation result is not too strong, except when

the probability is very close to 0 or 1. Fig. 3.5 shows three different facies realizations generated using the same probability map.



(a) Probability map (c) Realization 1 (d) Realization 2 (e) Realization 3

Figure 3.5: Generation of new facies distribution using the probability map.

From Fig. 3.5, we can see that even though we can perceive clear channel connections from the probability map, the realizations do not quite reproduce those connections. This is acceptable and probably desired for early iterations where we want to explore different channel configurations. However, for the last iteration, we need a well-defined facies distribution that is going to be used in stage 2, so some modifications are needed to make the MPS algorithm better honor the probability map. We have experimented with several ways of doing so and these procedures are discussed below:

– The least intrusive method is to directly use the probability map in the MPS algorithm as in early iterations, but apply a rejection algorithm to keep only the realizations that are consistent with the probability map. To do so, we can apply an algorithm in computer vision that detects and labels connected regions in a binary image. The algorithm is generally referred to as Connected Component Labeling (Samet and Tamminen, 1988). Fig. 3.6 schematically illustrates how this algorithm works. In this figure, there are two foreground black bodies (e.g. sand channels) on a white background (e.g. shale). The algorithm is able to detect and assign label 1 to grid cells that make up the first body and assign label 2 to grid cells that make up the second body. In the history-matching context, gridblocks of the same sand channel would receive the same label while

gridblocks of different channels would receive different labels. By checking if two wells are located in two gridblocks of the same label, we are able to determine if they are connected by a sand channel or not; see Fig. 3.7. The Connected Component Labeling algorithm is applied to both the probability map and the facies realizations to make sure that the well connections are consistent. Note that, for the probability map, we have to convert from the continuous image to a binary image; gridblocks with sand probability above some certain threshold become foreground; everything else is defined as background. There are many different ways to implement this algorithm. We do not implement it ourself but use an open source implementation based on a contour tracing technique (Chang et al., 2004). In our experience, this method only works when the probability map is very clear, i.e., when most area (gridblocks) having a 100% probability of being sand or a 100% probability of being shale; otherwise, most of the realization generated by MPS would be rejected and it would take a long time to obtain enough realizations. As will be shown later, the probability maps obtained in our examples are not clear enough, so this method is actually not used in our examples.

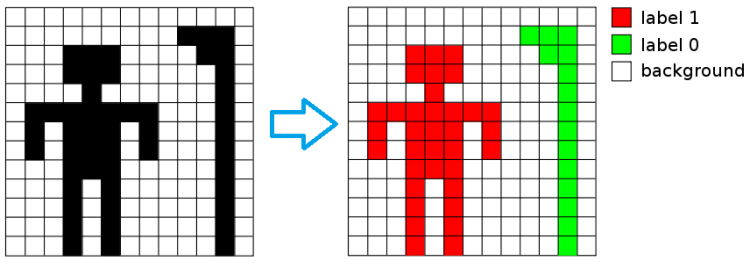
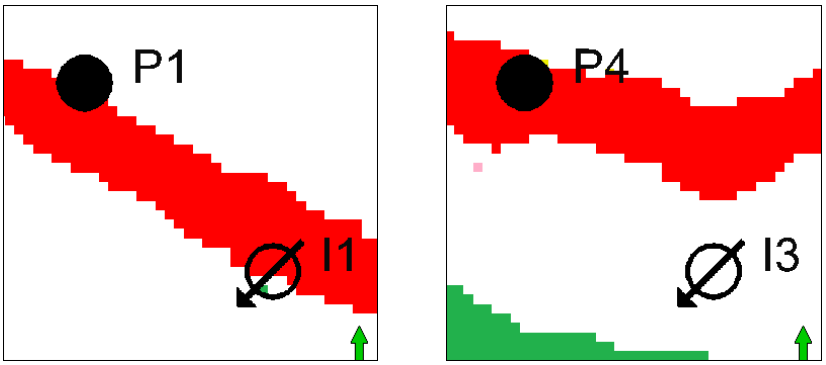


Figure 3.6: Connected component labeling illustration, reproduced from Wikipedia.

– A more intrusive way to enhance the probability map is to convert it into a hard data map and use the hard data map in the MPS algorithm instead. Here, by hard data, we mean specifying the facies of certain gridblocks. We can do so by using a threshold to separate the probability map into sand, shale and undefined regions. Fig. 3.8 shows the hard data generated from the same probability map

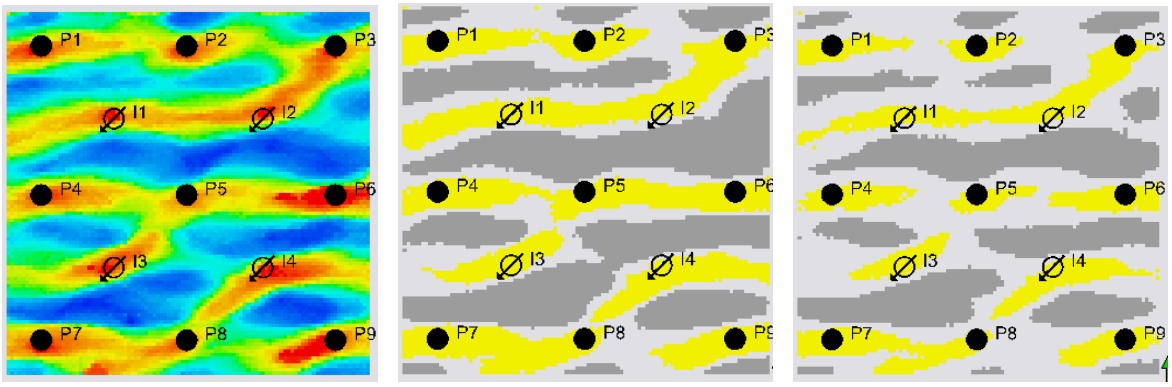


(a) Connected (b) Unconnected

Figure 3.7: Connected component labeling algorithm applied to detect well connectivity.

using different truncation thresholds. A 70% threshold means that gridblocks with values of 0.7 or higher on the probability map are assigned to sand while gridblocks with values of 0.3 or lower on the probability map are assigned to shale. Note that the white region corresponds to the remaining gridblocks where the facies is not assigned to either sand or shale. Which facies will occupy these gridblocks are determined by the MPS algorithm. As we can see from Fig. 3.8b, for a smaller threshold value, there is not much freedom left for the MPS algorithm and the realizations are dictated primarily by the hard data map. When we increase the threshold, fewer hard data will be generated, so we give more freedom to the MPS algorithm but the connections between wells may be less well-honored. For example, the connection between wells P5 and P6 in Fig. 3.8c is broken while they are connected by a continuous sand streak in Fig. 3.8b. Therefore, one disadvantage of using this approach is that we have to choose a reasonable threshold value so that there are not too many hard data but the connections between wells are still captured.

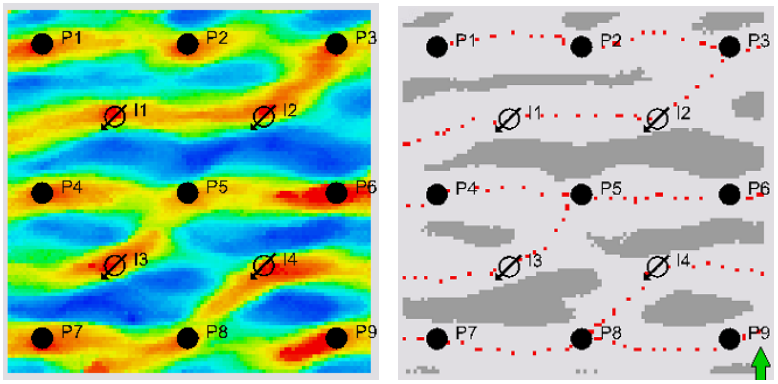
– Perhaps, a better method of converting the probability map into a hard data map is to sparsely draw the sand facies between wells based on the connections we perceive from the probability map. This is done manually at the moment but for practical application it will be necessary to implement an automatic algorithm.



(a) Probability map (b) Hard data map using 70% threshold (c) Hard data map using 80% threshold

Figure 3.8: Hard data map generated using different truncation threshold.

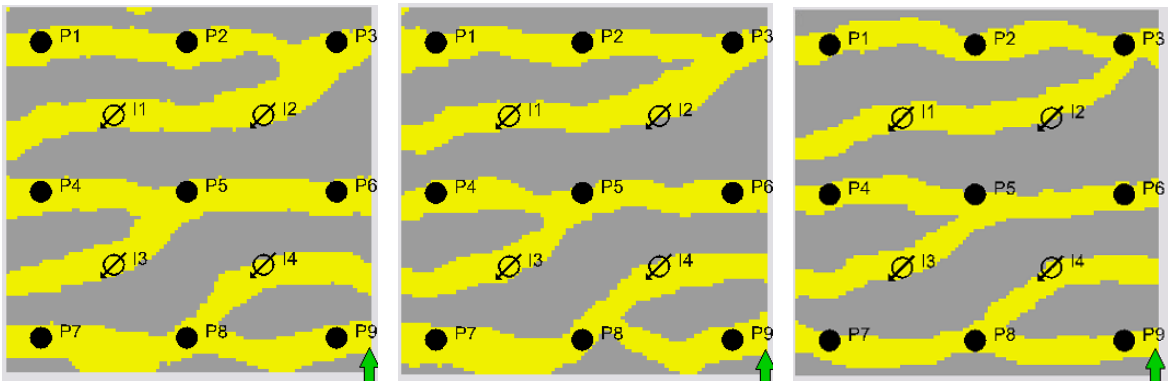
If we use this approach instead of the threshold approach, the resulting hard data map would be similar to Fig. 3.9. Note that in Fig. 3.9b, the connections representing the continuity of the channel facies between wells are colored red instead of yellow. Fig. 3.10 compares the realizations generated from the hard data maps obtained using different approaches. It can be seen that using the hard data obtained with the sparse drawing method in MPS results in realizations that are slightly more consistent with the training image (Fig. 3.1) than those obtained using the threshold method in terms of the channel width.



(a) Probability map (b) Hard data map obtained by sparse drawing

Figure 3.9: Hard data map generated using different truncation threshold.

- *Step 2(d)*: This step is done using sequential Gaussian simulation conditioned to the facies distributions generated in step 2(c). As discussed in step 2(a), we do not use



(a) 70% threshold (b) 80% threshold (c) Sparse drawing

Figure 3.10: Random realizations of the facies distribution generated using the hard data map generated using various methods.

the updated means and covariances obtained from EM but simply use the prior means and covariances to generate new permeability field.

3.1.2 Stage 2

In this research, we use ES-MDA 8x with decreasing inflation factors (1000, 100, 50, 20, 7, 5, 4, 3). Note that the the inflation factors are chosen so that their inverses sum to unity.

3.1.3 Notes on Sampling Correctness in the Linear Gaussian Case

Even though we develop the procedure in this chapter specifically to history match models with multiple facies, there are certain situations when the problem may revert to a linear-Gaussian problem or become almost equivalent to one. These situations include the following:

- There is only one facies.
- The Gaussian distributions within each facies are very similar and there is little correlation between gridblocks. The second requirement is necessary to avoid discontinuities in the correlation between gridblocks.

Because these situations are unpredictable, it is important to consider whether our procedure

is guaranteed to sample correctly should the problem revert to a linear-Gaussian one. In stage 1 of our procedure, after every 4 iterations, we generate new facies and permeability realizations based on the prior distributions. This it is equivalent to generating a new prior ensemble when applied to a linear-Gaussian problem. Note that, even though we indirectly update the facies pdf by using the probability map in the MPS algorithm, the updated facies pdf does not affect the newly generated permeability field when there is only one facies or the distributions of permeability within each facies are identical. Because the number of iterations in stage 1 has to be divisible by four, stage 1 is concluded with a newly generated ensemble. Then, what we use for the inflation factors in stage 1 does not affect the sampling correctness. Stage 2 is where the inflation factors matter and our procedure requires that the inverse inflation factors in stage 2 have to sum to unity, which guarantees that our method samples correctly in the linear Gaussian case.

3.2 Case Study 1

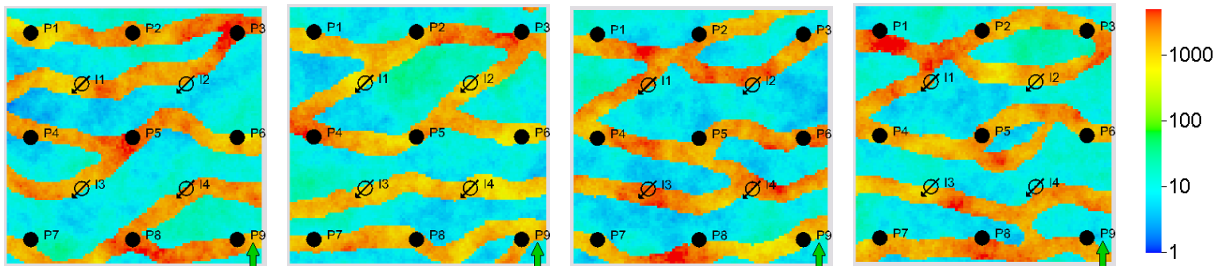
We define a 2D water-flooding problem with the following properties:

- $100 \times 100 \times 1$ gridblocks, each of which is of size 100 ft $\times$ 100 ft $\times$ 50 ft.
- Constant porosity of 0.20.
- The model consists of two facies, "shale" and "sand," in a channelized reservoir. The facies at well locations are assumed to be all sand.
- The means and covariances of the permeability distributions used to generate the prior realizations are the same as those used to generate the true model
- The permeability within each facies follows a log-normal distribution with the standard deviations equal to 0.5 in log domain, i.e., the mean of $\ln k = 0.5$, for both facies. The geometric means of the permeability distributions are equal to 10 md and 2000 md for the shale and sand facies, respectively. This is the same as saying the means of the log permeability are equal to 2.3 and 7.6, respectively. Note that all history matching

operations on the permeability are done in the log domain even though we report permeability values in the regular domain for the reader's convenience. The variogram is defined with an exponential covariance function, and the major and minor correlation lengths both are equal to 5000 ft.

- Four injectors and nine producers arranged in a 5-spot pattern, all under constant pressure control.
- Observed data consist of the water injection rates at the injectors and the oil and water production rates at the producers.
- The history-matching period is 20 years, followed by a 10-year forecast period.

The true model and three unconditional realizations are shown in Fig. 3.11. The corresponding prior ensembles of predicted data are shown in Fig. 3.12.



(a) True model (b) Prior realization 1 (c) Prior realization 2 (d) Prior realization 3

Figure 3.11: True model and three prior realizations (case 1)

The proposed procedure is applied using an ensemble of size 400 to attempt to minimize sampling error and the need of covariance localization. The number of iterations in stage 1 is 24 with inflation factors equal to 24 at all data assimilation steps. The objective function evolution and the probability map evolution in stage 1 are shown in Figs. 3.13 and 3.14, respectively. Looking at the objective function evolution, we realize that the objective function goes down each time we apply ES-MDA, but jumps up to the initial value when we generate new facies distributions and populate them with permeability values sampled from the prior distributions. The main reason for this behavior is that the probability maps

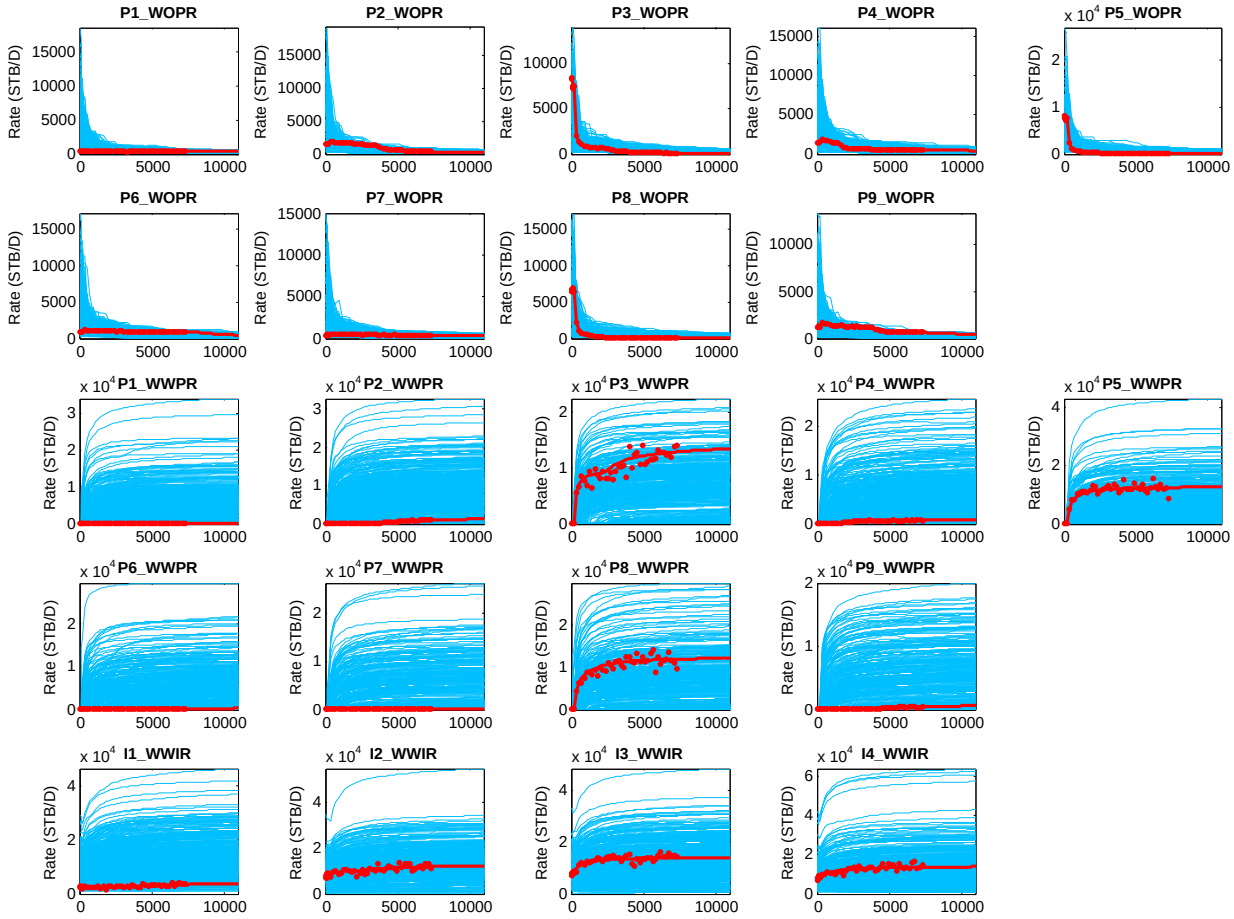


Figure 3.12: Prior ensembles of simulated well data (case 1)

(Fig. 3.14) are not clear enough for the MPS algorithm to generate facies distributions that are very close to the true model. This means that every time the facies is generated, we obtain essentially random facies distributions again. This behavior is different from what was reported in Jafapour and Khodabakhsi (2011), where very clear probability maps could be obtained. This is probably due to the higher complexity of our channel system where the the number of channels is higher, which leads to more complex interaction between the wells. This also highlights one limitation of the Tau model (at least with the current implementation in Petrel), where very high probability values are needed to significantly affect the MPS process.

On the positive side, the probability map becomes a little clearer with each iteration; enough for us to analyze maps and determine the continuity between wells. Even though

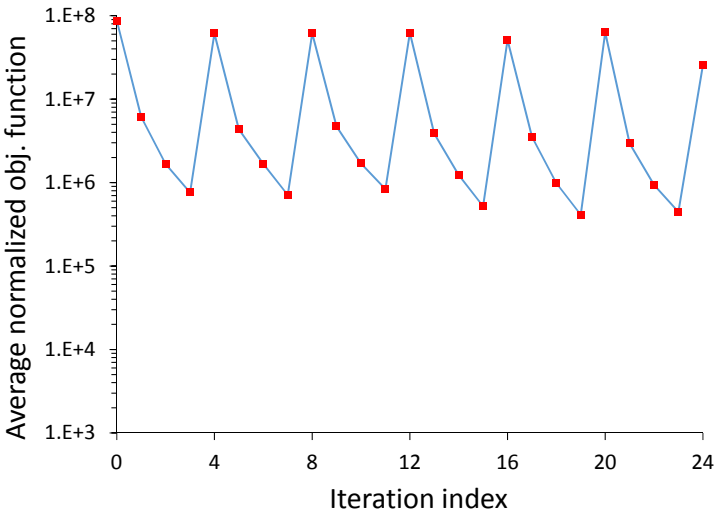
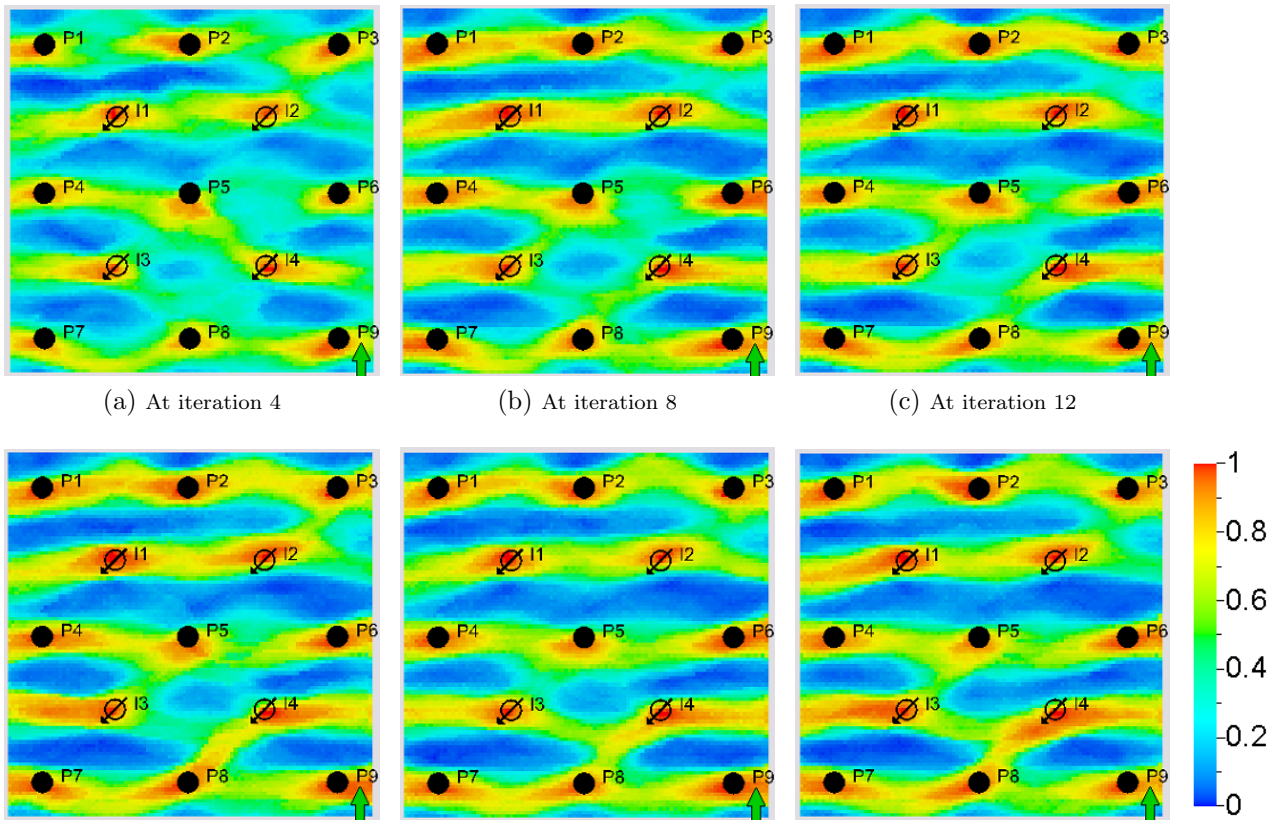


Figure 3.13: Average normalized objective function (case 1)



(a) At iteration 4 (b) At iteration 8 (c) At iteration 12

(e) At iteration 16 (f) At iteration 20 (g) At iteration 24

Figure 3.14: Probability maps obtained in stage 1 (case 1)

there are still some unclear connections, by logical deduction, we can make the following statement:

- I2 is connected to P3 by channel facies, otherwise there is no water supply to P3 and subsequently to P2 and P1.
- I3 is connected to P5, otherwise there is no water supply to P5.
- I4 may or may not connect to I3.

Using the sparse drawing method (see step 2(c)), we convert the final probability map at iteration 24 to a hard data map, which is shown in Fig. 3.15. Note that we do not either explicitly create or block the connections between I3 and I4, which allows the MPS algorithm to generate realizations with either channel or shale between I3 and I4. Fig. 3.16 shows the realizations obtained by using the hard data map and we can observe both channel configurations as expected.

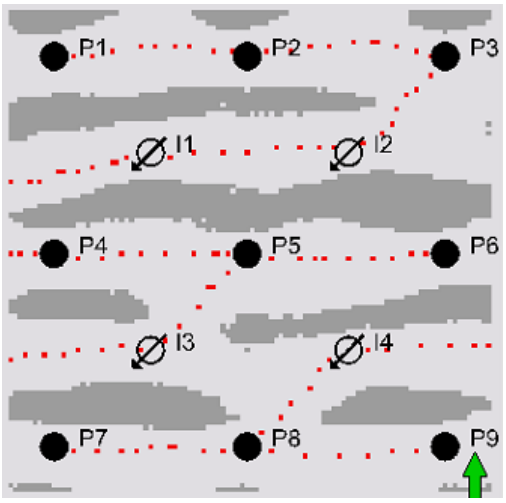
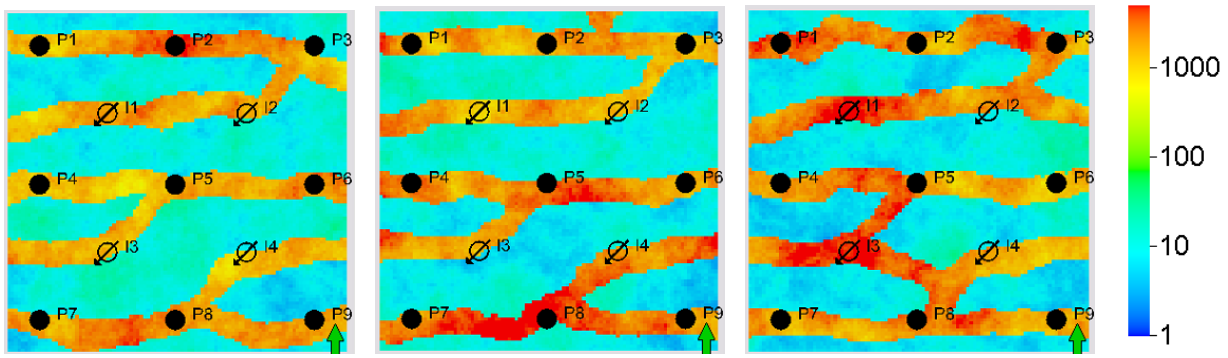


Figure 3.15: Hard data map (case 1)

Before we continue with stage 2, we want to point out that, even though realizations 1 and 2 in Fig. 3.16 and many other realizations look very similar to the true model, the data match is still poor. Fig. 3.17 shows the well data predictions at the end of stage 1, where a large discrepancy between the observed data and the simulated data can be observed. This happens for two reasons. First, we miss the channel curvatures between P5 and P6, between P7 and P8 and between P8 and P9. This is probably subtly reflected in the probability maps by the low uncertainty of the channel connections between these three pairs of wells



(a) Realization 1 (b) Realization 2 (c) Realization 3

Figure 3.16: Realizations at the end of stage 1 (case 1)

but it is difficult to recognize this because we do not expect connections between pairs of two producers to be strongly resolved. The lack of the channel curvature in our models makes the water break through much faster at P6, P7 and P9, which leads to poor data match. The second reason for the poor data match is that we have not resolved the permeability distributions within each facies. Both of these problems will be addressed in stage 2.

In stage 2, we simply apply ES-MDA 8x to the ensemble of realizations obtained at the end of stage 1. Figs. 3.18 and 3.19, respectively, show three realizations obtained at the end of stage 2 and the predicted data for all the wells at the end of stage 2. Comparing Figs. 3.16 and 3.18, we can observe some notable changes. First, the connections between I3 and I4 has been completely erased. Secondly, the lack of channel curvature has been accounted for by the use of lower permeability values. Thirdly, the distribution of permeability within each facies has been adjusted. For example, the permeability to the left of P3 and P5 is now higher than in other areas, which is consistent with the true model (Fig. 3.11a). Overall, the adjustments made during stage 2 have resulted in a very good data match as shown in Fig. 3.19. We also get most of the future forecast right except for the water rates at P6, P7 and P9, where the future forecast is either biased (P9) or highly uncertain (P6, P7). These poor predictions occur because we are not able to capture the sinuosity of the channel between P5 and P6, between P7 and P8 and between P8 and P9. To illustrate this point, we plot the water saturation profiles of the true model and the first posterior model at the

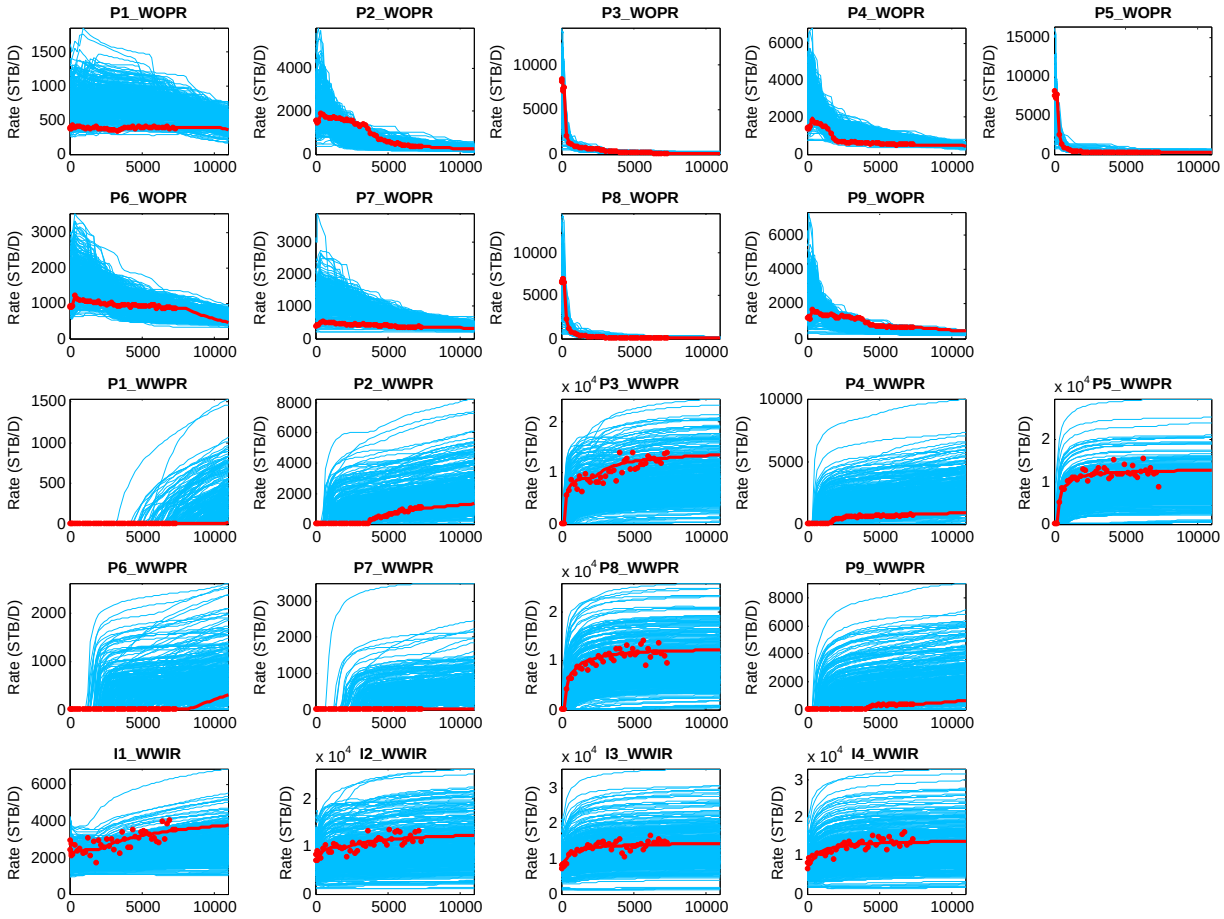
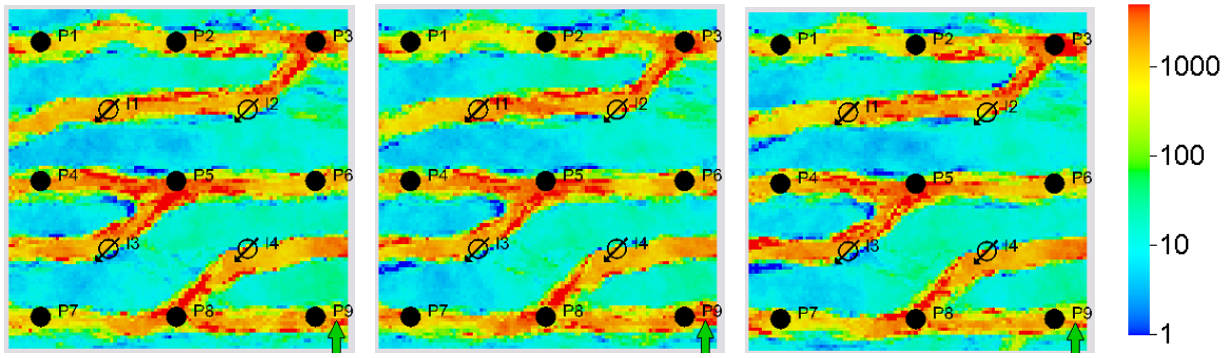


Figure 3.17: Ensembles of well data at the end of stage 1 (case 1)

end of the history matching period and the end of the prediction period in Fig. 3.20. It can be seen that even though, at the end of history matching period, water has not reached P7 in both the first posterior realization and the true model, at the end of prediction period, water has already passed P7 in the first posterior realization while it has just barely reached P7 in the true model.



(a) Realization 1 (b) Realization 2 (c) Realization 3

Figure 3.18: Realizations at the end of stage 2 (case 1)

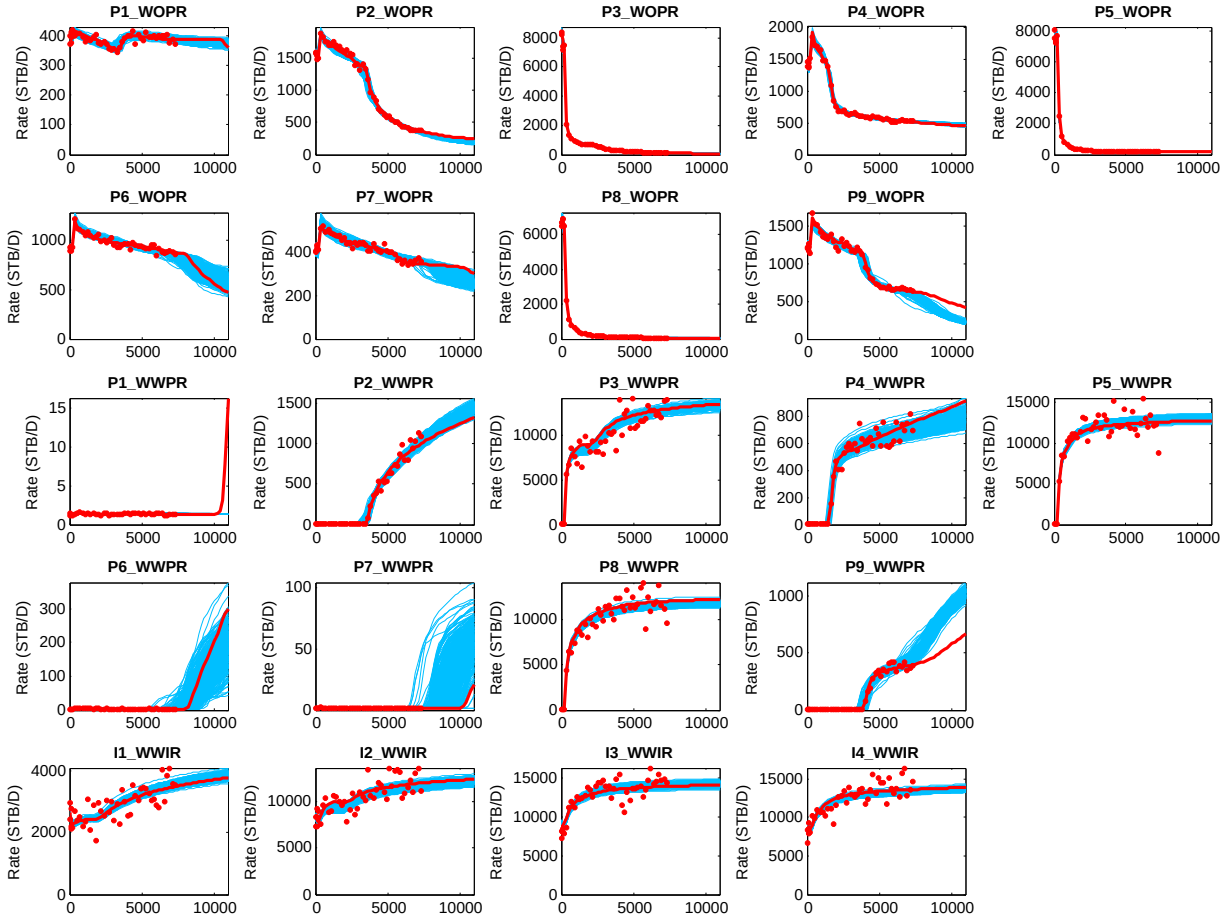
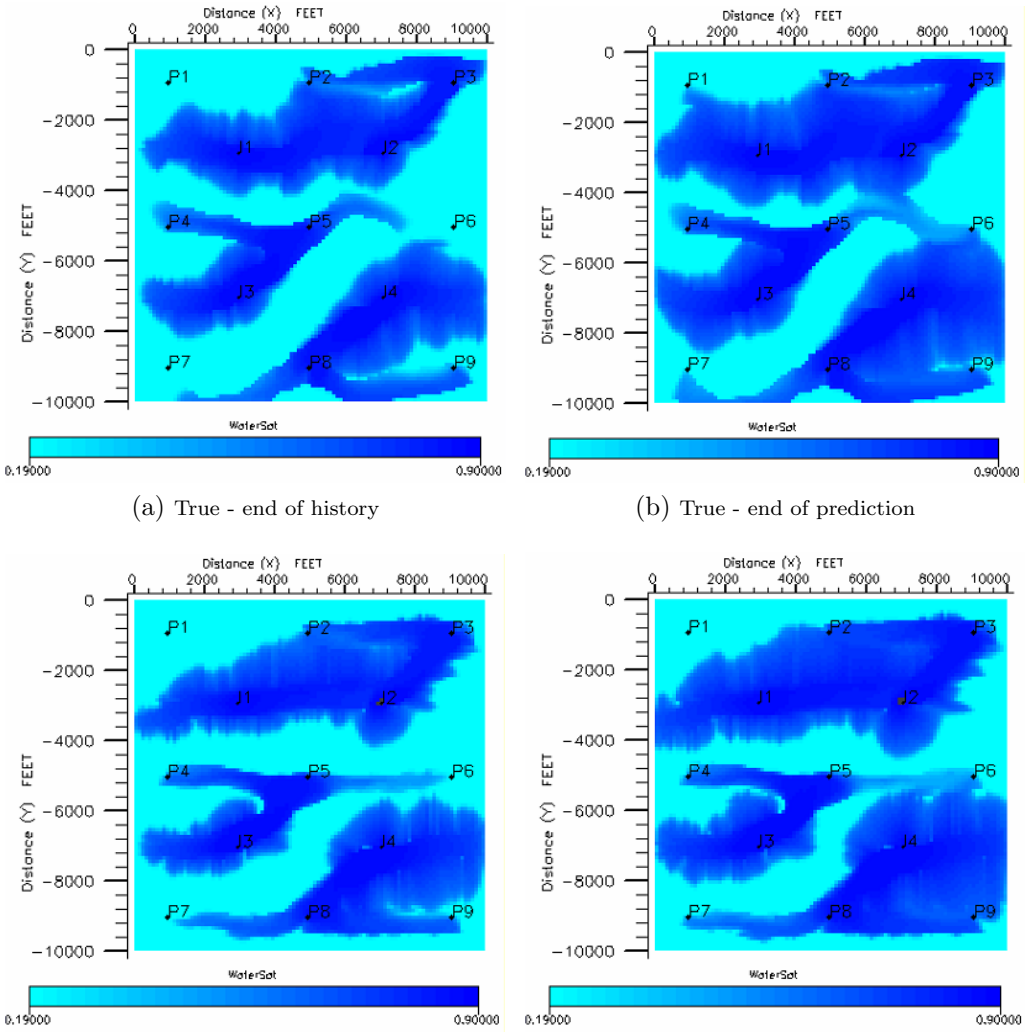


Figure 3.19: Ensembles of well data at the end of stage 2 (case 1)



(a) True - end of history

(b) True - end of prediction

(c) Realz 1 - end of history

(d) Realz 1 - end of prediction

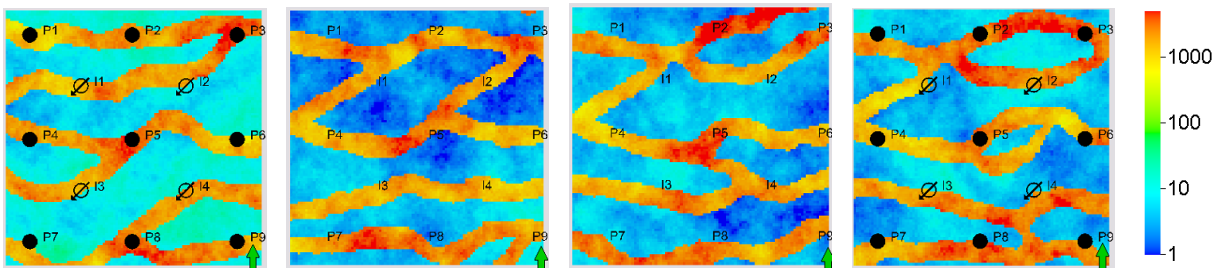
Figure 3.20: Water saturation profiles for the true model and the first posterior realization (case 1).

