THE UNIVERSITY OF TULSA

THE GRADUATE SCHOOL

ENSEMBLE KALMAN FILTER METHOD FOR GAUSSIAN AND NON-GAUSSIAN PRIORS

by
Yong Zhao

A dissertation 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

2008

THE UNIVERSITY OF TULSA

THE GRADUATE SCHOOL

ENSEMBLE KALMAN FILTER METHOD FOR GAUSSIAN AND NON-GAUSSIAN PRIORS

by
Yong Zhao

A DISSERTATION

APPROVED FOR THE DISCIPLINE OF

PETROLEUM ENGINEERING

By Dissertation Committee

______________________________ , Chairperson
Albert C. Reynolds

____
Gaoming Li

____
Richard A. Redner

____
Jagan Mahadevan

COPYRIGHT STATEMENT

ABSTRACT

Yong Zhao (Doctor of Philosophy in Petroleum Engineering)

Ensemble Kalman Filter Method for Gaussian and Non-Gaussian Priors

Directed by Albert C. Reynolds

291 pp. Chapter 9

(347 words)

The objective of this work is to find an efficient and robust way to implement the ensemble Kalman filter (EnKF) to assimilate production and seismic data for both Gaussian and truncated pluri-Gaussian geological models.

Truncated pluri-Gaussian models have proven to be useful for generating realistic geological models of facies distributions. In this work, we will specifically test a new idea for modeling of a channelized reservoir (fluvial system with two facies, channel facies and non-channel facies). EnKF is used to adjust the facies distribution (e.g. channel and non-channel facies) as well as the porosity and permeability of each facies to match production data and seismic data. For two and three-dimensional pluri-Gaussian models, we present a new procedure to ensure that facies observations at wells are honored at each data assimilation step.

As the erroneous saturation distribution obtained with EnKF may result from nonlinearity or the failure of the assumption that the ensemble of predictions is approximately Gaussian, we investigate the application of a global and local normal score transform to transform water saturation to a Gaussian variables before applying the EnKF analysis step. We also apply an iterative EnKF scheme to obtain more plausible saturations distributions. To improve water cut data matches, we consider matching breakthrough times directly before matching watercut data.

The integration of seismic data poses problems because of the large number of data that are assimilated. With a global assimilation procedure based on subspace projection, filter divergence becomes severe. On the other hand, our implementation of a local updating method to reduce filter divergence results in an unrealistic rough facies map. We introduce a projection method to obtain a more realistic map of the facies distribution, which retains the inherent smoothness of the underlying geological model.

The characterization of measurement error is important if one uses a Bayesian approach to condition reservoir models to dynamic data. We use Savitzky-Golay smoother and wavelet smoother to estimate the measurement error in the production data, and use a modified EM (Expectation-Maximization) algorithm combined with a quadratic fitting to estimate the measurement error in the 4-D seismic data.

ACKNOWLEDGEMENTS

I would like to express my sincere appreciation to Dr. Albert C. Reynolds, Jr., Professor of Petroleum Engineering and Mathematical Science of the University of Tulsa, and Dr. Gaoming Li, Assistant Professor of Petroleum Engineering of the University of Tulsa, for their support, guidance, insights, and encouragement during my five years' Ph.D. study in the University of Tulsa. I specially thank Dr. Richard A. Redner, Professor of Mathematical Science of the University of Tulsa and Dr. Jagan Mahadevan, Assistant Professor of Petroleum Engineering of the University of Tulsa for serving on my committee and for their helpful suggestions and comments. I would like to express my thanks and appreciations to my fellow brothers Dr. Guohua Gao and Dr. Shi Chen for their generous helps on the many critical concepts on the automatic history matching problem and petroleum engineering courses when I was starting my PH.D. study. I greatly appreciate the helps from Mrs. Judy Teal and Mrs. Loreta M. Watkins during my five years study in the University of Tulsa. I specially thank Dr. Yudou Wang and all students in TUPREP group for their generous helps and encouragements.

I would like to thank all the faculty members in the Petroleum Engineering department of the University of Tulsa who offer all the knowledge and education for me to pass the PH.D. qualifying exam and grow up as a petroleum engineer. I gratefully acknowledge financial support from the Department of Energy (Award No. DE-FC26-04NT15517) and the member companies of TUPREP.

I should thank my wife Mrs. Jing Li for her endless love, patience and support for the last five years. This thesis is dedicated to my parents and my parents in law and all other members in my family.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Page

CHAPTER 1

INTRODUCTION

Assisted or automatic history matching methods in a Bayesian setting require a sound geological model with clear definition of a prior probability density function (PDF), a reasonable estimation of measurement error and an efficient algorithm to modify geological models so that they match the data. This study will use a truncated pluri-Gaussian geological model to model facies distribution as well as Gaussian random fields to model porosity and permeability distributions. The ensemble Kalman filter (EnKF) and an iterative scheme (EnRML) are used to condition the models to both production data and seismic data. Savitzky-Golay smoothing, the wavelet transform and the EM algorithm are used to estimate measurement errors in production and seismic data.

Geologists are often able to provide a training image of the reservoir facies distributions. Conditioning these to production data by history matching is difficult because the sharp change in rock properties across facies boundaries causes a discontinuity in the derivative of production data to grid block properties (sensitivities). This discontinuity in derivatives can be avoided by computing the exact location of facies boundaries within a simulator grid block ([84]) or by introducing artificial smooth changes in facies properties across facies boundaries ([71, 53]), but these procedures are generally not feasible for complex three-dimensional geological models. In this study, we use a pluri-Gaussian geological model to describe the facies distribution, and use EnKF to assimilate production and seismic data. We also consider the case of Gaussian priors. We present new computationally efficient procedures for integrating both production data and seismic data into facies maps to obtain a more accurate reservoir description and evaluation of uncertainties and to provide better guidance for reservoir development and production optimization.

EnKF is an efficient data assimilation scheme. After data assimilation at a time step, EnKF continues with prediction by restarting the simulator with the analyzed ensemble and does not require simulating from time zero with the updated model. This assumes that the updated state vector (including primary variables from the simulator) is identical with those we obtain if we simulate from time zero with updated model parameters. This makes the method very efficient, but this property can only be formally established when the dynamical equations are linear, modeling errors are negligible and there is a linear relation between data and the state vector [77]. However, the reservoir simulator equations are non-linear, which means primary reservoir simulator variables in the updated state vector may be non-physical. This error is most obvious in phase saturations, which accumulates over assimilation time steps and affects the material balances and predictions of future performance. In pluri-Gaussian cases, we have found that high water saturation regions are not consistent with the channel facies region because of an inconsistent adjustment between model parameter and saturations during analysis. Because of the non-linearity of the system, the saturations in the updated ensemble can become non-Gaussian. For example, in the water flooding case, the histogram of water saturation near the water front may have a high value mode and a low value mode, so the Gaussian assumption is not valid.

Assimilating seismic data or inverted seismic data with EnKF is a special problem because EnKF has limited freedom (typically 100) to match a large number of data (in every gridblock), and technically, inverting a large matrix ($N_d \times N_d$, where N_d is the number of data) is not feasible. In this study, we will apply and compare two approaches to this problem, i.e. global analysis and local analysis of seismic data.

1.1 Objectives

The objective of this work is to find an efficient and robust way to implement the ensemble Kalman filter (EnKF) to assimilate production and seismic data for both Gaussian and truncated pluri-Gaussian geological models. For the pluri-Gaussian geological model, both the facies distributions as well as the porosity and permeability of each facies are model parameters, and within the Bayesian framework we use, are treated as random variables. Although a channelized reservoir (fluvial system with channel facies and non-channel facies) can only be described very roughly with a pluri-Gaussian model, the model is sufficiently reasonable to investigate whether EnKF can correctly characterize the connectivity between injection and production wells. We consider different ways to improve the updates of saturations. In particular, we apply a normal score transform to make the transformed saturations follow Gaussian distribution [8, 35], and consider an iterative scheme to alleviate non-linearity effects [49, 34]. We include methods to estimate the covariance matrix of the measurement error for different types of production data and inverted seismic data.

1.2 Literature Review and Discussion

1.2.1 Ensemble Kalman Filter and Pluri-Gaussian Geological Models

The ensemble Kalman filter provides an efficient method to assimilate the dynamic data into reservoir models and its application to reservoir history matching problems is an intense focus of several research groups. The theory and algorithms for implementing various procedures for implementing EnKF have been presented in many publications by Evensen [22, 23, 24, 27] and other researchers working in atmospheric physics and the most important results can be found the recent book of Evensen [26]. The pioneering work of Loc'h and Galli [55] on pluri-Gaussian provides a way to model the geological facies of a reservoir. Although, pluri-Gaussian models are able to generate a wide range of geological descriptions [5], our objective here is to be able to integrate production data and seismic data with geological facies models, particularly those for fluvial systems, to obtain a more accurate map of the most productive facies for reservoir management.

The application of the ensemble Kalman filter (EnKF) [22] to forecast the state of a system has been widely explored in the ocean dynamics and atmospheric physics literatures; see [23], [40] [59, 24, 41] and the exhaustive citation list of Evensen [27]. Since its introduction into the petroleum engineering literature [62] for sequential or real-time automatic history matching, it has been applied to a number of synthetic problems for estimation or stochastic simulation of reservoir model parameters and reservoir simulator primary variables [61, 33, 31, 56, 75, 78, 79, 68, 81, 77] as well as to field cases [39, 28, 10].

In these three field cases, EnKF provided a better match of data than was obtained from a traditional history matching approach. Nonetheless, the EnKF method can encounter problems in matching water cut data [28], and it has been shown that for toy multi-modal problems, the method does not sample correctly and that the estimated mean may lie in a set which has zero probability [80]. For such problems and for highly nonlinear problems, there is evidence that an improved data match, an improved sampling of the conditional pdf and an improved estimate of future production can be obtaining by applying an iterative EnKF method [68, 49]. The iterative methods of these last two papers, however, require an adjoint method to compute the gradient of a least-squares objective function of data mismatch terms. For the truncated pluri-Gaussian model considered here, however, the derivative of this objective function does not exist along truncation lines or curves because as a point (pair of Gaussian random variables for a simulation gridblock) moves across a truncation line, the facies occupying the gridblock changes so that there is a discontinuous change in the objective function. This discontinuity in the gradient can be avoided if the problem is sufficiently simple so that the location of facies boundaries within a reservoir simulator gridblock can be computed [83] or, for the more complicated cases considered here, by using a transition region to introduce artificial smooth changes in facies properties across facies boundaries [71, 54]. For two dimensional two phases problem, Liu and Oliver [54] successfully implemented pluri-Gaussian models in dynamic data assimilation using both the ensemble Kalman filter (EnKF) and a gradient based method. In the pluri-Gaussian model, three intersecting lines were used to define the truncations for three facies. Gaussian random fields were adjusted to match the dynamic data assuming the porosity and permeability of each facies are known. In the EnKF method, the hard data (observed facies at well locations) were appended as the observed data in each data assimilation step to guarantee the hard data are always satisfied. In gradient based method similar to Schaff et al. [71], Liu and Oliver [53] added a transition zone to the truncation map to make it possible to calculate the sensitivity of facies with respect to Gaussian random fields. In their implementation, the gradients were calculated using the adjoint method [50, 52], and the objective function is optimized using the LBFGS method [82]. However, for the two-dimensional two-phase flow problems considered, Liu and Oliver [54] found that EnKF was not only more efficient than the gradient based approach but also gave better data matches.