3.3 Case Study 2

This case is similar to the first case except for the following modifications:

- To generate the prior realizations, we estimate the geometric mean of the sand permeability from the hard data at the well locations. The result is 2200 md, which is reasonably close to the mean in the true model, which is 2000 md.
- The geometric mean of the shale facies is unknown, but we assume the value of 4 md in the prior model, whereas the true model was generated with a value of 10 md.

The true model, which is the same as in case 1, and three prior realizations are shown in Fig. 3.21. It can be clearly seen that the shale permeability in the prior realizations is lower than the shale permeability in the true model. The corresponding prior ensembles of predicted data are shown in Fig. 3.22.



(a) True model (b) Prior realization 1 (c) Prior realization 2 (d) Prior realization 3

Figure 3.21: True model and three prior realizations (case 2)

Similar to case 1, in stage 1, we use an ensemble size of 400 and perform 24 data assimilation steps with inflation factors equal to 24 for all iterations. The objective function evolution and the probability map evolution are shown in Figs. 3.23 and 3.24, respectively. From the two figures, it can be seen that the objective function in case 2 behaves in a similar manner as in case 1 but the probability map obtained in this case is less clear than that in the previous case. For example, in case 1, we can be quite certain that there is no connection between I4 and P5 because probability of channel occurrence between the two wells is only about 10% (Fig. 3.14g) but in case 2, the probability of channel occurrence between the two

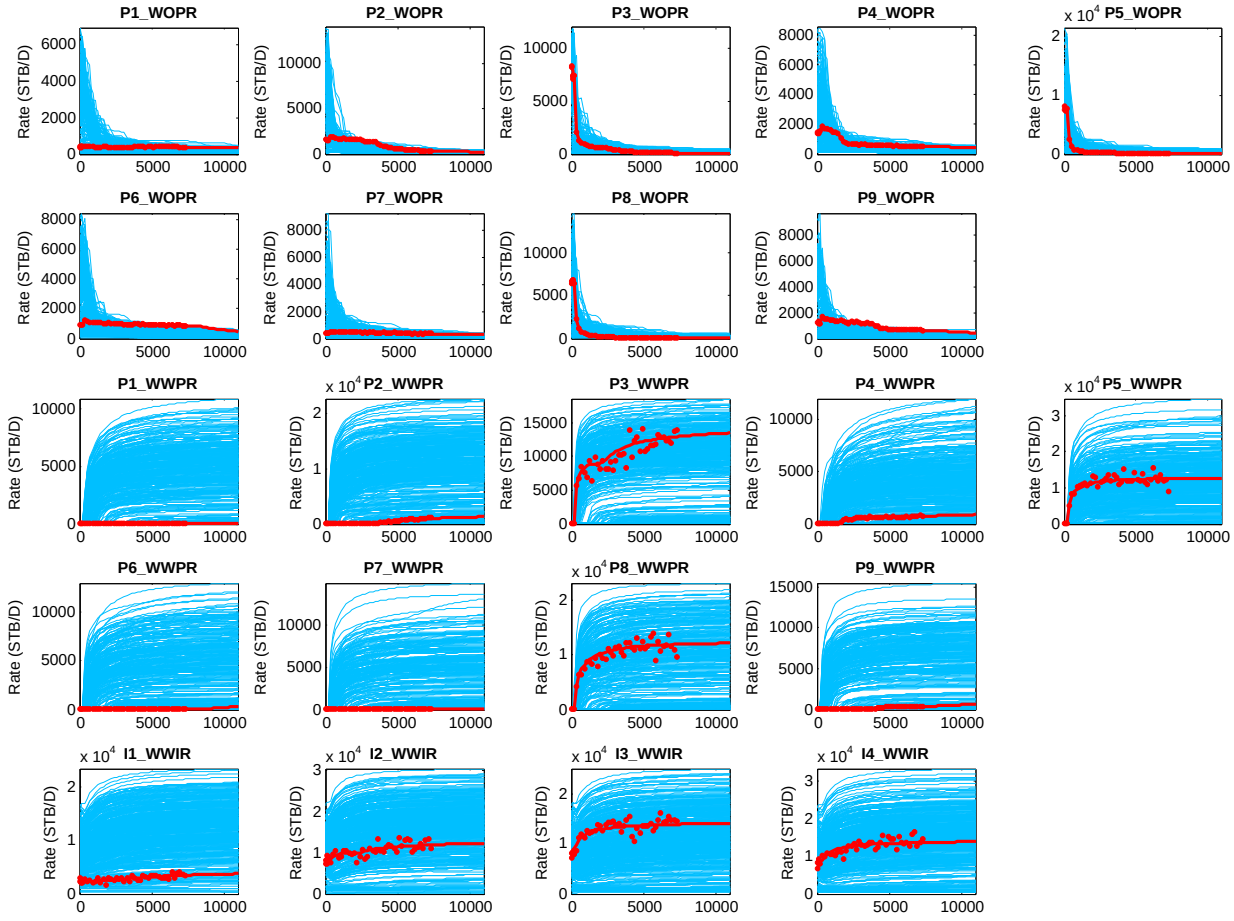


Figure 3.22: Prior ensembles of simulated well data (case 2)

wells is about 50% (Fig. 3.24g). The probability map also does not get better after iteration 12. Again, by logical deduction, we can make the following statements:

- I2 is connected to P3, otherwise there is no water supply to P3 and subsequently P2 and P1.
- I4 is at least connected to P8, otherwise there is no water supply to P8 and subsequently P7 and P9.
- I4 may or may not connect to I3 and P5.

Base on the above deductions, we investigate two possible scenarios: (1) I4 is connected to only P8 and (2) I4 is connected to both P5 and P8 with a possible connection to I5. For each scenario, we draw a hard data map, which is shown in Fig. 3.32. Then we use

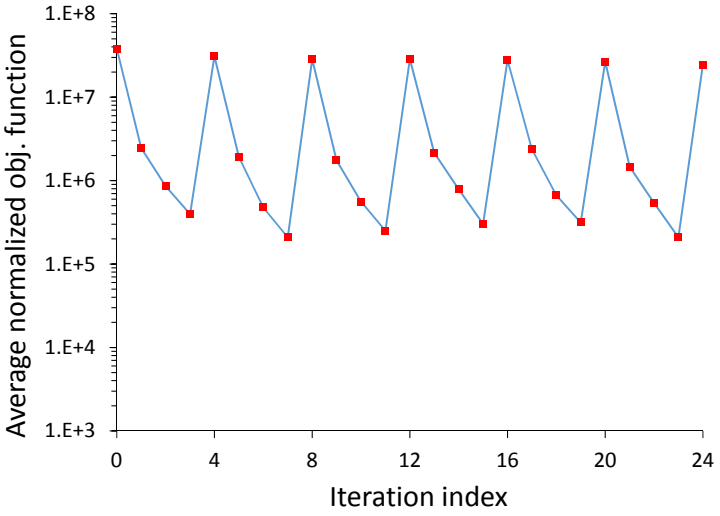
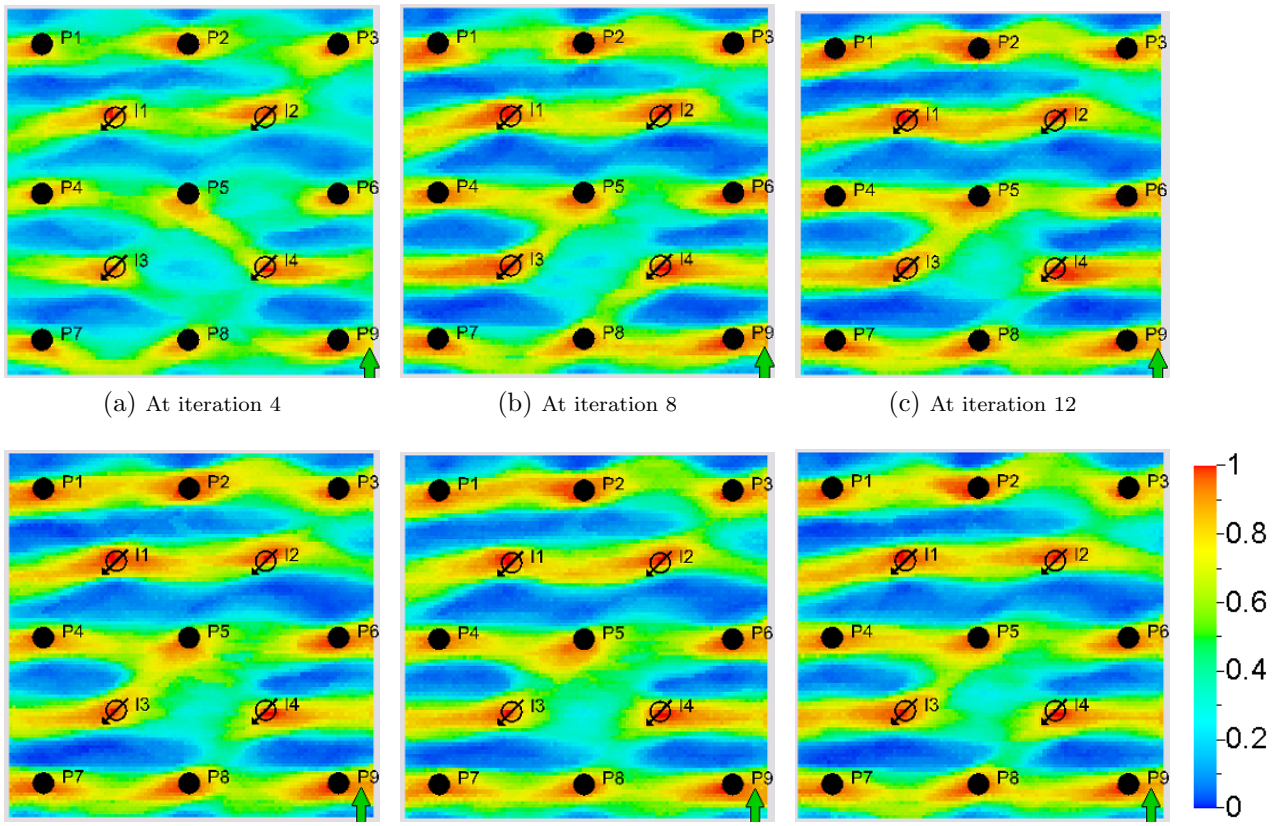


Figure 3.23: Average normalized objective function (case 2)

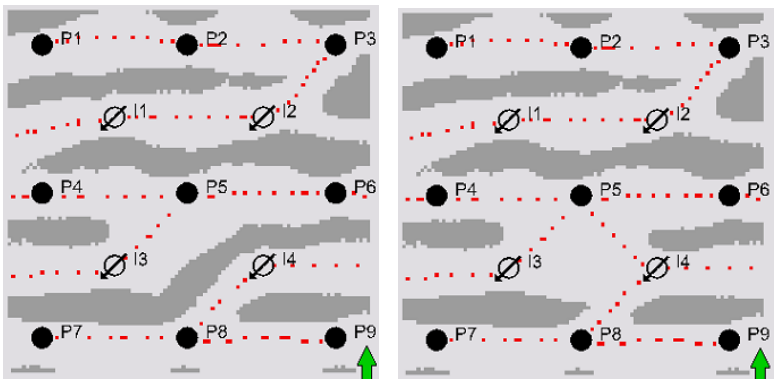


(a) At iteration 4 (b) At iteration 8 (c) At iteration 12

(e) At iteration 16 (f) At iteration 20 (g) At iteration 24

Figure 3.24: Probability maps obtained in stage 1 (case 2)

the hard data map in combination the MPS algorithm to generate 400 realizations of the facies distribution and populate these facies distributions with permeability values drawn

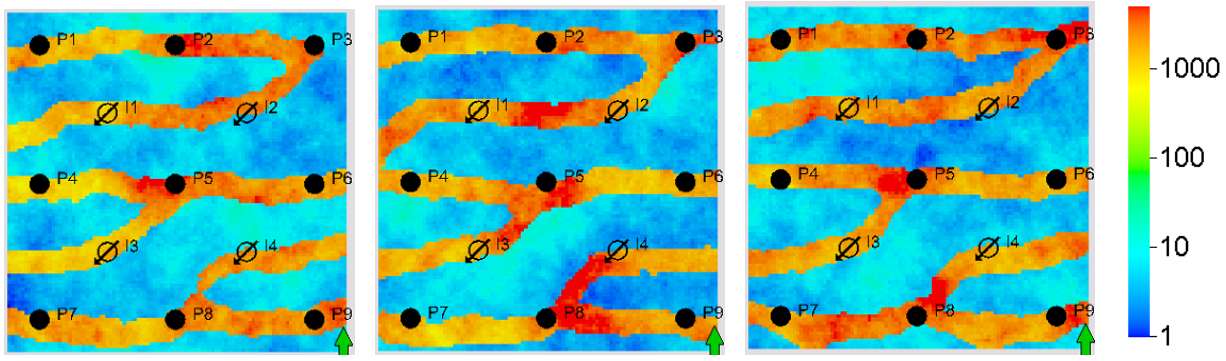


(a) Scenario 1 (b) Scenario 2

Figure 3.25: Hard data map for two scenarios (case 2)

from the prior permeability distributions, which gives us the final ensemble of realizations of stage 1, which will be used as the prior for stage 2. For case 2, we do not show the data match at the end of stage 1 but they are similar to that in case 1.

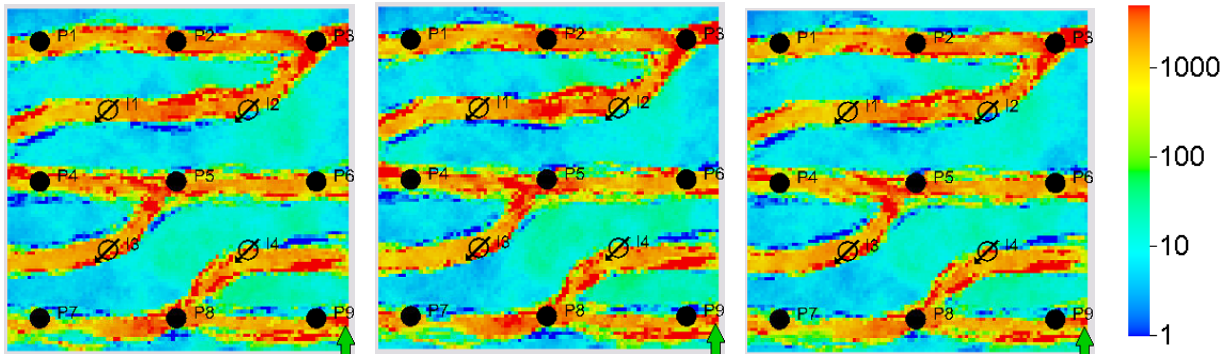
For scenario 1, Figs. 3.26, 3.27, and 3.28, respectively, show three realizations at the end of stage 1, three realizations at the end of stage 2 and the predicted data for all the wells at the end of stage 2. Comparing Figs. 3.26 and 3.27, we can see that the shale permeabilities have been increased to roughly the same values as in the true model (Fig. 3.21a). The overall data match is very good even though some wells suffer from poor predictions for the same reason as mentioned in case 1.



(a) Realization 1 (b) Realization 2 (c) Realization 3

Figure 3.26: Realizations at the end of stage 1 (scenario 1, case 2)

For scenario 2, Figs. 3.29, 3.30, and 3.31, respectively, show three realizations at the end of stage 1, three realizations at the end of stage 2 and the predicted data for all the



(a) Realization 1 (b) Realization 2 (c) Realization 3

Figure 3.27: Realizations at the end of stage 2 (scenario 1, case 2)

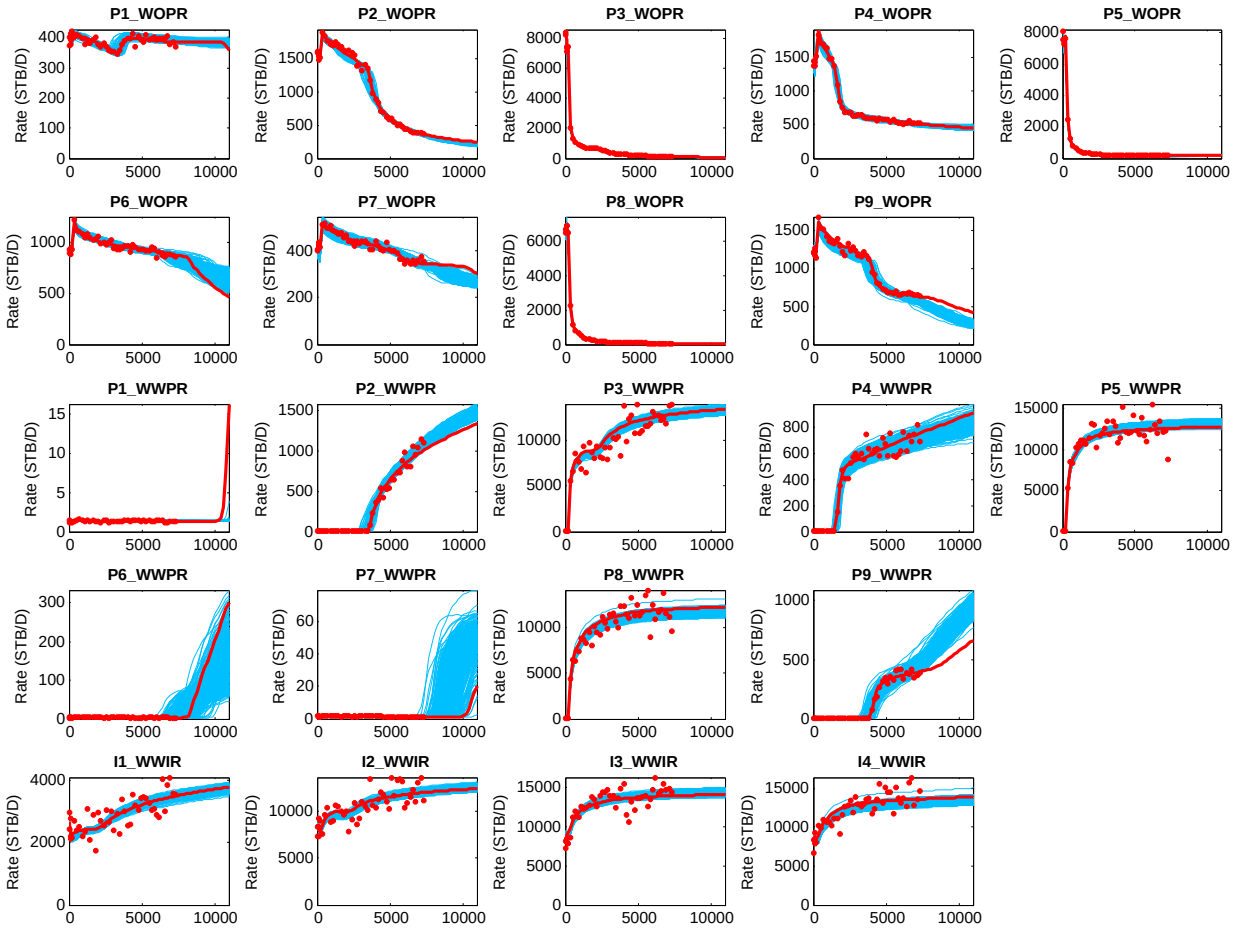
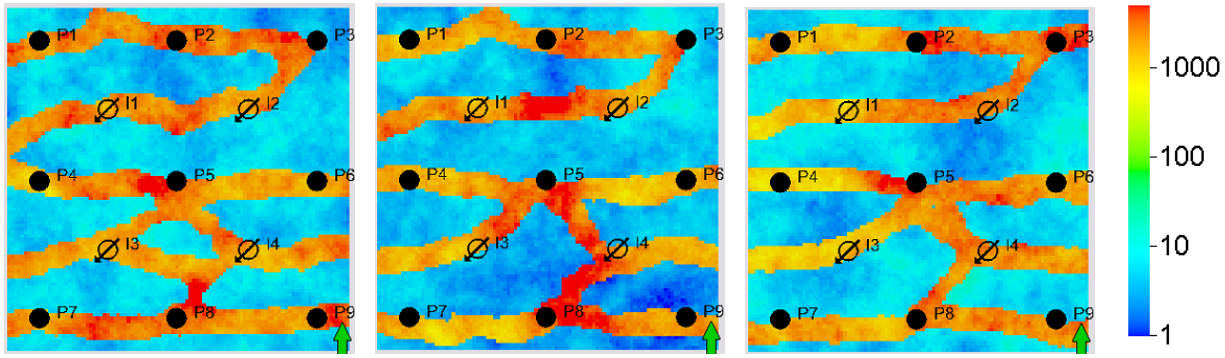


Figure 3.28: Ensembles of well data at the end of stage 2 (scenario 1, case 2)

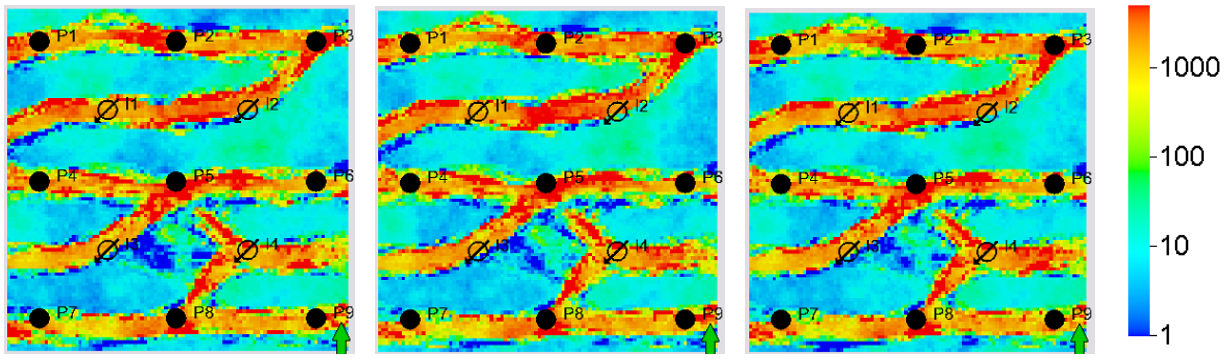
wells at the end of stage 2. It is interesting to note that the connections between I4 and P5 and between I4 and I3 have been effectively erased in the posterior models, which means the first scenario is most likely the correct scenario. With the redundant connections erased, the

data match for this scenario is also very good even though the forecast is somewhat worse than scenario 1.



(a) Realization 1 (b) Realization 2 (c) Realization 3

Figure 3.29: Realizations at the end of stage 1 (scenario 2, case 2)



(a) Realization 1 (b) Realization 2 (c) Realization 3

Figure 3.30: Realizations at the end of stage 2 (scenario 2, case 2)

3.3.1 A slightly different approach

Here, we investigate what happens if, in step 2(c) of stage 1, instead of using the probability map directly in the MPS algorithm at iterations that are not the final iteration, we first convert it to a hard data map. Note that we are redoing everything thing from the beginning of stage 1, so it is not related to either of the scenarios above. We use the threshold method with a 85% threshold to transform from the probability map to a hard data map. Fig. 3.32 shows the probability maps and the corresponding hard data maps at the end of iterations 4 and 8 of stage 1. It can be seen that the model converges very quickly when the

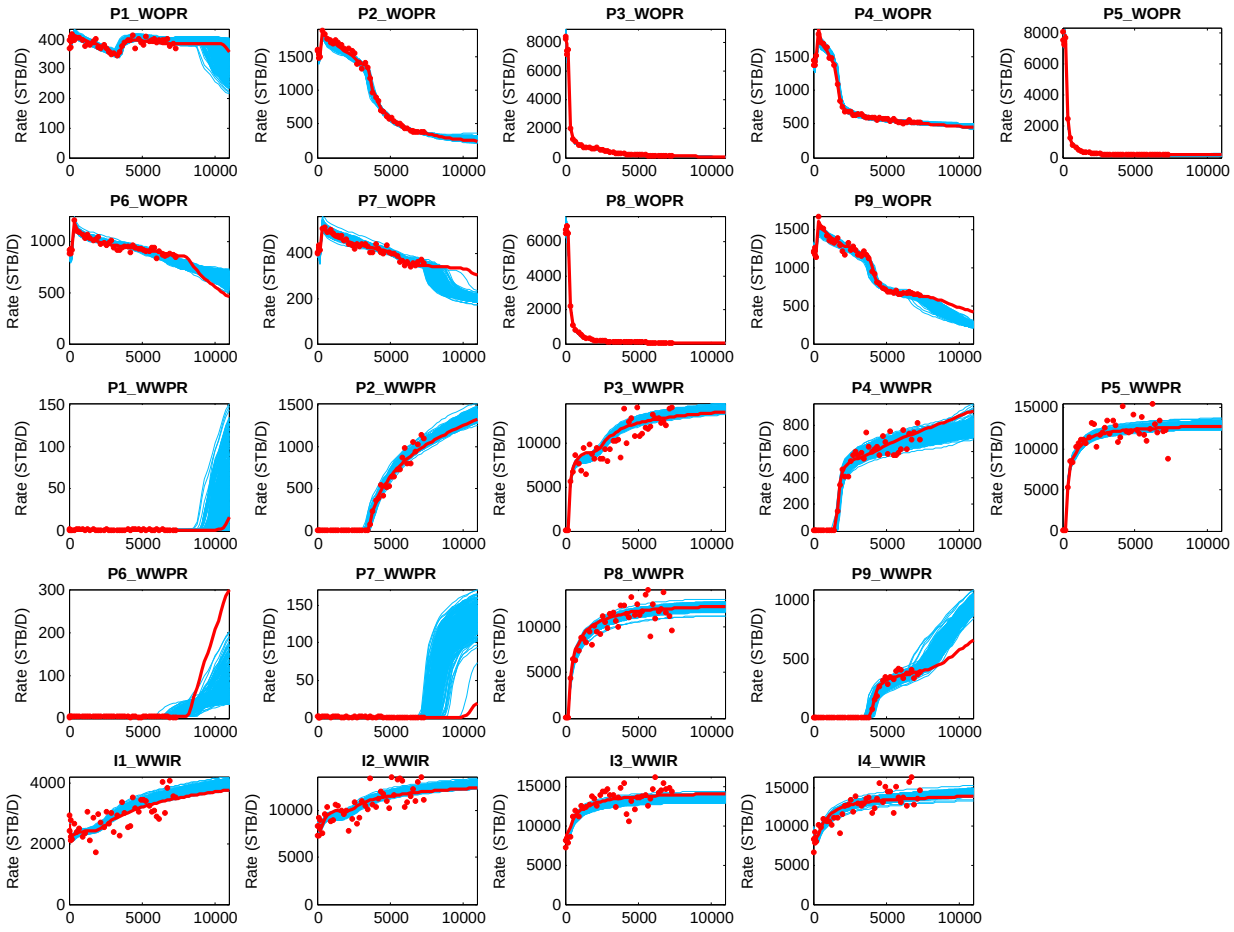
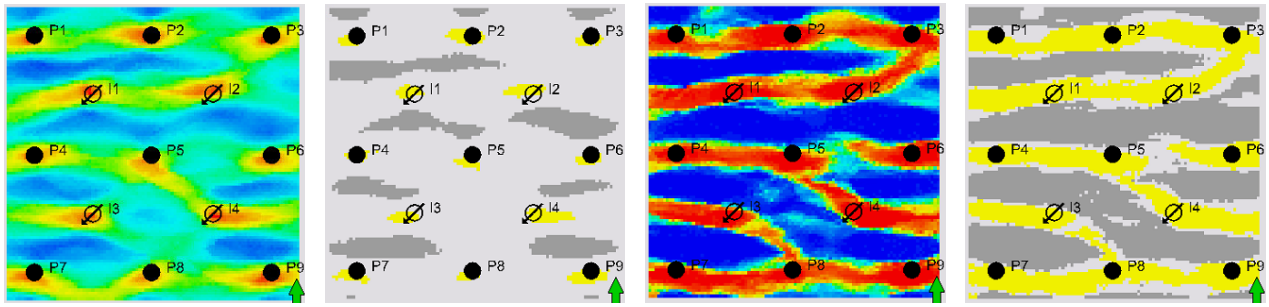


Figure 3.31: Ensembles of well data at the end of stage 2 (scenario 2, case 2)

hard data map is used instead of the probability map; at iteration 8, the probability map is mostly dominated by 100% shale and 100% sand regions. However, the channel configuration in the probability map is inverted compared to the true model (Fig. 3.21a); I4 now supplies water to P5, P4 and P6 while I3 supplies water to P8, P7, P9. The reason this happens is because at iteration 4, a small bias is introduced in the probability map where I4 has high chance of being connected to P5. If we use the probability map instead of the hard data map in the MPS algorithm, this bias is corrected at iteration 8; see Fig. 3.24b. However, by using the hard data map, we force the bias to be carried over to subsequent iterations.

Nevertheless, the well configuration we obtain using hard data maps is still reasonable since I3 and I4 still supply water to the same number of producers, so we end stage 1 after iteration 8 and move to stage 2. In stage 2, we apply ES-MDA 8x to the ensemble of



(a) Probability map at iter 4 (b) Hard data map at iter 4 (c) Probability map at iter 8 (d) Hard data map at iter 8

Figure 3.32: Probability maps and hard data maps obtained in stage 1 (a slightly different approach, case 2)

realizations obtained at the end of stage 1, and the data match at the end of stage 2 is shown in Fig. 3.33. It can be seen that even though most of the data matches are good, we cannot match the water rates at well P4 due to the inverted well configuration. In this configuration, P4's water production is provided by I4 instead of I3 but because I4 is much farther from P4 than I3 is, it cannot provide enough water to match the observed data.

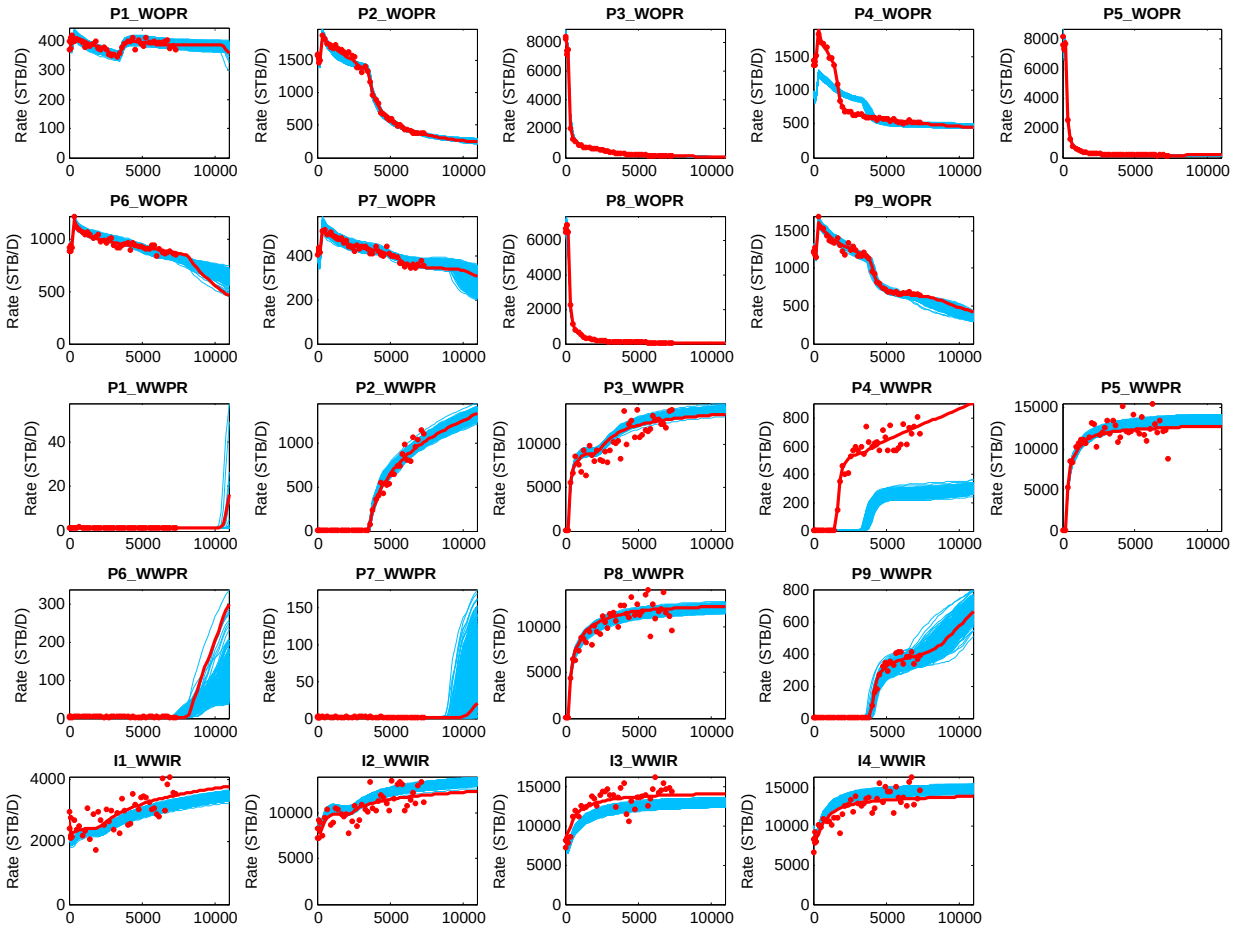
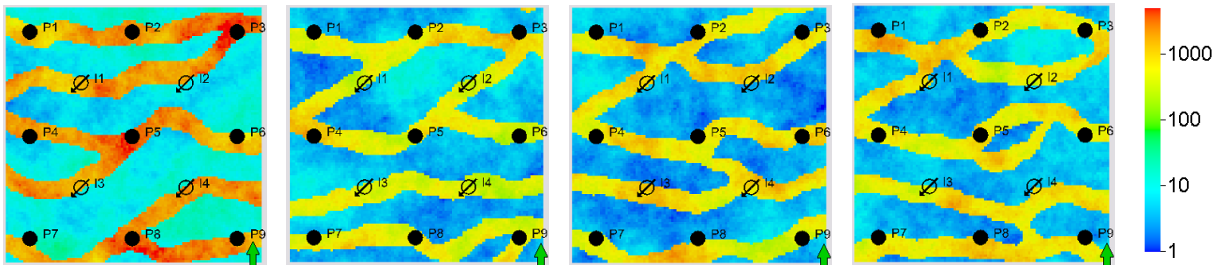


Figure 3.33: Ensembles of well data at the end of stage 2 (a slightly different approach, case 2)

3.4 Case Study 3

Case 3 is identical to case 2 except that we assume the permeability hard data at the well locations are not available and the best guess of the geometric mean of the permeability is 700 md as opposed to the true value of 2,000 md. The true model and three prior realizations are shown in Fig. 3.34. It can be seen that both the shale and sand permeability in the prior realizations are lower than those in the true model.



(a) True model (b) Prior realization 1 (c) Prior realization 2 (d) Prior realization 3

Figure 3.34: True model and three prior realizations (case 3)

In stage 1, similar to the previous cases, we use an ensemble size of 400 and perform 24 data assimilation steps with inflation factors equal to 24 for all iterations. The objective function evolution is shown in Fig. 3.35. What happens here is similar to cases 1 and 2; every four iterations when new realizations are regenerated, the data mismatch increases sharply and the algorithm falls into an endless cycle. The probability map evolution is shown in Fig. 3.36. An unexpected observation is that the final probability map obtained in case 3 (Fig. 3.36g) is clearer than the final probability maps obtained in case 1 (Fig. 3.14g) and case 2 (Fig. 3.24g). We currently do not have a explanation for this phenomenon.

From the final probability map, we create a hard data map using the sparse drawing method (step 2(c) of final iteration), which is shown in Fig. 3.37. Then we use the hard data map in the MPS algorithm to create 400 realizations of the facies field and populate them with permeability values sampled from the prior Gaussian distributions. Fig. 3.38 shows the first three realizations of the permeability field obtained at the end of stage 1.

In stage 2, we apply ES-MDA 8x to the ensemble of realizations obtained at the end of stage 1. Figs. 3.39 and 3.40, respectively, show three realizations at the end of stage 2 and

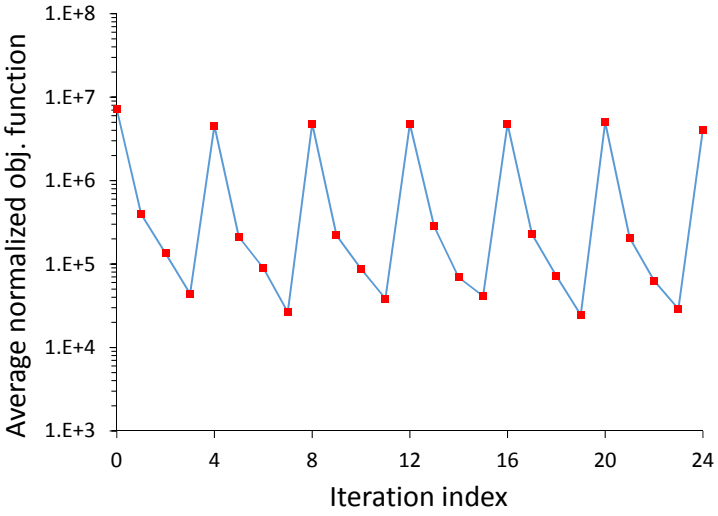
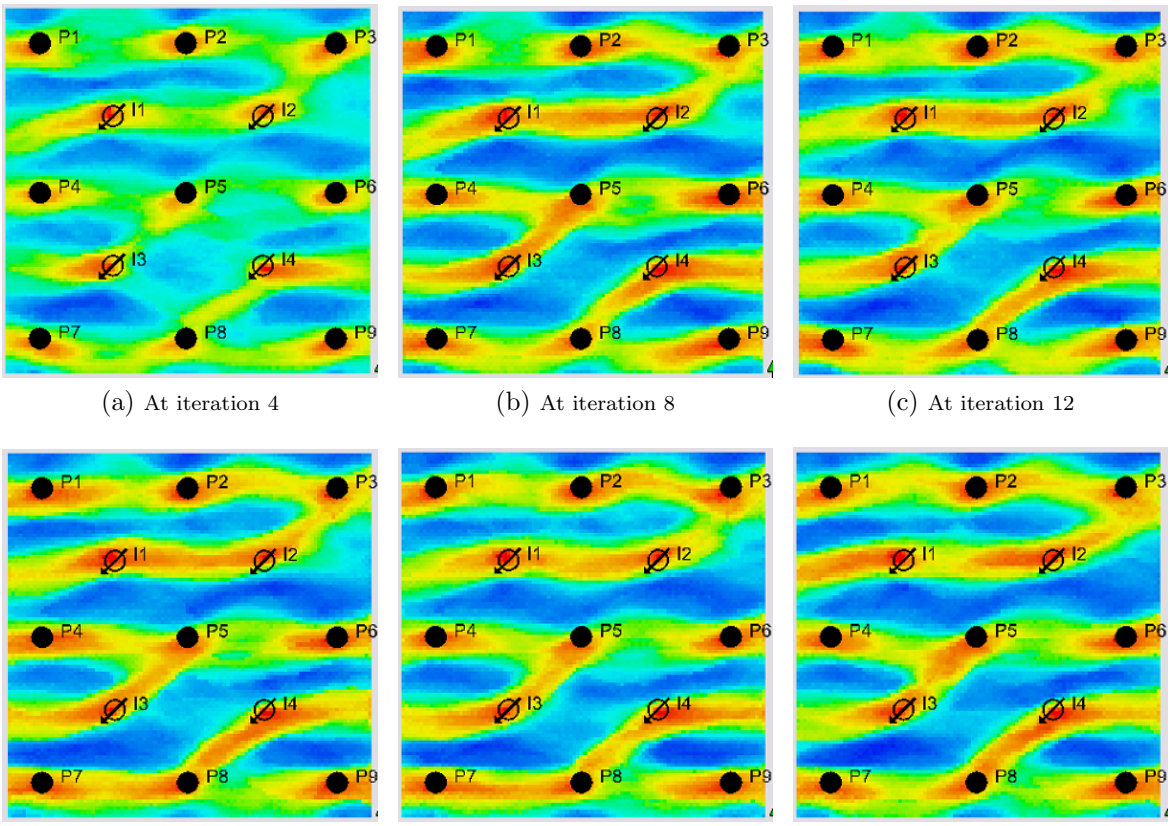


Figure 3.35: Average normalized objective function (case 3)



(a) At iteration 4 (b) At iteration 8 (c) At iteration 12

(d) At iteration 16 (e) At iteration 20 (f) At iteration 24

Figure 3.36: Probability maps (case 3)

the predicted data for all the wells at the end of stage 2. It can be seen that despite starting with the wrong permeability values in both shale and sand, stage 2 is able to increase the

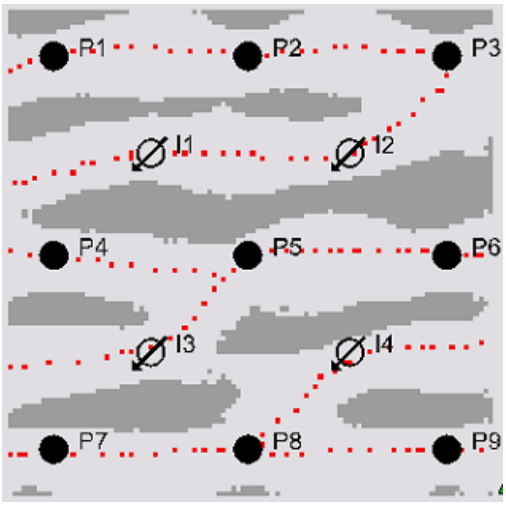
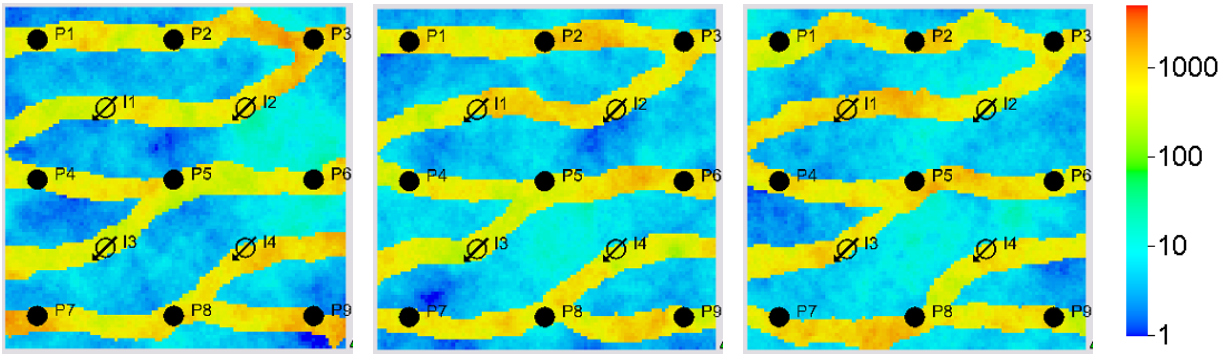


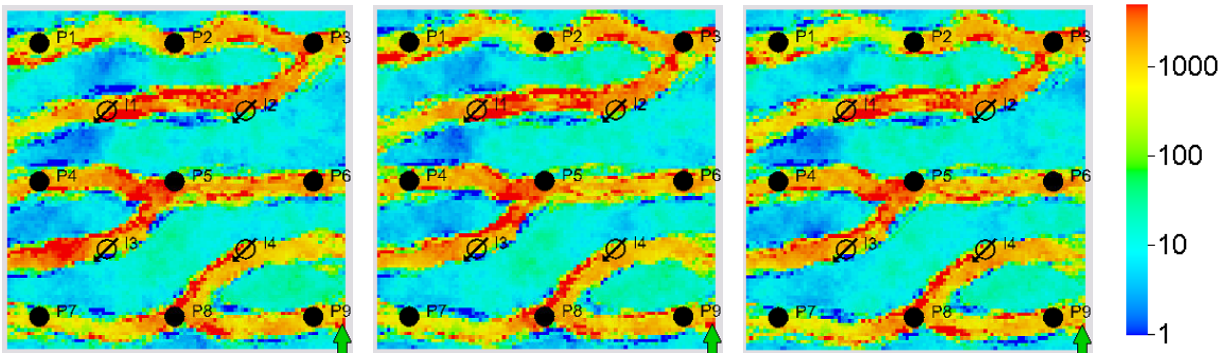
Figure 3.37: Hard data map (case 3)



(a) Realization 1 (b) Realization 2 (c) Realization 3

Figure 3.38: Permeability realizations at the end of stage 1 (case 3)

permeability in both regions to get a good data match.



(a) Realization 1 (b) Realization 2 (c) Realization 3

Figure 3.39: Permeability realizations at the end of stage 2 (case 3)

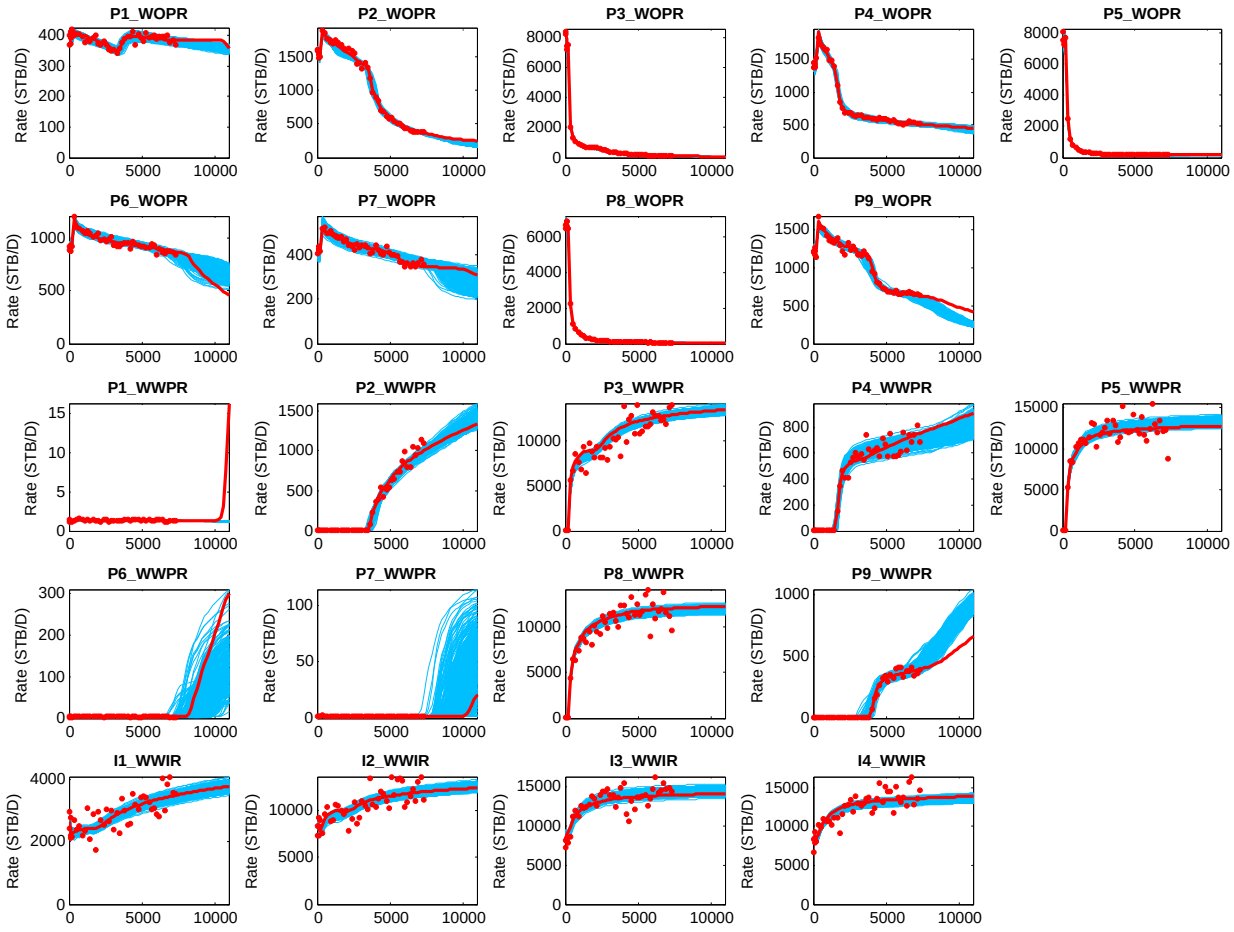


Figure 3.40: Ensembles of well data at the end of stage 2 (case 3)

3.4.1 A modified procedure

Here, we try modifying our procedure to adjust the means of the permeability distributions in stage 1, as opposed to waiting until stage 2 as in the original procedure. Even though our modified procedure does not work as expected, for research purposes, we still present our modifications and explain why they do not work. Our modifications are as follows:

1. In step 2(d), when generating a new permeability field, we add one random parameter to the mean of the shale and sand permeability to account for the uncertainty in the mean. In other words, in the log domain, the shale and sand permeability distributions are, respectively, $N(\text{shale\_mean} + N(0, 1.21), 0.25)$ and $N(\text{sand\_mean} + N(0, 1.21), 0.25)$.

2. Also in step 2(d), we use the updated means of the permeability distributions obtained from EM in step 2(b) to populate the gridblocks with new permeability values. We do this with the hope that our ES-MDA iterations will drive the overall permeability means closer to the their true values. This, however, does not work well due to the complexity of the channels and will be explained later.

3. Since the idea of updating the permeability means through EM above does not work, we try adding a screening step to find the correct permeability means near the end of stage 1. To do this, between steps 2(c) and 2(d) of the final iteration of stage 1, we take a single facies distribution among the ensemble obtained at step 2(c) and populate the sand and shale facies with different but constant permeability values. Then we run simulations for these realizations and pick the pair of permeability values that result in the best data match and consider them to be the correct means of the shale and sand permeability. Then we continue step 2(d) of the last iteration of stage 1 but, instead of using the prior means, we use the means obtained in the screening step. This again does not work and the permeability values we obtain are very far away from the true permeability means.

In stage 1, we again use an ensemble size of 400 and perform 24 data assimilation steps with inflation factors equal to 24 for all iterations. All three modifications are incorporated in the procedure at their respective steps. With modification #2, every four iterations we use the updated means of the distributions from EM to populate gridblocks with new permeability values. Fig. 3.41 shows the evolution of the geometric means of the sand and shale permeability. After 24 iterations, the geometric sand mean decreases from 1000 to around 700 mD while the shale mean, after some fluctuation, ends up essentially at the same 4 mD value. This result is different from what we expect. Because both the prior means for sand and shale are smaller than the true means, we expect them to increase instead of decreasing.

After some investigation into this issue, we find that this happens because the water

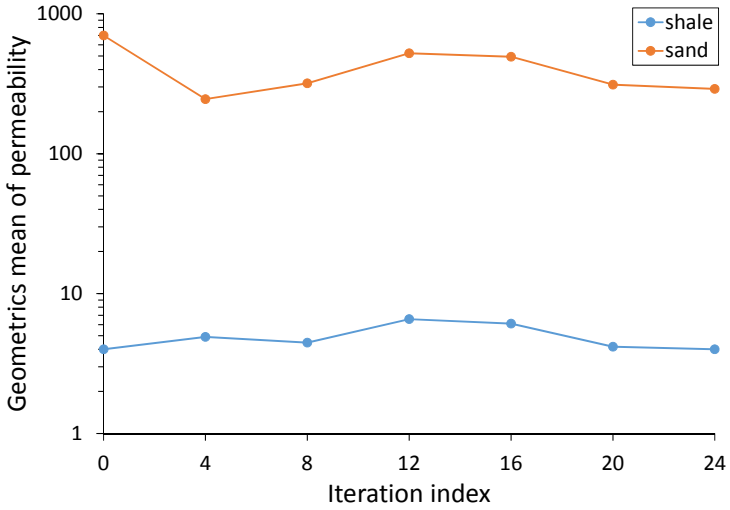


Figure 3.41: The evolution of the permeability means (a modified procedure, case 3)

breaks through at some wells much earlier than it should due to an incorrect channel configuration in the prior ensemble. To correct for this, the ES-MDA algorithm, which is based on the linear-Gaussian assumption, decreases the overall permeability in the reservoir. To prove this point, we try redoing stage 1 using larger measurement errors for the water rates. In the original problem, the standard deviation of the measurement error for the water rates is at a minimum of 0.1 STB/D when there is no water production. We redo the problem using minimum values of 1.0 STB/D and 10.0 STB/D for the standard deviation of water rate measurement error. The results are shown in Fig. 3.42 and 3.43. From these figures, it can be seen that the permeability means are updated much better when the water rate measurement error is 10 STB/D, where estimates of the geometric sand and shale means are 1,000 mD and 9 mD, respectively. There is a reason why the sand permeability mean cannot increase to the correct value of 2,000 mD. This is because our procedure cannot capture the channel sinuosity between wells, so it has to make the permeability at those locations smaller, thus decreasing the overall sand permeability mean. We will experience this phenomenon again when we try to find the correct permeability means using screening; see the description of the third modification we make to the procedure.

Back to the original problem where the minimum water rate measurement error is 0.1 STB/D, the objective function and the probability map evolutions of stage 1 are, respectively,

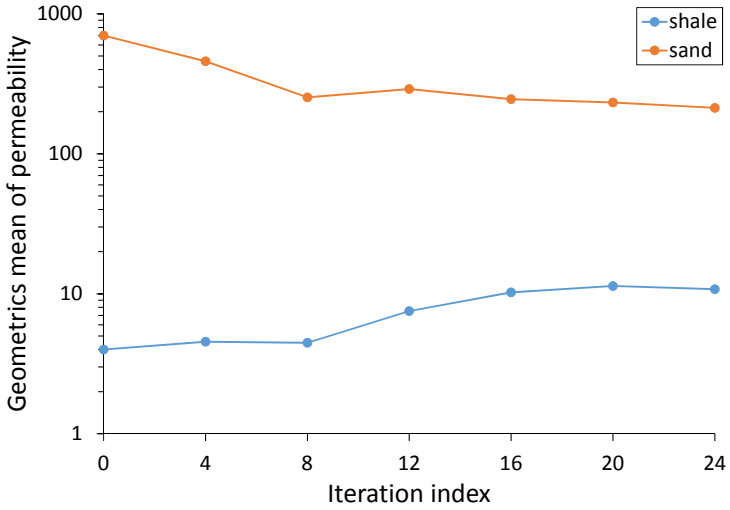


Figure 3.42: The evolution of the permeability means when minimum water rate measurement error is 1.0 STB/D (a modified procedure, case 3)

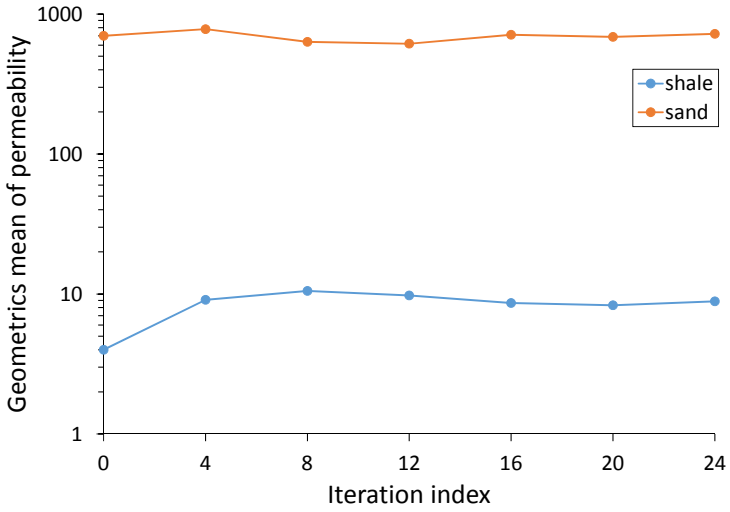


Figure 3.43: The evolution of the permeability means when minimum water rate measurement error is 10.0 STB/D (a modified procedure, case 3)

shown in Figs. 3.44 and 3.45. One interesting thing we observe from the plot of the objective function is that the data mismatch at the end of iteration 4 is higher than the initial data mismatch. This happens due to the first modification. Instead of generating the shale and sand permeability from the prior Gaussian distributions, we add a random parameter to account for the uncertainty in the means. This helps create a new ensemble with a larger range of permeability than the range based on the prior ensemble. Unfortunately, some ensemble members have extreme permeability values, which results in a very poor data

match and leads to a higher data mismatch. On the other hand, despite not being able to improve the data match, the probability map is still able to capture all important connections in the true model. From the final probability map, we create a hard data map using the sparse drawing method (step 2(c)), which is shown in Fig. 3.46, then we use the hard data map in the MPS algorithm to create 400 realizations of the facies field (still step 2(c)). Note that we have not performed step 2(d) of the final iteration yet. Before doing so we have have to perform the third modification where a screening step is inserted between steps 2(c) and 2(d) of the final iteration.

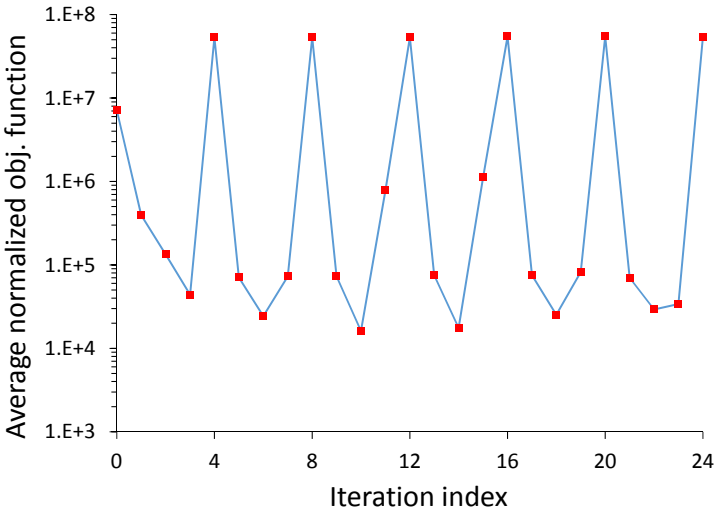
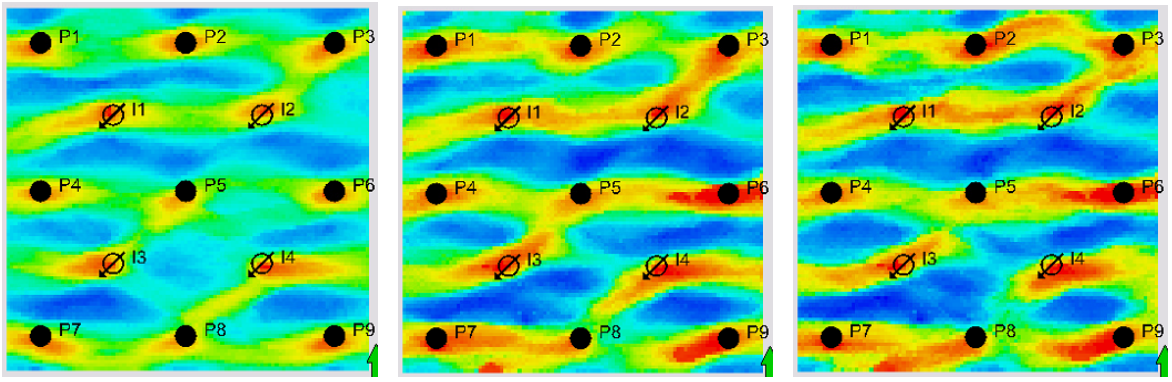
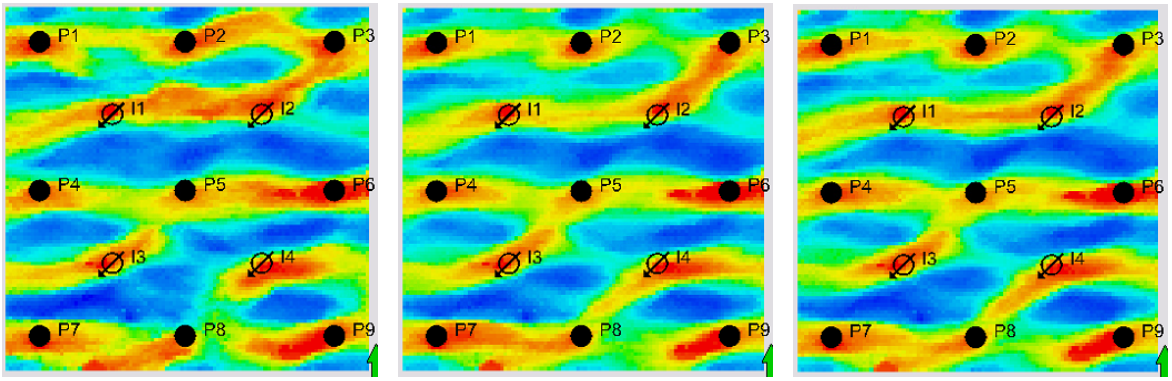


Figure 3.44: Average normalized objective function (a modified procedure, case 3)

As mentioned earlier, the purpose of the screening step is to try to determine the best means for the sand and shale permeability. From the ensemble of facies distributions generated at the end of stage 1, we pick one representative facies distribution (in this case, facies realization #2), and then assign different, constant permeability values to the sand and shale facies. We create a total of 1,000 realizations of the sand-shale permeabilities for the single fixed facies distribution where the sand and shale log-permeability values generated, respectively, from $N(1.39, 1.21)$ and $N(6.55, 1.21)$. Three such realizations are shown in Fig. 3.47. We run simulations for these realizations and pick the one with the best data match. We would hope that the one with the best data match would have the shale and sand permeabilities closest to the true permeability means. However, the best data match is



(a) At iteration 4 (b) At iteration 8 (c) At iteration 12



(d) At iteration 16 (e) At iteration 20 (f) At iteration 24

Figure 3.45: Probability maps (a modified procedure, case 3)

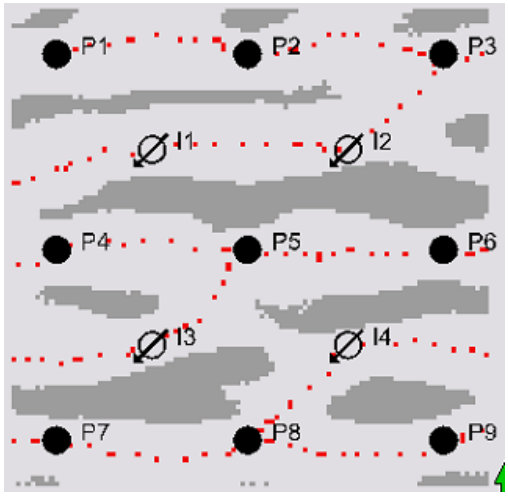
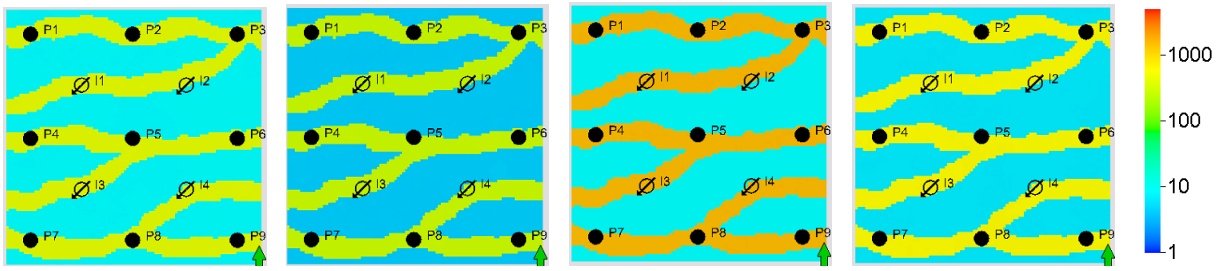


Figure 3.46: Hard data map (a modified procedure, case 3)

obtained by realization 88 that has the permeabilities for the shale and sand, respectively, being 6.4 md and 590 md, which are much smaller than the true permeability means, which are 10 md for shale and 2000 md for sand. Realization 24, which has the permeability

values closest to the true means, ranks very low in term of the data match. To explain this behavior, recall that we have mentioned several times that our procedure does not capture the sinuosity of the channels between P5 and P6, between P7 an P8 and between P8 and P9. These sinuosities delay the time at which the water reaches P6, P7 and P9 in the true model, i.e. water production at these wells remains zero for most of the history period. Since the standard deviation of measurement errors is based on the observed value, the measurement error is very small for the water rates at these wells, thus putting a significant weight on these water rates in the data mismatch calculation. Therefore, the permeability value that gives the best data match is one that is significantly smaller than the true permeability in order to slow down the water production at P6, P7 and P9 in the absence of channel sinuosity.



(a) Realization 1 (b) Realization 3 (c) Realization 24 (d) Realization 88

Figure 3.47: Realizations created for screening process (a modified procedure, case 3)

At this point, we realize that all modifications we make to the stage 1 of the procedure do not give the desirable results. Therefore, we stop here and do not proceed further. At this point in the research, it is recommended that the unmodified procedure is used instead.

CHAPTER 4

SURVEILLANCE OPTIMIZATION

As mentioned in the Introduction section, even though the exhaustive history matching procedure is a reasonably rigorous way to estimate the expected uncertainty reduction associated with a given surveillance scenario, the required computational cost makes it unsuitable for real problems. In this chapter, we pursuit a more efficient way to answer the two questions pertaining to surveillance optimization: (1) What are the best data to collect, i.e., which surveillance operation among those that are viable, will yield the greatest reduction in uncertainty? and (2) What is the expected value of the uncertainty reduction? We use a concept in information theory, namely mutual information (Shannon, 1948), to quantify the strength of the relationship between observed data and the reservoir variable J of interest. To estimate mutual information, we experiment with two different methods. The first one is a rigorous method based on the nearest neighbor distance (Kraskov et al., 2004). This method works well for the first two problems where the number of observed data is small. However, when applied to problems with large number of observed data (problems 3 and 4), the nearest neighbor method shows its limitation. For these problems, we adopt a less rigorous but more efficient method based on the linear-Gaussian assumption. The value of the mutual information provides a direct answer to the first question. To answer the second question, we develop a procedure to calculate an approximation of the expected value of P90-P10 of J (the difference between the 90th and the 10th percentiles of the pdf of J) conditional to observed data from the estimated value of mutual information. This chapter is organized as follows: first we present some basic concepts in information theory. Then we present the methodology, followed by the application of the proposed method to four examples.

4.1 Basic Concepts in Information Theory

4.1.1 Discrete Form

Consider a discrete random variable X, with the set of all possible realizations of X given by

$$\{x_i, i = 1, 2, ..., N\}. \tag{4.1}$$

Let $P_X(x_i)$ denote the probability of $X = x_i$, we must have $\sum_{i=1}^{N} P_X(x_i) = 1$.

The *information content* of x_i is defined as

$$I(x_i) \equiv \log\left(\frac{1}{P_X(x_i)}\right) = -\log P_X(x_i), \tag{4.2}$$

and may be thought of as a measure of one's surprise at the outcome of an experiment that produces the sample x_i. Note that the base of the logarithm can be any real number greater than 1; the information content has different units depending on what base is used. Most commonly, $\log_2$ and natural logarithm are used and the corresponding units are bits and nats. We use the natural logarithm throughout this research.

There are some appealing properties of the above definition. First, this definition is intuitive because when the probability of x_i is high, observing x_i does not provide as much information as when the probability of x_i is low. In the extreme case, the information content of x_i is 0 if $P_X(x_i) = 1$; the receiver of a message knows that the message will be x_i, he does not get any information when he receives the message. Secondly, the information is always nonnegative. It is not possible to lose information by receiving a message.

The information measure considers each *outcome* of the random variable X but does not say anything about the random variable itself. This can be accomplished by calculating the average information of the random variable X which is given by

$$H(X) \equiv \sum_{i=1}^{N} P_X(x_i) I(x_i) = -\sum_{i=1}^{N} P_X(x_i) \log P_X(x_i). \tag{4.3}$$

$H(X)$ is called the *entropy* of X and is a measure of uncertainty about X. It is a *real number* that represents the average information content one is missing when one does not know the value of the random variable X. $H(X)$ may also be thought of as the expected information gain one will obtain by conducting an experiment to sample X.

Now we introduce a second discrete random variable Y, which, for example, can be an output signal from the same system as X. We assume the set of all possible realizations of Y is given by

$$\{y_j, j = 1, 2, ..., M\}, \tag{4.4}$$

where $\sum_{j=1}^{M} P_Y(y_j) = 1$.

Suppose that we can measure the value of Y and use that measurement to make an inference about the value of X. This means once the value of Y is known, say $Y = y_j$, the probability of each x_i, $i = 1, 2, ..., N$, will change, often in a way that will cause several x_i's to have much a higher probability than the rest. The posterior distribution of X conditioned to y_j will have a new entropy value, which is called the posterior entropy or specific conditional entropy of X conditioned to $Y = y_j$ and denoted by $H(X|Y = y_j)$ or $H(X|y_j)$. $H(X|Y = y_j)$ is given by

$$H(X|Y = y_j) = -\sum_{i=1}^{N} P_{X|Y}(x_i|y_j) \log P_{X|Y}(x_i|y_j). \tag{4.5}$$

Note that, for each plausible measurement outcome of Y, the probability distribution of X may change in a different way, which means the specific conditional entropy $H(X|Y = y_j)$ varies depending on which y_j is observed, $j = 1, 2, ..., M$. The average of $H(X|y_j)$ for different y_j's, $j = 1, 2, ..., M$, is called the *conditional entropy* of X given Y. It is calculated as

$$\begin{aligned} H(X|Y) &\equiv \sum_{j=1}^{M} H(X|Y = y_j) P_Y(y_j) \\ &= -\sum_{j=1}^{M} P_Y(y_j) \left(\sum_{i=1}^{N} P_{X|Y}(x_i|y_j) \log P_{X|Y}(x_i|y_j) \right) \\ &= -\sum_{i=1}^{N} \sum_{j=1}^{M} P_{X,Y}(x_i, y_j) \log P_{X|Y}(x_i|y_j). \end{aligned} \tag{4.6}$$

Conditional entropy is a measure of the average information in X given that Y is known. In other words, it is the average remaining uncertainty of X after observing Y.

The difference between $H(X)$ and $H(X|Y)$ is called *mutual information* and denoted by $I(X;Y)$. It represents the average information about X gained when observing Y, or, a representation of how much knowledge of one of the random variables reduces the uncertainty in the other,

$$I(X;Y) = H(X) - H(X|Y). \tag{4.7}$$

As shown in Appendix C.1, mutual information can be expressed as:

$$I(X;Y) = \sum_{i=1}^{M}\sum_{j=1}^{N} P_{X,Y}(x_i, y_j) \log \frac{P_{X,Y}(x_i, y_j)}{P_X(x_i)P_Y(y_j)}, \tag{4.8}$$

so $I(X;Y) = I(Y;X)$.

As its name implies, mutual information is a representation of the common information between the two variables. $I(X;Y)$ quantifies the dependence between X and Y, e.g., if X and Y are independent, then $P(X|Y) = P(X)$, so $H(X|Y) = H(X)$ and Eq. 4.7 gives $I(X;Y) = 0$, which is the minimum value of $I(X;Y)$.

The relationship between different quantities in information theory can be summarized by the Venn diagram in Fig. 4.1. The area contained by the left circle (red and violet) is the entropy $H(X)$. The area contained by the right circle (blue and violet) is the entropy $H(Y)$. The intersection of two circles is $I(X,Y)$. The union of two circles is $H(X,Y)$, which is the entropy of the bivariate random variable, (X, Y). The red and blue areas, respectively, are $H(X|Y)$ and $H(Y|X)$.

4.1.2 Differential Form

When X is a continuous random variable, the *differential entropy* of X can be written as

$$h(X) = -\int_R p_X(x) \log p_X(x) dx, \tag{4.9}$$

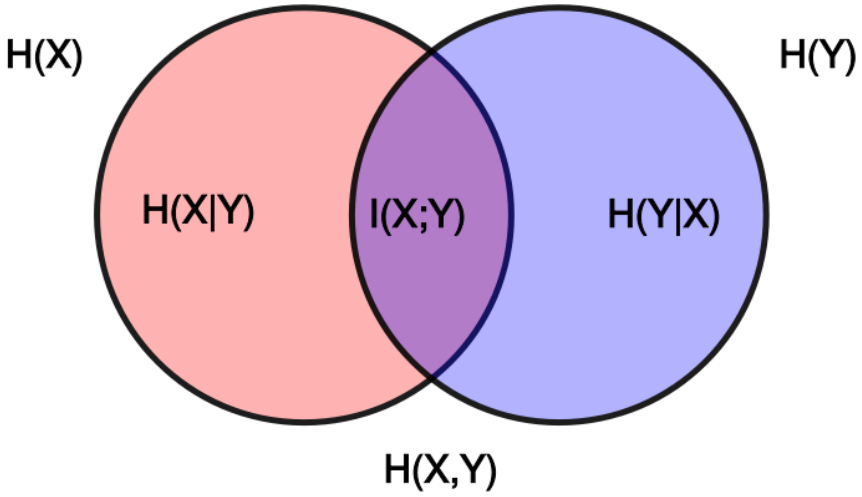


Figure 4.1: Venn diagram representing the relationship between different quantities in information theory.

where $p_X(x)$ is the probability density function of X. Throughout, small h is used for differential entropy while capital H is used for discrete entropy.

The conditional entropy of X given Y, where both X and Y are continuous random variables, is given by

$$h(X|Y) = \int_R \int_R p_{XY}(x,y) \log p_{X|Y}(x|y) dxdy, \tag{4.10}$$

where $p_{XY}(x,y)$ is the joint pdf of X and Y and $p_{X|Y}(x|y)$ is the conditional pdf of X given $Y = y$.

The mutual information between continuous random variables X and Y is then

$$I(X;Y) = h(X) - h(X|Y) = \int_R \int_R p_{XY}(x,y) \log \left(\frac{p_{XY}(x,y)}{p_X(x) p_Y(y)} \right) dxdy. \tag{4.11}$$

For the case where X and Y are vectors of dimensions N and M, respectively, the mutual information can be written in the same way:

$$I(X;Y) = \int_{R^N} \int_{R^M} p_{XY}(x,y) \log \left(\frac{p_{XY}(x,y)}{p_X(x) p_Y(y)} \right) dxdy, \tag{4.12}$$

where the integral takes place over all dimensions of X and Y.

Consider the special case where X and Y are Gaussian random vectors, with means μ_X and μ_Y and covariances C_{XX} and C_{YY}. The probability density function for X is:

$$p_X(x) = \frac{1}{(2\pi)^{N/2}} \frac{1}{\sqrt{\det(C_{XX})}} \exp(-\frac{1}{2}(x-\mu_X)^T C_{XX}^{-1}(x-\mu_X)). \tag{4.13}$$

Substituting the above equation into Eq. 4.9, we obtain the expression for the differential entropy of X (Cover and Thomas, 1991, Table 16.1):

$$h(X) = \frac{1}{2}\log((2\pi e)^N |C_{XX}|), \tag{4.14}$$

where $|*|$ denotes the determinant of a matrix. Similarly, the expressions for $h(X)$ and $h(X,Y)$, respectively, are

$$h(Y) = \frac{1}{2}\log((2\pi e)^M |C_{YY}|), \tag{4.15}$$

and

$$h(X,Y) = \frac{1}{2}\log((2\pi e)^{N+M} |C|), \tag{4.16}$$

where

$$C = \begin{bmatrix} C_{XX} & C_{XY} \\ C_{YX} & C_{YY} \end{bmatrix}, \tag{4.17}$$

and $C_{YX} = C_{XY}^T$ is the cross-covariance matrix between X and Y.

Using the above expressions and the relationship depicted by Fig. 4.1, the mutual information between two Gaussian vectors X and Y is then

$$\begin{aligned} I(X;Y) &= h(X) + h(Y) - h(X,Y) \\ &= \frac{1}{2}\log((2\pi e)^N |C_{XX}|) + \frac{1}{2}\log((2\pi e)^M |C_{YY}|) - \frac{1}{2}\log((2\pi e)^{N+M}|C|) \\ &= \frac{1}{2}\log\left(\frac{|C_{XX}||C_{YY}|}{|C|}\right). \end{aligned} \tag{4.18}$$

4.1.3 Properties of Entropy and Mutual Information

Entropy is the measurement of the uncertainty of a variable which qualitatively reflects the shape of its probability density function. Because the definition of entropy is $h(x) = \int p(x) \log(\frac{1}{p(x)}) dx$, the entropy is equal to the expectation of $\log(\frac{1}{p(x)})$.