Before assimilating dynamic data using EnKF, we need to be able to generate initial ensemble of Gaussian random fields that honor the hard data (observed facies at well locations). Unfortunately, the distribution of Gaussian random field values at well locations is non-Gaussian. In this work, the pairs of data at well locations are randomly drawn from data banks of corresponding facies, which is generated from a set of unconditional realizations of Gaussian random fields, and the values at unsampled locations are generated from a sequential Gaussian simulation algorithm. The only assumption is that the wells are far enough apart so that the Gaussian random field values are uncorrelated with each other.

In this study, we specifically designed a truncation with two intersecting ellipses, which can generate channel facies and non-channel facies. Although the results obtained do not look like realistic channels, the procedure gives us a way to investigate whether we can characterize the connectivity of a channelized reservoir.

1.2.2 Matching Both Production Data and Seismic Data

As the size of the ensemble approaches infinity, the ensemble Kalman filter provides a correct sample of the conditional pdf if the prior state vector (reservoir model parameters plus simulation primary variables) is Gaussian and the relation between the state vector and predicted data is linear [27]. In the examples considered here, predicted production data will be generated with a reservoir simulator. For such problems, the saturation distribution in a reservoir simulation gridblock through which a front is moving is non-Gaussian. Moreover, when saturations are updated at an analysis step of EnKF, it is not uncommon to find analyzed (updated) saturations to be nonphysical, for example, greater than one or less than zero. In such cases, it is common to simply truncate these values to physically plausible values before predicting forward to the next data assimilation time with the simulator. For problems where variables are non-Gaussian, Bertino et al. [8] suggested replacing those variables with transformed variables obtained by applying a normal score transform before assimilating data. When the normal score transform is applied to predicted saturation values before data assimilation, one can ensure that the analyzed values will be within the specified physical limits. However, for a simple one-dimensional waterflooding problem, Gu and Oliver [35] obtained nonphysical oscillations in the saturation profile. We have experimented with a different construction of the normal score transform than the one used by Gu and Oliver [35], but the modification yields no improvement, and introduces an additional problem.

When assimilating production and seismic data to generate facies maps, the reservoir model variables are the pair of Gaussian random variables at each gridblock and, for some examples in this work, the porosity and permeability of each facies. Hard data, however, may include the observation of a particular facies at a well block. As the facies type is not one of the model variables, a method must be introduced to ensure that the values of the two Gaussian random fields at a well gridblock are consistent with observed facies at each data assimilation step. If, when assimilating production data with the EnKF analysis step, well observations of facies are not honored, Liu and Oliver [54] use EnKF with iteration to ensure observed facies data are honored. Here, we use a different approach, which avoids multiple iterations to honor well observations of facies. If the normal EnKF update violates the observed facies, we compute feasible values of the well block Gaussian random values that are "close" to the updated values and also are consistent with the observations of facies. Then we select pairs of these values of the Gaussian random variables, incorporate them as hard data into the data vector and redo the EnKF analysis step.

The assimilation of seismic data or inverted seismic data with EnKF poses special problems because the number of data is large, for example, one may have values of acoustic impedance and Poisson's ratio at every reservoir simulation gridblock. Because EnKF is typically applied with an ensemble size on the order of 100 and each updated (analyzed) model must be a linear combination of the initial ensemble of models [27], there are not sufficient degrees of freedom to match all seismic data. The degrees of freedom may decrease when data are assimilated due to a loss of rank in the updated covariance matrix when the covariance matrix for measurement error that appears in the Kalman gain is approximated using a finite sample [43, 25]. Moreover, even if a full rank version of the measurement error covariance matrix is used, the matrix that must be inverted in the data assimilation step is $N_d \times N_d$ where N_d is the number of data to be assimilated at a particular step. When assimilating seismic data, N_d is often so large that it is not feasible to solve an $N_d \times N_d$ matrix problem. One approach to deal with this difficulty is to apply a subspace EnKF inversion scheme [43, 25]. However, this approach may require a very large ensemble size to obtain reasonable results as shown by Skjervheim et al. [75] as well as by results presented later in this study. If the ensemble is not large, after assimilation of seismic data, all ensemble members may be very similar which suggests that uncertainty has been significantly underestimated. Another way to reduce the problem of insufficient degrees of freedom and avoid inverting a large matrix is to use local updating [59, 24, 64, 75]. As discussed later, in local updating, model parameters at a gridblock are updated using only data at nearby gridblocks. When Skjervheim et al. [75] used this approach to update values of gridblock permeabilities, porosities and clay ratios, they found that one could obtain non-smooth property fields. In this work, we introduce a projection method with local updating that eliminates the inherent lack of smoothness that can occur when using local updating. For the facies models considered in this study, this loss of smoothness is severely exacerbated.

As mentioned previously, for highly nonlinear or non-Gaussian problems, EnKF may not give a reasonable sample of the conditional pdf's we wish to sample, but results may be improved via iteration [68, 49]. The iteration methods in these papers require adjoint code for computing gradients of the sum of squared production data mismatch terms, and for the truncated Gaussian model, this gradient is discontinuous. Gu and Oliver [34] have presented an iterative method motivated by the maximum likelihood ensemble filter of Zupanski [86] which uses an approximation to an average sensitivity matrix. The method of Zupanski and Gu and Oliver both use a Taylor series approximation to find an "average" sensitivity matrix and thus avoid the direct approximation of gradients. Although they applied the method to only one true reservoir flow problem (a linear one-dimensional waterflooding problem) we show that a modification of their algorithm can sometimes improve results when assimilating production data.

1.2.3 Measurement Error Estimation

In order to characterize reservoir description, make performance predictions, track the movement of injected fluids, characterize the uncertainty in reservoir descriptions and performance predictions and optimize production, it is desirable to integrate all available data including static data (core, well logs) and dynamic production data (wellbore pressure, WOR, GOR or phase production rates) and seismic data. A popular approach to characterize uncertainty involves generating multiple reservoir models (realizations) using randomized maximum likelihood (RML) [63], [45], [69], and ensemble Kalman filter (EnKF) [22], [62], [60], [24]. An individual realization is constructed by minimizing an objective function (constructed by Baysian inversion theory), which is composed of a prior model mismatch term and a data mismatch term. The data mismatch term typically involves dynamic production and/or seismic data. The relative weights of different types of data in the objective function are determined by the covariance matrix of the measurement error. The balance among different types of data affects the suite of realizations obtained, so having a reasonable estimate of the measurement error is important. It is also true for ensemble Kalman filter method (EnKF), which assimilate data sequentially in time, because it is also based on Bayesian inversion theory. Estimation of the measurement error is required for both of these methods.

We wish to estimate the measurement error directly from the observed (measured) data which are the sum of the true data plus error. Our ability to do so rests on the underlying assumption that the measurement error for a particular data type (e.g. wellbore pressure versus time measured with a pressure gauge) has much higher frequency than the true underlying signal. Thus, we can smooth the data to estimate the true signal. Subtracting the estimated true signal from the measured data gives an estimate of the measurement error. Note this estimated measurement error can include components of processing errors, e.g., in 4D seismic data, provided the processing errors are of much higher frequency than the underlying true signal. Throughout, our focus is on the estimation of measurement error in 4D acoustic impedance change and Poisson's ratio change data, which are derived from time-lapse seismic surveys, and production WOR and GOR data.

Aanonsen et al. [1] applied a moving average with equal weights to estimate measurement error in both production and seismic data with some success. The moving average method can introduce bias in the estimation when the underlying true signal is not a linear function of time or space. The bias becomes worse when the true signal contains discontinuities, because the simple moving average smoothes out discontinuities and thus some of the true signal is contained in the estimated measurement error. Intuitively, to accurately estimate measurement error in this case, it is necessary to find these discontinuities or group the data into regions to exclude the discontinuities and implement smoothing algorithm within each region.

For 1-D measurement of production data, i.e., WOR, GOR, pressure etc., a Savitzki-Golay smoothing method and a wavelet transform method are used for smoothing. We will show that if outliers and discontinuities exist in the measurements, the smoothing can cause overestimation of measurement errors. We can not separate them correctly unless we recognize and remove their effects. We combined mid-point smoothing, the second derivative, and wavelet smoothing to remove outliers and identify discontinuities. Finally, Savitzky-Golay smoothing and wavelet transform smoothing were applied to corrected measurements free of outliers and discontinuities.

For 4-D seismic data, the identification of discontinuities is also important to avoid overestimation of measurement error by smoothing across these discontinuities. An expectation-maximization (EM) algorithm [37, 13, 67, 66, 57, 58, 47] will be applied to statistically separate measurements by assuming they are sampled from a Gaussian mixture model. However, this algorithm is traditionally used to group measurements by measured values, and we need to assume the number of groups are known. We will introduce a spatial constraint (consistent with Besag [9]) and design a grouping quality coefficient so that the EM algorithm can group seismic data. The idea of using a spatial constraint is that two data should be in the same group if the data are close in value and the distance between the locations of the measurements is relatively small for spatially measured data. The grouping quality coefficients designed in this study is used to find the approximate number of groups, which is an very important issue because we normally do not know how to partition the data into groups to avoid smoothing across discontinuities.

CHAPTER 2

GAUSSIAN AND PLURI-GAUSSIAN GEOLOGICAL MODELS

Geological models are used to represent heterogeneous property fields. Realizations from a stochastic geologic model (conditioned to hard data) represent the uncertainty in reservoir description prior to dynamic data assimilation. An initial ensemble of these realizations is the starting point of any sequential data assimilation method such as the ensemble Kalman Filter (EnKF), which is the focus of this research. In this chapter, the two types of geological models used in this study will be introduced: 1) a Gaussian geological model is most commonly used in history matching study, in which porosity and permeability fields are modeled as Gaussian random fields (GRFs) with specified variograms; 2) a pluri-Gaussian geological model, which uses two Gaussian random fields (GRFs) and a truncation scheme to generate facies distributions. Unlike the Gaussian geological models which incorporate a gradual spatial change, a pluri-Gaussian geological model can characterize reservoirs with clear boundaries between different rock types. In this study, two ellipses are used for truncation to model a two facies system consisting of a channel facies and a non-channel facies, and three truncation lines are used to model three facies in the reservoir. However, the general procedure presented can be applied to model a large number of facies. As examples, we consider a two-dimensional channelized reservoir and a three-layer three facies case. For each case, the production data are assimilated using EnKF to condition the initial set of realizations to production data. The PUNQS3 example is used as an synthetic case for a prior Gaussian geological model.

2.1 Gaussian Geological Model

In the Gaussian geological model considered here, porosity and log-permeability fields are modeled as Gaussian random fields. The true porosity and log-permeability fields are assumed to be a realization of random vectors that follow a Gaussian distribution, i.e., $\Phi = [\phi_1, \phi_2, \cdots, \phi_N]^T$ for porosity field, and $\ln k_h = [\ln k_{h,1}, \ln k_{h,2}, \cdots, \ln k_{h,N}]^T$ for horizontal log-permeability field, and $\ln k_z = [\ln k_{z,1}, \ln k_{z,2}, \cdots, \ln k_{z,N}]^T$ for vertical log-permeability field, where the subscript i refers to the ith reservoir simulation gridblock. These random variables are spatially correlated, and their probability density function of $M = (\Phi^T, \ln {k_h}^T, \ln {k_z}^T)^T$ are defined as Gaussian with prior mean vector $\overline{M} = (\overline{\Phi}^T, \overline{\ln K_h}^T, \overline{\ln K_z}^T)^T$ and prior covariance matrix C_M. Both $\overline{M}$ and C_M are given as prior information describing the geology of the reservoir. Normally, C_M is constructed from variograms and cross-variograms for the porosity field and the two log-permeability fields. In this study, a variogram is defined using the following information [14]: (1) variances of each model type, i.e., porosity, $\ln k_x$ and $\ln k_z$; (2) horizontal correlation direction; (3) long and short correlation distance (horizontal ranges); (4) vertical correlation range; and (5) type of the spatial variogram function. The most common types of variogram are exponential, spherical and Gaussian [14]. The definitions of these three variograms are normally the same except for the variances so that the cross-variograms between different model types, i.e. porosity, $\ln k_x$ and $\ln k_z$, can be constructed using the correlation coefficients. Therefore, with the three predefined variograms, the covariance of any two random variables in M are related to their spatial distances and correlation coefficient between porosity and log-permeability.