As shown in Cover and Thomas (1991, Chapter 11), among all pdf's of a random variable x such that the support of each pdf is [a, b], the uniform distribution of x is the pdf of maximum entropy. Fig. 4.2 shows two pdf's defined on [0, 4]. The red one is a uniform distribution and it has higher entropy than the green one. Also shown in Cover and Thomas (1991, Chapter 11), among all distributions with the same standard deviation σ, the normal distribution has the highest entropy. Or, to put it another way, among all distributions with the same entropy, the normal distribution has the lowest standard deviation.

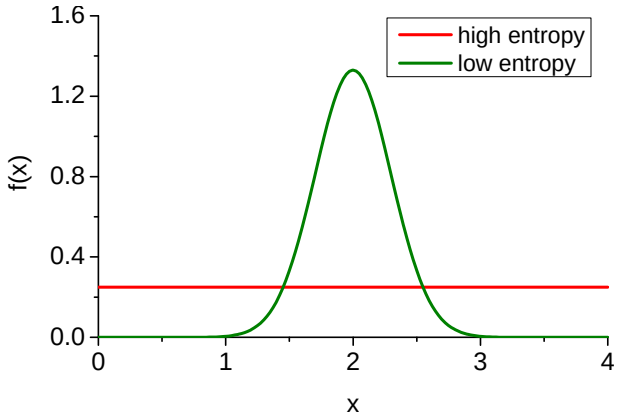


Figure 4.2: High and low entropies

The definition of entropy, Eq. 4.3 or Eq. 4.9, indicates that entropy does not depend on the actual value of X but only on the distribution of X. For a multi-modal distribution, entropy depends on the relative height of the modes, not on the distance between the modes (Fig. 4.3). This is an important property of entropy that makes it different from the popular standard deviation approach.

4.1.4 Mutual Information as Uncertainty Reduction in a Linear Inverse Problem

We start by defining a typical linear inverse problem where the vector of model parameters m is an N_m-dimensional Gaussian random vector, $m \sim N(m_{\text{prior}}, C_M)$. In a linear inverse problem, the vector of predicted data is a linear transformation of m, i.e.

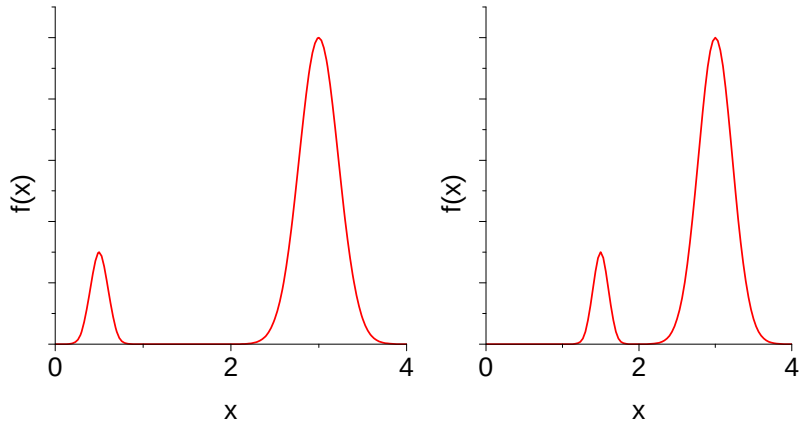


Figure 4.3: Two Gaussian mixture distributions having almost the same entropy even though distances between the modes are not the same. They are not exactly equal because the values of the pdf between the modes are not exactly zero.

$D = Gm$, where G is a $[N_D \times N_m]$ matrix. We assume that the noise ϵ_D in the observed data vector has a Gaussian distribution, $\epsilon_D = N(0, C_D)$ and that ϵ_D is independent of m. The random observed data D_{obs} is a function of both the model parameter and the noise, $D_{obs} = Gm + \epsilon_d$. An inverse problem is the problem of determining the posterior pdf of m conditioned to a realization d_{obs} of the vector of observed data, D_{obs}. Note that D_{obs} is used to denote a random vector whereas d_{obs} represents a realization of D_{obs}. An inverse problem does not usually have a unique solution but an infinite number of equally good solutions. Therefore, in solving an inverse problem, we are interested in quantifying the uncertainty of m conditioned to the observed data, which in practice is usually done by generating a sample set of realizations $\{m_j\}_{j=1}^{N_e}$ which represent an (approximate) sampling of $p(m|d_{obs})$. For a general (non-linear) inverse problem, the posterior uncertainty (pdf) of m is dependent on the actual measurement (that is, different realizations of D_{obs} will lead to different posterior pdf's for m). However, for a linear inverse problem, the posterior uncertainty of m is independent of the value of d_{obs}. As shown in Oliver et al. (2008), for the linear case, the probability distribution of m conditional to d_{obs} is a multivariate Gaussian distribution with mean m_∞ and covariance matrix C_{MAP}. Even though the expression for C_{MAP} is derived using a specific realization d_{obs} of D_{obs}, it does not contain d_{obs} and therefore

is independent of the actual realization.

It can be shown (see Appendix C.2) that the mutual information between m and D_{obs}, $I(m, D_{obs})$ is related to the MAP estimate of m by the following equation:

$$I(m, D_{obs}) = \frac{1}{2} \log \left(\frac{|C_M|}{|C_{MAP}|} \right). \tag{4.19}$$

Note that mutual information defines the relationship two random variables or vectors (not between, say, a random variable and a realization of the other) and it takes into account all possible realizations of both variables/vectors. Therefore, although mutual information is related to C_{MAP} in the linear-Gaussian case, its nature is very different from that of C_{MAP}, which is the covariance of the posterior model when a particular realization of the vector of observed data is assimilated.

The relationship of Eq. 4.19 also applies to any variable that is a linear function of m. Suppose J is a reservoir variable that we want to investigate and J is also a linear transformation of m, $J = G_J m$, where G_J is a $[N_J \times N_m]$ matrix. Let C_J denote the prior covariance of J and C_J^u be the posterior covariance of J. It can be shown (see Appendix C.2) that the mutual information between J and D_{obs}, $I(J, D_{obs})$ is related to the posterior covariance matrix of J by the following equation:

$$I(J, D_{obs}) = \frac{1}{2} \log \left(\frac{|C_J|}{|C_J^u|} \right). \tag{4.20}$$

Again, for the linear-Gaussian case, C_J^u is independent of the actual realization (d_{obs}) of D_{obs}.

When J is a scalar variable, e.g. net present value or cumulative water production, the above equation reduces to

$$I(J, d_{obs}) = \log \left(\frac{\sigma_J}{\sigma_J^u} \right), \tag{4.21}$$

where σ_J and σ_J^u are the prior and posterior standard deviation of J.

4.2 Proposed Method to Quantify Uncertainty Reduction

Let J be the reservoir variable of interest. This method is primarily applicable for the cases where J is an 1-dimensional variable, e.g. cumulative oil production or net present value at the time of interest. If, for example, we are interested in the cumulative oil production at several different times, the procedure can be repeated for the cumulative oil production at each time. Let μ_J, σ_J and P90 - P10 be the mean, standard deviation and the difference between the 90th and the 10th percentiles of the pdf of J. Now suppose that we are going to perform a surveillance operation to obtain the vector of observed data D_{obs}. The mean of the posterior pdf of J depends on the outcome of the measurement (a realization of D_{obs}), even for the linear-Gaussian case. However, it can be shown that the average of the means of J conditioned to different plausible measurement outcomes is still equal to the mean of the prior model (Appendix C.3), for both linear and nonlinear cases. Therefore we are not interested in investigating the change in the mean of J from the prior model to the posterior model. We will instead try to quantify the average change in the standard deviation (or the change in P90 - P10) of J. Our proposed method can be outlined as follows:

1. Generate N_e realizations from the prior reservoir model and run the simulation for each of those realizations

2. Let N_d be the number of observed data that we would collect for a certain surveillance scenario. From the simulation results, extract the values of the predicted data and add noise to each vector of predicted data by sampling $N(0, C_D)$ to obtain N_e realizations of D_{obs}. Then, arrange these realizations into a matrix X of size $N_e \times N_d$. Each column of this matrix this matrix represents one of the N_e realizations of the vector of observed data D_{obs}. Also extract the values of the reservoir variable J from the N_e simulation outputs and arrange these values in a matrix Y of size $N_e \times 1$. It is important to note that this procedure generates N_e specific pairs of (J, d_{obs}), each of which corresponds to a specific realization of m.

3. Calculate the prior entropy of J, $h(J)$, from Y as discussed later.

4. Calculate the mutual information between J and D_{obs}, $I(J, D_{obs})$, using N_e associated (J, d_{obs}) pairs in X and Y using one of the procedures discussed later. Note that $I(J, D_{obs})$ is the mutual information between a random variable and a random vector and we only obtain a single scalar value by using N_e pairs of (J, d_{obs}). The more (J, d_{obs}) pairs we have, the more accurate the mutual information estimate is expected to be.

   Then, calculate the conditional entropy of J given D_{obs}, $h(J|D_{obs})$, by

$$h(J|D_{obs}) = h(J) - I(J, D_{obs}). \tag{4.22}$$

5. Reconstruct the posterior pdf of J under the condition that it has an entropy equal to $h(J|D_{obs})$. This pdf represents the average uncertainty of J if D_{obs} is measured. As shown in details later, we propose three different ways of doing this, with increasing degrees of accuracy and complexity.

6. Using the methodology discussed later, calculate the standard deviation (or P90 - P10) of the reconstructed posterior pdf of J conditional to D_{obs}. Multiply this result by a correction factor between 1.0 and 1.359 (Appendix C.4) to obtain an estimate for the expectation of the posterior standard deviation (or P90 - P10) of J. This correction factor is empirical and as discussed later, the extent of its general applicability is not known.

In the following subsections, we show how each step is performed.

4.2.1 Step 1: Generate realizations

To generate realizations of the prior reservoir model, we first require a description of the prior reservoir model. Here, we assume that the prior pdf for the vector of model variables generally has a Gaussian probability density function

$$m \sim N(m_{\text{pr}}, C_M), \tag{4.23}$$

where m_{pr} is the N_m-dimensional mean of the vector of model variables, m, and C_M is the prior covariance matrix.

For reservoir problems of interest, C_M specifies the correlation between reservoir model variables. For example, in a problem where we consider $\ln k$, ϕ and $\ln k_z$ as model variables, the covariance matrix can be written as:

$$C_M = \begin{bmatrix} C_{\ln k} & C_{\ln k,\phi} & C_{\ln k,\ln k_z} \\ C_{\ln k,\phi} & C_{\phi} & C_{\phi,\ln k_z} \\ C_{\ln k,\ln k_z} & C_{\phi,\ln k_z} & C_{\ln k_z} \end{bmatrix}, \tag{4.24}$$

If we let L denote the lower triangular part of the Cholesky decomposition of C_M, i.e., $C_M = LL^T$, then the realizations of m can be created using

$$m_j = m_{prior} + LZ_j, j = 1, 2, ..., N_e, \tag{4.25}$$

where Z_j is a vector of independent identically distributed (i.i.d.) standard random normal deviates.

It is important to note that if the prior model is not Gaussian, our overall procedure still applies but requires some form of the prior distribution that lets us generate a set of N_e realizations.

4.2.2 Step 2: Obtain values of observed data and reservoir variable J

To obtain the observed data, we need to extract the predicted data from simulation results and add noise to them. For the predicted data that are directly output by the simulator such as water rate, oil rate and cumulative oil production, we simply read off the values from the simulator output files (Eclipse in this research). For the data type that is not a direct output of the simulator, post-processing of the simulator results is often required. Here, the only such type of data considered is the location of the water front in a 1-D water flooding problem (which is assumed to be measurable using nano-tracers). It is calculated

as follows:

- At the simulation time of interest, output water saturation values at all gridblocks between the injector and producer. For a one-dimensional problem where water is injected at the left and produced at the right, the water saturations will gradually change from 1 - S_{or} at the leftmost gridblock to S_{wi} at the rightmost gridblock, where S_{or} and S_{wi} are the residual oil saturation and the initial water saturation, respectively.

- Define a threshold value for the water saturation at the water front. In our case, we use 0.5 as the threshold value

- Proceed from the rightmost block to the leftmost block to determine the index of the first gridblock whose water saturation exceeds 0.5. Interpolate between this gridblock and the gridblock to its right to determine the position where the water saturation is exactly 0.5 (we treat the position of the water front as a floating point value). Record this value as the location of the water front.

Values of reservoir variable J are obtained directly from the simulation results with no noise being added.

4.2.3 Step 3: Calculate the prior entropy of J

There are multiple methods to calculate the entropy from an ensemble of data points. The methods range from simple methods such as binning to more sophisticated methods such as sample spacing. Based on the comments of Kraskov et al. (2004), the most reliable method is seemingly based on sample spacings (Beirlant et al., 1997). This method takes into account the distance between sample points. It is expected that for a pdf of low entropy, the distance between sample points will be small and for a variable of high entropy, the distance between sample points will be large. Although this method is only applicable to one dimensional data, it is applies for the problems considered here because the reservoir variable J is only one-dimensional. The method is outlined as follows:

- Sort all values of J from low to high.

- Calculate the entropy using the estimator

$$h(J) = \frac{1}{n}\sum_{i=1}^{n-k} \ln\left(\frac{n}{k}(J_{i+k} - J_i)\right) - \psi(k) + \ln(k), \tag{4.26}$$

where n is the total number of values of J, k is the order of spacing and $\psi(x)$ is the digamma function, $\psi(x) = \frac{d}{dx}\ln\Gamma(x) = \frac{\Gamma'(x)}{\Gamma(x)}$, where $\Gamma(x)$ is the gamma function (generalized factorial function). Fig. 4.4 shows plot of $\psi(x)$ versus x.

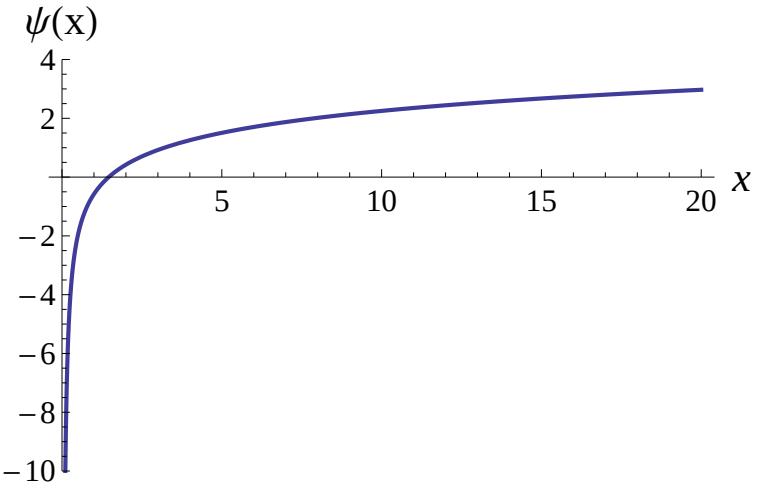


Figure 4.4: Plot of $\psi(x)$ versus x

It is not generally clear how to choose the optimal value of k as it is dependent on the probability distribution of J and the number of values of J in the ensemble. Small values of k will lead to low bias (also called systematic error, which is the difference between the true value of $h(J)$ and the expectation of the estimator of Eq. 4.26) but high variance (also called statistical error, which is the variance of the estimated values over different data sets) and vice versa. Here, we do not delve into investigating this issue but simply choose $k = 1$ to maintain a low bias.

Before applying our estimator to a real problem, we test its reliability by calculating the entropies of 10,000 data sets (ensembles) of size 100 each. The data sets are all sampled from the standard Gaussian distribution $N(0, 1)$. Note that one data set gives one entropy value and 10,000 data sets give 10,000 entropy values. The bias of the method is defined

as difference between the average of the 10,000 calculated entropy values and the analytical entropy value of the standard Gaussian distribution. The variance (or standard deviation) of the method is defined as the variance (or standard deviation) of the 10,000 entropy values. We then repeat the test using 10,000 data sets of size 1,000 and 10,000 data sets of size 10,000. Table 4.1 shows the results of our test. While the bias is quite small regardless of the ensemble size, the standard deviation is quite high for small ensemble sizes. For ensembles of size 100, the standard deviation of the estimate is 7.4% of the analytical value, which means the entropy estimate using a single ensemble of size 100 can be quite different from the analytical value. Therefore, we recommend using an ensemble size of at least 1,000 to estimate the entropy.

Table 4.1: Performance of entropy estimation (h) using sample spacing. The analytical entropy of a standard normal distribution is calculated using Eq. 4.14 and has a value of 1.419.

	E[h]	bias	stdev[h]
$N_e = 100$	1.376	-0.043	0.106
$N_e = 1{,}000$	1.412	-0.007	0.034
$N_e = 10{,}000$	1.418	-0.001	0.011

4.2.4 Step 4: Calculate mutual information between J and D_{obs}

Obtaining a good estimate of the estimate of mutual information is a crucial step in our procedure. Many papers exist (Roulston, 1999; Darbellay and Vajda, 1999; Cellucci et al., 2005; Kraskov et al., 2004) which focus on the estimation of mutual information. However, because mutual information represents the general relationship between variables, there is no explicit expression to calculate mutual information, and the estimation of mutual information is prone to errors. Even for the most simple case involving one dimensional data, i.e. estimating $I(X, Y)$ where X and Y are both real random variables, the estimation error may be significant and many papers have been devoted to the development of correction terms or alternate methods to reduce the error (see, for example, Roulston (1999); Cellucci et al. (2005); Darbellay and Vajda (1999)). Most of these methods only apply for the

one dimensional case. Several methods exist to estimate mutual information for the high dimensional case but they are not without their limitations. Because of the complexity involved, we devote a separate section (Section 4.3) to review the available methods and discuss their applicability.

4.2.5 Step 5: Reconstruct the posterior pdf of J

Recall that the conditional entropy, $h(J|D_{obs})$, calculated in the previous step is the expectation of the posterior entropy of J conditioned to different realizations of D_{obs}. Although the conditional entropy by itself is a measure of uncertainty, it is desirable that other formulations to quantify uncertainty, e.g. standard deviation or P90 - P10, can be estimated. We propose a method that can relate the value of conditional entropy to the average value of the posterior standard deviation (or P90 - P10). This involves reconstructing a pdf that represents the average of all plausible posterior pdf's of J. We require that the pdf we are about to construct satisfies two conditions: (1) Its entropy is equal to the conditional entropy of J, $h(J|D_{obs})$, and (2) It has a shape similar to the shapes of the plausible posterior pdf's of J. The reason we need to impose the second condition is because pdf's of different shapes can have different standard deviation (or P90 - P10) values even though they all have the same entropy. Note that, in this context, "shape" does not refer to the actual height and spread of the pdf, only the the relative height of one part of the pdf to others. Rigorously, two pdf's are said to have the same shape if the ratio of the heights at any two percentiles of one pdf is equal to that at the same two percentiles of the other pdf. Intuitively, two pdf's are said to have the same shape if they can be stretched/compressed and translated to look the same as the other. For example, any two Gaussian pdf's have the same shape (Fig. 4.5)

How to determine a reasonable shape is a difficult question for two reasons: (1) The shapes of the posterior pdf's are often not Gaussian and (2) Different plausible posterior pdf's can have different shapes. There are several methods to tackle this problem, given J is an 1-dimensional variable:

1. Use a Gaussian shape for the posterior pdf. This is the easiest but is not accurate if

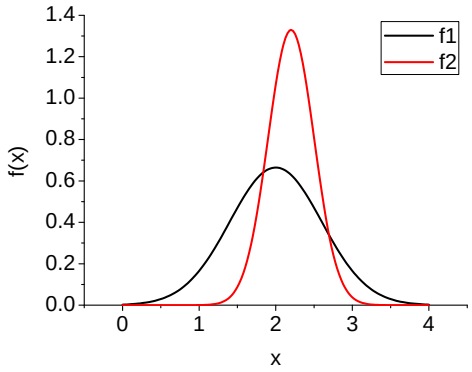


Figure 4.5: Two Gaussian distributions have the same shape but different standard deviation and mean

the posterior pdf is far away from a Gaussian distribution.

2. Assume the posterior shape of the pdf of J does not change from its prior shape. This can lead to a high error if the posterior shape deviates significantly from the prior shape.

3. For each potential surveillance scenario, perform history matching 3 times, each time with a different prior ensemble and a different realization of D_{obs}. The shapes of the three posterior ensembles are then used to construct three pdf's that have entropy equal to the conditional entropy calculated in step 4; see below for details how to do this. In our experience, this is the most accurate method to tackle this problem.

For the first method, since we can directly calculate the standard deviation (or P90 - P10) of the Gaussian distribution given a specific entropy (see Eq. 4.14), there is nothing else we need to do. For the other two methods, we either have one prior ensemble of values of J (method 2) or 3 posterior ensembles of values of J (method 3), where the histogram of each ensemble represents a plausible shape of the posterior pdf of J. We need to modify the values in each ensemble so that its entropy is equal to the conditional entropy $h(J|D_{obs})$. This can be thought of as stretching the histogram so that its entropy reaches our desired value. This can be performed as follows:

1. From the ensemble of values, calculate the ensemble mean μ.

2. Move all points closer to or farther from the ensemble mean, μ, by replacing their current values with new values:

$$J_i \rightarrow k(J_i - \mu) + \mu, \tag{4.27}$$

where k is the stretching factor. Ideally, k should be equal to the ratio of the standard deviation of the target distribution to the current distribution, i.e. $k = \frac{\sigma_{\text{target}}}{\sigma_{\text{current}}}$, then this step only needs to be performed once. In reality, since we do not know how calculate the standard deviation of the target distribution from its entropy unless it is a Gaussian, we have to perform the "stretching" iteratively, each time with a value of k that is approximately equal to $\frac{\sigma_{\text{target}}}{\sigma_{\text{current}}}$. The value of k at each iteration is calculated by assuming the distribution is Gaussian and by making use of Eq. 4.14. From Eq. 4.14, the current and target standard deviations, respectively, are approximated as

$$\sigma_{\text{current}} \approx \sqrt{\frac{\exp(2 * h_{\text{current}})}{2\pi e}}, \tag{4.28}$$

and

$$\sigma_{\text{target}} \approx \sqrt{\frac{\exp(2 * h_{\text{target}})}{2\pi e}}, \tag{4.29}$$

where h_{current} and h_{target}, respectively, are the current and the target entropies of the distribution. Taking the ratio of σ_{target} to σ_{current} gives us the value of k:

$$k = \sqrt{\exp\left(h_{\text{target}} - h_{\text{current}}\right) * 2}. \tag{4.30}$$

3. Repeat step 2 until we obtain a pdf with entropy equal to our target entropy.

At the end of this procedure, we obtain 1 or 3 ensembles of values of J, each of which approximate the expectation of the pdf of J conditional to D_{obs} and, in step 6, statistical analysis can be done on these pdf's to obtain metrics such as standard deviation or P90 - P10. In the case there are 3 ensembles, we calculate the standard deviation or P90 - P10 of

all of them, and then average the results.

4.2.6 Step 6: Calculate standard deviation and P90 - P10

If we have more than 1 ensemble that represent the expected posterior pdf, we calculate the standard deviation or P90 - P10 for each ensemble, and then average the results. The estimator for calculating standard deviation is well-known and can be found in most statistics books, so it will not be repeated here. To calculate P90 - P10 from a set of samples, we simply calculate the values of the 90^{th} and 10^{th} percentiles and then subtract P10 from P90. The procedure we use to calculate the value of the p^{th} percentile of a list of N values is identical to the one used in Microsoft Excel®. Suppose we have an ensemble of N_e values of J, the procedure is as follows:

- Sort the values of J in ascending order, $J_1 \leq J_2 \leq \cdots \leq J_{N_e}$
- Calculate the rank:

$$n = \frac{p}{100}(N_e - 1) + 1. \tag{4.31}$$

- If n is an integer then the p^{th} percentile is simply the n^{th} value of the ordered list. If n is a floating point value, let k be the integer part and and d be the decimal part of n. The p^{th} percentile is calculated using interpolation between the k^{th} and $(k+1)^{\text{th}}$ values of the ordered list:

$$J\ (p\text{th percentile}) = J_k + d(J_{k+1} - J_k). \tag{4.32}$$

There is one last issue, that is, our computational experiments show that the values of σ_J and P90 - P10 calculated from our reconstructed pdf are usually smaller than the actual average values of σ_J and P90 - P10. We found that a correction factor from 1.0 to 1.359 should be used to bring our estimated value closer to the true value. Appendix C.4 shows how this correction factor is developed. This correction factor is calculated based on the value of the mutual information, so each scenario has one correction factor, regardless

of how many ensembles (1 or 3) are used to estimate the shape of the posterior pdf in that scenario.

4.3 Estimating Mutual Information

4.3.1 The straightforward binning method

We will first show how this method is done for the case where X and Y are both 1-D variables, i.e. scalars, then show how the method can be extended to higher dimensional cases. This method is directly based on the definition of the mutual information for continuous variables which is given by

$$I(X;Y) = \int_R \int_R p_{XY}(x,y) \log\left(\frac{p_{XY}(x,y)}{p_X(x)p_Y(y)}\right) dxdy. \tag{4.33}$$

The values of X and Y can be partitioned into bins of finite sizes. Let k_X and k_Y respectively be the number of bins for X and Y. The mutual information can be approximated by the finite sum

$$I_{binning}(X;Y) = \sum_{i=1}^{k_X} \sum_{j=1}^{k_Y} P_{XY}(i,j) \log\left(\frac{P_{XY}(i,j)}{P_X(i)P_Y(j)}\right), \tag{4.34}$$

where $P_{XY}(i,j) = \int_i \int_j p_{XY}(x,y)dxdy$, $P_X(i) = \int_i p_X(x)dx$, $P_Y(j) = \int_j p_Y(Y)dy$ and $\int_i$ is the integral over bin i. These integrals are approximated as discussed in the next paragraph.

Let n be the total number of sample points of (X, Y). Let $n_X(i)$ be the number of points falling into the i^{th} bin of X, $n_Y(j)$ be the number of points falling into the j^{th} bin of Y, and $n_{XY}(i,j)$ be the number of points that belong to both the i^{th} bin of X and the j^{th} bin of Y. The value of $P_{XY}(i,j)$, $P_X(i)$, and $P_Y(j)$ can simply be approximated as:

$$\begin{aligned} P_{XY}(i,j) &= n_{XY}(i,j)/n, \\ P_X(i) &= n_X(i)/n, \\ P_Y(j) &= n_Y(j)/n. \end{aligned} \tag{4.35}$$

The value of this estimate is dependent on the binning scheme and is expected to converge to the true value when $n \to \infty$. Choosing an appropriate number of bins is important as it can significantly impact the value of the estimate. Cellucci et al. (2005) summarize available formulas to calculate an appropriate number of bins. However, even when the proper number of bins is used, the method still suffers from systematic error. Roulston (1999) and Steur et al. (2002) presented procedures to partially correct for these errors.

Although the aforementioned issues with the binning method are not negligible, the biggest shortfall of this method is actually its inability to accurately estimate mutual information when X and Y are multi-dimensional vectors. Theoretically, we can still apply the same principle to estimate mutual information, except now each dimension of X will be partitioned into k_{X_i} bins, $i = 1, 2, ..., N$ and each dimension of Y will be partitioned into k_{Y_i} bins, $i = 1, 2, ..., M$, where N and M , respectively, are the dimensions of X and Y. We can easily see that this method is not practical unless N and M are small or we generate a very large number of sample points. For example, consider the case where X is 5 dimensional and Y is 1 dimensional. If we partition each dimension of X and Y into 2 bins (the minimum possible number of bins), it would result in a total of $2^{5+1} = 64$ hyper-rectangles in the 6-D space. Calculating mutual information requires the estimate of the probability that (X, Y) is within each of these 64 hyper-rectangles. Because of the exponential relationship, the number of possible hyper-rectangles increases sharply as the number of dimensions increases and there would be too few points in each hyper-rectangle for the calculated probability to be representative unless n, the number of samples of (X, Y), is impractically large.

4.3.2 The k^{th} nearest neighbor method

The k^{th} nearest neighbor algorithm (Kraskov et al., 2004) is a nonparametric method to estimate mutual information. The procedure of this method is summarized as follows:

- Let X and Y be N-dimensional and M-dimensional data, respectively and let $Z = (X, Y)$.

- For the the i$^{\text{th}}$ sample of Z, find the k$^{\text{th}}$ closest point to this i$^{\text{th}}$ point. Here the distance between two points is defined as $||z - z'|| = \max\{||x - x'||, ||y - y'||\}$, where any norm can be used for $||x - x'||$ and $||y - y'||$ (the norms do not need to be the same).

- Let $\epsilon(i)/2$ be the distance from the i$^{\text{th}}$ point, z_i, to its k$^{\text{th}}$ nearest neighbor. Using this distance, draw horizonal and vertical bands around the i$^{\text{th}}$ point as illustrated in Fig. 4.6. More generally, we would form a hypersquare centered at z_i with each dimension equal to $\epsilon(i)/2$.

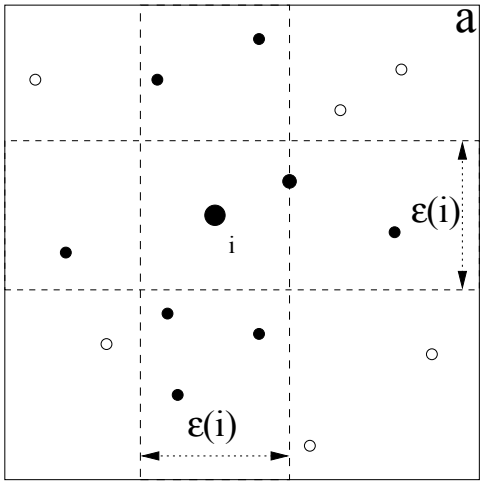


Figure 4.6: Estimation of mutual information using k$^{\text{th}}$ nearest neighbor for k = 1. In this example, $n_x = 5$ and $n_y = 3$

- Count n_x and n_y, which respectively represent the number of points that strictly lie inside the vertical and horizontal bands, respectively. It is expected that the value of n_x and n_y will be small for high mutual information and will be large for low mutual information. For example, consider the case where X and Y are 1-dimensional and perfectly correlated, i.e. the points on the cross-plot of X and Y follow a line. The band drawn around the i$^{\text{th}}$ point based on the distance to the nearest neighbor point will contain only the nearest neighbor point (excluding the i$^{\text{th}}$ point). On the other hand, when X and Y have little correlation, the band drawn in the same way will contain significantly more points as the points are more scattered.

- Repeat the above steps for all points in the cross plot and apply the following formula

to calculate mutual information:

$$I(X,Y) = \psi(k) - \langle \psi(n_x + 1) + \psi(n_y + 1) \rangle + \psi(n), \tag{4.36}$$

where $\langle ... \rangle$ denotes the average over all the points, k is the order of neighborhood, n is the number of points and $\psi(x)$ is the digamma function, $\psi(x) = \frac{d}{dx} \ln \Gamma(x) = \frac{\Gamma'(x)}{\Gamma(x)}$, where $\Gamma(x)$ is the gamma function (generalized factorial function).

The accuracy of this algorithm is somewhat dependent of the choice of k. Small values of k will lead to low bias but high variance and vice versa. The optimal choice of k varies with the number of data points and the nature of the problem. Here, we simply choose $k = 1$ to try to minimize the bias.

4.3.3 Using PCA to Improve Estimate of Mutual Information

Even though the nearest neighbor method is applicable to cases of any number of dimensions and, in our experience, it performs better than other methods given the same ensemble size, the estimate can still be inaccurate if the number of dimensions is too high for the number of realizations available. One way to partially remedy this problem is to use the Principal Component Analysis (PCA) to reduce the number of dimensions. The general concepts and a standard procedure to perform PCA can be found in Appendix C.5.

For high dimensional problems, even though we have no theoretical result that shows the nearest neighbor algorithm underestimates the value of mutual information, all computational experience we have to date indicates that this is the case. Intuitively, we believe this is due to the fact that for large dimensional problems, we can never generate enough samples of the probability density function (pdf) for (J, D_{obs}) to determine the nonlinear relationship between J and D_{obs}. We apply PCA with the hope that, by reducing the number of dimensions, it will make the relationship between J and D_{obs} easier to be recognized, thus increase the mutual information estimate. However, we have to be careful when using PCA because we could lose information when using PCA for two reasons: (1) PCA elimi-

nates the principle components that have the lowest variances and (2) PCA cannot preserve non-linear relationship. When this happens, some relationship between J and D_{obs} is lost and our estimate of mutual information is reduced, which is the exact opposite of what we wish to achieve.

When PCA is applied to the right problem, the error due to loss of information is expected to be negligible compared to the error caused by including non-informative dimensions. Here, we compare the mutual information estimate before and after PCA. If the mutual information estimate increases after applying PCA, it suggests that applying PCA is appropriate, otherwise we discard the mutual information estimate using PCA and keep the one obtained without PCA. There are also some other precautions that we should pay attention to when applying PCA:

1. Only apply PCA to the case where there is significant difference between the magnitudes of the greatest and the smallest singular values and only abandon the dimensions corresponding to the smallest singular values; we only want to abandon the dimensions of little significant so that we do not lose much information. If all singular values are roughly the same, this indicates the information contained in all dimensions are significant. In this case, abandoning even one dimension can lead to significant loss of information. Therefore, this case is not suitable for PCA.

2. One must be careful when applying PCA to a data set consisting of different observation types, e.g. a data set that contains both water cut data and grid block pressures. One major problem arises because of the difference in units and magnitudes of measurement values, although we can generally ameliorate this problem by normalization of data to generate dimensionless data which are all roughly on the same scale. We do this by dividing the data values in each dimension by the standard deviation of that dimension. A more important issue that occurs when reducing the number of data by PCA is that PCA is applied directly to the data set and does not account for the relationship between data and the reservoir variable J. Suppose that we have a data set that

contains 4 water rate measurements and 8 pressure measurements so that data vector is 12-dimensional, and that after PCA, the data vector is reduced to 3 dimensional, 1 of which comes from water rate measurements and 2 of which come from pressure measurements. Now suppose that when we estimate mutual information between our reduced data set and J, we find that the 2 pressure dimensions contribute little to the estimate of mutual information, i.e. there is no relation between pressures and J. Therefore it turns out that keeping 2 measurements of pressure is not useful and it would be better if we keep 3 dimensions from water rate instead. This illustrates the difficulty in trying to keep only the data that is most relevant using PCA.

3. There are also times when abandoning even one dimension leads to a significant underestimation of mutual information even though the singular value corresponding to that dimension is small. This can happen when each individual dimension does not provide useful information but the combination of all dimensions provide significant information about J.

As shown in the next two sections, the combination of the nearest neighbor method and PCA works well for the toy problem and the 1-D water flooding problem. However, the accuracy of the nearest neighbor method tends to decrease as the dimension of the observed data vector increases because the method relies on obtaining an accurate representation of the distribution of the observation vectors in a R^{N_d+1} space, where N_d is the dimension of the observed data vector and J is a real random variable. Therefore, the estimate is prone to error when the number of dimensions is so high that it is not computationally feasible to generate enough samples of the joint distribution of J and D_{obs} to fully represent the relationship between J and D_{obs}. For the last two problems, the number of dimensions can be as high as a few hundred so the applicability of the nearest neighbor algorithm is tenuous. For this reason, an alternative, more efficient method is needed to estimate the mutual information between J and D_{obs}.

4.3.4 Estimate Mutual Information Using Linear-Gaussian Assumption

In addition to the nearest neighbor algorithm, we explore a simple method based on the linear-Gaussian assumption. We refer to the resulting procedure for estimating mutual information as the linear-Gaussian method. Using Eq. 4.18, the mutual information between X and Y, where $X \subset R^{N_X}$ and $Y \subset R^{N_Y}$, can be calculated as follows,

$$I(X;Y) = \frac{1}{2}\log\left(\frac{|C_{XX}||C_{YY}|}{|C|}\right), \tag{4.37}$$

where

$$C = \begin{bmatrix} C_{XX} & C_{XY} \\ C_{YX} & C_{YY} \end{bmatrix}. \tag{4.38}$$

Here, C_{XX} and C_{YY}, respectively, are the auto-covariance of X and Y, $C_{YX} = C_{XY}^T$ is the cross-covariance matrix between X and Y and the vertical bars around a matrix denote the determinant of the matrix.

For a linear-Gaussian problem, the calculation of mutual information using Eq. 4.37 is exact. It can also be shown that, for the linear-Gaussian case, the mutual information is equal to the natural logarithm of the ratio of the prior standard deviation to the posterior standard deviation of the random variable J of interest (see Appendix C.2), thus the value of mutual information provides one way to quantify uncertainty reduction. For a general problem, however, Eq. 4.37 essentially ignores the non-linear relationship in the data. Therefore, we expect this method to underestimate the mutual information if there is a nonlinear relationship present in the data. This is actually what we experienced in the 1D water flooding problem discussed later, where the linear-Gaussian estimator gave a much lower estimate compared to the nearest neighbor estimator. However, our experience indicates that the nearest neighbor algorithm tends to significantly underestimate mutual information for large-dimensional problems. Thus, it is possible that Eq. 4.37 will be more accurate than the nearest neighbor algorithm when N_d is on the order of 50 or larger. We should note that Eq. 4.37 will underestimate mutual information whenever the relationship between J and

D_{obs} is dominated by nonlinearities.

The use of Eq. 4.37 requires the calculation of several matrices's determinants, which can be difficult if the matrices are large in size. Note that, in our problems, C_{XX}, C_{YY} and C are all symmetric and positive definite. The procedure we use to calculate the determinant of a symmetric, positive definite matrix C of size $n \times n$ is as follow:

1. Calculate the singular value decomposition (SVD) of C.

$$C = U\Lambda V^T, \tag{4.39}$$

   where Λ is a diagonal matrix whose diagonal elements, $\lambda_i, i = 1, ..., N$, are the singular values of C. The singular values of C are sorted from largest to smallest with λ_1 being the largest and λ_n being the smallest. Because C is symmetric and positive definite, the diagonal elements of Λ are also C's eigenvalues and are all positive.

2. Calculate the determinant of C by taking the product of all eigenvalues (singular values) of C in the following order:

$$|C| = \lambda_1\lambda_n\lambda_2\lambda_{n-1}..., \tag{4.40}$$

   which is done to prevent the immediate value of the product from going over the maximum value representable by the double-precision floating-point format (about 10^{308}) at any point during the calculation.

4.4 Toy Problem

The toy problem is a simple analytical problem where the model parameter m is a real variable, $m \sim N(2.4, 0.1)$. The ith component of the vector of observed data, $(D_{obs})_i$, is a nonlinear function of m and i, with measurement error $\epsilon_D \sim N(0, 0.01)$;

$$(D_{obs})_i = \hat{g}(m, i) + \epsilon_D = 1 - \frac{9}{2}\left(m - 2\frac{\pi}{3}\right)^2 + (i-1)\sin(m) + \epsilon_D, \tag{4.41}$$

where $\hat{g}(m,i) = 1 - \frac{9}{2}\left(m - 2\frac{\pi}{3}\right)^2 + (i-1)\sin(m)$. Note that D_{obs} is a function of m only:

$$D_{obs} = g(m) = \begin{bmatrix} \hat{g}(m,1) + \epsilon_D \\ \hat{g}(m,2) + \epsilon_D \\ ... \\ \hat{g}(m,N_d) + \epsilon_D \end{bmatrix}, \tag{4.42}$$

where N_d is the number of dimensions of D_{obs}. In this example, we perform experiments using $N_d = 1, 2, ..., 5$.