Fig. 2.1 are the spatial correlation coefficients (covariances divided by variances) for these three types of variograms as a function of normalized spatial distance (distance divided by correlation range). We can see that the correlation drops rapidly for the exponential variogram, and it drops slowly for a Gaussian variogram.

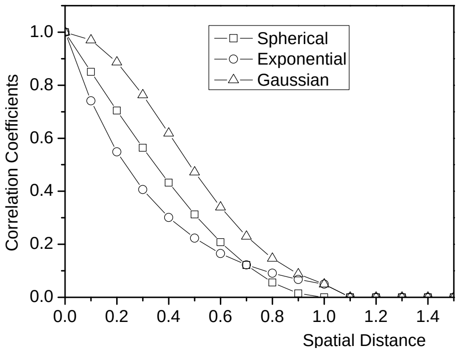
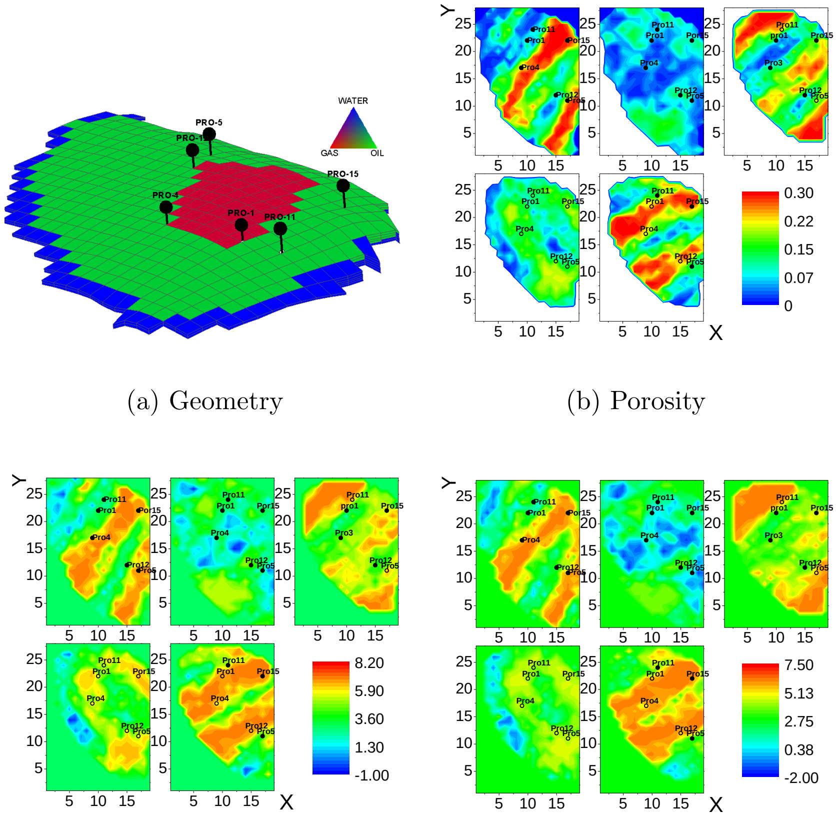


Figure 2.1: Three types of spatial of correlation coefficients.

PUNQS3 is a reservoir model with 5 layers, but these layers are uncorrelated and have different correlation directions and ranges. We assume each field in each layer follows spherical variogram. In this case, the uncertain model parameters include porosity (Φ) and log-permeability in horizontal and vertical directions ($\ln k_h$ and $\ln k_z$). In the next section, a Gaussian type of variogram will be used in a pluri-Gaussian model to generate facies distribution because, among the three choices, it gives the smoothest Gaussian random fields and the smoothest facies boundaries.

2.1.1 PUNQS3 Geological Model

The PUNQS3 reservoir was constructed based on a real field operated by Elf Exploration & Production. The detailed information of this reservoir can be found in references [7], [29], and [30]. Fig. 2.2(a) shows the reservoir is of dome-like shape, and bounded by a fault located at the east and south side of the reservoir. This figure also shows the initial saturation distribution. At the north and west sides, there is a strong acting aquifer shown in blue. A small gas cap is located at the top structure of the reservoir shown in red. Six wells are drilled in the oil column around the gas cap. The size of the reservoir grid is $19 \times 28 \times 5$ with horizontal grid size of 590.55 ft × 590.55 ft. The vertical grid sizes vary between 2.5 feet to 30 feet for different gridblocks, and are around 15 ft for most gridblocks. Fig. 2.2(b), 2.2(c) and 2.2(d) are porosity and log-permeabilities distributions in the five layers. In



(a) Geometry

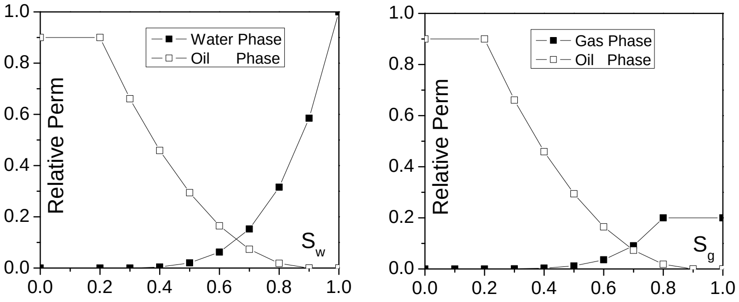
(b) Porosity

(c) $\ln k_h$

(d) $\ln k_z$

Figure 2.2: True model of PUNQS3: geometry, porosity, $\ln k_h$, $\ln k_z$.

each layer, the six circular symbols indicate locations of the six wells. Solid symbols indicate not perforated, and open circles indicate perforated. As the figures show, no wells are perforated in the first two layers, and all are perforated in the fourth layer. Only producer 5 and producer 11 are perforated in the third layer, and only producer 1 and producer 4 are perforated in the fifth layer. The truth model is a realization of prior geological model described in Table. 2.1, which is also used to generate initial realizations for ensemble Kalman Filter (EnKF). The range is in the number of gridblocks, and correlation angle is in degree of long correlation angle counterclock wise from x-direction. In this study, hard data are not used. Fig. 2.3 shows the relative permeability curves.



(a) Water-Oil

(b) Gas-Oil

Figure 2.3: PUNQS3 relative permeability curves.

Layers	Layer 1	Layer 2	Layer 3	Layer 4	Layer 5
Mean Φ	0.143	0.099	0.143	0.117	0.143
STD Φ	0.101	0.029	0.101	0.056	0.101
Mean $\ln k_h$	4.54	3.91	4.54	4.37	4.54
STD $\ln k_h$	2.49	0.93	2.49	1.35	2.49
Mean $\ln k_z$	3.55	2.76	3.55	3.04	3.55
STD $\ln k_z$	2.49	0.93	2.49	1.35	2.49
Long Range (GDBK)	19.44	4.167	33.33	8.33	20.83
Short Range (GDBK)	5.56	4.167	8.33	4.167	6.94
Corr. Angle (DEG)	60.0	60.0	45.00	-60.0	30.0
Φ, $\ln k_h$ Corr.	0.8	0.8	0.8	0.8	0.8
Φ, $\ln k_z$ Corr.	0.8	0.8	0.8	0.8	0.8

Table 2.1: PUNQS3 geological model

2.1.2 PUNQS3 Production History and Data

The production history of the 6 wells are identical, and are all bound on oil rate constraints (targets) as is shown in Fig. 2.4. Some observed data are available in the first 2936 days, and predictions to 6023 days are made after data assimilation. As a synthetic example, the observed data are generated by simulating the truth model and then adding noise. Figs. 2.5, 2.6 and 2.7 respectively show the producing bottomhole pressure, GOR and watercut respectively, in which the black lines are true predictions from simulator, and red symbols are observed data with noise. The standard deviation of measurement error is 50 psi in pressure data, and 5 percent of

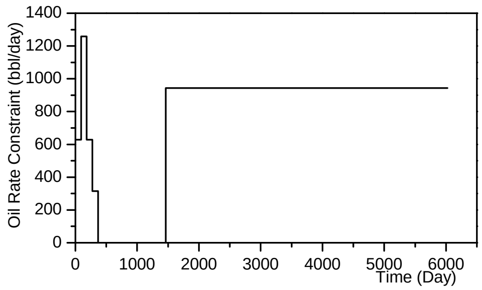
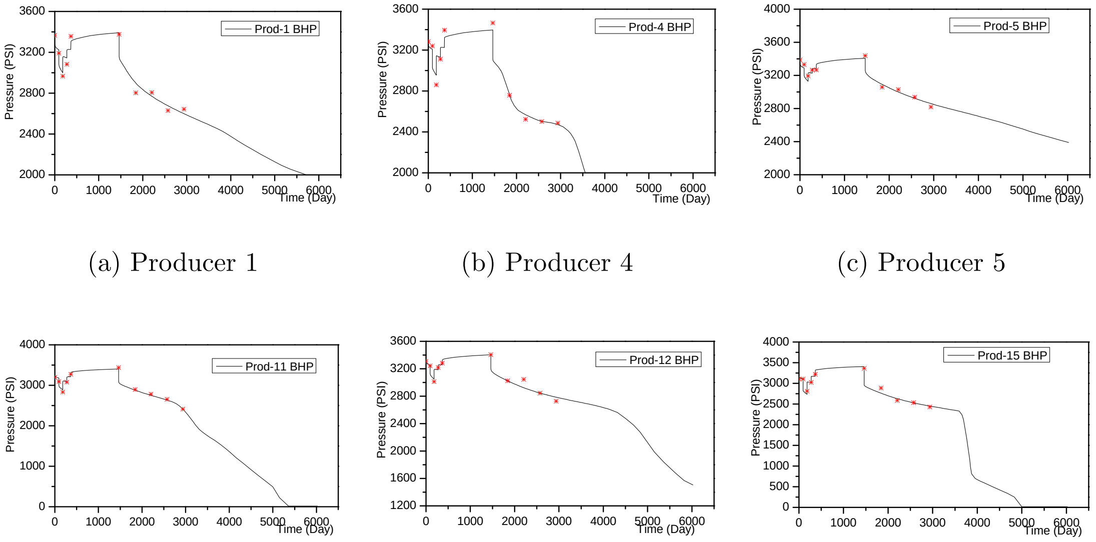


Figure 2.4: Production history of PUNQS3, for each well.



(a) Producer 1

(b) Producer 4

(c) Producer 5

(d) Producer 11

(e) Producer 12

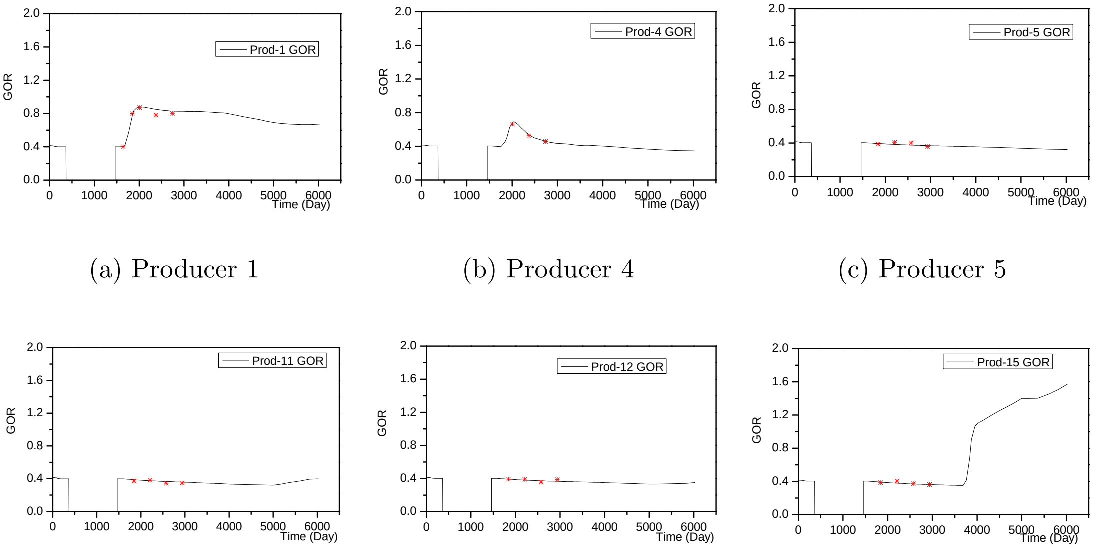
(f) Producer 15

Figure 2.5: Predictions and measurements of bottomhole pressure from the truth model (PUNQS3).

the truth in GOR and water cut data.

Acoustic impedance is implicitly a function of rock property, pressure and phase saturations. With the calculated pressures and saturations from the simulator, the acoustic impedance value for each gridblock is generated using the method used by Dong and Oliver [18], which is based on the rock physics by Gassmann [32] and Han [36]. In this model, acoustic impedance of a grid block is computed from Eq. 2.1.

$$Z = \rho V_p = \sqrt{\rho(K + \frac{4}{3}G)} = \sqrt{\rho K + \frac{4}{3}\rho^2 V_s^2}, \tag{2.1}$$



(a) Producer 1

(b) Producer 4

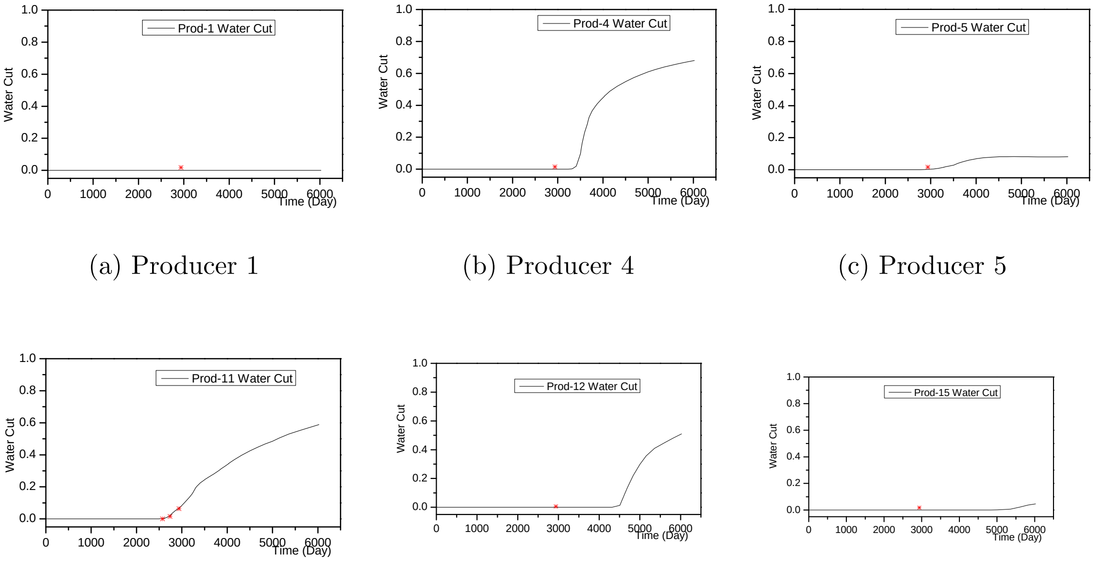
(c) Producer 5

(d) Producer 11

(e) Producer 12

(f) Producer 15

Figure 2.6: Predictions and measurements of producing GOR from the truth model (PUNQS3).



(a) Producer 1

(b) Producer 4

(c) Producer 5

(d) Producer 11

(e) Producer 12

(f) Producer 15

Figure 2.7: Predictions and measurements of water cut from the truth model (PUNQS3).

where Z is the acoustic impedance of a certain grid-block, V_p and V_s are P-wave and S-wave velocity respectively. K and G are bulk modulus and shear modulus of the rock. Therefore, expressions for K, ρ and V_s are needed to calculate the acoustic impedance of each grid-block.

$$K = K_{\text{grain}} \frac{K_{\text{frame}} + \beta}{K_{\text{grain}} + \beta}, \tag{2.2}$$

$$\beta = \frac{K_{\text{fluid}}(K_{\text{grain}} - K_{\text{frame}})}{\phi(K_{\text{grain}} - K_{\text{fluid}})}, \tag{2.3}$$

where ϕ is the porosity, and K_{grain}, K_{frame} and K_{fluid} represent the bulk modulus of the rock grain, dry frame and fluid in the rock, respectively. These three moduli can be written as

$$K_{\text{fluid}} = \left(\frac{S_w}{K_w} + \frac{S_o}{K_o} + \frac{S_g}{K_g} \right)^{-1}, \tag{2.4}$$

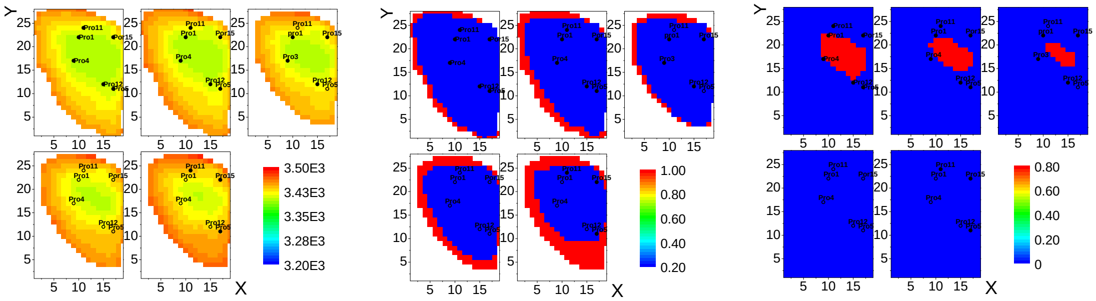
$$K_{\text{grain}} = \frac{1}{2}\left[\gamma K_c + (1-\gamma)K_s + \frac{K_s K_c}{K_s \gamma + K_c(1-\gamma)} \right], \tag{2.5}$$

$$K_{\text{frame}} = 10^{\log(K_{\text{grain}}) - 4.25\phi}, \tag{2.6}$$

where S_w, S_o and S_g represent the saturation of the three phases, and K_w, K_o and K_g represent the bulk modulus of water, oil and gas. K_s and K_c represent the bulk modulus of sand and clay, and γ is the shaliness of the rock. The saturations are obtained dynamically from simulations. For the other terms, in this example, we use the values used by Dong and Oliver [17]: $K_w = 2.39 \times 10^9 (Pa \cdot N/m^2)$, $K_o = 6.71 \times 10^8 (Pa \cdot N/m^2)$, $K_g = 3.94 \times 10^7 (Pa \cdot N/m^2)$, $K_s = 3.80 \times 10^{10} (Pa \cdot N/m^2)$, $K_c = 2.12 \times 10^{10} (Pa \cdot N/m^2)$, and $\gamma = 0.2$.

According to Han [36], the shear wave velocity can be written as a linear function of porosity and shaliness,

$$V_s = A - B\phi - C\gamma, \tag{2.7}$$

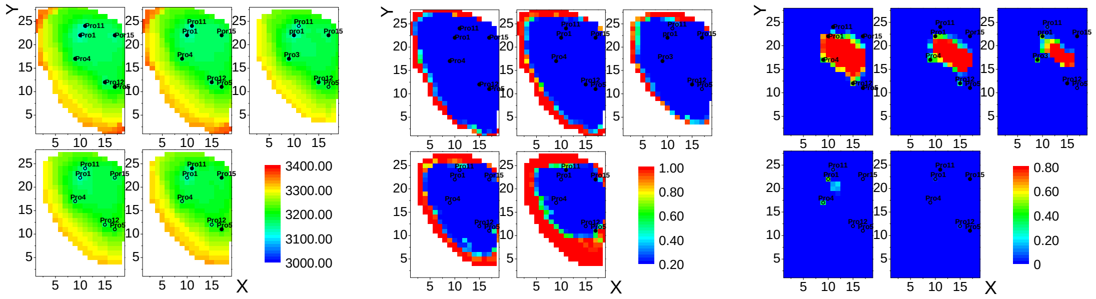


(a) Pressure ($t = 0$)

(b) S_w ($t = 0$)

(c) S_g ($t = 0$)

Figure 2.8: Pressure, water and gas saturation at time 0 (PUNQS3).



(a) Pressure ($t = 2008$)

(b) S_w ($t = 2008$)

(c) S_g ($t = 2008$)

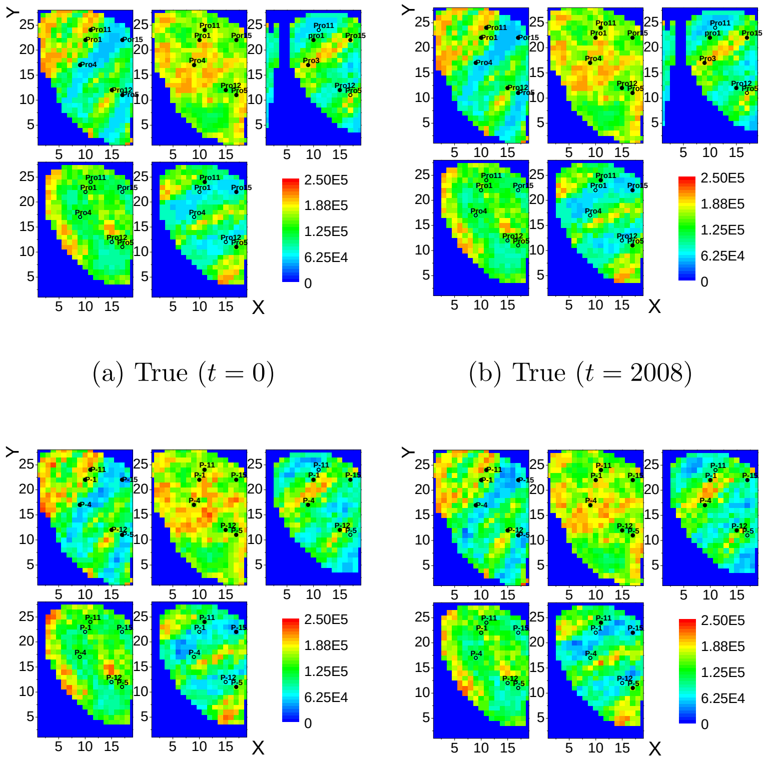
Figure 2.9: Pressure, water and gas saturation at 2008 days (PUNQS3).

where A, B and C are constant values, and the unit of V_s is m/s. We use the values used by Dong and Oliver [17]: $A = 3520,\ B = 4910,\ C = 1890$. The body density ρ can be written as:

$$\rho = (\rho_g S_g + \rho_o S_o + \rho_w S_w)\phi + (1 - \phi)\rho_{\text{solid}}, \tag{2.8}$$

where ρ_w, ρ_o and ρ_g represent the density of water, oil and gas, respectively. Also, ρ_{solid} is the density of the solid rock. The densities of the three phases are related to the fluid pressure, and can be obtained dynamically from simulators. Here, we set $\rho_{\text{solid}} = 2.65 \times 10^3 \text{kg/m}^3$.