The variable J is also a nonlinear function of m and is given by

$$J = h(m) = m^3. \tag{4.43}$$

Our objective is to quantify the average uncertainty reduction in J when D_{obs} is obtained. Following our proposed procedure (section 4.2), in step 1, we generate 10,000 realizations of m from our prior model. Then, in step 2, we calculate the corresponding D_{obs} and J for each realization using Eqs. 4.42 and 4.43. In step 3, the prior entropy of J, $h(J)$ is calculated using the sample spacing method; see section 4.2.3. The result of our estimate is 3.090 nats. To validate this result, we calculate the prior entropy of J using a more rigorous method, using the definition of entropy, i.e.,

$$h(J) = -\int_R f_J(J) \log f_J(J) dJ. \tag{4.44}$$

To perform this integration, the analytical expression of $f_J(J)$ is needed. Note that because J is a monotonic function of m, each value of m corresponds to exactly one value of J. Furthermore, all values of m between two arbitrary numbers a and b correspond to all values of J between $h(a) = a^3$ and $h(b) = b^3$. Therefore, the probability that m falls between a and b is equal to the probability that J falls between $h(a)$ and $h(b)$. In other words, the area under the curve of $f_M(m)$ from m to $m + dm$ must be equal the area under the curve

of $f_J(J)$ from J to $J + dJ$, where $J = h(m)$ and $dJ = h(m + dm) - h(m)$. We can write this as

$$|f_J(J)dJ| = |f_M(m)dm|\,, \tag{4.45}$$

or

$$f_J(J) = \left|\frac{dm}{dJ}\right| f_M(m) = \left|\frac{1}{h'(m)}\right| f_M(m), \tag{4.46}$$

where $m = h^{-1}(J)$. After obtaining the pdf of J, numerical integration using the iterative trapezoidal rule is used to obtain the value of $h(J)$. The result of the numerical integration is 3.099 nats. Compared to the result obtained with the sample spacings method of 3.090 nats, we have good agreement between the two results.

In step 4 of our procedure, the mutual information between J and D_{obs}, $I(J, D_{obs})$, is calculated using the k$^{\text{th}}$ nearest neighbor method with $k = 1$; see section 4.3.2. Then, the conditional entropy of J given D_{obs}, $h(J|D_{obs})$ is calculated by subtracting the value of mutual information from the value of the prior entropy. We estimate the conditional entropy for 5 cases, where the number of observed data, i.e. the number of dimensions of D_{obs}, varies from 1 to 5. The results are shown in Table 4.2 but before discussing the results, we show how we validate the results. The conditional entropy of J given D_{obs} can also be calculated using numerical integration. Recalling that the conditional entropy is simply the expectation of the posterior entropy of J conditioned to D_{obs}, we can calculate the posterior entropy for different realizations of D_{obs} and take the average value. For this purpose, we use the 10,000 realizations of the vector of D_{obs} that we generated earlier as our plausible measurement outcomes. The pdf of m conditional to a realization of the N_d-dimensional vector of observed data, d_{obs}, is expressed as:

$$f_M(m|d_{obs}) = a\exp(-\frac{1}{2}(m-2.4)^T C_M^{-1}(m-2.4) - \frac{1}{2}(g(m)-d_{obs})^T C_D^{-1}(g(m)-d_{obs})) \tag{4.47}$$

where a is a normalizing constant and the expression for $g(m)$ is defined by Eq. 4.42. The posterior pdf of J conditioned to the i^{th} plausible measurement outcome can be calculated

using an equation similar to Eq. 4.46 :

$$f_J(J|d_{obs}) = \left|\frac{1}{h'(m)}\right| f_M(m|d_{obs}). \tag{4.48}$$

We can now perform numerical integration to obtain the entropy of each plausible posterior pdf of J and then average the results to obtain the conditional entropy $h(J|D_{obs})$. We consider the estimates of $h(J|D_{obs})$ obtained from this rigorous procedure as the true values and are shown in Table 4.2 along with the estimates using the k^{th} nearest neighbor method we obtained earlier.

Table 4.2: Comparison of $h(J|D_{obs})$ using 2 different methods

	$I(J;D_{obs})$ estimate	$h(J\|D_{obs})$ estimate	$h(J\|D_{obs})$ rigorous
$N_d = 1$	1.926	1.164	1.200
$N_d = 2$	2.389	0.701	0.689
$N_d = 3$	2.748	0.342	0.326
$N_d = 4$	2.991	0.099	0.060
$N_d = 5$	3.110	-0.02	n/a*

*numerical integration subroutine did not give reliable results because the slope of the pdf of $f(J|D_{obs,i})$ is too steep.

When the number of dimensions is small ($N_d = 1$ or 2), our estimation of $h(J|D_{obs})$ agrees very well with the true values. When the dimension of the problem is higher ($N_d =$ 3 or higher), our estimation of $h(J|D_{obs})$ over-predicts the true values, which is the direct result of our mutual information estimation under-predicting the true values. This illustrates a major problem with any method that estimates the mutual information; the mutual information is often underestimated when the dimension of the problem is high. Overall, the comparison is good which verifies that our estimation of mutual information is reasonably accurate. Because $h(J|D_{obs})$ represents the average uncertainty in J given D_{obs}, the trend of $h(J|D_{obs})$ in the table also agrees with our expectation; as more observed data is obtained, the conditional entropy of J is further reduced. Note that the entropy can take on both positive and negative values, so the negative value of the estimated conditional entropy using 5

observed data is not an indicator of estimation error.

Next, in step 5 of our proposed procedure, we want to translate the conditional entropy into the expected value of the standard deviation of J conditional to D_{obs}. This is not an easy task because the posterior pdf of J for the toy problem under consideration can be either unimodal or bimodal depending on the value of d_{obs}, which is a realization of D_{obs}. Fig. 4.7 shows the posterior pdf's obtained using Eq. 4.48 and with different values of d_{obs} for the case where $N_d = 1$. Note that the three posterior pdf's shown in Fig. 4.7 have distinctively different shapes and the pdf corresponding to $d_{obs} = 0.58$ has two modes. Also as we mentioned earlier, the entropy value does not carry information about the distance between the nodes. Therefore, we cannot use conditional entropy to estimate average standard deviation of a multi-modal distribution. By examining the posterior pdf of J, we found that when using 3 or more observed data ($N_d = 3$ or higher), the posterior pdf's of J conditioned to D_{obs} are mostly unimodal and in that case, we can translate from conditional entropy into average standard deviation using our proposed method.

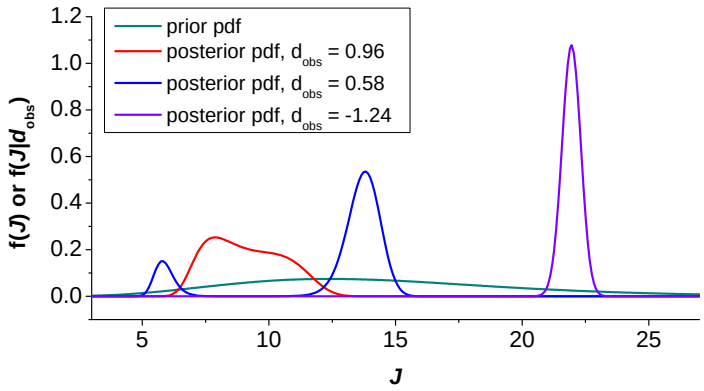


Figure 4.7: Plot of the posterior pdf's of J conditional to three realizations of D_{obs} for the case $N_d = 1$.

Despite the problems when the observed data vector is one or two dimensional, we will use a Gaussian shape to reconstruct our posterior pdf of J. Because we use a Gaussian shape, there is actually no reconstruction necessary. Instead, the standard deviation of our reconstructed pdf can be calculated directly using Eq. 4.14. We then multiply the estimated standard deviation with the correction factor between 1.0 and 1.359; see Step 6 of our pro-

posed method. We compare the resulting estimates to the average of the posterior standard deviations of J conditioned to 1000 random truth cases (rigorous method). Table 4.3 shows the result of this comparison.

Table 4.3: The expected standard deviation of J obtained with our proposed procedure and with the rigorous method.

	$\overline{\sigma_J}$ estimate	$\overline{\sigma_J}$ rigorous	$\overline{\sigma_J}_{,i}/\overline{\sigma_J}_{,i-1}$ estimate	$\overline{\sigma_J}_{,i}/\overline{\sigma_J}_{,i-1}$ rigorous	Ratio difference
$N_d = 1$	1.053	2.258	n/a	n/a	
$N_d = 2$	0.663	0.967	0.630	0.428	0.202
$N_d = 3$	0.463	0.540	0.698	0.558	0.140
$N_d = 4$	0.363	0.348	0.784	0.644	0.140
$N_d = 5$	0.322	0.257	0.887	0.738	0.149

We can make three observations from the result of Table 4.3:

- For $N_d = 1$ or 2, the average standard deviation estimated by our proposed method is significantly smaller than that calculated from the rigorous method. This can be explained using the fact that the posterior pdf of J often has two modes for 1 and 2 observed data, so we underestimate the standard deviation if we assume the posterior pdf is Gaussian with a single mode. This means our Gaussian assumption for the posterior pdf of J is not good enough, which leads to a high difference between the estimated $\overline{\sigma_J}$ and the value computed with the rigorous method.

- For $N_d = 3$ to 5, the comparison is much closer. This is because the posterior pdf of J is, in most case, unimodal and thus can be approximated by a Gaussian distribution. For $N_d = 5$, we observe a reverse trend in the comparison, where our estimated value of $\overline{\sigma_J}$ is greater than the value calculated rigorously. This is because our mutual information estimator underestimates the true value, which leads to an erroneously high value of conditional entropy which automatically results in an erroneously high value of the average standard deviation (for a Gaussian pdf, standard deviation is directly proportional to the entropy according to Eq. 4.14).

- The third observation is that the estimated $\overline{\sigma_J}$ values mimic the qualitative behavior of the values of $\overline{\sigma_J}$ computed from the rigorous method in the sense that in both cases $\overline{\sigma_J}$ decreases as the number of observed data increases as it obviously should. The fourth and fifth columns of Table 4.3 show the values of $\overline{\sigma_J}_{,i}/\overline{\sigma_J}_{,i-1}$, i = 2, 3, 4, 5 obtained from the two methods, i.e., the average standard deviation obtained when assimilating i data divided by the average standard deviation obtained assimilating $i-1$ data. The trends of this standard deviation ratio for the two procedures are again the same but the values of this ratio obtained from the approximate method is always greater than the corresponding value obtained from the rigorous method. Column six of Table 4.3 records the value in column four minus the value in column five. Note this value ranges only from 0.14 to 0.20 and the values in the last three rows are all very close to 0.14. A value of 0.14 means that the rigorous method indicates a 14% greater reduction in uncertainty ($\overline{\sigma_J}$) by adding one more measurement than is indicated by the approximate method.

4.5 One-dimensional Water Flooding Problem

We define a simple 1D water flooding problem with the following properties:

- $21 \times 1 \times 1$ grid blocks, each of which is of size 100 ft $\times$ 20 ft $\times$ 20 ft.
- All grid blocks have porosity equal to 0.18. The permeability follows a log-normal distribution with mean equal to 5 and standard deviation equal to 0.5. The spherical covariance function with prior correlation length of 10 grid blocks is used to generate realizations from the prior Gaussian pdf.
- One water injector is placed on the left in grid block (1, 1, 1), with constant injection pressure of 5500 psi.
- One producer is placed on the right in grid block (21, 1, 1), with constant production pressure of 4500 psi.

- One observation well is placed in the middle in grid block (11, 1, 1).
- The simulation time is 8 years.

Our objective is to quantify the uncertainty reduction in cumulative oil production at the end of simulation time (8 years) using observed data that can be obtained from various surveillance operations. From now on, whenever we mention cumulative oil production without specifying measurement time, we mean the value obtained at the end of year 8. The data types we consider in this research are water front location, water production rate at the production well, water saturation measured at the observation well, and pressure measured at the observation well. Specifications of each type of data are shown in Table 4.4.

Table 4.4: Specifications of measurement data

	Measurement Freq.	Units	Measurement Error
Water front location	Every 12 months	gridblock index	$N(0, 0.3)$
Water production rate	Every 3 months	STB/D	$N(0, 1.0)$
Water saturation	Every 3 months	dimensionless	$N(0, 0.02)$
Pressure	Every 3 months	psi	$N(0, 5.0)$

In step 1, we generate 10,000 realizations of the prior model, the first four of which are shown in Fig. 4.8. In step 2, we run simulations (using Eclipse) to obtain observed data and cumulative oil production for all realizations. Figs. 4.9 and 4.10, respectively, show the pressure and water saturation profiles after one month, one year and three years for the first 4 realizations. Fig. 4.11 shows the water rates at the production well for the first 4 realizations. From Figs. 4.10 and 4.11, we can see that, among the first 4 realizations, no realization experiences water breakthrough after 1 year and only one realization experiences water breakthrough after 3 years, which means that the water front location should provide valuable information about the water movement when we measure it at the end of the first year and at the end of 3 years.

In step 3, we calculate the prior entropy of cumulative oil production. For the water flooding problem defined at the beginning of this section, the prior distribution of cumulative

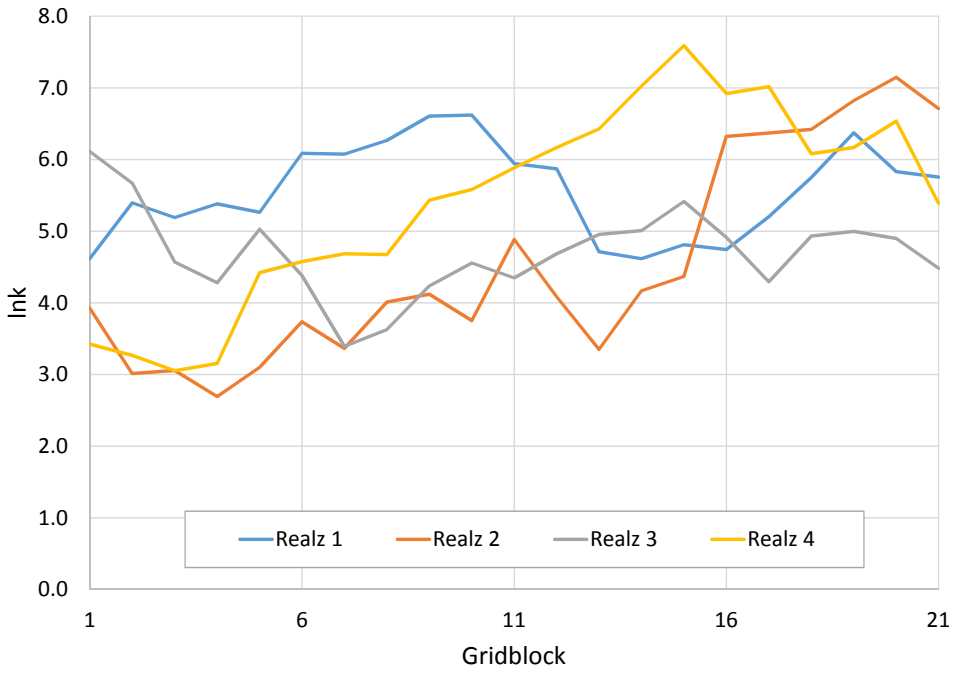
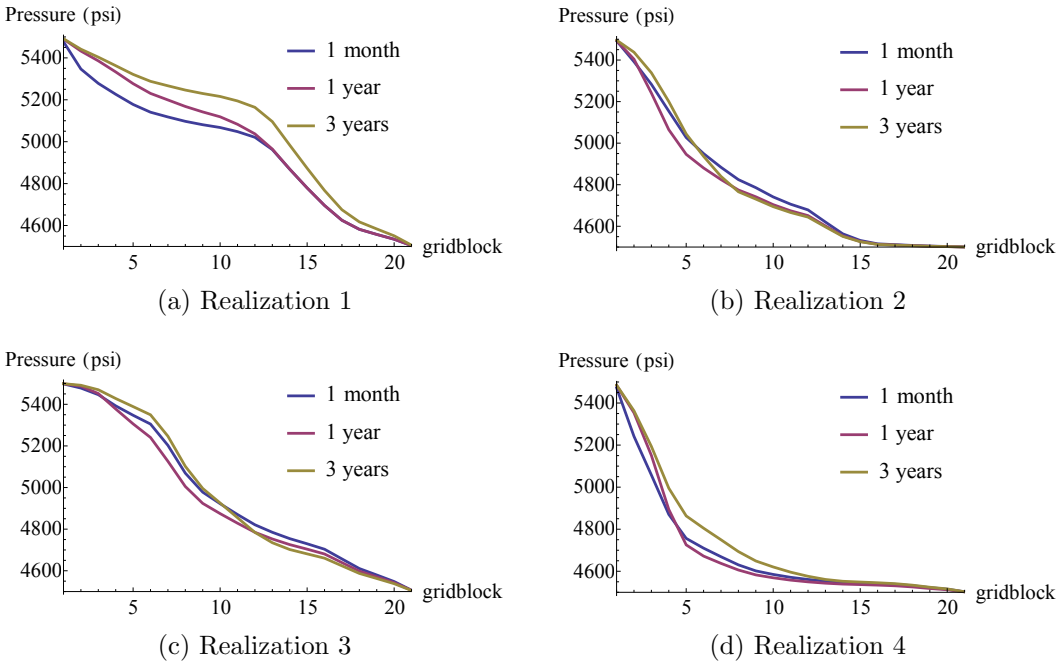


Figure 4.8: First four realizations of the log permeability field.



(a) Realization 1

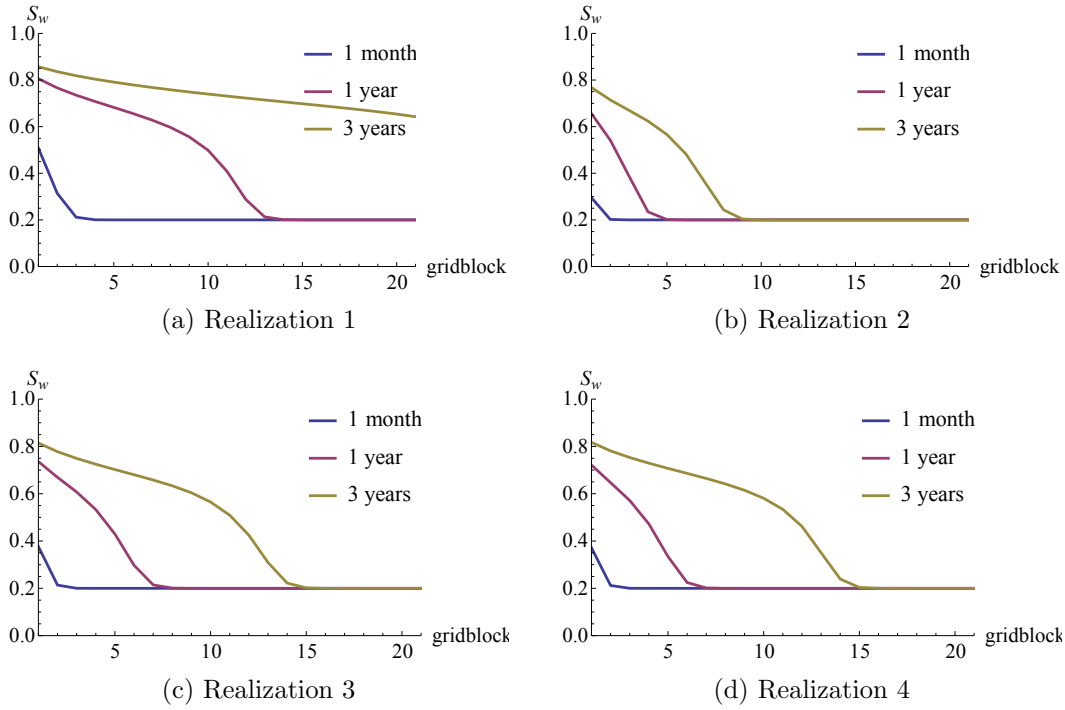
(b) Realization 2

(c) Realization 3

(d) Realization 4

Figure 4.9: Pressure profiles of the first four realizations.

oil production at the end of year 8 is similar to a Gaussian distribution but skewed to the right with a long tail at the left (Fig. 4.12). The prior uncertainty of cumulative oil production, in term of P90 - P10, is 6134 STB. The estimation of prior entropy based on sample spacings



(a) Realization 1 (b) Realization 2

(c) Realization 3 (d) Realization 4

Figure 4.10: Water saturation profiles of the first four realizations.

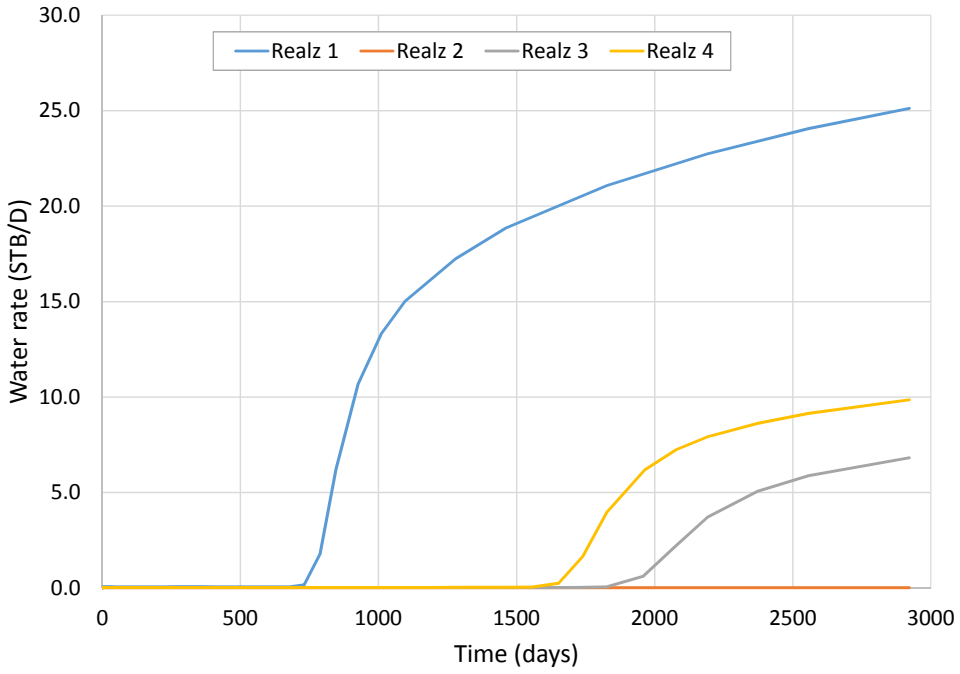


Figure 4.11: The water production rates of the first four realizations.

give us a result of 8.94. We consider this result reliable because it is estimated from a set of 10,000 values.

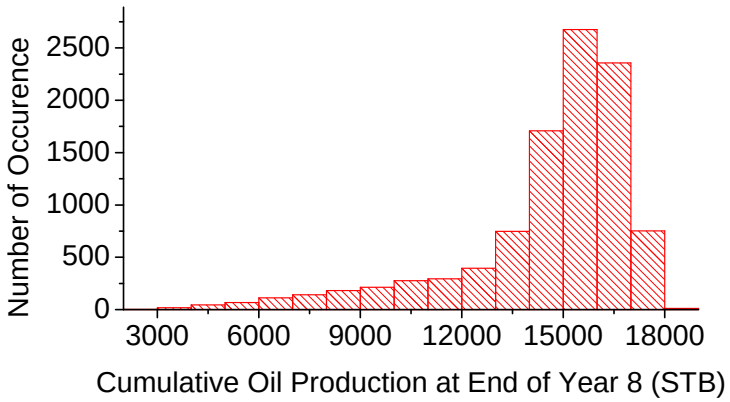


Figure 4.12: Histogram of prior knowledge of cumulative oil production

In step 4, using the k^{th} nearest neighbor method (Kraskov et al., 2004), we calculate the mutual information between cumulative oil production and each type of observed data. We consider 2 cases: (1) data are observed in the first year, and (2) data are observed in the first 3 years. This means that we will have either 1 or 3 observed data for the water front location, and either 4 or 12 observed data for the water rate at the production well, 4 or 12 observed data for the water saturation at the observation well and 4 or 12 observed data for the pressure at the observation well. Note that, even though other types of data are also available, such as the pressures at the production well, they are not considered in this research. Also, in this example, we only consider the effect of each type of observed data separately. In example 3 and 4, we will consider the combined effect of different types of observed data. The results are shown in Table 4.5.

Table 4.5: Mutual information between cumulative oil production and different types of observed data. The superscript * denotes that PCA has been applied to improve the estimation of mutual information.

	1 year data	3 year data
Water front location	1.63	2.21
Water production rate	0.02*	1.10*
Water saturation	0.53*	1.82*
Pressure	0.59	1.90

From Table 4.5, it can be seen that the information content carried in the water front

locations are greatest among 4 types of observed data. The water production rate provides the least amount of information, especially if we use only the data for the first year. This is easy to understand as there is almost no water production for most realizations during the first year (see the plots above for the first four realizations) and even when there is, the production rate is too small compared to the noise level. The information content carried in the pressure and water saturation are roughly equal and fall between those of the water front location and the water production rate.

The conditional entropy of cumulative oil production conditioned to observed data is calculated as the difference of prior entropy and mutual information. These results are shown in Table 4.6. Note that the sum of any cell of Table 4.6 with the cell of the same row and column indices in Table 4.5 is equal to 8.94, which is the value of the prior entropy.

Table 4.6: Conditional entropy of cumulative oil production given different types of observed data.

	1 year data	3 year data
Water front location	7.31	6.73
Water production rate	8.92	7.84
Water saturation	8.41	7.12
Pressure	8.35	7.04

Now we want to cross-check the above results with another method, similar to what we did in the toy problem. We cannot use the numerical integration approach because there is no explicit expression for J as a function of m. Instead we will use history matching to update the prior model with different plausible measurement outcomes to obtain the posterior pdf's of J. Our choice of history matching method for this problem is Ensemble Smoother with Multiple Data Assimilation (ES-MDA) (Emerick and Reynolds, 2013c) as it has been shown to give the most accurate uncertainty quantification among ensemble-based methods (Emerick and Reynolds, 2013c) for a similar 1-D water flooding problem. The steps are outlined below:

1. Divide our 10,000 realizations we created earlier into 100 ensembles of 100 realizations

each.

2. Generate another 100 independent realizations, which represents 100 plausible truth cases. For each of these truth cases we generate a vector of plausible observed data by running the reservoir simulator and adding noise.

3. Match each of the 100 ensembles generated in step 1 with one vector of observed data generated in step 2 to obtain the updated ensemble. Each updated ensemble of 100 models approximately represents one plausible posterior probability density function (pdf).

4. For each updated ensemble, calculate the posterior entropy using the sample spacing method (Beirlant et al., 1997).

5. Average the posterior entropies over the 100 ensembles to obtain the conditional entropy. Compare this value with our estimated value.

When applying the ES-MDA algorithm to our problem, we do not use covariance localization because the ensemble size (100) is considerably larger than the number of model parameters (21), so spurious correlations are expected to be insignificant. The number of data assimilations in ES-MDA method is chosen so that the final objective function is smallest. We found that 3-6 data assimilations are the optimal value for our problems. The first 4 plausible posterior distributions of J conditioned to each type of observed data are shown in Fig. 4.13-4.16.

The conditional entropy of cumulative oil production given different types of observed data calculated through rigorous history matching are shown in Table 4.7. Comparing the results of Tables 4.6 and 4.7, we see that the two estimates of conditional entropy agree quite well. The difference can be the consequence of any of the following: (1) error in estimating mutual information (2) error in estimating entropy (3) error in sampling of posterior distributions using ES-MDA.

Next, we would like to estimate the average P90 - P10 of J (steps 5 and 6) and

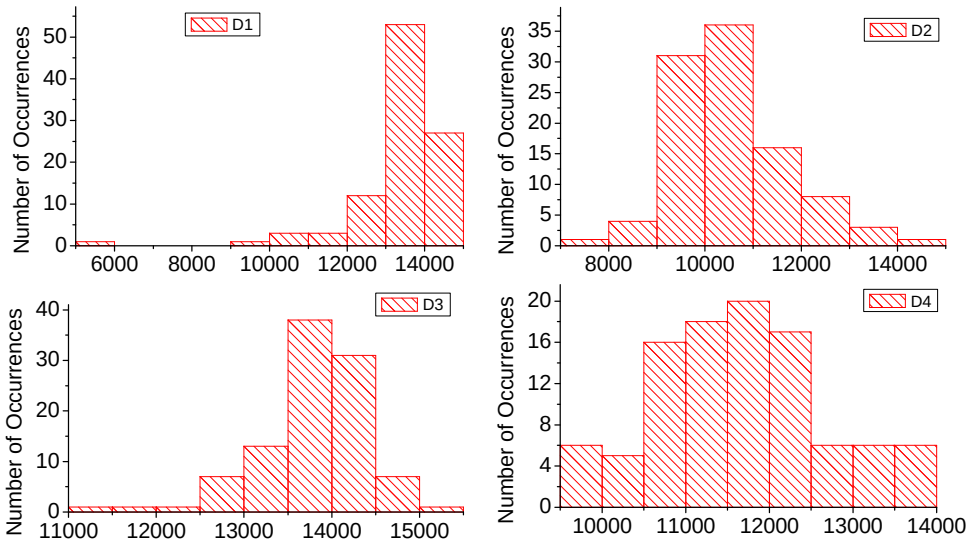


Figure 4.13: The plausible posterior distributions of cumulative oil production conditioned to 3 years of water front data.

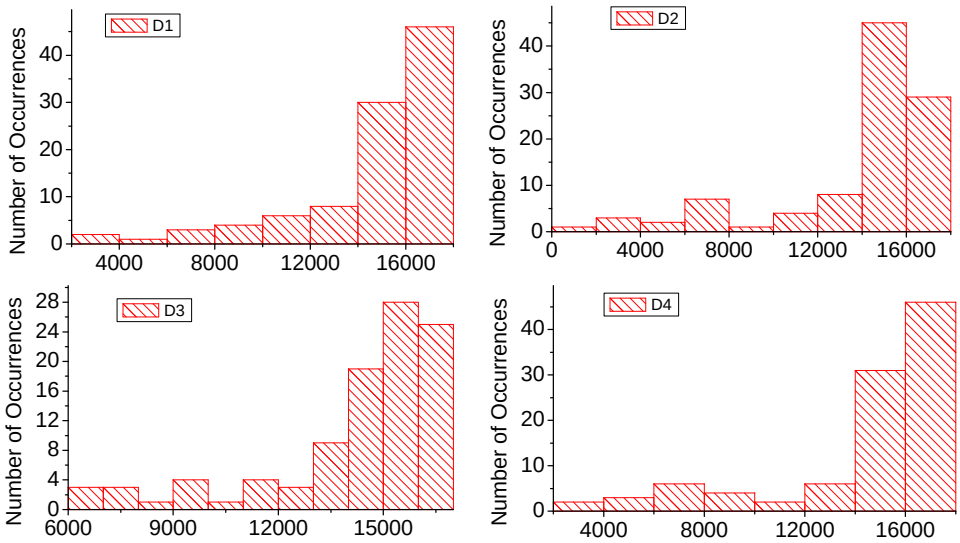


Figure 4.14: The plausible posterior distributions of cumulative oil production conditioned to 3 years of water production rates.

compare it with the true value. For each type of observed data, the average P90 - P10 over 100 posterior ensembles of J obtained using ES-MDA is considered to be the true value. These values are shown in Table 4.8. It can be seen from Table 4.8 that using the water front location gives us the smallest posterior uncertainty while using the water rate gives us the highest degree of uncertainty.

To obtain a reliable estimate of the average P90 - P10 of J from our estimates of

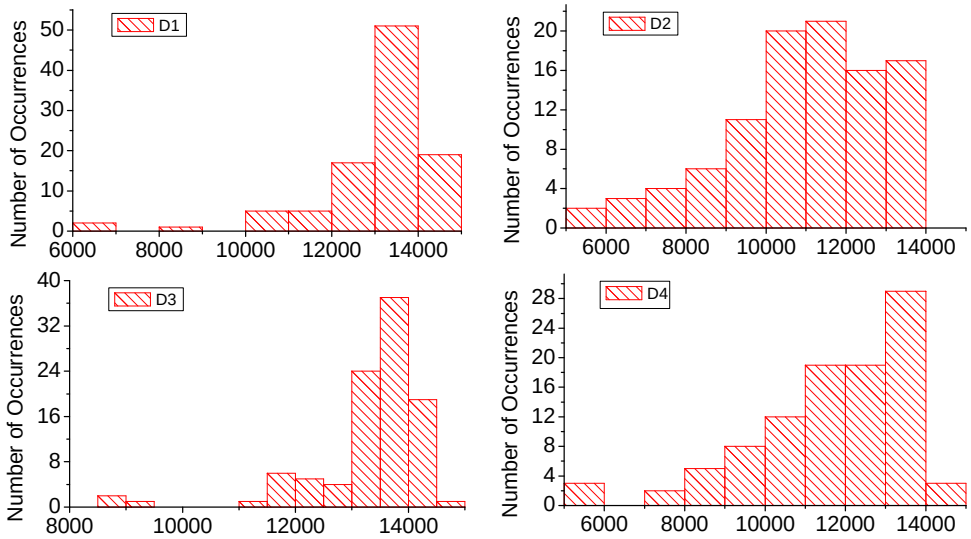


Figure 4.15: The plausible posterior distributions of cumulative oil production conditioned to 3 years of water saturation data.

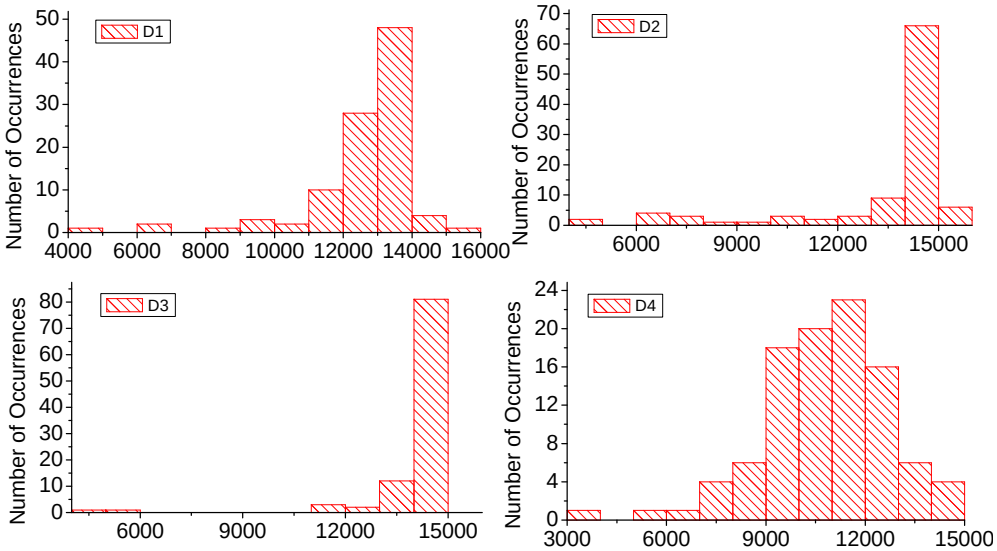


Figure 4.16: The plausible posterior distributions of cumulative oil production conditioned to 3 years of pressure data.

conditional entropy, we need information about the shape of the posterior pdf's. Using a Gaussian shape will not work for this case because, based on our history matching results, the posterior pdf's are far from Gaussian. The most simple alternative is to assume that the shapes of the posterior pdf's do not deviate significantly from the prior shape (Fig. 4.12) and reconstruct our posterior pdf based on the prior shape; see the discussion in section 4.2.5. By making this assumption, we can completely avoid history matching and our procedure

Table 4.7: Conditional entropy of cumulative oil production given different types of observed data, calculated using rigorous history matching.

	1 year data	3 year data
Water front location	7.40	7.00
Water production rate	8.78	7.96
Water saturation	8.25	7.13
Pressure	8.24	6.77

Table 4.8: The true average value of P90 - P10 of cumulative oil production obtained using history matching. The unit of measurement is STB.

	1 year data	3 year data
Water front location	1447	904
Water production rate	5587	4789
Water saturation	4525	1259
Pressure	4112	1178

requires little computational cost other than the cost of generating the initial ensembles. Our estimation of average P90 - P10 of J is shown in Table 4.9. Note that in this table, we have already multiplied our estimates with the appropriate correction factor (between 1.0 and 1.359) calculated using the procedure shown in Appendix C.4. Because the determination of the correction factor is based on the value of the mutual information and each surveillance scenario has a different value of the mutual information, a different correction factor is applied to each surveillance scenario.

Table 4.9: The estimated average value of P90 - P10 of cumulative oil production obtained assuming the shapes of posterior pdf's are the same as the prior shape. The unit of measurement is STB.

	1 year data	3 year data
Water front location	1546	936
Water production rate	6007	2434
Water saturation	3936	1313
Pressure	3743	1225

As we can see by comparing the results of Table 4.9 with the "true" values of Table 4.8,

most of our estimates are in good agreement with the true values except when we use 3 years of water production rate data. This difference is most likely due to the deviations of the posterior shapes from the prior shape; we can see there is a big difference in the shapes of the pdf's in Fig. 4.14 and the shape of the prior pdf in Fig. 4.12.

A seemingly more reliable method to estimate the shape of the posterior pdf is to do so through history matching. This may sound like defeating the purpose of what we are trying to accomplish, that is, estimating uncertainty reduction without history matching. However, we argue that we may not need as many as 100 different posterior ensembles to obtain information about the shape of the posterior distribution and use only 3 ensembles. Using information about the shape of these ensembles combined with the conditional entropy calculated earlier, we can expect to obtain much more reliable estimate of the average P90 - P10 of J than by simply calculating the average of P90 - P10 of J over 3 ensembles. Note that the rigorous result is obtained by averaging the P90 - P10 of all 100 ensembles. Based on this reasoning, we randomly choose 3 posterior ensembles from the 100 posterior ensembles we obtained in history matching. Here, for simplicity, the first 3 ensembles are used. We then construct three new pdf's, each of which is based on one chosen ensemble and with entropy equal to our calculated conditional entropy. For each of the reconstructed pdf's we calculate P90 - P10 and take the average value. Table 4.10 shows the estimate using our proposed method.

Table 4.10: The estimated average value of P90 - P10 of cumulative oil production obtained using 3 history matches for each surveillance scenario. The unit of measurement is STB.

	1 year data	3 year data
Water front location	*1302*	847
Water production rate	*6947*	**3891**
Water saturation	**4746**	1281
Pressure	**4092**	*1540*

Comparing the estimates in Table 4.10 to the estimates in Table 4.9, we can see that some estimates are improved (bold text), some are worse (italic text) and some are roughly

the same (regular text). The case with biggest improvement is the case of 3 years of water production rate data, where the estimation error is reduced from 49% to 19%. It seems like having knowledge about the shape of the posterior pdf's helps us improve our estimates in this case. The case with the worst degradation is the case of 3 years of pressure data, where the estimation error increases from 4% to 31%. One possible reason is that the shapes of the three posterior ensembles obtained with history matching is not representative of the overall shape. If we look at the first 3 histograms of Fig. 4.16, which are the ones we actually use to estimate value of P90 - P10 in Table 4.10, we can see that the second and the third histograms are highly irregular. While most of the histogram concentrates in the high value region, there are some outliers scattered near 0, which greatly increases the estimated P90 - P10. This illustrates the difficulty to obtain a good estimate of P90 - P10 even when we have a good estimate of the conditional entropy.