Two seismic surveys are generated: the first one is before production (at



(a) True ($t = 0$)

(b) True ($t = 2008$)

(c) Observed ($t = 0$)

(d) Observed ($t = 2008$)

Figure 2.10: True and observed acoustic impedance at time 0 and 2008 days (PUNQS3).

time zero), and the second one is at 2008 days. Figs. 2.8 and 2.9 show the true pressure, water and gas saturations at the two seismic surveys from reservoir simulator. Figs. 2.10(a) and 2.10(b) are the computed true acoustic impedance using the model described above. The values of acoustic impedance are in field units, with the conversion to the SI unit as 1 Kg/m$^2/s$ = 0.01902 lb/ft$^2/s$. We added some noise to the true acoustic impedance data to generate the observed data shown in Figs. 2.10(c) and 2.10(d). The noise is Gaussian with zero mean and standard deviation of 1×10^4 lb/ft$^2/s$. The correlation length is 2 gridblocks in the horizontal direction, and uncorrelated vertically. The type of variogram is spherical. Noise is generated with sequential Gaussian simulation method (GsCosim, [14]).

The uncertain model parameters for the PUNQS3 example are the three fields (porosity and horizontal and vertical log-permeability). Relative permeability, water-oil and gas-oil contacts, etc., are assumed to be known.

2.2 Pluri-Gaussian Geological Model

Defining a pluri-Gaussian model requires the specification of two Gaussian random fields Z_1 and Z_2 on the simulation grids (each grid block has a pair of (z_1, z_2) values), and a truncation strategy defined on the (Z_1, Z_2) domain. It is convenient to choose the Gaussian random fields to have zero mean and unit variance, but they can have different correlation ranges and different principal directions or be based on a different type of variogram. In all examples presented here, the Gaussian random fields are based on a variogram of Gaussian type. This is because it generates the smoothest Gaussian random fields and hence tend to generate smooth boundaries between facies. Fig. 2.11 shows an example of two independent realizations of Gaussian random fields (GRFs), Z_1 and Z_2, which have the same Gaussian covariance functions, i.e., zero means, unit variances, and an isotropic correlation length of 20 grid-blocks. Sequential Gaussian simulation was used to generate the realizations of the two Gaussian random fields. In this example, our reservoir model is based on a 100×100 grid system and the two plots in Fig. 2.11 are the generated values for the pair of (Z_1, Z_2) on each gridblock. We will use lower case (z_1, z_2) to represent realizations of (Z_1, Z_2).

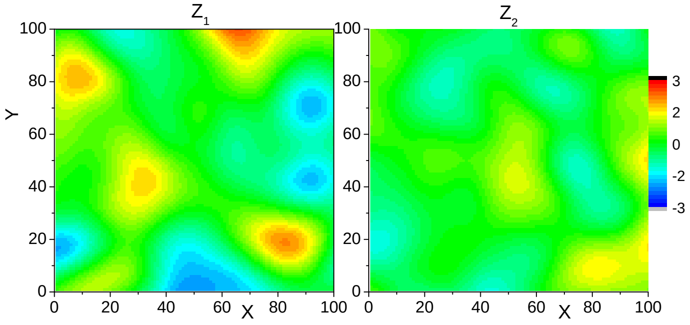
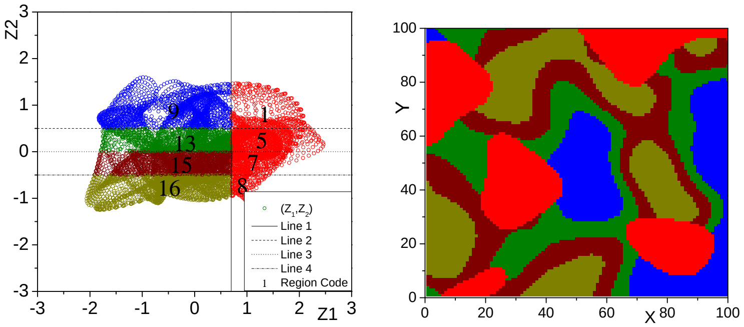


Figure 2.11: Gaussian random fields, correlation length is 20.

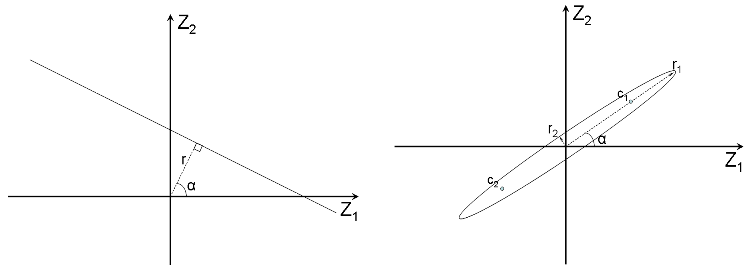


(a) Truncation

(b) Facies map

Figure 2.12: Truncation using 4 lines and facies maps.

The type and correlation range of the two variograms are set as a priori knowledge, and the two associated Gaussian random fields are defined to have zero means and unit variances. Because we have knowledge of the range of values that can be taken on by Z_1 and Z_2, it easier to define a truncation strategy. The truncation is defined on a plot of (Z_1, Z_2) values. The most commonly used truncation scheme is to divide the whole area into a few rectangular regions using perpendicular lines [48]. Figs. 2.12(a) shows a truncation with 4 lines. The numbers in this figure are codes for the regions. Each region (code) represents a particular facies, but it is possible to assign the same facies to two different regions. The truncation map and the realizations of the two Gaussian random fields (Fig. 2.11) determine which region (facies) is assigned to each gridblock. The set of colored points in the truncation map represent the set of all values of (z_1, z_2) pairs in the realization of the Gaussian random fields (Fig. 2.11). To obtain the facies in a gridblock, the pair of GRFs values in the gridblock is placed in the truncation map to find the region into which it falls, and we then assign the corresponding facies to this gridblock. Fig. 2.12(b) shows the generated facies; the color of each facies is the same as in the truncation map (Figs. 2.12(a)). Thus, when a grid block in the facies map is colored red, the associated point (z_1, z_2) must fall in region 1, 5, 7 or 8 on the truncation map of



(a) Truncation lines (b) Truncation ellipses

Figure 2.13: Definition of truncation lines and truncation ellipses.

Fig. 2.12(a). Similarly, if (z_1, z_2) for gridblock i is such that $z_1 \leq 0.7$ and $z_2 \leq -0.5$, then region 16 is assigned to gridblock i. With the color scheme used in Fig. 2.12, region 16 is represented by brown in the facies and truncation maps. Note because the brown region does not share a boundary with the blue or green regions in the truncation map of Fig. 2.12(a), brown can not have a common boundary with blue or green in the associated facies map of Fig. 2.12(b). Similarly, the blue facies can not be in contact with the wine or brown facies and the green facies can not be in contact with the brown facies. Also note that in the truncation map, regions 1, 5, 7 and 8 all correspond to the same facies and are thus assigned the same color, i.e., red. Even though there are eight regions, there are only five facies. Also note that if (z_1, z_2) for gridblock i is such that $z_1 > 0.7$, then gridblock i will contain the red facies regardless of the value of z_2. Also note that in the truncation map of Fig. 2.12(a), the red facies, which occupies four regions, is in contact with each of the other four facies. Hence, in the facies map of Fig. 2.12(b), the red region is in contact with each of the four other colors.

Although the thresholds conceivably can be adjusted according to the training image [54], here, the thresholds as well as the variograms of the Gaussian random fields are assumed known during dynamic data assimilation. In this study, in order to match dynamic data and seismic data, the facies boundary is moved by adjusting the two Gaussian random fields. We will show results for two pluri-Gaussian cases:

1. In case 1, similar to Liu and Oliver [54], three intersecting lines are used to generate a truncation map for a three facies model. To define a line in the truncation map, as shown in Fig. 2.13(a), two parameters are needed: (1) the distance from the line to the origin (r); (2) the angle (α) between the Z_1 axis and a line segment from the origin that is perpendicular to the truncation line; the angle (α) is measured counterclockwise from the positive Z_1 axis to line segment.

2. Channels in a fluvial system are geometrically characterized by some continuous and relatively narrow bands intersecting each other. We have found that two long, narrow ellipses can be used as truncations of the two independent Gaussian random fields to generate a facies map with roughly these features. To define an ellipse centered at the origin in the truncation map, as shown in Fig. 2.13(b), three parameters are needed: (1) one half the length of the long axis (r_1); (2) one-half the length of the short axis (r_2); (3) the angle (α) measured counterclockwise from the positive Z_1 axis to the long axis of the ellipse.

A line can divide the plane into two regions. To decide which region a given pair of Gaussian random field values (z_1,z_2) falls in, the following function is evaluated:

$$f_L(z_1, z_2) = \cos(\alpha) z_1 + \sin(\alpha) z_2 - r. \tag{2.9}$$

Note $f(z_1, z_2) = 0$ defines a line such as the one shown in Fig. 2.13(a). If $f(z_1, z_2) \leq 0$, the point (z_1, z_2) is located on, below, or to the left of the line, and if $f(z_1, z_2) > 0$ this point is located above or to the right of the line. An ellipse can also divide the whole region into two regions. An ellipse, as shown in Fig. 2.13(b), can be defined by $f_E(z_1, z_2) = 0$ where

$$f_E(z_1, z_2) = \frac{|Z - C_1| + |Z - C_2|}{2} - r_1. \tag{2.10}$$

In the preceding equation, C_1 and C_2 are the two focal points of the ellipse, and $|Z - C_1|$ is the distance between the point (z_1, z_2) and C_1, and $|Z - C_2|$ is that between this point and C_2. If $f_E(z_1, z_2) \leq 0$ the point (z_1, z_2) falls inside, and if $f_E(z_1, z_2) > 0$ it falls outside the ellipse. Here, the distance from C_1 (or C_2) to the origin is $\sqrt{r_1^2 + r_2^2}$ (r_1 and r_2 are defined in the discussion of Fig. 2.13(b)).

In order to use the truncation map for the purpose of specifying facies, the regions are coded by the dividers (lines or ellipses) in binary mode. Suppose we have N dividers where each divider is either a line or an ellipse. A binary array is defined as $B = \{b_1, b_2, \cdots, b_N\}$, in which the b_i's can only take one of the two values, 0 or 1. A specific region can then be coded so that all points in a specific region correspond to the same value of B. To do this, we consider the value of the N functions defining the dividers (truncation functions). If N lines are used for truncation, we let $f_{L,i}$ denote the ith truncation line. Then each specific region j in the truncation map has all its boundaries (except those at $\pm\infty$) defined by truncation lines. Thus, for each and every (z_1, z_2) in this specific region, either $f_{L,i}(z_1, z_2) < 0$ or $f_{L,i} \geq 0$, i.e., each $f_{L,i}$ is either negative or nonnegative throughout the region. If $f_{L,i} < 0$, we set $b_i = 0$ and if $f_{L,i} \geq 0$, we set $b_i = 1$. Having found the value of each b_i for region j, we assign region j the binary code B_j defined by

$$B_j = \sum_{i=1}^{N} b_i 2^{i-1}. \tag{2.11}$$

Note there are only 2^N possible values for B_j, namely, $\{0, 1, 2, \cdots, 2^N - 1\}$; thus, with N dividers we can have no more than 2^N regions.

If ellipses, $f_{E,i} = 0$, $i = 1, 2, \cdots N$ are used to define the truncation map, then for region j, we set $b_i = 0$ if for all (z_1, z_2) in region j, $f_{E,i}(z_1, z_2) > 0$ and set $b_i = 1$, if $f_{E,i}(z_1, z_2) \leq 0$. Once the b_i are determined, the binary code for region j is calculated from Eq. 2.11. With the same logic, given an EnKF updated value of (z_1, z_2) at a gridblock, we can calculate the value for the binary code for this (z_1, z_2) by determining the value of each truncation equation at this (z_1, z_2), and based on this sign, we calculate the value of each b_i by the same procedure used above to code the regions. With these values of the b_is, we compute

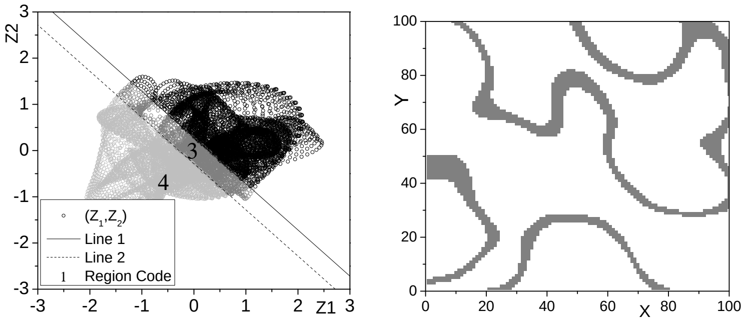
$$B_v = \sum_{i=1}^{N} b_i 2^{i-1}; \tag{2.12}$$

this value of B_v must be equal to one of the region B_j values. If $B_v = B_\ell$, then (z_1, z_2) falls into region ℓ on the truncation map and the facies corresponding to region ℓ is assigned to the gridblock.

We will show that a truncation with intersecting ellipses gives a rough model of a channelized reservoir. Then the methods for conditioning initial realizations to hard data (observed facies at well locations) will be introduced. This is because in order to assimilate production data using EnKF we need a suite of realizations of the Gaussian random fields that honor the hard data.

2.2.1 Truncation Scheme One : Intersecting Ellipses

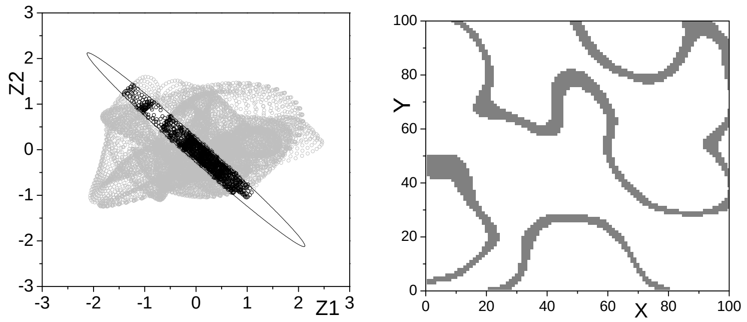
The idea of using ellipses for truncation comes from truncating the Gaussian random fields using two parallel lines ($r = 0.2$ for both lines). In this type of truncation, we can obtain some narrow and continuous bands which roughly represent channels as shown in Fig. 2.14. The two lines separated the truncation map into three regions. The region between the two lines (region 3) corresponds to the channel facies and separate the other two regions in the truncation plot. These two regions are both set equal to non-channel facies and are shown as white in the facies map, the gray channel facies in the facies map separates the white facies. We can also use an ellipse



(a) Truncation

(b) Facies map

Figure 2.14: Truncate the Gaussian random fields using two lines.



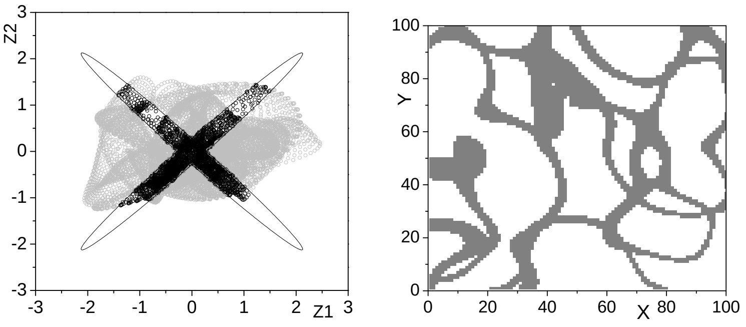
(a) Truncation

(b) Facies map

Figure 2.15: Truncate the Gaussian random fields using one ellipse.

($r_1 = 3.0$, $r_2 = 0.2$ and $\alpha = 45°$), to approximate the parallel lines. The grid blocks that have (z_1, z_2) values inside the ellipse on the truncation map are occupied by the channel facies. As shown in Fig. 2.15, the facies map is virtually identical to the one generated from the two parallel lines. Because defining an ellipse need 3 parameters while defining two parallel lines need 4 parameters with one redundant parameter (either angle for the two lines), we prefer ellipses to generate truncation for channel and non-channel facies.

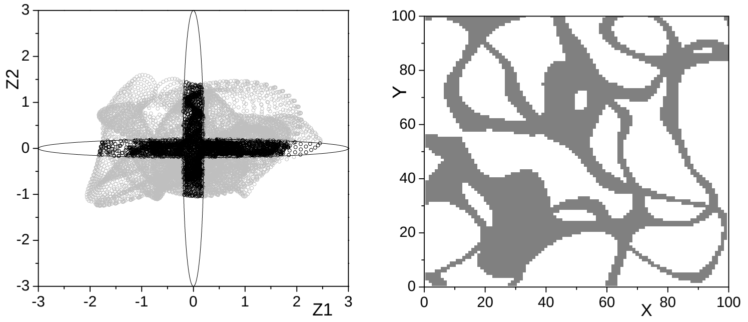
When two ellipses are used, we can generate a more complex facies map. Figs. 2.16(a) and 2.16(b) show the truncation map and resulting facies map if an additional ellipse with its longest axis perpendicular to the major axis of the first



(a) Truncation

(b) Facies map

Figure 2.16: Truncate the Gaussian random fields using two ellipses (45° and 135°).

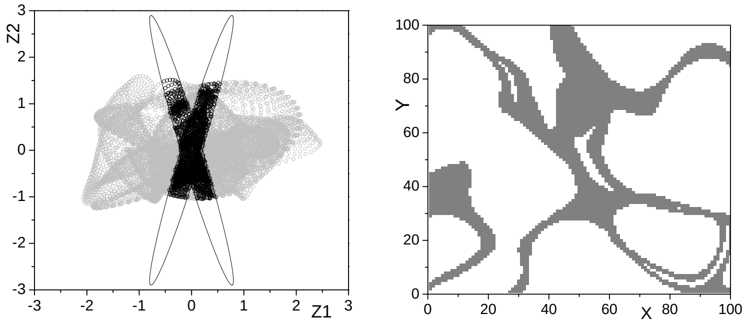


(a) Truncation

(b) Facies map

Figure 2.17: Truncate the Gaussian random fields using two ellipse (0° and 90°).

ellipse is applied to the same Gaussian random fields. Here, α is 45° for the first ellipse, and 135° for the second one. Figs. 2.17(a) and 2.17(b) shows another case where the two ellipses are perpendicular but have a different orientation (0° and 90° respectively); similar results are obtained. In this case, more channels appear, but unfortunately, this also yields some "thick blobs" of channel facies such as the one near the upper center in 2.16(b). We have not yet found a way to overcome this deficiency. To compare with the cases where major axis of the two truncation ellipses are perpendicular, Figs. 2.18(a) and 2.18(b) show a case where the angle between the major axes of the two ellipses is 30 degrees. In this case, more (z_1, z_2) values fall inside one or both ellipses so the resulting facies map looks even less like a channel



(a) Truncation (b) Facies map

Figure 2.18: Truncate the Gaussian random fields using two ellipse (75° and 105°).

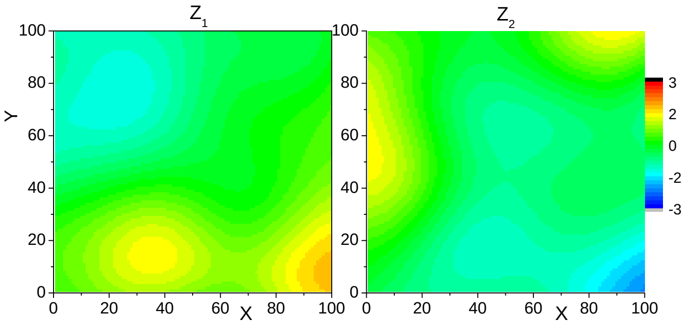
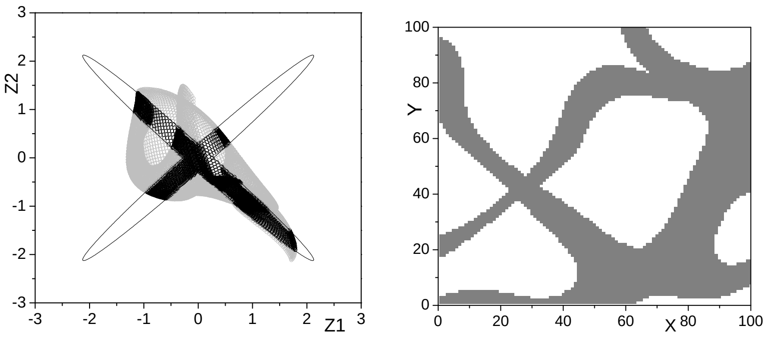


Figure 2.19: Gaussian random fields, correlation length is 40.

system than the results obtained when the two ellipses are perpendicular. We also found, but their results are not shown here, that if the center of two ellipses are off the (0,0) point of truncation map, the shape of channels will become less realistic. Figs. 2.20 and 2.21 show two additional examples, where the Gaussian random fields have a longer correlation length (40 grid blocks) as shown in Fig. 2.19. These two examples use two ellipses with different minor radii, i.e., 0.2 in Fig. 2.20(a), and 0.4 in Fig. 2.21(a). Note that for the case with the longer minor radius, most of the realizations of (z_1, z_2) fall inside the two ellipses so resulted facies map Fig. 2.21(b) show that most of the gridblocks are occupied by channel facies.