For comparison, we obtain another estimate of average P90 - P10 by simply calculating the average of P90 - P10 of the first 3 posterior ensembles, without using conditional entropy. This is the same way we calculated the true value of average P90 - P10 in Table 4.8, with the only difference being that we now use only 3 ensembles instead of 100 ensembles. The results of this simple method are shown in Table 4.11.

Table 4.11: The estimated average value of P90 - P10 of cumulative oil production obtained by calculating the average P90 - P10 of 3 ensembles. The unit of measurement is STB.

	1 year data	3 year data
Water front location	2869	2170
Water production rate	6032	7142
Water saturation	5351	3140
Pressure	5102	2997

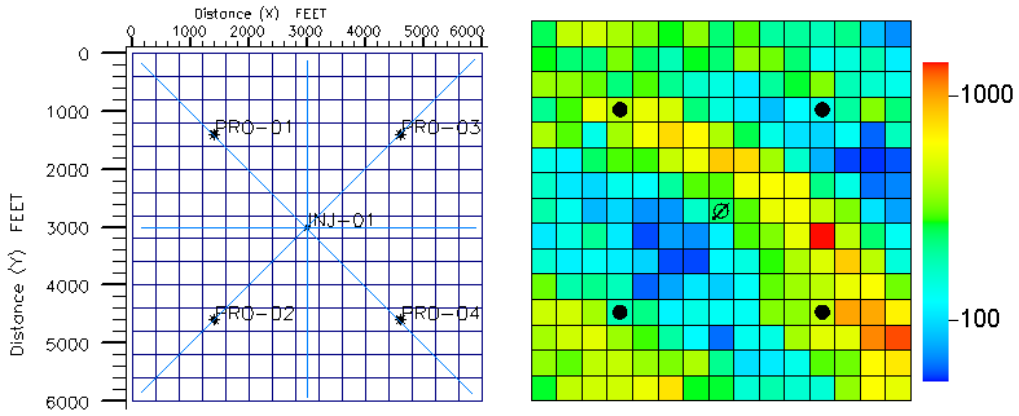
It can be seen that the values in Table 4.11 are very poor estimates of the true average P90 - P10 in Table 4.8. This means that 3 posterior pdf's are not sufficient to estimate the value of average P90 - P10, unless we combine the information provided 3 posterior pdf's (here the shape of the pdf's) with our calculated conditional entropy value.

4.6 Two-dimensional Water Flooding Problem

We now consider a two-dimensional, two-phase flow (oil and water) problem. The model is defined as follows:

- $15 \times 15 \times 1$ grid with all gridblocks of size 400 ft $\times$ 400 ft $\times$ 20 ft.
- One water injector is placed in the center grid block and operated with a constant injection pressure of 5500 psi. The injector is surrounded by four production wells, each of which is produced at a constant flowing bottom hole pressure equal to 2000 psi. Fig. 4.17a shows the locations of the injector and four producers. The blue lines through the injector are drawn for a purpose that will be explained later.
- The porosity field is homogeneous and with porosity equal to 0.18.
- The permeability field is isotropic. We assume that we have noisy measurements of horizontal permeability at gridblocks penetrated by wells, and in terms of $\ln(k)$, these measurements are 5.5 at well INJ-01 and are, respectively, 6.0, 5.0, 5.0, 6.0 at wells PRO-01 to PRO-04. The standard deviation of the measurement error of the hard data is assumed to be 0.1.
- The grid permeabilities follow a log-normal distribution with the mean equal to 5.0 and standard deviation equal to 1.0. The permeability field is generated conditional to the measured permeabilities at the well locations using randomized maximum likelihood (RML). A spherical covariance with a correlation length of 30 gridblocks in the major direction and 7 gridblocks in the minor direction and an angle of maximum continuity of 30 degree is used to generate the covariance matrix C_M. Note that the covariance matrix is in terms of the natural logarithm of the permeability. One realization of the permeability field is shown in Fig. 4.17b.
- The initial reservoir pressure is 5000 psi.
- Only two-phase flow occurs with irreducible water saturation and residual oil saturation given, respectively, by $S_{iw} = 0.1$ and $S_{or} = 0.1$

- The simulation time is 8 years.



(a) Well locations (b) Permeability of realization #1

Figure 4.17: Well locations and permeability of the first realization.

Our objective is to quantify the uncertainty reduction in cumulative oil production at the end of the simulation time (8 years) using the observed data that could be obtained from various surveillance operations during the first year. Throughout, whenever we mention cumulative oil production without specifying measurement time, we mean the value obtained at the end of year 8. The data we consider are water rate, oil rate, water front location, build-up pressure and several combinations of these data. The first two types of data are self-explanatory as they are simply the output of the reservoir simulator with some noise added, but we will need some clarification on the other two. The water front location is an advanced type of observed data that are presumably measurable using a new class of nano-sensors currently under development (Ullo, 2008; Chapman and Thomas, 2010). In this research, the water front locations are modeled deterministically. From the simulation results, we extract the water saturations at all gridblocks along the eight directions indicated by the blue lines in Fig. 4.17a. The water front locations are then calculated as the distances from the injector to the points where the water saturation is exactly 0.5 along the 8 directions. Finally, Gaussian random noise is added to these distances to obtain observed data for front locations. To obtain the build-up pressure, at the end of each year, we shut in the producers for 2 days while still keeping the injector open. The build-up pressure is modeled as the gridblock

pressure during this shut-in period. We use build-up measurements that are logarithmically distributed over the two-day shut-in period, i.e. the measured data are denser near the instance of shut-in and sparser later on, for a total of 10 pressure data per well during the two day shut-in period. Specifications of each type of data, including measurement frequency, number of observed data per year (for all wells), units of measurement and measurement error, are shown in Table 4.12. Measurement error refers to the standard deviation of the random variables and each measurement error is assumed to be Gaussian with mean zero.

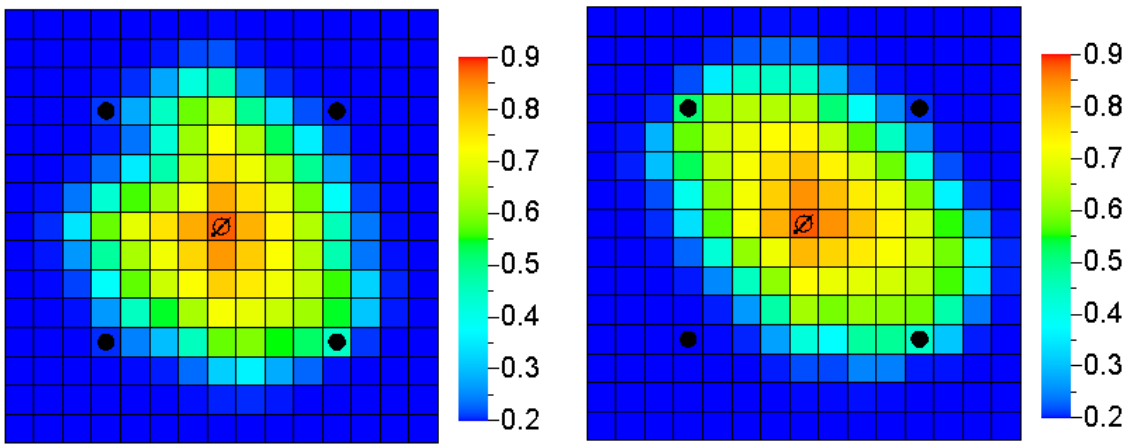
Table 4.12: Specifications of measurement data.

	Measurement Freq.	# of data/year	Units	Measurement Error
Oil rate	Every 3 months	16	STB/D	3% measured value
Water rate	Every 3 months	16	STB/D	10% measured value
Front location	Every 12 months	8	gridblock	0.1 grid block
Pressure	2 days per year	40	psi	10 psi

We generate 10,000 realizations of the prior model and run simulations (using Eclipse) to obtain the observed data and cumulative oil production for all realizations. Fig. 4.18 shows the water saturation distribution after two years for the first 2 realizations. Fig. 4.19a shows the cumulative oil production as a function of time for the first 100 realizations. The prior distribution of cumulative oil production at the end of year 8 is very similar to a Gaussian distribution (Fig. 4.19b). The prior P90 - P10 and the prior entropy of cumulative oil production are 821,800 STB and 14.085, respectively.

Using both the nearest neighbor and the linear-Gaussian methods, we estimate the mutual information between cumulative oil production at the end of year 8 and the data obtained during the first year under different surveillance scenarios. The results are shown in Table 4.13. Note that these results are estimated from the initial ensemble of 10,000 realizations with no history matching involved.

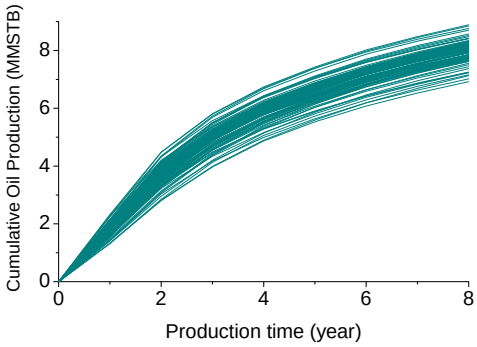
As we can see from the results of Table 4.13, the mutual information estimates using the two methods are comparable when the number of observed data is small (see rows 1, 2 and 4 of Table 4.13). Since we know from the results of sections 4.4 and 4.5 that the



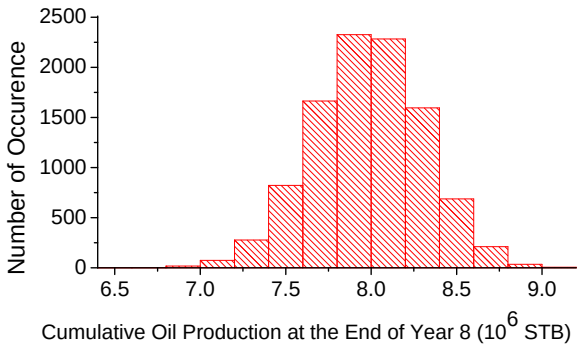
(a) Realization 1

(b) Realization 2

Figure 4.18: Water saturation distribution after two years for the first 2 realizations.



(a) Cumulative oil production as a function of time for the first 100 realizations from the prior pdf



(b) Histogram of cumulative oil production at year 8

Figure 4.19: Cumulative oil production based on realization from the prior pdf, example 1.

nearest neighbor estimator tends to work fairly well for a small number of observed data, the close agreement between the two methods gives us the confidence that our linear-Gaussian estimator may be reliable for this problem. This could be because the relationship between cumulative oil production and the observed data in this problem is reasonably linear.

The comparison between the two estimators when the number of observed data is high, however, tells a different story (rows 3, 5, 6, 7 of Table 4.13). This time, the estimates obtained using the nearest neighbor method are consistently smaller than the estimates obtained using the linear-Gaussian method. As we have previously discussed, the highest

Table 4.13: Estimates of mutual information between cumulative oil production and observed data during year 1.

Row	Data	Linear-Gaussian method	Nearest neighbor method
1	Water rate (16 data)	0.397	0.389
2	Water front (8 data)	0.507	0.491
3	Build-up pressure (40 data)	0.630	0.358
4	Oil rate (16 data)	0.445	0.449
5	Oil rate + water rate (32 data)	0.457	0.392
6	Oil rate + water front (24 data)	0.569	0.500
7	Oil rate + pressure (56 data)	0.658	0.462

estimates of mutual information are likely to be more accurate as both methods tend to underestimate mutual information. The results in Table 4.13 therefore suggests that the linear-Gaussian estimator tends to be more reliable than the nearest neighbor estimator for high-dimensional problems. If we take a closer look at the estimates using the nearest neighbor method, certain results seem to be clearly wrong. For example, when we combine the oil rate data with the water rate data, we would expect a higher value of mutual information compared to using only oil rate data. However, the results in rows 4 and 5 of Table 4.13 are the opposite of what we expect. This incorrect result may occur because the high dimensionality of the problem renders the nearest neighbor estimator much less accurate. To confirm the ineffectiveness of the nearest neighbor estimator in high-dimensional problems, we produce results similar to those of Table 4.13 but using three years of observed data instead of one, which effectively triples the number of observed data in all scenarios. The results are shown in Table 4.14, where we observe that the nearest neighbor estimator underestimates mutual information in all cases and many of the estimates are clearly wrong, e.g. the calculated mutual information between J and oil rate and water rate data combined is smaller than the calculated mutual information between J and water rate data alone. This again leads to the conclusion that the nearest neighbor method is inaccurate for high dimensional problems. It is also important to note that we tried to improve the estimates of the nearest neighbor method using PCA as in the previous example. However, the PCA

cannot accomplish the dual tasks of reducing the data set from more that 50 dimensions to 10 or so dimensions so that the nearest neighborhood algorithm may give a somewhat reasonably accurate estimate of mutual information while at the same time obtaining a reduced data set which includes the essential information content of the original data. Thus, for the remaining discussion and calculations in this research, we exclusively use the linear-Gaussian estimator to calculate mutual information.

Table 4.14: Estimates of mutual information between cumulative oil production and observed data during the first 3 years.

Row	Data	Linear-Gaussian method	Nearest neighbor method
1	Water rate (48 data)	0.567	0.380
2	Water front (24 data)	0.848	0.554
3	Build-up pressure (120 data)	0.968	0.534
4	Oil rate (48 data)	1.311	0.653
5	Oil rate + water rate (96 data)	1.327	0.459
6	Oil rate + water front (72 data)	1.442	0.497
7	Oil rate + pressure (168 data)	1.462	0.567

The relative importance of different types of data can be qualitatively interpreted from the values of mutual information in Table 4.13 and Table 4.14. Here we concentrate our discussion on the results in Table 4.14, i.e. with 3 years of observed data because they provide better contrast among different types of data. It appears that the water rate provides the least information about the cumulative oil production and the oil rate provides the most information (rows 1 and 4). The information provided by the water front and build-up pressure falls somewhere in between (rows 2 and 3). It is interesting to note that, in the industry, oil rate is usually measured, so the question of real interest is how much additional information the water rate, water front or build-up pressure can provide. The results in rows 5, 6 and 7 of Table 4.14 suggest that the answer seems to be "not much," as 1.327, 1.442 and 1.462 are not much greater than 1.311. In the later part of the section, we will consider this issue in greater details as the most important goal of acquiring data is to reduce uncertainty.

In the next step, using the estimated values of mutual information, we calculate the

conditional entropy of cumulative oil production for different surveillance scenarios. The conditional entropy of cumulative oil production conditional to data that could be measured is denoted by $h(J|D_{obs})$ and is simply the difference between prior entropy and mutual information. As discussed in section 4.1, $h(J|D_{obs})$ is a measure of the average remaining uncertainty that will exist in J after observing D_{obs}. Thus, the lower the value of $h(J|D_{obs})$, the greater the expected reduction in uncertainty that will be obtained by conducting the surveillance operation to obtain D_{obs} and then conditioning the reservoir model to D_{obs}. The results are shown in Table 4.15.

Table 4.15: Estimates of conditional entropy of cumulative oil production given observed data during the first 3 years.

Row	Data	Conditional entropy
1	Water rate (48 obs data)	13.518
2	Water front (24 obs data)	13.237
3	Build-up pressure (120 obs data)	13.117
4	Oil rate (48 obs data)	12.774
5	Oil rate + water rate (96 obs data)	12.758
6	Oil rate + water front (72 obs data)	12.643
7	Oil rate + pressure (168 obs data)	12.623

The conditional entropy itself provides only a qualitative measure of the uncertainty in the posterior model after data assimilation. To get a better quantification of the uncertainty, we still need to translate the estimated value of conditional entropy into the expected P90 - P10 of the cumulative oil production in the posterior model. Before doing so, we would first like to verify the accuracy of our conditional entropy estimates by using the exhaustive-history matching procedure discussed below. Estimation of the conditional entropy by exhaustive history matching is performed by the following five step procedure:

1. From the prior model, generate 100 ensembles with each ensemble consisting of N_e independent realizations.

2. Generate 100 plausible truth cases and run simulations to obtain the plausible vectors of observed data.

3. History match each ensemble generated in step 1 with one truth case from step 2 (using ES-MDA 4x (Emerick and Reynolds, 2013c,a)) to obtain 100 plausible pdf's of J.

4. Calculate the entropy of each of the 100 pdf's of J generated in the previous step using the sample spacings method (Beirlant et al., 1997) and calculate the average entropy value. The result is our conditional entropy.

5. Repeat the process to obtain the conditional entropy for all other surveillance scenarios.

To apply the exhaustive-history matching procedure with an ensemble-based method, we must decide how to choose an appropriate ensemble size N_e; see Evensen (2003, 2007); Aanonsen et al. (2009). Emerick and Reynolds (2013c) compared several ensemble-based history-matching methods for a 1D water flooding problem and found that ES-MDA provided the most accurate quantification of the uncertainty in the posterior model. However, for the 2D water flooding problem considered in this research, we are not certain that the uncertainty quantification obtained using ES-MDA is accurate. From our experience, different values of N_e will lead to different uncertainty in the posterior model, with higher values of N_e generally leading to higher values P90 - P10. Therefore, to account for the error in estimating the posterior model uncertainty using ES-MDA, we perform the exhaustive history-matching procedure two times, first with an ensemble size of 100, and then repeat the procedure for an ensemble size of 1000. The first 4 plausible posterior distributions of J conditional to each type of observed data based on an ensemble size of 1,000 are shown in Figs. 4.20-4.24. Note that the posterior distributions of J in all cases are very similar to a Gaussian distribution, which partly explains why our linear-Gaussian estimator for mutual information works reasonably well.

The conditional entropy of cumulative oil production calculated using the exhaustive history-matching procedure is shown in Table 4.16. From the results of this table, we see that the conditional entropies calculated using the exhaustive history matching procedure are slightly different for the two ensemble sizes. Since we have no reliable method to correctly sample the posterior pdf, except for the Markov Chain Monte Carlo method, which is not

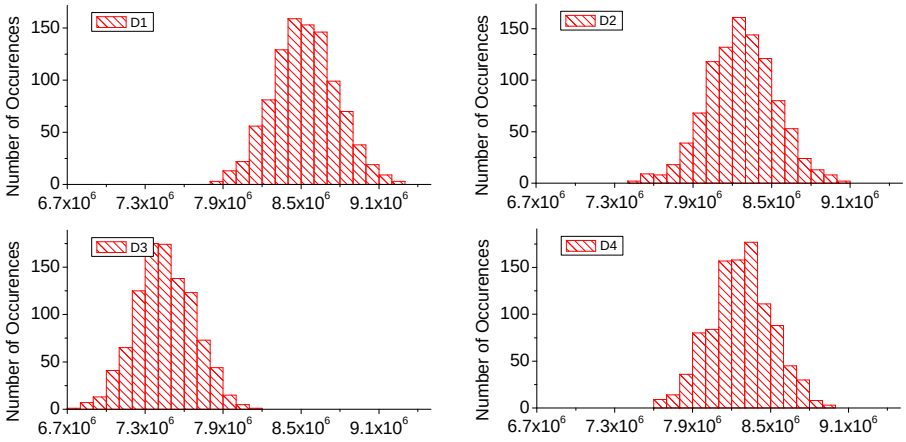


Figure 4.20: Four plausible posterior distributions of cumulative oil production conditional to water rate data.

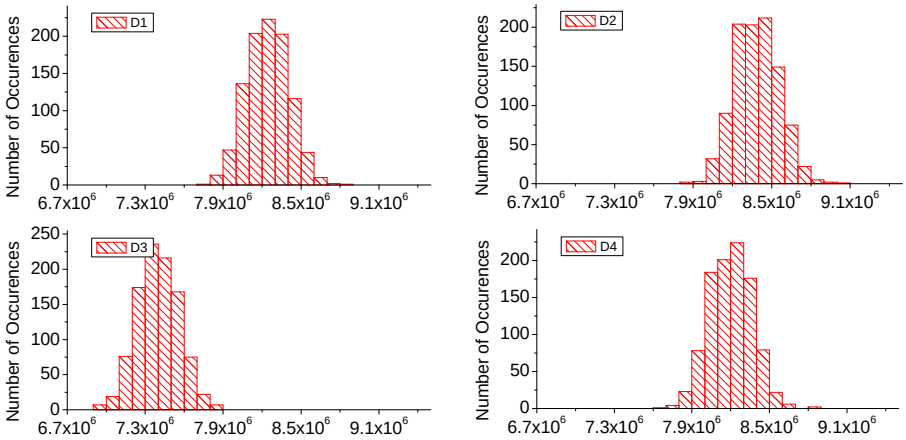


Figure 4.21: Four plausible posterior distributions of cumulative oil production conditional to water front data.

generally computationally feasible, we have to rely on the result of ES-MDA to verify our conditional entropy estimates. Here, we assume that the results with an ensemble size of 1000 are more accurate, but it may important to note that it is possible to over-estimate the uncertainty in future reservoir performance predictions when using results from an ensemble-based history-matching procedure (Emerick and Reynolds, 2012a). We therefore assume that the correct value of conditional entropy is not less than the results generated with an ensemble

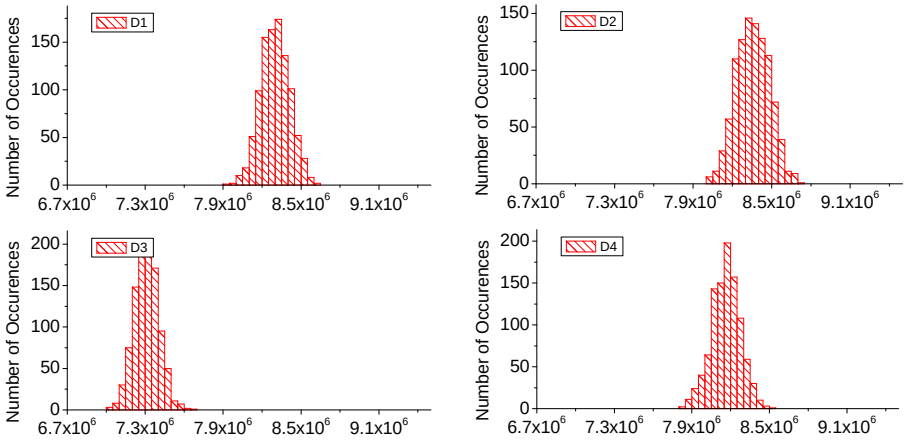


Figure 4.22: Four plausible posterior distributions of cumulative oil production conditional to pressure data.

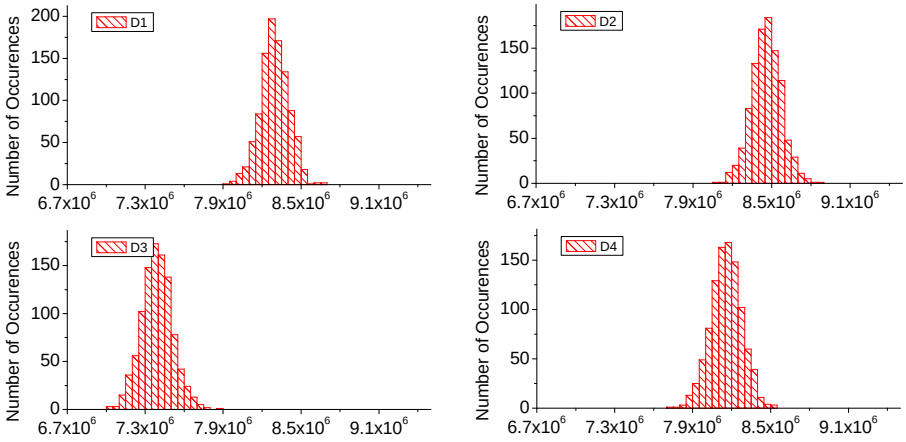


Figure 4.23: Four plausible posterior distributions of cumulative oil production conditional to oil rate data.

size of 100 and not much larger than those generated with an ensemble size of 1000.

Comparing each conditional entropy in Table 4.15 with the corresponding conditional entropy estimate in Table 4.16, we observe reasonable agreements between our estimates shown in Table 4.15 and those obtained using the exhaustive history-matching procedure. Our estimates for the first, second, fifth, sixth and seventh scenarios lie between the two extreme values obtained from exhaustive history matching, which suggests that they are

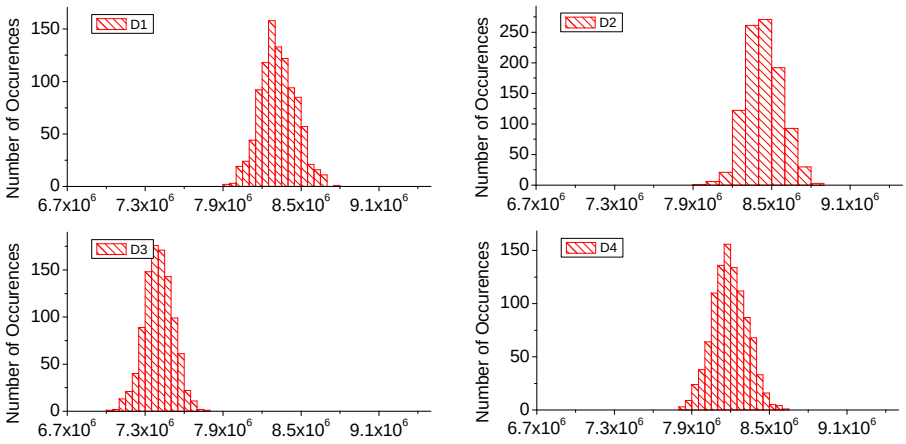


Figure 4.24: The plausible posterior distributions of cumulative oil production conditional to oil rate and water front data.

Table 4.16: Conditional entropy (given 3-year observed data) calculated using exhaustive history matching procedure.

Row	Data	Ensemble size 100	Ensemble size 1000
1	Water rate (48 obs data)	13.450	13.717
2	Water front (24 obs data)	13.174	13.382
3	Build-up pressure (120 obs data)	12.831	13.072
4	Oil rate (48 obs data)	12.869	12.996
5	Oil rate + water rate (96 obs data)	12.750	12.917
6	Oil rate + water front (72 obs data)	12.641	12.872
7	Oil rate + pressure (168 obs data)	12.611	12.667

reasonably good estimates. For the third scenario, our conditional entropy estimate is slightly higher than the history-matching result with an ensemble size of 1000. For the fourth surveillance scenario (oil rate data), our conditional entropy estimate is slightly smaller than value obtained using $N_e = 100$. However, as we have previously discussed, we do not expect the linear-Gaussian estimator to overestimate the mutual information, i.e., underestimate the conditional entropy, so one possible explanation here is that the exhaustive history matching does not match the data well enough and results in an overestimation of the conditional entropy value.

The final step in our proposed procedure is to translate the conditional entropy into

the expected P90 - P10. To do so, we "stretch" or "compress" the histogram of the prior distribution of cumulative oil production until its entropy is equal to the estimated conditional entropy; see section 4.2.5. The estimates of the expected P90 - P10 obtained from the reconstructed pdf's are shown in column 2 of Table 4.17. In the same table, we also show the average P90 - P10 obtained using the exhaustive history-matching procedure (column 3). Here we only use the results obtained from history matching with an ensemble size of 1,000 for comparison because a bigger ensemble size usually means a more accurate uncertainty characterization. Note that, with the exhaustive history-matching procedure, we are able to obtain one extra parameter that cannot be obtained with our proposed procedure, namely the standard deviation of the P90-P10 corresponding to 100 different truth cases. This standard deviation is shown inside the parentheses in column 3 of Table 4.17. It characterizes how much the uncertainty in the posterior model can be affected by different measurement outcomes. For a linear-Gaussian case, the uncertainty in the posterior model, C_{MAP}, does not depend on the measurement outcomes.

Table 4.17: Expected P90 - P10 of cumulative oil production (in thousands STB).

Row	Data	Proposed method	Exhaustive history matching ($N_e = 1000$)
1	Water rate (48 obs data)	481	578 ($\pm$34)
2	Water front (24 obs data)	354	409 ($\pm$19)
3	Build-up pressure (120 obs data)	314	301 ($\pm$30)
4	Oil rate (48 obs data)	223	280 ($\pm$15)
5	Oil rate + water rate (96 obs data)	219	258 ($\pm$14)
6	Oil rate + water front (72 obs data)	195	246 ($\pm$12)
7	Oil rate + pressure (168 obs data)	191	222 ($\pm$17)

From the results of Table 4.17, we can see that our estimates of expected P90 - P10 are fairly close to, although slightly smaller than, the values calculated from the exhaustive history matching procedure. This difference can be explained by noting that our estimated value of conditional entropy is also slightly smaller than the corresponding value obtained from history matching (column 2 of Table 4.15 versus column 3 of Table 4.16). This directly translates into smaller values of P90 - P10 using our proposed method. The small difference

in conditional entropy can be the cumulative effect of many factors, including the error in estimating mutual information as well as the possible error in quantifying uncertainty using an ensemble-based history-matching method. However, if we look at the big picture, the difference is small, and our proposed method is able to determine which type of observed data leads to the highest uncertainty reduction. More importantly, our method provides a reasonable quantification of the uncertainty in the posterior pdf for cumulative oil recovery. It is also important to note that our proposed method only requires 10,000 simulation runs for all seven surveillance scenarios while the exhaustive history matching method requires 2.8 million simulation runs to obtain the same result.

The expected values of P90 - P10 in Table 4.17, either using our proposed method or the exhaustive history-matching procedure, also confirms our comment earlier that obtaining additional data besides the oil rate does not result in a significant reduction in the uncertainty in cumulative oil recovery. In retrospect, it is not surprising that the measured oil rate would be the most useful data for predicting future oil rate. However, if our objective were to estimate which type of data would be most useful for estimating the uncertainty in the estimated net present value of future production data or the optimal location of an infill well, the relative value of data from different potential surveillance operations could be considerably different. To demonstrate one such case, we use our proposed method to obtain the estimates for the case where the reservoir variable J is the net present value instead of the cumulative oil production. In calculating the net present value, we use \$90/STB for the oil price, \$15/STB for the water disposal cost and \$10/STB for the water injection cost. The annual discount factor is 10%. The prior entropy and prior P90 - P10 for the net present value, respectively, are 18.107 and \$45.4 million. Table 4.18 shows the mutual information, conditional entropy and expected P90 - P10 of net present value conditional to 1 year of observed data.

From the results of Table 4.18, we see that the oil rate data are not the best type of data that we could collect during the first year of production. In fact, the water front and pressure build-up data now lead to lower uncertainty in the posterior net present value.

Table 4.18: Estimates of mutual information between net present value and observed data obtained during the first year, and conditional entropy and expected P90 - P10 of net present value (in millions of dollars) conditional to the observed data.

Row	Data	MI	Con. Entropy	Expected P90 - P10 ($million)
1	Water rate (16 data)	0.124	17.982	40.0
2	Water front (8 data)	0.225	17.881	36.2
3	Build-up pressure (40 data)	0.273	17.833	34.5
4	Oil rate (16 data)	0.137	17.970	39.6
5	Oil rate + water rate (32 data)	0.160	16.947	38.7
6	Oil rate + water front (24 data)	0.306	17.801	33.4
7	Oil rate + pressure (56 data)	0.334	17.773	32.5

When the oil rate data are combined with either water front or pressure build-up data, we obtain a much greater reduction in the uncertainty of the net present value than is obtained by using the oil rate data alone. Note that the choice of using only observed data in the first year has an impact on the relative importance of each surveillance scenario. During the first year, for most realizations of the permeability field, the injected water has not broken through at producers for most realizations of the permeability field and thus, the location of the water front is relatively more useful as it provides information on how much water will be produced in the future. After the water breakthrough, the usefulness of the water front location diminishes, and the oil production rate will have the dominant impact on the uncertainty reduction in net present value.

4.7 Three-dimensional Water Flooding Problem

The last problem we consider is the three-phase (oil, gas and water), three-dimensional PUNQS3 reservoir model (Floris et al., 2001; Barker et al., 2001; Gao et al., 2006). The pertinent details of this model are as follows:

- $19 \times 28 \times 5$ gridblocks; the size of each gridblock is 590.55 ft $\times$ 590.55 ft in the "x-y direction" but the thicknesses of the gridblocks vary. There are 1761 active gridblocks in the model.

- There are six producers with the locations and perforated intervals the same as in the original model (Floris et al., 2001; Barker et al., 2001). Fig. 4.25 shows the location of these wells. All production wells are produced at a constant flowing bottom hole pressure of 2000 psi.

- There is a strong analytical aquifer in the original model, but here, we replace the aquifer by six water injection wells along the periphery of the reservoir (Fig. 4.25). All injectors operate at a constant injection pressure equal to 4500 psi.

- The unconditional permeability and porosity fields are generated using sequential Gaussian co-simulation based on the geostatistical parameters given in Gao et al. (2006). One realization of the property fields is shown Fig. 4.26.

- The initial reservoir pressure is approximately equal to 3500 psi.

- The simulation time is 30 years.

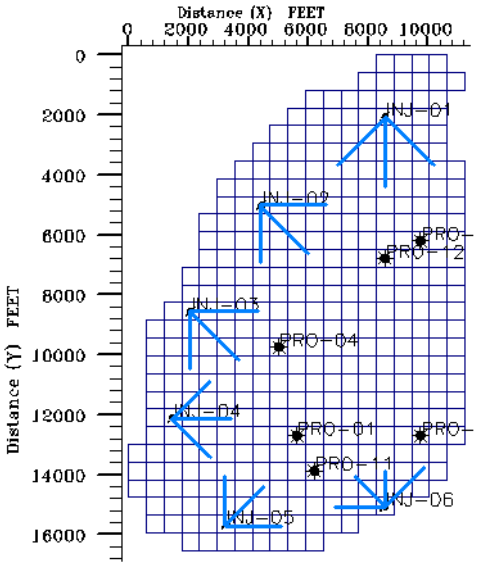
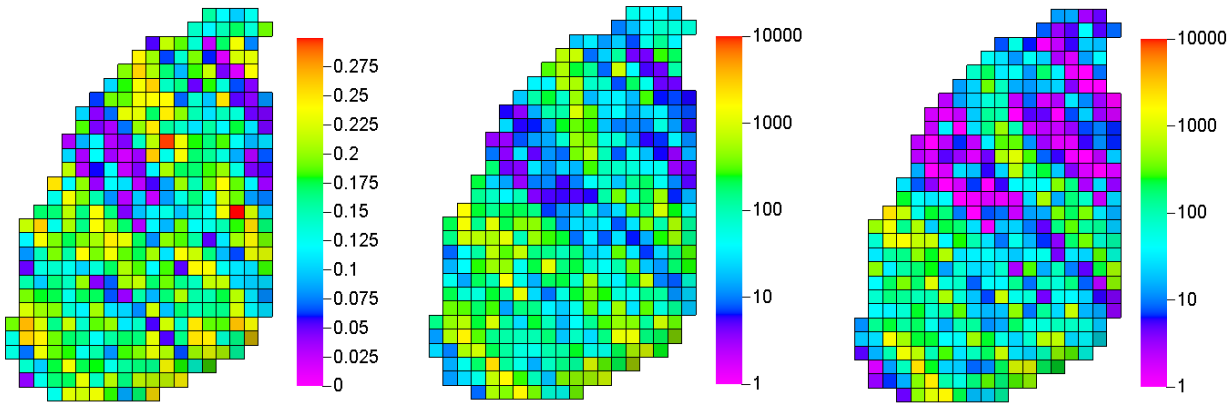


Figure 4.25: Well locations in the PUNQ case.

In this example, our reservoir variable of interest is again the cumulative oil production at the end of the total reservoir simulation time (30 years). We will use the surveillance



(a) Porosity (b) Horizontal permeability (c) Vertical permeability

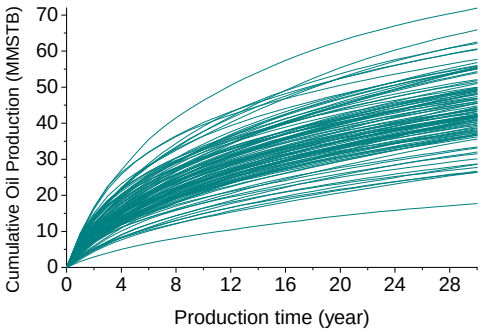
Figure 4.26: Porosity and permeability distribution in layer 1 of a realization.

data that can be obtained during the first 10 years to quantify the potential uncertainty reduction in cumulative oil production. Similar to the previous example, we consider the water rate, water front location, oil rate, build-up pressure and several combinations of these data. Due to the presence of multiple water injection wells, we cannot use the procedure in example 1 to model the water front because the water from different injectors can intermingle. Instead, we associate the water injected from each injector with one different type of tracer and calculate the water front location as the position where the tracer concentration from a particular well exceeds 0.01. These values are calculated along the 19 directions indicated by the blue lines in Fig. 4.25. We obtain the build-up pressure using the same procedure as in the first example, but this time, we shut-in the producers every two years instead of every year, and we also reduce the measurement error from 10 psi to 1 psi. The measurement frequency of other types of data has also been changed. The detailed specifications for each type of data are shown in Table 4.19.

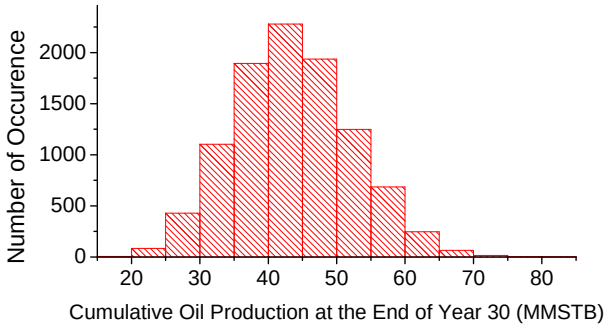
Table 4.19: Specifications of measurement data.

	Measurement Freq.	# of data	Units	Measurement Error
Oil rate	Every 6 months	12 per year	STB/D	3% measured value
Water rate	Every 6 months	12 per year	STB/D	10% measured value
pressure	2 days per 2 years	60 per 2 years	psi	1 psi
Front location	Every 12 months	19 per year	gridblock	0.1 grid block

We generate 10,000 realizations of the prior model and run simulations (using Eclipse) to obtain the cumulative oil production for all realizations. Fig. 4.27a shows the cumulative oil production as a function of time for the first 100 realizations. The associated prior distribution of cumulative oil production at the end of year 30 is shown in Fig. 4.27b. The prior P90 - P10 and the prior entropy of cumulative oil production are 22.6 MMSTB and 17.379, respectively.



(a) Cumulative oil production as a function of time for the first 100 realizations from the prior pdf



(b) Histogram of cumulative oil production at year 30

Figure 4.27: Cumulative oil production based on realization from the prior pdf, example 2.

In this example, we only use the linear-Gaussian method to estimate mutual information. Following the same procedure as above, we estimate the mutual information, conditional entropy and expected P90 - P10 of the cumulative oil production, conditional to 10 years of observed data. The results are shown in Table 4.20. The result of this table again shows that oil rate is the most useful data to obtain in order to reduce the uncertainty in cumulative oil production. Other types of data, such as water rate and pressure, when combined with oil rate can increase the reduction in uncertainty but not by a significant amount.

Next, we will attempt to verify the estimates of Table 4.20 using the exhaustive history-matching procedure. For this example, however, it may be more difficult to obtain an accurate quantification of the uncertainty in the posterior model because the number of model parameters (7980) is significantly greater than our computationally affordable ensemble size

Table 4.20: Estimates of the mutual information between cumulative oil production and 10 years of observed data, and the corresponding conditional entropy and expected P90 - P10 using the proposed procedure.