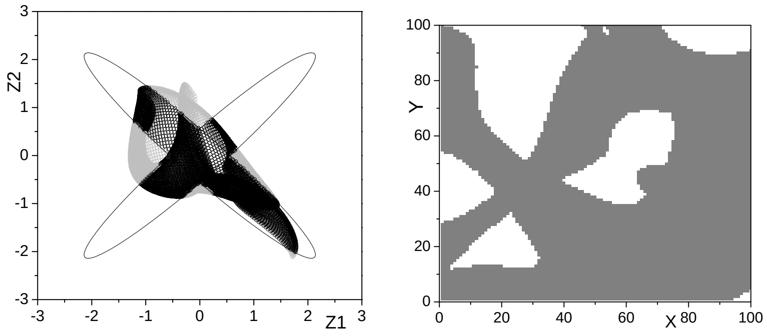
From these experiments, we note that Gaussian random fields with longer correlation lengths generate fewer but wider channels. Moreover, increasing the number of truncating ellipses results in a larger proportion of channel facies, but



(a) Truncation

(b) Facies Map

Figure 2.20: Truncate using two ellipses (correlation length is 40, short radius is 0.2, long radius is 3.0).

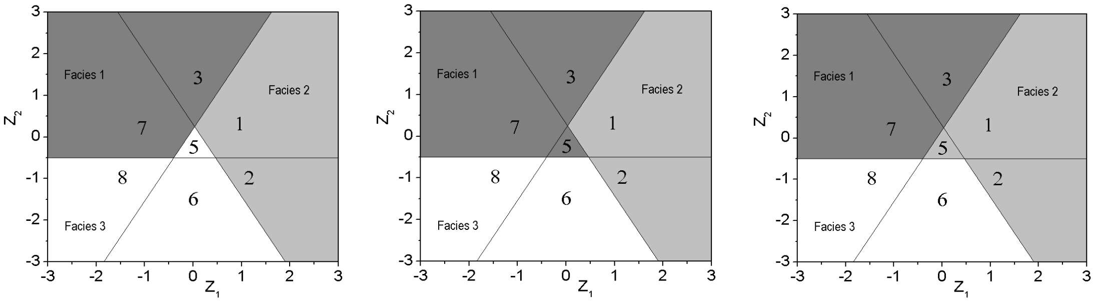


(a) Truncation

(b) Facies Map

Figure 2.21: Truncate using two ellipses (correlation length is 40, short radius is 0.4, long radius is 3.0).

the channels contain non-geological "bulges." Although the channels generated are not very geologically realistic, we use this model to explore whether we can properly characterize the connectivity of a channelized reservoir by assimilating production data with EnKF. To sum up, the short radius of each ellipse must be kept reasonably small, and we will use narrow and perpendicular ellipses with centers at the origin as shown in Fig. 2.16(a) to model the fluvial system. In this way, we can move the channel by adjusting the values of Gaussian random fields to assimilate the measured data. In this process, the truncations and variograms should be carefully chosen and



(a) Layer 1 truncation

(b) Layer 2 truncation

(c) Layer 3 truncation

Figure 2.22: Truncations for the 3 layers, facies specification for the 5 regions varies from layer to layer.

fixed as geological description of the reservoir during EnKF data assimilation.

2.2.2 Truncation Scheme Two : Intersecting Lines

Three intersecting lines were used by Liu and Oliver [54] to generate facies distributions for two dimensional system, and the same methods will be used in this study to construct a 3-layer model. In each layer, we will use three truncation lines to define facies as shown in Figs. 2.22(a), 2.22(b) and 2.22(c). In this example, for each layer, the three lines are the same, and have the same distance to the origin: $r = 0.2$, but have different angles: $\alpha = 30°, 150°, 270°$ respectively. From these three plots we can see that, for each layer, the facies specifications are the same for each layer except for the 5th region. Because the 5th region is located around the origin, the probability of generating (z_1, z_2) values in this region is high, thus varying the facies specification for this region has considerable impact on the proportion of each facies in each layer.

The Gaussian random fields of these three layers follow different variograms, and are all uncorrelated with each other, and therefore the facies are also uncorrelated. We will discuss this issue after presenting a new procedure to condition to hard data represented by observations of facies at a well.

2.2.3 Conditioning to Hard Data and Geological Models

With predefined variograms, we use sequential Gaussian simulation to generate an ensemble of Gaussian random fields. This ensemble can be used to generate a facies distribution by applying the truncation map as a prior, but should not be used as an initial ensemble for EnKF if the realizations fail to honor the hard data (observed facies at well locations). Here, conditioning to hard data is somewhat different from the traditional one in the sense that the observed hard data at wells is the facies type, not a specific realization of (Z_1, Z_2). For concreteness, assume that in the channel example, all wells penetrate the channel facies. If we use two ellipses to truncate, the values of the Gaussian random fields values at gridblocks containing wells simply have to fall within one of the two ellipses in the truncation map to honor the observation of a particular facies at a wells gridblock. This is also true for the three facies example truncated by three lines.

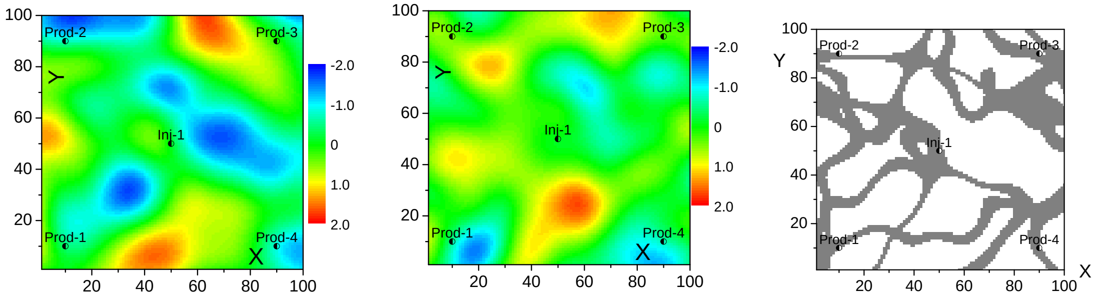
The pdf of a Gaussian random field conditioned to observed facies at the well location is not Gaussian. In the sequential Gaussian simulation algorithm, the values of hard data at the well locations need to be specified. In our case, these values can vary from realization to realization, and should follow a certain distribution controlled by the dividers in the truncation map. In this study, the pdf of each hard datum is approximated by a data bank which consists of a number of samples of Gaussian random pairs, so that the samples in each data bank can roughly represent the distribution of the Gaussian random field values at the location and the layer where the hard data are observed. For example, the 2-D channel example requires five data banks because 5 wells will be included; the 3-layer three facies example requires 15 data banks because there are 15 hard data observed in 5 wells and in 3 layers. We use the following three steps to generate these data banks at gridblocks containing wells where an observation of the facies type is available. This scheme assumes that the Gaussian random fields values at two distinct well locations are uncorrelated, i.e., the distance between wells should be greater than the spatial correlation length of the Gaussian random fields, and Gaussian random fields should be uncorrelated in the vertical direction.

1. Generate a set of unconditional realizations of the Gaussian random fields (Z_1 and Z_2), then separate the Gaussian random field values pairs gridblock by gridblock with the following process. If (z_1, z_2) is a pair in any one of the gridblocks, such that the truncation map indicates that (z_1, z_2) corresponds to the facies observed at well j, then we put (z_1, z_2) into the data bank for hard data at well j. In this way, each data bank can roughly represent the desired distribution by rejecting the pairs that do not satisfy the hard data. Through this process, we require that the number of pairs in each bank to be at least twenty times the number of realizations that will be used in the initial ensemble of realizations conditioned to hard data.

2. For each realization, randomly draw a pair of values (z_1, z_2) for each well gridblock from the corresponding data bank. Different realizations will have different sampled values.

3. To generate a realization of the Gaussian random fields conditioned to the hard data, apply sequential Gaussian simulation using the sampled values in step 2 as the hard data of Gaussian random field values at well gridblocks and continue until we have generated a complete realization of (Z_1, Z_2) at all gridblock locations. This gives one realization of (Z_1, Z_2) conditioned to hard data. Repeat the process to generate as many realizations as we wish to have in the initial ensemble for beginning the assimilation of production data with EnKF.

Using the two truncation schemes shown above, we will generate two synthetic examples, which should be a realization of the prior model. We will generate



(a) True Z_1

(b) True Z_2

(c) True Facies

Figure 2.23: Channel example, true model.

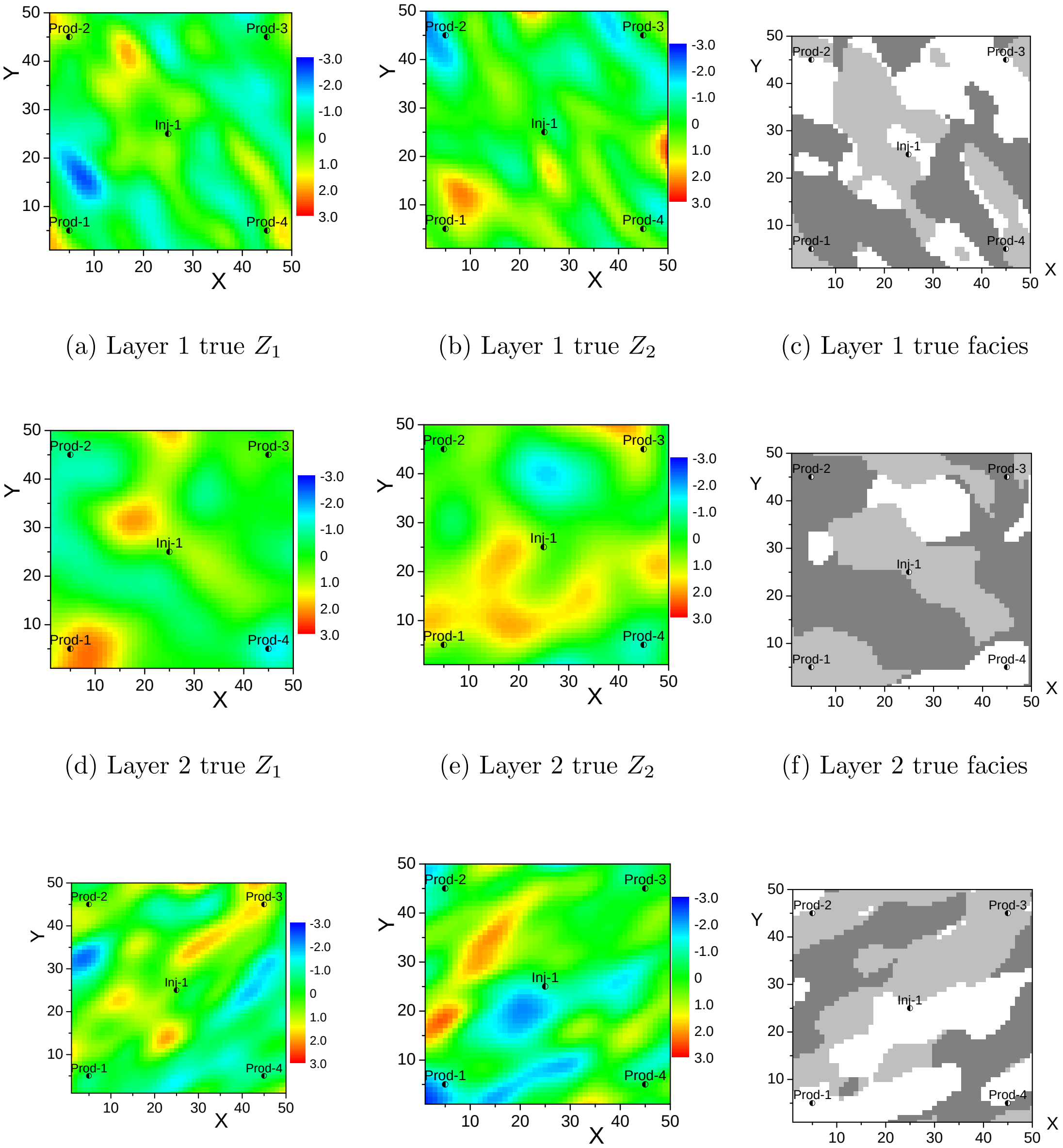
many realizations and pick one as the truth from which we can generate synthetic production data and seismic data. To get a complete prior, variograms of Gaussian random fields and hard data need to be specified.