Row	Data	MI	Con. Entropy	Expected P90 - P10 (MMSTB)
1	Water rate (120 data)	0.496	16.883	13.75
2	Build-up pressure (300 data)	0.624	16.755	12.10
3	Water front (190 data)	0.301	17.077	16.70
4	Oil rate (120 data)	1.778	15.601	3.81
5	Oil rate + water rate (240 data)	1.808	15.571	3.74
6	Oil rate + pressure (420 data)	1.867	15.511	3.49
7	Oil rate + water front (310 data)	1.820	15.558	3.66

(maximum of 1000). We first test our history matching method (ES-MDA 4x (Emerick and Reynolds, 2013c,a)) for the first plausible truth case for two different ensemble sizes with and without covariance localization. In the cases where localization is used, we follow the distance-based approach described in Emerick and Reynolds (2011a). The histograms of the posterior distribution of cumulative oil production conditional to water rate are shown in Fig. 4.28. The results of Fig. 4.28 pertain to a single ensemble. We assume the histograms obtained using an ensemble size of 1000 are more accurate than those obtained using an ensemble size of 100. With this assumption, the history-matching result using an ensemble size of 100 without localization seems to exhibit the worst performance; specifically, the location of the mode is wrong and the spread is too small when compared to the results obtained using an ensemble size of 1000. Using localization with an ensemble size of 100, the location of the mode becomes more accurate but the spread is still too small. For ensemble sizes of 1000, we obtain similar results regardless of whether we use localization or not. Motivated by the results of Fig. 4.28, we carry out the exhaustive history matching procedure with an ensemble size of 1000 without localization. It is important to note that it is possible that a larger ensemble size can lead to a greater spread in the posterior histogram, but an ensemble size of 1000 is our computational limit in this study because we need to repeat the history-matching process for multiple plausible truth cases and then multiple

surveillance scenarios.

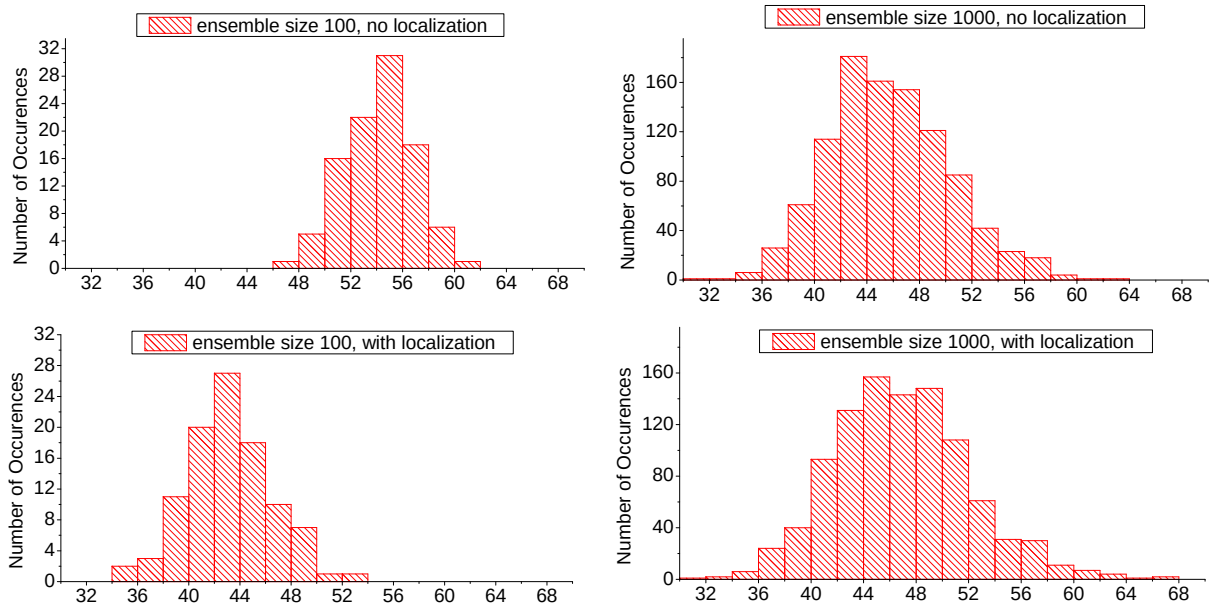


Figure 4.28: Histogram of cumulative oil production conditional to water rate (MMSTB).

The conditional entropy and average P90 - P10 calculated from the exhaustive history-matching procedure are shown in Table 4.21. For this example, the exhaustive history-matching method requires 2.8 million simulation runs compared to 10,000 simulations required by our proposed method. Comparing the results of Tables 4.20 and 4.21, our proposed method qualitatively agrees with the exhaustive history-matching procedure in ranking the five surveillance scenarios according to their posterior uncertainty, where the water rate data alone leads to the highest posterior uncertainty while the oil rate combined with pressure data leads to the smallest posterior uncertainty. Quantitatively, however, our proposed method agrees only approximately with the exhaustive history-matching procedure. The main issue here is that our conditional entropy estimates are greater than those calculated from the exhaustive history-matching procedure in several scenarios, and higher values of conditional entropy translate directly into greater expected values of P90 - P10. We have no provable explanation of this discrepancy. On the one hand, it is possible that computing mutual information using Eq. 4.37, which is based on the linear-Gaussian, tends to underestimate mutual information and thus overestimates conditional entropy and expected P90 - P10.

On the other hand, given d_{obs}, it can not be proved that ES-MDA samples the posterior conditional pdf correctly except in the linear case and ES-MDA often underestimate the posterior uncertainty. We believe the correct results may be somewhere between the results of Table 4.20 and Table 4.21.

Table 4.21: Estimates using exhaustive history matching (10 years of data).

Row	Data	Conditional Entropy	Expected P90 - P10 (MMSTB)
1	Water rate (120 data)	16.677	11.08 ($\pm$1.15)
2	Build-up pressure (300 data)	16.438	8.72 ($\pm$0.76)
3	Water front (190 data)	16.701	11.74 ($\pm$1.85)
4	Oil rate (120 daa)	15.645	3.92 ($\pm$0.61)
5	Oil rate + water rate (240 data)	15.532	3.50 ($\pm$0.51)
6	Oil rate + pressure (420 data)	15.215	2.56 ($\pm$0.35)
7	Oil rate + water front (310 data)	15.263	2.80 ($\pm$0.49)

CHAPTER 5

CONCLUSIONS

5.1 Adaptive ES-MDA

The adaptive ES-MDA algorithms developed here yield an improvement over the original ES-MDA algorithm in the sense that they provide sufficient damping to avoid over corrections to the property fields and improve the conditioning of the matrix that must be inverted in ES-MDA algorithms. The adaptive ES-MDA methods trade a preselection of the number of data assimilation and damping factors of the original ES-MDA algorithm for a preselection of a single parameter ρ, or a preselection of the number of standard deviations that we allow model parameters to change over an iteration. Although we believe the selection of a single parameter instead of several is desirable, how to choose ρ is not clear. Choosing ρ close to unity may cause the algorithms to require so many iterations that their application becomes computationally infeasible for large scale problems. Thus, it may still require some trial and error where we try a few iterations with $\rho = 0.5$ and then increase or decrease it based on the results. On the other hand, restricting the change in model parameters over an iteration to two standard deviations seems to work well in examples that we have tried but can sometime lead to more iterations than are necessary to obtain a good data match and smooth property fields.

Based on the example provided as well as computations not shown, the following conclusions are warranted:

1. The adaptive ES-MDA algorithms are more robust than the original ES-MDA in that they prevent over correction of the initial guesses and result in good data matches with reasonable rock property fields.

2. At early iterations, an exceeding large damping factor may be required. These damping factors are much larger than any we imagined when we developed and tested the original ES-MDA algorithm.

3. A good data match does not imply good or unbiased future predictions, although with the adaptive ES-MDA algorithms, most ensembles result in an ensemble with small bias in future predictions.

4. The adaptive ES-MDA algorithms show most advantages over the regular ES-MDA algorithms in the more difficult case with no hard data. In the more reasonable case with hard data, the regular ES-MDA algorithms work almost as well as the adaptive ES-MDA algorithms.

5. With the optimal choice of λ_{ini}, LM-EnRML can achieve a relatively decent data match while keeping the posterior relatively smooth. However, it still does not perform as well as the adaptive ES-MDA algorithms in the examples considered.

5.2 History Matching Non-Gaussian Facies

We proposed a two-stage procedure based on ES-MDA to history match data from non-Gaussian reservoir models, specifically those generated using multi-point statistics. In the first stage, a combination of the ES-MDA and MPS algorithms is used to obtained an approximation of the true facies distribution. In the second stage, ES-MDA is used to adjust the permeability within the facies to obtain a good data match. The proposed procedure was applied to three examples in a 2D channelized reservoir with different degrees of known information. After stage 1, we are able to obtain very good agreement between the facies distribution and the true model. The data match, however, is still poor after stage 1 due to the incorrect permeability distribution within each facies. After stage 2 is performed, the data match is greatly improved. We experimented with various ideas to modify the permeability means of the facies in stage 1, but found that it did not work well due to the incorrect channel configuration and the inability of our procedure to resolve the sinuosity of

the channels.

5.3 Surveillance Optimization

We proposed a procedure based on information theory to quantify the expected uncertainty reduction in a reservoir variable J, e.g., predicted cumulative oil production, that can be obtained by matching possible measured data from a reservoir surveillance operation before the actual measurement takes place. In this procedure, we estimated the conditional entropy of the reservoir variable of interest, J, by calculating the difference between its prior entropy and the mutual information between J and the vector of observed data, D_{obs}. Then we reconstructed the expected posterior pdf of J using the calculated conditional entropy and the assumed or calculated information about the shape of the posterior pdf of J. Metrics about the posterior uncertainty of J such as standard deviation or P90 - P10 were obtained directly from this reconstructed pdf. The proposed method is simple to implement and fast to execute as it only requires simulation runs for an ensemble of prior model realizations. The results obtained by our proposed method were verified with an exhaustive history-matching procedure. Based on theoretical arguments and results from the example problems, the following conclusions are warranted:

1. Our proposed method correctly ranks the potential surveillance scenarios according to the expected posterior uncertainty in cumulative oil recovery if the actual measurement takes place.

2. The estimates of our proposed method agree reasonably well with the exhaustive history-matching procedure in terms of the expected value of P90 - P10 of the cumulative oil production, but the information-theory-based procedure requires more than an order of magnitude less computational time.

3. The relative importance of each surveillance scenario is dependent on many factors, for example, the choice of reservoir variable J or the duration of data collection. Our proposed method provides an easy way to screen the possible surveillance scenarios to

determine the best one.

4. The computational cost of our proposed method is small and fixed regardless of the number of potential surveillance scenarios considered.

BIBLIOGRAPHY

Aanonsen, S. I., G. Nævdal, D. S. Oliver, A. C. Reynolds, and B. Vallés, Review of ensemble Kalman filter in petroleum engineering, *SPE Journal*, **14**(3), 393–412, 2009.

Abellan, A. and B. Noetinger, Optimizing subsurface field data acquisition using information theory, *Mathematical Geosciences*, **42**, 603–630, 2010, 10.1007/s11004-010-9285-6.

Agbalaka, C. and D. S. Oliver, Application of the EnKF and localization to automatic history matching of facies distribution and production data, *Mathematical Geosciences*, **40**(4), 353–374, 2008.

Aziz, K. and A. Settari, *Petroleum Reservoir Simulation*, Elsevier Applied Science Publishers, London, 1979.

Barker, J. W., M. Cuypers, and L. Holden, Quantifying uncertainty in production forecasts: Another look at the PUNQ-S3 problem, *SPE Journal*, **6**(4), 433–441, 2001.

Beirlant, J., E. J. Dudewicz, L. Györfi, and E. C. Meulen, Nonparametric entropy estimation: An overview, *International Journal of the Mathematical Statistics Sciences*, **6**, 17–39, 1997.

Bertino, L., G. Evensen, and H. Wackernagel, Sequential data assimilation techniques in oceanography, *International Statistical Review*, **71**(2), 223–241, 2003.

Bi, Z., D. S. Oliver, and A. C. Reynolds, Conditioning 3D stochastic channels to pressure data, *SPE Journal*, **5**(4), 474–484, 2000.

Bianco, A., A. Cominelli, L. Dovera, G. Nævdal, and B. Vallès, History matching and production forecast uncertainty by means of the ensemble Kalman filter: A real field application, in *Proceedings of the EAGE/EUROPEC Conference and Exhibition, London, U.K., 11–14 June*, SPE 107161, 2007.

Caers, J., History matching under training-image based geological model constraints, *SPE Journal*, **74716**, 218–226, 2003.

Caers, J. and T. Zhang, Multiple-point geostatistics: a quantitative vehicle for integrating geologic analogs into multiple reservoir models., *AAPG Mem*, **80**, 383–394, 2004.

Cellucci, C. J., A. M. Albano, and P. E. Rapp, Statistical validation of mutual information calculations: Comparison of alternative numerical algorithms, *Phys. Rev. E*, **71**, 066,208, 2005.

Chang, F., C.-J. Chen, and C.-J. Lu, A linear-time component-labeling algorithm using contour tracing technique, *Computer Vision and Image Understanding*, **93**(2), 206 – 220, 2004.

Chapman, D. and C. N. Thomas, Think big, get small, *Hart's E&P*, **83**(2), 2010.

Chen, C., G. Gao, B. Ramirez, J. Vink, and A. Girardi, Assisted history matching of channelized models using pluri-principal component analysis, in *2015 SPE Reservoir Simulation Symposium*, Society of Petroleum Engineers, Houston, Texas, USA, 2015.

Chen, Y. and D. Oliver, Cross-covariance and localization for EnKF in multiphase flow data assimilation, *Computational Geosciences*, **14**(4), 579–601, 2009.

Chen, Y. and D. Oliver, Levenberg-Marquardt forms of the iterative ensemble smoother for efficient history matching and uncertainty quantification, *Computational Geosciences*, **17**(4), 689–703, 2013.

Chen, Y. and D. Zhang, Data assimilation for transient flow in geologic formations via ensemble Kalman filter, *Advances in Water Resources*, **29**(8), 1107–1122, 2006.

Cover, T. M. and J. A. Thomas, *Elements of Information Theory*, Wiley-Interscience, New York, NY, USA, 1991.

Darbellay, G. and I. Vajda, Estimation of the information by an adaptive partitioning of the observation space, *Information Theory, IEEE Transactions on*, **45**(4), 1315 –1321, 1999.

Dempster, A. P., N. M. Laird, and D. B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, *Journal of The Royal Statistical Society: Series B*, **39**(1), 1–38, 1977.

Devegowda, D., E. Arroyo-Negrete, A. Datta-Gupta, and S. G. Douma, Efficient and robust reservoir model updating using ensemble Kalman filter with sensitivity-based covariance localization, in *Proceedings of the SPE Reservoir Simulation Symposium, Houston, Texas, 26–28 February*, SPE 106144, 2007.

Dong, Y., Y. Gu, and D. S. Oliver, Sequential assimilation of 4D seismic data for reservoir description using the ensemble Kalman filter, *Journal of Petroleum Science and Engineering*, **53**(1–2), 83–99, 2006.

Dovera, L. and E. Della Rossa, Ensemble Kalman filter for Gaussian mixture models, in *Extended abstracts book of Petroleum Geostatistics 2007, Cascais, Portugal, September, 10–14*, 2007.

Emerick, A. A. and A. C. Reynolds, Combining sensitivities and prior information for covariance localization in the ensemble Kalman filter for petroleum reservoir applications, *Computational Geosciences*, **15**(2), 251–269, 2011a.

Emerick, A. A. and A. C. Reynolds, History matching a field case using the ensemble Kalman filter with covariance localization, *SPE Reservoir Evaluation & Engineering*, **14**(4), 423–432, 2011b.

Emerick, A. A. and A. C. Reynolds, Combining the ensemble Kalman filter with Markov chain Monte Carlo for improved history matching and uncertainty characterization, *SPE Journal*, **17**(3), 639–659, 2012a.

Emerick, A. A. and A. C. Reynolds, History matching time-lapse seismic data using the ensemble Kalman filter with multiple data assimilations, *Computational Geosciences*, **16**(3), 639–659, 2012b.

Emerick, A. A. and A. C. Reynolds, Ensemble smoother with multiple data assimilations, *Computers & Geosciences*, **55**, 3–15, 2013a.

Emerick, A. A. and A. C. Reynolds, History-matching production and seismic data in a real field case using the ensemble smoother with multiple data assimilation, in *Proccedings of the SPE Reservoir Simulation Symposium, The Woodlands, Texas, USA, 18-20 February*, SPE SPE-163645, 2013b.

Emerick, A. A. and A. C. Reynolds, Investigation of the sampling performance of ensemble-based methods with a simple reservoir model, *Computational Geosciences*, **17**(2), 325–350, 2013c.

Evensen, G., 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.

Evensen, G., The ensemble Kalman filter: Theoretical formulation and practical implementation, *Ocean Dynamics*, **53**, 343–367, 2003.

Evensen, G., *Data Assimilation: The Ensemble Kalman Filter*, Springer, Berlin, 2007.

Evensen, G., J. Hove, H. C. Meisingset, E. Reiso, K. S. Seim, and O. Espelid, Using the EnKF for assisted history matching of a North Sea reservoir model, in *Proceedings of the SPE Reservoir Simulation Symposium, Houston, Texas, 26–28 February*, SPE 106184, 2007.

Floris, F. J. T., M. D. Bush, M. Cuypers, F. Roggero, and A.-R. Syversveen, Methods for quantifying the uncertainty of production forecasts: A comparative study, *Petroleum Geoscience*, **7**(SUPP), 87–96, 2001.

Galli, A., H. Beucher, G. Le Loc'h, B. Doligez, and H. Group, The pros and cons of the truncated Gaussian method, in *Geostatistical Simulations*, pp. 217–233, Kluwer Academic, Dordrecht, 1994.

Gao, G. and A. C. Reynolds, An improved implementation of the LBFGS algorithm for automatic history matching, *SPE Journal*, **11**(1), 5–17, 2006.

Gao, G., M. Zafari, and A. C. Reynolds, Quantifying uncertainty for the PUNQ-S3 problem in a Bayesian setting with RML and EnKF, *SPE Journal*, **11**(4), 506–515, 2006.

Gu, Y. and D. S. Oliver, History matching of the PUNQ-S3 reservoir model using the ensemble Kalman filter, *SPE Journal*, **10**(2), 51–65, 2005.

Han, M., Y. Zhao, and G. Li, Application of EM algorithm ofr seismic facies classification, *Computational Geosciences*, **15**(3), 421–429, 2011.

Hanke, M., A regularizing Levenberg-Marquardt scheme, with applications to inverse groundwater filtration problems, *Inverse Problems*, **13**(1), 79, 1997.

Hanke, M., The regularizing Levenberg-Marquardt scheme is of optimal order, *Journal of Integral Equations and Applications*, **22**(2), 259–283, 2010.

Hartley, H., Maximum likelihood from incomplete data, *Biometrics*, **14**(1), 174–194, 1958.

Haskett, W. J., Optimal appraisal well location through efficient uncertainty reduction and value of information techniques, in *Proceedings of the SPE Annual Technical Conference and Exhibition, Denver, Colorado, USA*, 2003.

Haugen, V., G. Nævdal, L.-J. Natvik, G. Evensen, A. M. Berg, and K. M. Flornes, History matching using the ensemble Kalman filter on a North Sea field case, *SPE Journal*, **13**(4), 382–391, 2008.

Houtekamer, P. L. and H. L. Mitchell, Ensemble Kalman filtering, *Quarterly Journal of the Royal Meteorological Society*, **131**, 3269–3289, 2005.

Hu, L. Y., Y. Zhao, Y. Liu, C. Scheepens, and A. Bouchard, Updating multipoint simulations using the ensemble Kalman filter, *Computers & Geosciences*, **51**, 7–15, 2013.

Iglesias, M. A., Iterative regularizing ensemble Kalman methods for inverse problems, in *Proceedings of the International EnKF Workshop, Bergen, Norway, 23-25 June*, 2014.

Iglesias, M. A., Iterative regularization for ensemble-based data assimilation in reservoir models, *Computational Geosciences*, **19**, 177–212, 2015.

Iglesias, M. A. and C. Dawson, The regularizing Levenberg-Marquardt scheme for history matching of petroleum reservoirs, *Computational Geosciences*, **17**(6), 1033–1053, 2013.

Jafapour, B. and M. Khodabakhsi, A probability conditioning method (pcm) for nonlinear flow data integration into multipoint statistical facies simulation, *Mathematical Geology*, **43**(2), 133–164, 2011.

Jafarpour, B. and D. B. McLaughlin, Reservoir characterization with the discrete cosine transform, *SPE Journal*, **14**(01), 2009.

Jansen, J. D., S. D. Douma, D. R. Brouwer, P. M. J. V. den Hof, and A. W. Heemink, Closed-loop reservoir management, in *Proceedings of the SPE Reservoir Simulation Symposium, The Woodlands, Texas, 2–4 February*, SPE 119098, 2009.

Jolliffe, I. T., *Principal Component Analysis*, second edn., Springer, 2002.

Journel, A., Combining knowledge from diverse sources: An alternative to traditional data independence hypotheses, *Mathematical Geology*, **34**(5), 573–596, 2002.

Keppenne, C. L. and M. M. Rienecker, Assimilation of temperature into an isopycnal ocean general circulation model using a parallel ensemble Kalman filter, *Journal of Marine Systems*, **40–41**, 363–380, 2003.

Kitanidis, P. K., Quasi-linear geostatistical theory for inversing, *Water Resources Research*, **31**(10), 2411–2419, 1995.

Kraskov, A., H. Stogbaurer, and P. Grassberger, Estimating mutual information, *Physcial Review E*, **69**(066138), 1–15, 2004.

Kung, S., M. Mak, and S. Lin, *Biometric Authentication: A Machine Learning Approach*, Prentice Hall, 2004.

Le, D. H. and A. C. Reynolds, Estimation of depths of initial fluid contacts, relative and absolute permeabilities and porosity using iterative ensemble smoothers and EnKF, TUPREP presentation, The University of Tulsa, 2013.

Le, D. H., A. C. Reynolds, and R. Younis, A history matching procedure for non-gaussian facies based on es-mda, TUPREP research report, The University of Tulsa, 2015.

Le Loc'h, G., H. Beucher, A. Galli, B. Doligez, and H. Group, Improvement in the truncated Gaussian method: Combining several Gaussian functions, in *Proceedings of ECMOR IV, Fourth European Conference on the Mathematics of Oil Recovery*, 1994.

Le Loc'h, G. and A. Galli, Truncated plurigaussian method: Theoretical and practical points of view, in *Geostatistics Wollongong '96*, (edited by E. Y. Baafi and N. A. Schofield), vol. 1, pp. 211–222, Kluwer Academic, 1997.

Leone, F. C., L. S. Nelson, and R. B. Nottingham, The folded normal distribution, *Technometrics*, **3**(4), pp. 543–550, 1961.

Li, G. and A. C. Reynolds, Iterative ensemble Kalman filters for data assimilation, *SPE Journal*, **14**(3), 496–505, 2009.

Li, L. and B. Jafarpour, A sparse Bayesian framework for conditioning uncertain geologic models to nonlinear flow measurements, *Advances in Water Resources*, **33**(9), 1024 – 1042, 2010.

Li, R., A. C. Reynolds, and D. S. Oliver, History matching of three-phase flow production data, *SPE Journal*, **8**(4), 328–340, 2003.

Liu, G., Y. Chen, and D. Zhang, Investigation of flow and transport processes at the MADE site using ensemble Kalman filter, *Advances in Water Resources*, **31**(7), 975–986, 2008.

Liu, N. and D. S. Oliver, Critical evaluation of the ensemble Kalman filter on history matching of geologic facies, *SPE Reservoir Evaluation & Engineering*, **8**(4), 470–477, 2005.

Lødøen, O. P. and H. Omre, Scale-corrected ensemble kalman filtering applied to production-history conditioning in reservoir evaluation, *SPE Journal*, **13**(2), 177–194, 2008.

Lorentzen, R. J., K. K. Fjelde, J. Frøyen, A. C. Lage, G. Nævdal, and E. H. Vefring, Underbalanced and low-head drilling operations: Real time interpretation of measured data and operational support, in *Proceedings of the SPE Annual Technical Conference and Exhibition, New Orleans, Louisiana, 30 September–3 October*, SPE 71384, 2001.

Meng, X.-L., Maximum likelihood estimation via the ECM algorithm: A general framework, *Biometrika*, **80**(2), 267–278, 1993.

Meng, X.-L., On the rate of convergence of the ECM algorithm, *Annals of Statistics*, **22**(1), 326–339, 1994.

Nævdal, G., L. M. Johnsen, S. I. Aanonsen, and E. H. Vefring, Reservoir monitoring and continuous model updating using ensemble Kalman filter, *SPE Journal*, **10**(1), 66–74, 2005.

Nævdal, G., T. Mannseth, and E. H. Vefring, Near-well reservoir monitoring through ensemble Kalman filter, in *Proceedings of the SPE/DOE Improved Oil Recovery Symposium, 13–17 April*, SPE 75235, 2002.

Oliver, D. S. and Y. Chen, Improved initial sampling for the ensemble Kalman filter, *Computational Geosciences*, **13**(1), 13–27, 2009.

Oliver, D. S. and Y. Chen, Recent progress on reservoir history matching: a review, *Computational Geosciences*, **15**(1), 185–221, 2010.

Oliver, D. S., N. He, and A. C. Reynolds, Conditioning permeability fields to pressure data, in *Proccedings of the European Conference for the Mathematics of Oil Recovery*, 1996.

Oliver, D. S., A. C. Reynolds, and N. Liu, *Inverse Theory for Petroleum Reservoir Characterization and History Matching*, Cambridge University Press, Cambridge, UK, 2008.

Peters, L., R. Arts, G. Brouwer, C. Geel, S. Cullick, R. Lorentzen, Y. Chen, K. Dunlop, F. Vossepoel, R. Xu, P. Sarma, A. Alhuthali, and A. Reynolds, Results of the Brugge benchmark study for flooding optimisation and history matching, *SPE Reservoir Evaluation & Engineering*, **13**(3), 391–405, 2010.

Reichle, R. H., D. B. McLaughlin, and D. Entekhabi, Hydrologic data assimilation with the ensemble Kalman filter, *Monthly Weather Review*, **130**(1), 103–114, 2002.

Reynolds, A. C., M. Zafari, and G. Li, Iterative forms of the ensemble Kalman filter, in *Proceedings of 10th European Conference on the Mathematics of Oil Recovery, Amsterdam, 4–7 September*, 2006.

Roulston, M. S., Estimating the errors on measured entropy and mutual information, *Physica D: Nonlinear Phenomena*, **125**(34), 285 – 294, 1999.

Sakov, P. and P. R. Oke, A deterministic formulation of the ensemble Kalman filter: an alternative to ensemble square root filters, *Tellus A*, **60**(2), 361–371, 2008.

Samet, H. and M. Tamminen, Efficient component labeling of images of arbitrary dimension represented by linear bintrees, *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **10**(4), 579–586, 1988.

Sarma, P., L. J. Durlofsky, and K. Aziz, Kernel principal component analysis for efficient differentiable parameterization of multipoint geostatistics, *Mathematical Geosciences*, **40**, 3–32, 2008.

Schlumberger, *Petrel 2010 User Guide*, Houston, TX, 2010.

Sebacher, B., A. Stordal, and R. Hanea, Bridging multipoint statistics and truncated Gaussian fields for improved estimation of channelized reservoirs with ensemble methods, *Computational Geosciences*, **19**(2), 341–369, 2015.

Shannon, C. E., A mathematical theory of communication, *Bell system technical journal*, **27**, 1948.

Skjervheim, J.-A., G. Evensen, S. I. Aanonsen, B. O. Ruud, and T. A. Johansen, Incorporating 4D seismic data in reservoir simulation models using ensemble Kalman filter, *SPE Journal*, **12**(3), 282–292, 2007.

Steur, R., J. Kurths, C. O. Daub, J. Weise, and J. Selbig, The mutual information: Detecting and evaluating dependencies between variables, *Bioinformatics*, **18**(066138), S331–S240, 2002.

Strebelle, S., Conditional simulation of complex geological structures using multiple-point statistics, *Mathematical Geology*, **34**(1), 1–22, 2002.

Szunyogh, I., E. J. Kostelich, G. Gyarmati, D. J. Patil, B. R. Hunt, E. Kalnay, E. Ott, and J. A. Yorke, Assessing a local ensemble Kalman filter: perfect model experiments with the national centers for environmental prediction global model, *Tellus A*, **57**, 528–545, 2005.

Tarantola, A., *Inverse Problem Theory and Methods for Model Parameter Estimation*, SIAM, Philadelphia, USA, 2005.

Tavakoli, R. and A. C. Reynolds, History matching with parameterization based on the SVD of a dimensionless sensitivity matrix, *SPE Journal*, **15**(12), 495–508, 2010.

Tavakoli, R. and A. C. Reynolds, Monte Carlo simulation of permeability fields and reservoir performance predictions with SVD parameterization in RML compared with EnKF, *Computational Geosciences*, **15**(1), 99–116, 2011.

Thulin, K., G. Li, S. I. Aanonsen, and A. C. Reynolds, Estimation of initial fluid contacts by assimilation of production data with EnKF, in *Proceedings of the SPE Annual Technical Conference and Exhibition, Anaheim, California, 11–14 November*, SPE 109975, 2007.

Ullo, J., Computational challenges in the search for and production of hydrocarbons, *Scientific Modeling and Simulation SMNS*, **15**(1-3), 313–337, 2008.

van Leeuwen, P. J. and G. Evensen, Data assimilation and inverse methods in terms of a probabilistic formulation, *Monthly Weather Review*, **124**, 2898–2913, 1996.

Vo, H. X. and L. J. Durlofsky, A new differentiable parameterization based on principal component analysis for the low-dimensional representation of complex geological models, *Mathematical Geosciences*, **46**(7), 775–813, 2014.

Wang, Y., G. Li, and A. C. Reynolds, Estimation of depths of fluid contacts by history matching using iterative ensemble-Kalman smoothers, *SPE Journal*, **15**(2), 2010.

Wolfram, S., *The Mathematica Book (4th edition)*, Cambridge University Press, New York, NY, USA, 1999.

Zafari, M. and A. C. Reynolds, Assessing the uncertainty in reservoir description and performance predictions with the ensemble Kalman filter, *SPE Journal*, **12**(3), 382–391, 2007.

Zhang, F., A. C. Reynolds, and D. S. Oliver, Evaluation of the reduction in uncertainty obtained by conditioning a 3D stochastic channel to multiwell pressure data, *Mathematical Geology*, **34**(6), 713–740, 2002.

Zhang, T. and A. Journel, Merging prior geological structure and local data: the mp geostatistics answer, SCRF annual report, 2003.

Zhao, Y., G. Li, and A. C. Reynolds, Characterization of the measurement error in time-lapse seismic data and production data with an EM algorithm, *Oil & Gas Science and Technology - Rev. IFP*, **62**(2), 181–193, 2007.

Zhao, Y., A. C. Reynolds, and G. Li, Generating facies maps by assimilating production data and seismic data with the ensemble Kalman filter, in *Proccedings of the SPE Improved Oil Recovery Symposium, Tulsa, Oklahoma, 20–23 April*, SPE 113990, 2008.

APPENDIX A

APPENDIX ON ADAPTIVE ES-MDA

A.1 LM-EnRML

To derive the LM-EnRML equation, we first start with the standard Gauss-Newton update equation:

$$\delta m_{\ell+1} = -(C_M^{-1} + G_\ell^T C_D^{-1} G_\ell)^{-1} \left[C_M^{-1}(m_\ell - m_{uc,j}) + G_\ell^T C_D^{-1}(g(m_\ell) - d_{uc,j}) \right]. \tag{A.1}$$

where $\delta m_{\ell+1}$ is the change in the vector model parameters, C_M is the prior covariance matrix of model parameters, G_ℓ is the sensitivity matrix, C_D is the covariance matrix of measurement error, m_ℓ is the current vector of model parameters, $m_{uc,j}$ is the vector of perturbed prior model parameters, g is the forward model, $d_{uc,j}$ is the vector of perturbed observed data. Chen and Oliver (2013) replace C_M^{-1} in the Hessian term by $(C_M^\ell)^{-1}$,

$$\delta m_\ell = -\left[(C_M^\ell)^{-1} + G_\ell^T C_D^{-1} G_\ell \right]^{-1} \left[C_M^{-1}(m_\ell - m_{uc,j}) + G_\ell^T C_D^{-1}(g(m_\ell) - d_{uc,j}) \right]. \tag{A.2}$$

The equation is then transformed to Levenberg-Marquardt form by the addition of an inflation factor,

$$\delta m_\ell = -\left[(1+\lambda_\ell)(C_M^\ell)^{-1} + G_\ell^T C_D^{-1} G_\ell \right]^{-1} \left[C_M^{-1}(m_\ell - m_{uc,j}) + G_\ell^T C_D^{-1}(g(m_\ell) - d_{uc,j}) \right]. \tag{A.3}$$

At early iterations, we want λ to be large so that the model update is highly influenced by the prior model. As the data assimilation progresses, we want to gradually reduce λ_ℓ to increase the relative influence of the observed data on the model update. To change λ,

we multiple or divide it by a factor γ between iterations. The choice of the initial λ and the factor γ can affect performance of the algorithm. Using matrix inversion formula, the equation can be rewritten as

$$\delta m_\ell = -\left[(1+\lambda_\ell)(C_M^\ell)^{-1} + G_\ell^T C_D^{-1} G_\ell\right]^{-1} C_M^{-1}(m_\ell - m_{uc,j}) \\ - C_M^\ell G_\ell^T \left[(1+\lambda_\ell)C_D + G_\ell C_M^\ell G_\ell^T\right]^{-1}(g(m_\ell) - d_{uc,j}). \quad \text{(A.4)}$$

Finally, we apply standard ensemble-based approximation for the covariance matrix of model parameters and the sensitivity matrix,

$$\delta m_\ell = -\left[(1+\lambda_\ell)(\Delta M^\ell \Delta M^{\ell^T})^{-1} + (\Delta M^\ell)^{-T}\Delta D^{\ell^T} C_D^{-1}\Delta D^\ell (\Delta M^\ell)^{-1}\right]^{-1} C_M^{-1}(m_\ell - m_{uc,j}) \\ - C_{MD}^\ell \left[(1+\lambda_\ell)C_D + C_{DD}^\ell\right]^{-1}(g(m_\ell) - d_{uc,j}). \quad \text{(A.5)}$$

A.2 Measurement of Model Difference

We consider the special case when the prior realizations and the true model, m_{true} are generated from the same Gaussian distribution, $N(m_{pr}, C_M)$, i.e.,

$$m_j = m_{pr} + C_M^{1/2} z_j, \quad \text{(A.1)}$$

and

$$m_{\text{true}} = m_{pr} + C_M^{1/2} z_{\text{true}}, \quad \text{(A.2)}$$

where m_j is an unconditional prior realization; z_j is the vector of standard normal deviates used to generate m_j; z_{true} is the vector of standard normal deviates used to generate m_{true}. In general, we only generate one true model for each history matching problem versus multiple prior realizations, so z_{true} can be considered constant in comparison to z_j. Following the

definition of Eq. 2.35, the model difference between m_j and m_{true} can be written as

$$\begin{aligned} R_j &= \frac{1}{N_m}||C_M^{-1/2}(m_j - m_{\text{true}})||_1 \\ &= \frac{1}{N_m}||C_M^{-1/2}\left((m_j - m_{pr}) - (m_{\text{true}} - m_{pr})\right)||_1 \\ &= \frac{1}{N_m}||C_M^{-1/2}\left(C_M^{1/2}z_j - C_M^{1/2}z_{\text{true}}\right)||_1 \\ &= \frac{1}{N_m}||(z_j - z_{\text{true}})||_1 \\ &= \frac{1}{N_m}\sum_{i=1}^{N_m}|z_{j,i} - z_{\text{true},i}|, \end{aligned} \tag{A.3}$$

where N_m is the number of model parameters, $z_{j,i}$ and $z_{\text{true},i}$ are respectively the ith components of z_j and z_{true}. The average model difference over the whole ensemble, $\bar{R}$, is,

$$\bar{R} \approx E[R_j] = E\left[\frac{1}{N_m}\sum_{i=1}^{N_m}|z_{j,i} - z_{\text{true},i}|\right] = \frac{1}{N_m}\sum_{i=1}^{N_m}E\left[|z_{j,i} - z_{\text{true},i}|\right], \tag{A.4}$$

where $E[*]$ is the expectation operator. Since $z_{j,i}$ follows the standard normal distribution $N(0,1)$, $|z_{j,i} - z_{\text{true},i}|$ follows the folded normal distribution (Leone et al., 1961) with mean $-z_{\text{true},i}$ and unit standard deviation. Its expectation, $E[|z_{j,i} - z_{\text{true},i}|]$, is dependent on the value of $z_{\text{true},i}$ and equal to

$$E[|z_{j,i} - z_{\text{true},i}|] = \sqrt{\frac{2}{\pi}}\exp\left(\frac{-z_{\text{true},i}^2}{2}\right) - (-z_{\text{true},i})\text{erf}\left(\frac{z_{\text{true},i}}{\sqrt{2}}\right), \tag{A.5}$$

so

$$\bar{R} \approx \frac{1}{N_m}\sum_{i=1}^{N_m}\left[\sqrt{\frac{2}{\pi}}\exp\left(\frac{-z_{\text{true},i}^2}{2}\right) - (-z_{\text{true},i})\text{erf}\left(\frac{z_{\text{true},i}}{\sqrt{2}}\right)\right]. \tag{A.6}$$

For a large value of N_m, which is generally true for most history matching problems of

interest, we have

$$\begin{aligned}\bar{R} &\approx E\left[\sqrt{\frac{2}{\pi}}\exp\left(\frac{-z_{\text{true},i}^2}{2}\right) - (-z_{\text{true},i})\text{erf}\left(\frac{z_{\text{true},i}}{\sqrt{2}}\right)\right] \\ &= E\left[\sqrt{\frac{2}{\pi}}\exp\left(\frac{-z_{\text{true},i}^2}{2}\right)\right] + E\left[z_{\text{true},i}\text{erf}\left(\frac{z_{\text{true},i}}{\sqrt{2}}\right)\right].\end{aligned} \tag{A.7}$$

For conciseness, we replace $z_{\text{true},i}$ by z since each $z_{\text{true},i}$ is independently sampled from the standard Gaussian distribution,

$$\bar{R} \approx E\left[\sqrt{\frac{2}{\pi}}\exp\left(\frac{-z^2}{2}\right)\right] + E\left[z\text{erf}\left(\frac{z}{\sqrt{2}}\right)\right]. \tag{A.8}$$

By noting that z follows the standard normal distribution with the probability density function $f_Z(z)$ and the cumulative density function $F_Z(z)$, the first term can be rewritten as,

$$\begin{aligned}E\left[\sqrt{\frac{2}{\pi}}\exp\left(\frac{-z^2}{2}\right)\right] &= \sqrt{\frac{2}{\pi}}\int_{-\infty}^{\infty}\exp\left(\frac{-z^2}{2}\right)f_Z(z)\text{d}z \\ &= \sqrt{\frac{2}{\pi}}\int_{-\infty}^{\infty}\exp\left(\frac{-z^2}{2}\right)\frac{1}{\sqrt{2\pi}}\exp\left(\frac{-z^2}{2}\right)\text{d}z \\ &= \frac{1}{\pi}\int_{-\infty}^{\infty}\exp\left(-z^2\right)\text{d}z \\ &= \frac{1}{\pi}\sqrt{\pi} \\ &= \frac{1}{\sqrt{\pi}}.\end{aligned} \tag{A.9}$$

The second term can be rewritten as

$$\begin{aligned}E\left[z\text{erf}\left(\frac{z}{\sqrt{2}}\right)\right] &= \int_{-\infty}^{\infty} z\text{erf}\left(\frac{z}{\sqrt{2}}\right)f_Z(z)\text{d}z \\ &= 2\int_{-\infty}^{\infty} z\left[\frac{1}{2}\left(1+\text{erf}\left(\frac{z}{\sqrt{2}}\right)\right)\right]f_Z(z)\text{d}z - \int_{-\infty}^{\infty} zf_Z(z)\text{d}z \\ &= 2\int_{-\infty}^{\infty} zF_Z(z)f_Z(z)\text{d}z - 0,\end{aligned} \tag{A.10}$$

where $F_Z(z) = \frac{1}{2}\left(1 + \text{erf}\left(\frac{z}{\sqrt{2}}\right)\right)$ is the cumulative distribution function (cdf) for $f_Z(z)$. We continue the derivation:

$$\begin{aligned}
E\left[z\text{erf}\left(\frac{z}{\sqrt{2}}\right)\right] &= 2\int_{-\infty}^{\infty} zF_Z(z)\frac{1}{\sqrt{2\pi}}\exp\left(\frac{-z^2}{2}\right)\mathrm{d}z \\
&= \frac{\sqrt{2}}{\sqrt{\pi}}\int_{-\infty}^{\infty} F_Z(z)\exp\left(\frac{-z^2}{2}\right)\mathrm{d}\left(\frac{-z^2}{2}\right)(-1) \\
&= -\frac{\sqrt{2}}{\sqrt{\pi}}\int_{-\infty}^{\infty} F_Z(z)\mathrm{d}\left(\exp\left(\frac{-z^2}{2}\right)\right) \\
&= -\frac{\sqrt{2}}{\sqrt{\pi}}\left\{F_Z(z)\exp\left(\frac{-z^2}{2}\right)\Big|_{-\infty}^{\infty} - \int_{-\infty}^{\infty}\exp\left(\frac{-z^2}{2}\right)f_Z(z)\mathrm{d}z\right\} \\
&= \frac{\sqrt{2}}{\sqrt{\pi}}\left\{0 - 0 + \int_{-\infty}^{\infty}\exp\left(\frac{-z^2}{2}\right)\frac{1}{\sqrt{2\pi}}\exp\left(\frac{-z^2}{2}\right)\mathrm{d}z\right\} \\
&= \frac{1}{\pi}\int_{-\infty}^{\infty}\exp\left(-z^2\right)\mathrm{d}z \\
&= \frac{1}{\pi}\sqrt{\pi} \\
&= \frac{1}{\sqrt{\pi}}. \qquad \text{(A.11)}
\end{aligned}$$

Using Eqs. A.9 and A.11 in A.8, we have

$$\bar{R} \approx \frac{2}{\sqrt{\pi}}. \qquad \text{(A.12)}$$

APPENDIX B

APPENDIX ON HISTORY MATCHING NON-GAUSSIAN FACIES

B.1 Neighborhood Expectation-Maximization Algorithm

Before listing steps in the algorithm, it is useful to recap some notations. The Gaussian mixture model is defined as

$$\Theta = (\pi_j, f_j),\ j = 1, ..., M, \tag{B.13}$$

where M is the number of facies, i.e., number of Gaussian models; f_j denotes the jth Gaussian probability density function (pdf), and π_j denotes the probability of each facies or each Gaussian distribution, i.e. the probability that a gridblock is occupied by facies j. The jth Gaussian distribution, f_j, has mean μ_j and covariance matrix C_j. The dimension of μ_j is equal to the number of petrophysical properties. In our case, we only have permeability, so μ_j is a real number. The covariance matrix C_j in our case is also a real number, which is equal to the variance of the permeability within each facies. Let N be the number of gridblocks and m_i, $i = 1, 2, ..., N$, be the permeability value at gridblock i. The steps of the algorithm are as follows:

1. Set the initial guess for the facies classification of each gridblock. This is done based on the gridblock permeability values, where facies 1 is assigned to the N/M lowest permeability gridblocks, facies 2 is assigned the next N/M lowest permeability gridblocks and so on.

2. Calculate the initial facies proportion π_j, mean μ_j and variance C_j of each Gaussian distribution based on the initial guess of the facies classification.

3. Set the initial value of the membership matrix h, which is of size $N \times M$: if, in the initial guess, facies j is assigned to gridblock i, then set $h_i^j = 1$, otherwise set $h_i^j = 0$. Here, h_i^j denotes the element at the ith row and jth column of h; h_i^j dictates how likely gridblock i belongs to facies j and has the value from 0 to 1. The sum of all elements in one row has to be equal to 1, i.e. $\sum_{j=1}^{M} h_i^j = 1$ for all i's. Let h_0 denote the initial value of h.

4. OUTER LOOP - FOR $\ell = 1$ TO $\ell_{\max}$:

   (a) Calculate a new value for h, h_ℓ. This is called the expectation step and requires an iterative procedure. The starting value for h_ℓ is denoted by $h_{\ell,0}$ and is set equal to the value of h obtained at the previous iteration, $h_{\ell-1}$. h_ℓ is then updated iteratively as follows:

   INNER LOOP - FOR $m = 1$ TO $m_{\max}$:

   i. Update each of component of h individually:

$$(h_i^j)_{\ell,m} = \frac{\pi_j f_j(m_i) \exp(\beta \sum_{k=1}^{N} (h_k^j)_{\ell,m-1} v_{ik})}{\sum_{j'=1}^{M} \left\{ \pi_{j'} f_{j'}(m_i) \exp(\beta \sum_{k=1}^{N} (h_k^{j'})_{\ell,m-1} v_{ik}) \right\}} \tag{B.14}$$

   where v_{ik} defines the neighborhood, i.e.,

$$v_{ik} = \begin{cases} 1, & \text{if } i \text{ and } k \text{ are the indices of two neighbor gridblocks} \\ & \text{and } i \neq k, \\ 0, & \text{otherwise.} \end{cases} \tag{B.15}$$

   Note that $f_j(m_i)$ are evaluated using the current value of μ_j and C_j.

   ii. Calculate the change in h.

$$\delta h = \sum_{i=1}^{N} \sum_{j=1}^{M} \left\{ (h_i^j)_{\ell,m} - (h_i^j)_{\ell,m-1} \right\}. \tag{B.16}$$

If δh is smaller than 0.01, we break out of the inner loop before $m_{\max}$ is reached.

END OF INNER LOOP.

(b) Calculate new parameters for the Gaussian mixture model. This is called the maximization step.

$$(\mu_j)_\ell = \frac{\sum_{i=1}^{N}(h_i^j)_\ell m_i}{\sum_{i=1}^{N}(h_i^j)_\ell}, \tag{B.17}$$

$$(C_j)_\ell = \frac{\sum_{i=1}^{N}(h_i^j)_\ell (m_i - (\mu_j)_\ell)(m_i - (\mu_j)_\ell)^T}{\sum_{i=1}^{N}(h_i^j)_\ell}, \tag{B.18}$$

$$(\pi_j)_\ell = \frac{1}{N}\sum_{i=1}^{N}(h_i^j)_\ell. \tag{B.19}$$

END OF OUTER LOOP.

5. Determine the facies classification of each gridblock using a stochastic process. For each gridblock i, we sample a number u from the uniform distribution $U(0,1)$. For the problems in this research where there are only two facies, if $u < h_i^1$, we assign facies 1 to gridblock i, otherwise we assign facies 2 to gridblock j.

B.2 Tau Model

Originally, the multi-point statistics algorithm (Strebelle, 2002) calculated the probability of a facies entirely based on the training image using multi-point statistics. With the need to incorporate different sources of data, Journel (2002); Zhang and Journel (2003) propose the Tau model which allows the calculation of the facies probability based on both the training image and user-specified soft data (probabilities). In addition, the users can specify the relative weight of the probability calculated from the training image and from soft data by changing the τ parameters, hence the name Tau model. Let p be the probability that facies i occupies gridblock j; p can be determined using the following implicit equation:

$$p = \frac{1}{1 + f_p^{1-\tau_1-\tau_2} f_{p1}^{\tau_1} f_{p2}^{\tau_2}}, \tag{B.1}$$

where

- p_1 = probability that facies i occupies gridblock j as calculated from the training image using multi-point statistics,
- p_2 = probability that facies i occupies gridblock j as specified by the probability map (soft probability),
- τ_1 = weight of p_1,
- τ_2 = weight of p_2,
- $f_p = \frac{1}{p} - 1$,
- $f_{p_1} = \frac{1}{p_1} - 1$,
- $f_{p_2} = \frac{1}{p_2} - 1$.

Note that in the above equation, p appears on both side of the equation. The relative weight of the soft data increases as τ_1 decreases and τ_2 increases. To illustrate the effect of τ_1 and τ_2 on the value of p, we performed a few experiments:

- If either p_1 or p_2 is 1, p is 1 regardless of the value of τ_1 and τ_2.
- Suppose $p_1 = 0.1$, $p_2 = 0.8$:
  - If $\tau_1 = \tau_2 = 1$, $p = 0.4$, which is about the midway between 0.1 and 0.8.
  - If $\tau_1 = \tau_2 = 2$ or $\tau_1 = \tau_2 = 3$, p is still equal to 0.4.
  - If $\tau_1 = 1$ and $\tau_2 = 3$, $p = 0.62$, which is closer to the value specified by the probability map.
  - If $\tau_1 = 1$ and $\tau_2 = 9$, $p = 0.73$, which is even closer to the value specified by the probability map.
  - If $\tau_1 = 3$ and $\tau_2 = 1$, $p = 0.21$, which is closer to the value calculated from the training image.

In this research, we use $\tau_1 = 1$ and $\tau_2 = 9$ so that the new realizations honor the probability map better.

APPENDIX C

APPENDIX ON SURVEILLANCE OPTIMIZATION

C.1 Derivation of Expression for Mutual Information

We start from the definition

$$
\begin{aligned}
H(X|Y) &\equiv \sum_{j=1}^{M} P(Y_j) H(X|Y = Y_j) \\
&= -\sum_{j=1}^{M} P(Y_j) \sum_{i=1}^{N} P(X_i|Y_j) \log P(X_i|Y_j) \\
&= -\sum_{j=1}^{M} \sum_{i=1}^{N} P(Y_j) P(X_i|Y_j) \log P(X_i|Y_j) \\
&= -\sum_{j=1}^{M} \sum_{i=1}^{N} P(X_i, Y_j) \log P(X_i|Y_j) \\
&= -\sum_{j=1}^{M} \sum_{i=1}^{N} P(X_i, Y_j) \log \frac{P(X_i, Y_j)}{P(Y_j)} \\
&= -\sum_{i=1}^{N} \sum_{j=1}^{M} P(X_i, Y_j) \log \frac{P(X_i, Y_j)}{P(Y_j)}.
\end{aligned}
\tag{C.1}
$$

The expression for mutual information can be written as

$$
\begin{aligned}
I(X;Y) &= H(X) - H(X|Y) \\
&= -\sum_{i=1}^{N} P(X_i)\log P(X_i) + \sum_{i=1}^{N}\sum_{j=1}^{M} P(X_i,Y_j)\log\frac{P(X_i,Y_j)}{P(Y_j)} \\
&= -\sum_{i=1}^{N}\left(\sum_{j=1}^{M} P(X_i,Y_j)\right)\log P(X_i) + \sum_{i=1}^{N}\sum_{j=1}^{M} P(X_i,Y_j)\log\frac{P(X_i,Y_j)}{P(Y_j)} \\
&= \sum_{i=1}^{N}\sum_{j=1}^{M}\left(-P(X_i,Y_j)\log P(X_i) + P(X_i,Y_j)\log\frac{P(X_i,Y_j)}{P(Y_j)}\right) \\
&= \sum_{i=1}^{N}\sum_{j=1}^{M}\left(P(X_i,Y_j)\log\frac{1}{P(X_i)} + P(X_i,Y_j)\log\frac{P(X_i,Y_j)}{P(Y_j)}\right) \\
&= \sum_{i=1}^{M}\sum_{j=1}^{N} P(X_i,Y_j)\log\frac{P(X_i,Y_j)}{P(X_i)P(Y_j)}.
\end{aligned}
\tag{C.2}
$$

The derivation is complete.

C.2 Derivation of Expressions for Mutual Information in Linear Gaussian Case

First we derive the expression for $I(m, D_{obs})$. Note that D_{obs} is not the traditional d_{obs}, which is $Gm_{\text{true}} + \epsilon_d$, but a random vector of plausible observed data. In other words, d_{obs} is a realization of the random vector D_{obs}. Because m and D_{obs} are both Gaussian, following the derivation of Eq. 4.18, the mutual information between m and D_{obs} can be written as:

$$
I(m, D_{obs}) = \frac{1}{2}\log\left(\frac{|C_M||C_{D_{obs}}|}{|C|}\right), \tag{C.3}
$$

where

$$
C = \begin{bmatrix} C_M & C_{MD_{obs}} \\ C_{D_{obs}M} & C_{D_{obs}} \end{bmatrix}. \tag{C.4}
$$

Assuming that the random vectors ϵ_d and m are independent, the expression for $C_{D_{obs}}$

are easy to find. First,

$$D_{obs} = Gm + \epsilon_d \sim N(Gm_{prior}, GC_M G^T) + N(0, C_D), \tag{C.5}$$

which implies that

$$C_{D_{obs}} = GC_M G^T + C_D. \tag{C.6}$$

The expression for $C_{MD_{obs}}$ can be derived as follows:

$$\begin{aligned} C_{MD_{obs}} &= \mathrm{cov}(m, D_{obs}) \\ &= \mathrm{cov}(m, Gm) + \mathrm{cov}(m, \epsilon_d) \\ &= \mathrm{cov}(m, Gm) + 0 \\ &= E[(m - E[m])(Gm - E[Gm])^T] \\ &= E[(m - m_{\mathrm{prior}})(Gm - Gm_{\mathrm{prior}})^T] \\ &= E[(m - m_{\mathrm{prior}})(m - m_{\mathrm{prior}})^T G^T] \\ &= C_M G^T. \end{aligned} \tag{C.7}$$

The expression for $|C|$ becomes

$$\begin{aligned} |C| &= \left| \begin{bmatrix} C_M & C_{MD_{obs}} \\ C_{D_{obs}M} & C_{D_{obs}} \end{bmatrix} \right| \\ &= \left| \begin{bmatrix} C_M & C_M G^T \\ GC_M & C_D + GC_M G^T \end{bmatrix} \right|. \end{aligned} \tag{C.8}$$

Using the formula for the determinant of a block matrix,

$$\left| \begin{bmatrix} A & B \\ C & D \end{bmatrix} \right| = |D||A - BD^{-1}C|), \tag{C.9}$$

we have

$$|C| = \left|C_D + GC_M G^T\right| \left|C_M - (C_M G^T)(C_D + GC_M G^T)^{-1}(GC_M)\right|. \tag{C.10}$$

For the linear inverse problem, Oliver et al. (2008) show that the posterior covariance of m after conditioned to a realization d_{obs} of D_{obs} is always

$$C_{MAP} = C_M - C_M G^T (C_D + GC_M G^T)^{-1} GC_M, \tag{C.11}$$

regardless of the true value of m and the realization d_{obs}.

Therefore,

$$|C| = \left|C_D + GC_M G^T\right| |C_{MAP}|. \tag{C.12}$$

Using the above result, the expression of $I(m, D_{obs})$ in Equation C.3 becomes

$$\begin{aligned} I(m, D_{obs}) &= \frac{1}{2} \log \left(\frac{|C_M||GC_M G^T + C_D|}{|C_D + GC_M G^T|\,|C_{MAP}|} \right) \\ &= \frac{1}{2} \log \left(\frac{|C_M|}{|C_{MAP}|} \right). \end{aligned} \tag{C.13}$$

The expression for $I(J, D_{obs})$ can be derived in a similar manner. If J and D_{obs} are both Gaussian, the mutual information between J and D_{obs} can be written as:

$$I(J, D_{obs}) = \frac{1}{2} \log \left(\frac{|C_J||C_{D_{obs}}|}{|C|} \right), \tag{C.14}$$

where

$$C = \begin{bmatrix} C_J & C_{JD_{obs}} \\ C_{D_{obs}J} & C_{D_{obs}} \end{bmatrix}. \tag{C.15}$$

Because we assume that $J = G_J m$, the covariance matrix C_J is given by

$$C_J = G_J C_M G_J^T. \tag{C.16}$$

The expression for $C_{JD_{obs}}$ can be derived as follows:

$$\begin{aligned} C_{JD_{obs}} &= \mathrm{cov}(J, D_{obs}) \\ &= \mathrm{cov}(J, Gm) + \mathrm{cov}(J, \epsilon_d) \\ &= \mathrm{cov}(J, Gm) + 0 \\ &= E[(J - E[J])(Gm - E[Gm])^T] \\ &= E[(G_J m - G_J m_{\text{prior}})(Gm - Gm_{\text{prior}})^T] \\ &= E[G_J(m - m_{\text{prior}})(m - m_{\text{prior}})^T G^T] \\ &= G_J C_M G^T. \end{aligned} \tag{C.17}$$

The expression for $|C|$ becomes:

$$\begin{aligned} |C| &= \left| \begin{bmatrix} C_J & C_{JD_{obs}} \\ C_{D_{obs}J} & C_{D_{obs}} \end{bmatrix} \right| \\ &= \left| \begin{bmatrix} G_J C_M G_J^T & G_J C_M G^T \\ G C_M G_J^T & C_D + G C_M G^T \end{bmatrix} \right|. \end{aligned} \tag{C.18}$$

Again, using the formula for the determinant of a block matrix, we have

$$\begin{aligned} |C| &= \left|C_D + G C_M G^T\right| \left|G_J C_M G_J^T - (G_J C_M G^T)(C_D + G C_M G^T)^{-1}(G C_M G_J^T)\right| \\ &= \left|C_D + G C_M G^T\right| \left|G_J \left(C_M - C_M G^T (C_D + G C_M G^T)^{-1} G C_M) G_J^T\right)\right| \\ &= \left|C_D + G C_M G^T\right| \left|G_J C_{MAP} G_J^T\right|. \end{aligned} \tag{C.19}$$

Using the above result, the expression for $I(J, D_{obs})$ in Equation C.14 becomes

$$\begin{aligned} I(J, D_{obs}) &= \frac{1}{2} \log \left(\frac{|C_J||G C_M G^T + C_D|}{|C_D + G C_M G^T| \, |G_J C_{MAP} G_J^T|} \right) \\ &= \frac{1}{2} \log \left(\frac{|C_J|}{|G_J C_{MAP} G_J^T|} \right). \end{aligned} \tag{C.20}$$

The denominator inside the logarithm is the expression for $|C_J^u|$ (similar to Equa-

tion C.16) but for the posterior model instead of the prior model), thus

$$I(J, D_{obs}) = \frac{1}{2} \log \left(\frac{|C_J|}{|C_J^u|} \right). \tag{C.21}$$

We have completed our derivations.

C.3 Average of Means of J

The average (expectation) of the means of J over all conditioned posterior distributions is defined as

$$\overline{\mu_J^u} = E(E(J|d_{obs})) \tag{C.22}$$

where the superscript u stands for updated. The inner expectation sign is with respect to J for a particular value d_{obs}. The outer expectation sign is with respect to D_{obs}, which is a random vector, of which d_{obs} is a realization. For continuous random vectors, the definition of $\overline{\mu_J^u}$ is

$$\begin{aligned} \overline{\mu_J^u} &= \int E(J|d_{obs}) f(d_{obs}) dd_{obs} \\ &= \int \left(\int J f(J|d_{obs}) dJ \right) f(d_{obs}) dd_{obs} \end{aligned} \tag{C.23}$$

Using Bayes theorem, $f(J|d_{obs}) = \dfrac{f(d_{obs}|J) f(J)}{f(d_{obs})}$, we have:

$$\begin{aligned} \overline{\mu_J^u} &= \int \left(\int J \frac{f(d_{obs}|J) f(J)}{f(d_{obs})} dJ \right) f(d_{obs}) dd_{obs} \\ &= \int \int J f(d_{obs}|J) f(J) dJ dd_{obs} \\ &= \int \left(\int f(d_{obs}|J) dd_{obs} \right) J f(J) dJ \\ &= \int (1) J f(J) dJ \\ &= \mu_{J_{prior}} \end{aligned} \tag{C.24}$$

This completes our proof that the average of the expectation of J over all plausible posterior distributions is equal to the prior mean.

C.4 Correction Factor for Estimating Standard Deviation and P90 - P10

In this section, we evaluate the relationship between the standard deviation calculated using our proposed method and the value calculated using the rigorous history matching method. To make everything simple, we assume all the plausible posterior pdf's of J are Gaussian. Because for Gaussian distribution, P90 - P10 is directly proportional to standard deviation, we will assume our result can be extended to P90 - P10 as well.

Recall the two approaches to quantify uncertainty reduction are as follows:

1. Rigorous approach:

   - History matching the prior models with N_t plausible vectors of observed data to obtain N_t posterior pdf's.
   - Calculate the standard deviation of each posterior pdf.
   - Calculate the arithmetic average $\bar{\sigma}$ of N_t standard deviations.

2. Information theory approach:

   - Calculate the prior entropy of J, $h(J)$.
   - Calculate the mutual information between J and D_{obs}, $I(J, D_{obs})$.
   - Calculate the conditional entropy of J given D_{obs} is known, $h(J|D_{obs})$. This should be equal to the average of the entropies of the posterior pdf's obtained with the rigorous approach.
   - Construct a pdf with entropy equal to $h(J|D_{obs})$ and with a reasonable shape.
   - Calculate the standard deviation σ^* of the pdf constructed in the previous step. We wish to use this σ^* as an estimate for the average standard deviation $\bar{\sigma}$.

Because all the pdf's obtained from history matching are assumed to be Gaussian, for the information theory approach, it is reasonable to construct a Gaussian distribution with the estimated conditional entropy. The problem is, even for this simplified situation, there is a difference between the value of σ^* and $\bar{\sigma}$. We will explore why this happens.

Suppose $f_\Sigma(\sigma)$ is the pdf of the standard deviation of all posterior pdf's. In approach 1, the average standard deviation can be calculated as

$$\bar{\sigma} = \int_{-\infty}^{\infty} \sigma f_\Sigma(\sigma) d\sigma. \tag{C.25}$$

In approach 2, because posterior pdf's are all Gaussian, the entropy of each posterior pdf's is calculated as

$$h(J|d_{obs,i}) = \frac{1}{2}\ln(2\pi e\sigma_i^2), \tag{C.26}$$

where $d_{obs,i}$ is the i$^{\text{th}}$ realization of the random vector D_{obs}

The conditional entropy $h(J|D_{obs})$, which is the average of the entropies of all posterior pdf's, can be calculated as:

$$\begin{aligned} h(J|D_{obs}) &= \frac{1}{N_t}\sum_{i=1}^{N_t} h(J|d_{obs,i}) \\ &= \frac{1}{N_t}\sum_{i=1}^{N_t} \frac{1}{2}\ln(2\pi e\sigma_i^2) \\ &\approx \int_{-\infty}^{\infty} \frac{1}{2}\ln(2\pi e\sigma^2) f_\Sigma(\sigma) d\sigma. \end{aligned} \tag{C.27}$$

The standard deviation of the reconstructed pdf at the end of method 2 is:

$$\begin{aligned} \sigma^* &= \sqrt{\frac{e^{2h(J|D_{obs})}}{2\pi e}} \\ &\approx \sqrt{\frac{e^{2\int_{-\infty}^{\infty}\frac{1}{2}\ln(2\pi e\sigma^2) f_\Sigma(\sigma)d\sigma}}{2\pi e}}. \end{aligned} \tag{C.28}$$

At this point, there is no way to argue that $\sigma^* \approx \bar{\sigma}$ because the equations representing them are totally different. The question is how different they can be. This can be quantified by taking the ratio of $\bar{\sigma}$ to σ^* to obtain

$$\frac{\bar{\sigma}}{\sigma^*} = \frac{\int_{-\infty}^{\infty} \sigma f_\Sigma(\sigma) d\sigma}{\sqrt{\frac{\exp(2\int_{-\infty}^{\infty}\frac{1}{2}\ln(2\pi e\sigma^2) f_\Sigma(\sigma)d\sigma)}{2\pi e}}}. \tag{C.29}$$

The ratio of $\bar{\sigma}$ to σ^* is a function of σ alone but to evaluate it we need an explicit expression for $f_\Sigma(\sigma)$. We will consider two cases for $f_\Sigma(\sigma)$, a uniform distribution and a Gaussian distribution, although we have no way to show that either expression is correct.

a. $f_\Sigma(\sigma)$ is a uniform distribution.

Suppose $f_\Sigma(\sigma)$ is a uniform distribution between σ_{min} and σ_{max}, so that the expression for $f_\Sigma(\sigma)$ can be written as

$$f_\Sigma(\sigma) = \frac{1}{\sigma_{max} - \sigma_{min}}, \tag{C.30}$$

and $\bar{\sigma}$ becomes

$$\bar{\sigma} = \int_{\sigma_{min}}^{\sigma_{max}} \sigma \frac{1}{\sigma_{max} - \sigma_{min}} d\sigma = \frac{\sigma_{max} + \sigma_{min}}{2}. \tag{C.31}$$

$h(J|D_{obs})$ becomes

$$\begin{aligned} h(J|D_{obs}) &= \int_{\sigma_{min}}^{\sigma_{max}} \frac{1}{2} \ln(2\pi e\sigma^2) \frac{1}{\sigma_{max} - \sigma_{min}} d\sigma \\ &= \frac{\sigma_{min} - \sigma_{max} + \sigma_{max} \ln(2\pi\sigma_{max}^2) - \sigma_{min} \ln(2\pi\sigma_{min}^2)}{2(\sigma_{max} - \sigma_{min})}. \end{aligned} \tag{C.32}$$

The ratio of $\bar{\sigma}$ to σ^* can be written as

$$\frac{\bar{\sigma}}{\sigma^*} = \frac{\frac{\sigma_{max}+\sigma_{min}}{2}}{\sqrt{\frac{\exp\left(\frac{\sigma_{min} - \sigma_{max} + \sigma_{max} \ln(2\pi\sigma_{max}^2) - \sigma_{min} \ln(2\pi\sigma_{min}^2)}{\sigma_{max} - \sigma_{min}}\right)}{2\pi e}}}. \tag{C.33}$$

It can be shown that the above expression is a function of $\frac{\sigma_{min}}{\sigma_{max}}$ only. It can also be shown that $\frac{\bar{\sigma}}{\sigma^*}$ achieves maximum value of approximately 1.359 when $\frac{\sigma_{min}}{\sigma_{max}} \to 0$ and minimum value of 1 when $\frac{\sigma_{min}}{\sigma_{max}} = 1$.

b. $f_\Sigma(\sigma)$ is a Gaussian distribution.

Suppose $f_\Sigma(\sigma)$ is a Gaussian distribution with mean $\bar{\sigma}$ and standard deviation s, the

expression for $f_\Sigma(\sigma)$ can be written as

$$f_\Sigma(\sigma) = \frac{1}{\sqrt{2\pi}s} \exp\left(-\frac{(\sigma-\bar{\sigma})^2}{2s^2}\right). \tag{C.34}$$

Then $h(J|D_{obs})$ becomes

$$\begin{aligned} h(J|D_{obs}) &= \int_{-\infty}^{\infty} \frac{1}{2}\ln(2\pi e\sigma^2)\frac{1}{\sqrt{2\pi}s}\exp\left(-\frac{(\sigma-\bar{\sigma})^2}{2s^2}\right)d\sigma \\ &= -\frac{1}{2}\left(-1+\gamma-\ln(\pi s^2)+{}_1F_1^{(1,0,0)}(0,\frac{1}{2},-\frac{\bar{\sigma}^2}{2s^2})\right), \end{aligned} \tag{C.35}$$

where γ is the Euler's constant ($\gamma \approx 0.577$), ${}_1F_1^{(1,0,0)}(a,b,z)$ is the first-order derivative of the Kummer confluent hypergeometric function ${}_1F_1(a,b,z)$ (Wolfram, 1999, Section 3.2) with respect to the first variable a.

The ratio of $\bar{\sigma}$ to σ^* can be written as:

$$\begin{aligned} \frac{\bar{\sigma}}{\sigma^*} &= \frac{\bar{\sigma}}{\sqrt{\frac{\exp\left(-\left(-1+\gamma-\ln(\pi s^2)+{}_1F_1^{(1,0,0)}(0,\frac{1}{2},-\frac{\bar{\sigma}^2}{2s^2})\right)\right)}{2\pi e}}} \\ &= \frac{\bar{\sigma}}{\sqrt{\frac{\exp\left(1-\gamma+\ln(\pi s^2)-{}_1F_1^{(1,0,0)}(0,\frac{1}{2},-\frac{\bar{\sigma}^2}{2s^2})\right)}{2\pi e}}}. \end{aligned} \tag{C.36}$$

Again it can be shown that the above expression is a function of the ratio $s/\bar{\sigma}$ only. Note that, because σ follows a Gaussian distribution with mean $\bar{\sigma}$ and standard deviation s, the practical minimum value of σ is $\bar{\sigma} - 2s$ (assuming σ can spread up to 2 standard deviations away from $\bar{\sigma}$). Also note that, because σ represents the standard deviation of a plausible posterior pdf of J, it has to be greater that 0, which mean

$$\bar{\sigma} - 2s > 0, \tag{C.37}$$

or

$$\frac{s}{\bar{\sigma}} < 0.5. \tag{C.38}$$

Figure C.1 shows the plot of $\bar{\sigma}/\sigma^*$ as a function of $s/\bar{\sigma}$. It can be seen from the figure that $\bar{\sigma}/\sigma^*$ is an increasing function of $s/\bar{\sigma}$ on the interval [0, 0.5], with $\bar{\sigma}/\sigma^* = 1$ at $s/\bar{\sigma} = 0$ and $\bar{\sigma}/\sigma^* = 1.189$ at $s/\bar{\sigma} = 0.5$

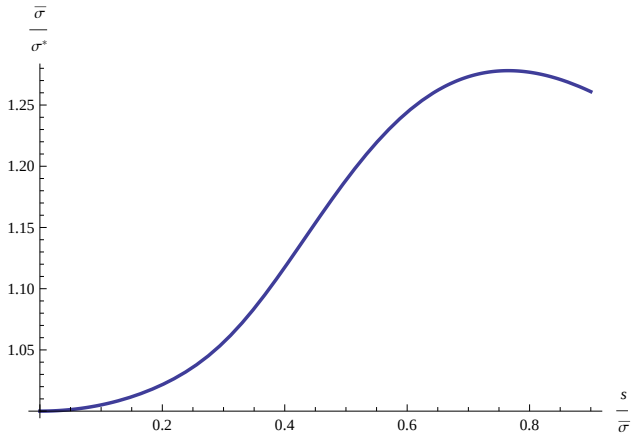


Figure C.1: The plot of $\bar{\sigma}/\sigma^*$ as a function of $s/\bar{\sigma}$

The trends of $\bar{\sigma}/\sigma^*$ in both the uniform and the Gaussian cases above are consistent with our expectation. We expect the ratio $\bar{\sigma}/\sigma^*$ to approach 1 when all plausible posterior pdf's are the same, i.e. have the same standard deviation. This is intuitive because when all posterior pdf's are identical, there is no difference whether we calculate the average standard deviation directly or calculate the standard deviation of a reconstructed pdf. As the standard deviations of the posterior pdf's become more different, we expect the ratio $\bar{\sigma}/\sigma^*$ to deviate from 1. The development above does not only confirm this expectation that but also shows that the ratio $\bar{\sigma}/\sigma^*$ is always greater than 1.

Using the results above, it is reasonable to say that we should apply a correction factor from 1 to 1.359 to bring our estimate σ^* closer to $\bar{\sigma}$. The question now is how much correction we should apply. Recall from the previous paragraph, when the differences between the standard deviations of the plausible posterior pdf's are big, the ratio $\bar{\sigma}/\sigma^*$ is higher than when the differences are small, so the correction factor should be based on differences between plausible posterior pdf's. Note that when the mutual information between observed data and the reservoir variable J is small, each plausible posterior pdf will not change much from the prior pdf and therefore the differences between the plausible posterior pdf are small.

On the other hand, when the mutual information is high, each plausible posterior pdf will deviate significantly from the prior pdf and we expect their differences are big. Based on these arguments, we use the following simple rule to determine the correction factor:

- When mutual information is zero, use correction factor equal to 1.0
- When mutual information is equal or greater than 2.0, use correction factor equal to 1.359
- When mutual information is greater than 0 and smaller than 2.0, use linear interpolation to determine the correction factor, i.e.

$$\text{correction factor} = 1.0 + \frac{1.359 - 1.0}{2.0}(\text{mutual information}) \tag{C.39}$$

The limit of 2.0 is based on our experience in working with mutual information. We have observed from the computational experiments done to date that when the mutual information is greater than or equal to 2.0, there are enough differences in plausible posterior pdf's so that the maximum value of the correction factor should be used.

We hypothesize that the relationship developed above approximately applies when the posterior distributions are not Gaussian, that is, as long as we reconstruct the pdf in the information theory approach based on some reasonable assumption about the shape of the posterior pdf's, we should still use a value between 1.0 and 1.359 as the correction factor to predict the average of the standard deviations of the posterior pdf's. We also hypothesize that the above relationship also applies when we want to estimate the difference between P90 and P10. It seems plausible that, for non-Gaussian distributions, this correction factor approach will hold better when applied to estimate P90 - P10 than when applied to estimate the standard deviation because P90 - P10 is not as heavily affected by the skewness of the distribution as the standard deviation is.

C.5 Principal Component Analysis (PCA)

Principal component analysis (PCA) is a mathematical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of uncorrelated variables called principal components. The number of principal components is less than or equal to the number of original variables. This transformation is defined in such a way that the first principal component has as high a variance as possible, i.e, accounts for as much of the variability in the data as possible, and each succeeding component in turn has the highest variance possible under the constraint that it be orthogonal to (uncorrelated with) the preceding components. PCA is very sensitive to the relative scaling of the original variables, i.e., variables with big values will dominate variables with small values. Thorough understanding of the method can be achieved by reading in Jolliffe (2002). In this appendix, we only present the procedure to perform PCA using SVD for a set of data.

Let X be a $N_m \times N_e$ matrix of data, where each row of X represents one random variable and consists of N_e observations of that random variable. There are a total of N_m variables. Now assume that the data in each row are centered, i.e., the mean of all values in each row is 0. If it is not zero, the mean of the observations of each variable should be subtracted from the data set for that variable. There is also a possibility that we can divide the values in each row by their variance or standard deviation but, since this changes the relative magnitudes of the variables, it is uncertain if this is a good thing to do. In our current implementation of PCA, we do not divide the data in each rows by their variance or standard deviation. We decompose X using SVD

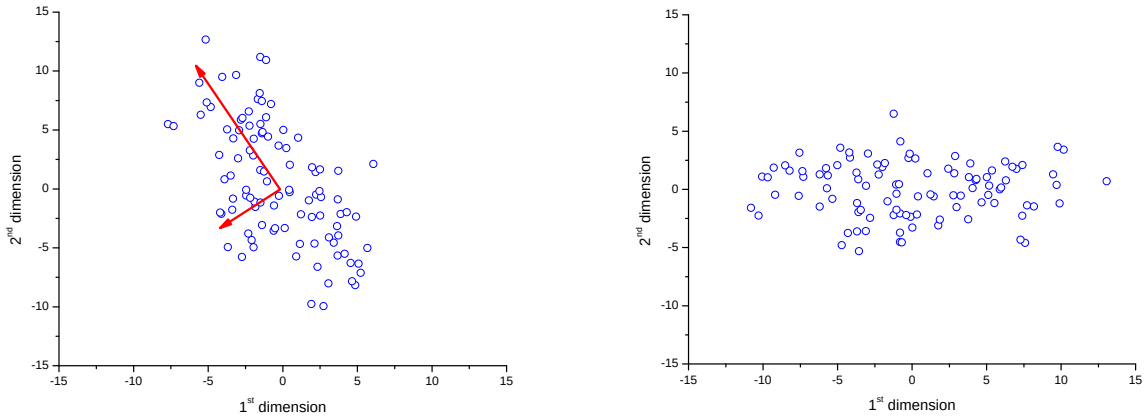
$$X = U\Lambda V^T, \tag{C.40}$$

where U is an $N_m \times N_m$ matrix, Λ is an $N_m \times N_e$ matrix and V^T is an $N_e \times N_e$ matrix The PCA transformation that preserves dimensionality (same number of variables as the original

data set) is given as

$$\begin{aligned} Y &= U^T X \\ &= U^T U \Lambda V^T \\ &= \Lambda V^T. \end{aligned} \tag{C.41}$$

The resulting data set is essentially the original data set except that all the points have been rotated around the origin so that the first few principal directions contain as much variance as possible (Figure C.2).



(a) Original data set with 2 variables (2 dimensions)

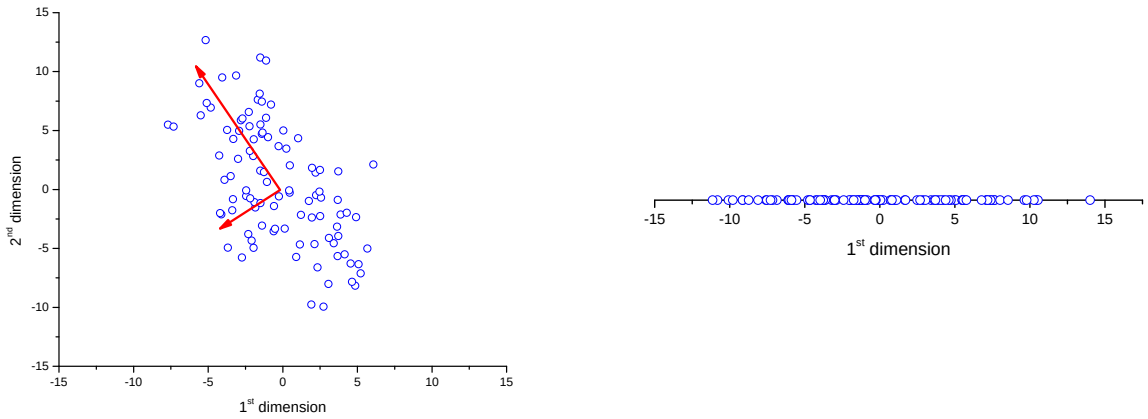
(b) Same data set after applying PCA but keeping same number of dimensions

Figure C.2: Illustration of PCA being applied to a 2-dimensional data set without reduction of dimensions

If dimension reduction is desired, we can project X to the reduced space by using the first L singular vectors in U, i.e.,

$$\begin{aligned} Y &= U_L^T X \\ &= U_L^T U \Lambda V^T \\ &= \Lambda_L V^T, \end{aligned} \tag{C.42}$$

where U_L is an $N_m \times N_L$ matrix whose columns corresponds to the first L columns of U, Λ_L is an $N_L \times N_m$ rectangular diagonal matrix whose diagonal elements correspond to the first L diagonal elements of Λ (the first L singular values). When we use this equation to reduce the number of dimensions of the data set in Figure C.2 to 1, we would obtain the new data set shown in Figure C.3.



(a) Original data set with 2 variables (2 dimensions)

(b) Same data set after applying PCA and reducing number of dimensions

Figure C.3: Illustration of PCA being applied to a 2-dimensional data set with reduction of dimensions

The procedure above is called linear PCA. It can only discover linear correlations between variables in a data set. A more advanced method to discover non-linear correlations between variables exists and is called kernel PCA. In this research, when necessary, we apply linear PCA to reduce the dimension of the matrix of plausible realizations of the observed data and then calculate the mutual information between the dimension-reduced matrix of plausible observed data and the reservoir variable J. All history matching works are still performed using the original plausible observed data before PCA.