1. **2-D channel model.** In this case, Gaussian variograms are used to define the Gaussian random fields where both variograms are isotropic with ranges equal to 20 gridblocks. The elliptical truncation map of Fig. 2.16(a) is used. The grid is 100×100, with four production wells near the corners, and a water injector at the center. Table 2.2 shows the gridblock location of wells and the observed facies (hard data).

Well Name	X	Y	Facies
Prod1	10	10	Channel
Prod2	10	90	Channel
Prod3	90	90	Channel
Prod4	90	10	Channel
Inj1	50	50	Channel

Table 2.2: Channel example hard data.

The dimension of each gridblock is 25 feet in both horizontal directions, and 5 feet in the vertical direction. The porosity and permeability are assumed to be homogeneous in each facies. For all examples in this study except those in Chapter 6, porosity and $\ln k_h$ for each facies are equal to the true values, i.e.,



(a) Layer 1 true Z_1

(b) Layer 1 true Z_2

(c) Layer 1 true facies

(d) Layer 2 true Z_1

(e) Layer 2 true Z_2

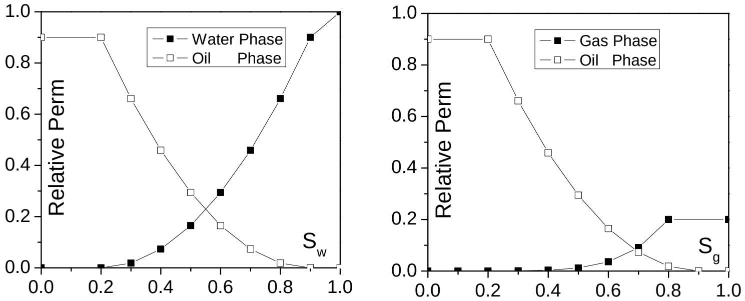
(f) Layer 2 true facies

(g) Layer 3 true Z_1

(h) Layer 3 true Z_2

(i) Layer 3 true facies

Figure 2.24: Three facies example, true model.



(a) Water-Oil (b) Gas-Oil

Figure 2.25: Relative permeability curves for the two pluri-Gaussian examples.

channel porosity equals to 0.212, non-channel porosity equals to 0.162, channel permeability equals to 1420.8 md, and non-channel permeability equals to 11.5 md.

2. **Three facies three layer model.** The variograms used to define the two Gaussian random fields for the three layers are Gaussian with the parameters given in Table 2.4. The truncation map shown in Fig. 2.22 is used. The simulation grid is 50 × 50 × 3, with four production wells near the corners, and a water injector at the center. All wells are perforated in all three layers. Table 2.3 shows the location of wells and the observed facies (hard data) for each layers.

Well Name	X	Y	Layer 1	Layer 2	Layer 3
Prod1	5	5	Facies 1	Facies 2	Facies 3
Prod2	5	45	Facies 3	Facies 1	Facies 2
Prod3	45	45	Facies 3	Facies 1	Facies 2
Prod4	45	5	Facies 2	Facies 3	Facies 1
Inj1	25	25	Facies 2	Facies 2	Facies 3

Table 2.3: Three facies example hard data.

The dimension of each gridblock is 50 feet in both horizontal directions, and 2 feet in the vertical direction. The porosity and permeability are assumed to

Variogram	Layer 1	Layer 2	Layer 3
Long Range	10	10	10
Short Range	5	10	5
Angle to X	120°	–	30°

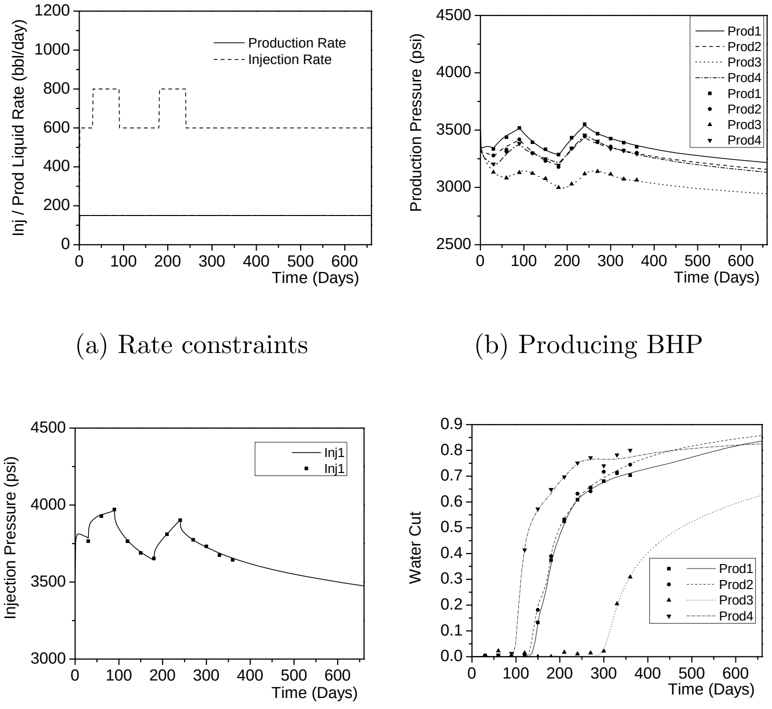
Table 2.4: Three facies example variograms.

be homogeneous in each facies. Except in chapter 6, true values will be used for the facies porosity and $\ln k_h$. The porosity of facies 1, 2 and 3, respectively, are 0.06, 0.127 and 0.213, and the permeability values of facies 1, 2 and 3, respectively, are 10.7 md, 99.88 md, 527.95 md. Vertical permeability is always set equal to 10% of horizontal permeability.

These two examples share the same fluid and rock properties, which are the same as for the PUNQS3 example except for porosity and permeability and water relative permeability. Fig. 2.25 shows the relative permeability curves. Fig. 2.23 and Fig. 2.24 show the true Gaussian random fields and facies distribution for the two examples.

2.2.4 Production History and Data

The production history and measured data for the channel example are shown in Fig. 2.26. The four producers have the same rate schedule: a constant liquid rate production target constraint of 150 STB/day, and the injector operates at a water injection rate of 600 STB/day with two short periods at a higher rate of 800 STB/day. The total simulation time considered is 660 days. Measurements are available during the first 360 days with 108 data, and the prediction for the following 300 days will be estimated. Figs. 2.26(b), 2.26(c) and 2.26(d) show the predictions from the truth model (lines) and the measurements (symbols). GOR is not included because the pressure is always above bubble point so that the GOR is constant and equals to 0.4 Mscf/STB. The standard deviation of measurement error is 10 psi in the pressure



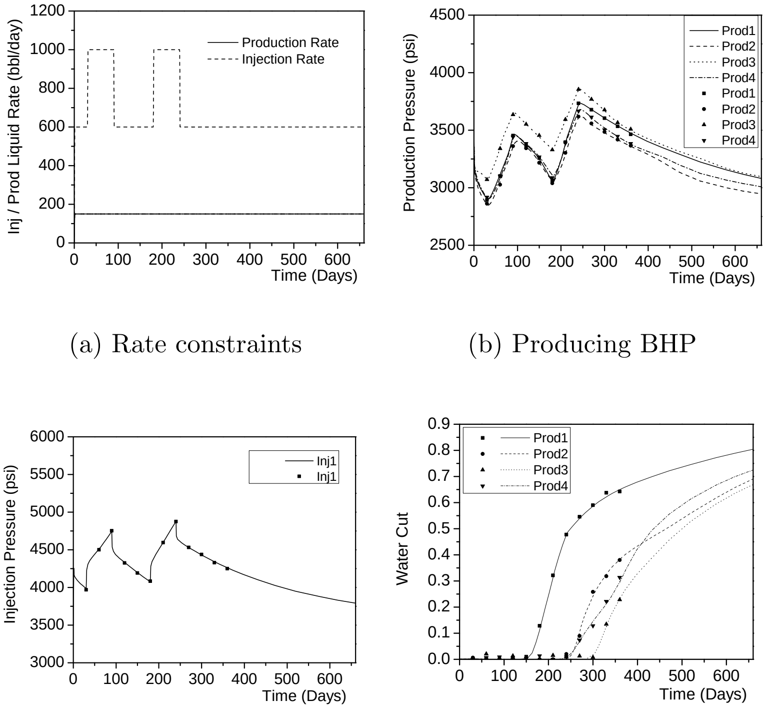
(a) Rate constraints

(b) Producing BHP

(c) Injection BHP

(d) Producing water cut

Figure 2.26: Channel model rate schedule true and measured data.



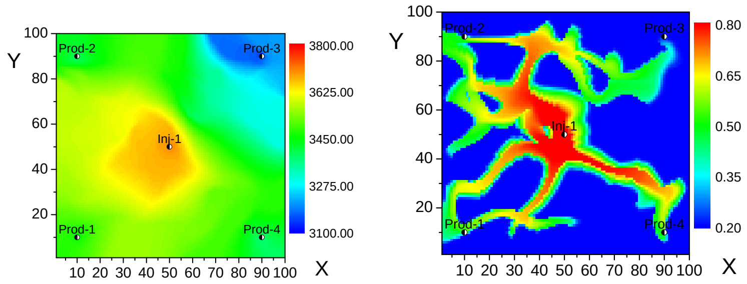
(a) Rate constraints

(b) Producing BHP

(c) Injection BHP

(d) Producing water cut

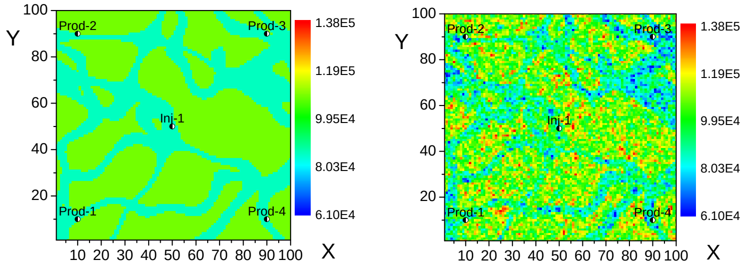
Figure 2.27: Three facies model rate schedule true and measured data.



(a) Pressure

(b) S_w

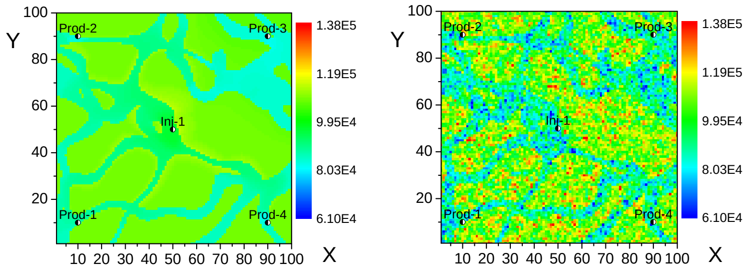
Figure 2.28: Pressure and water saturation at 300 days (channel example).



(a) True acoustic impedance

(b) Measured acoustic impedance

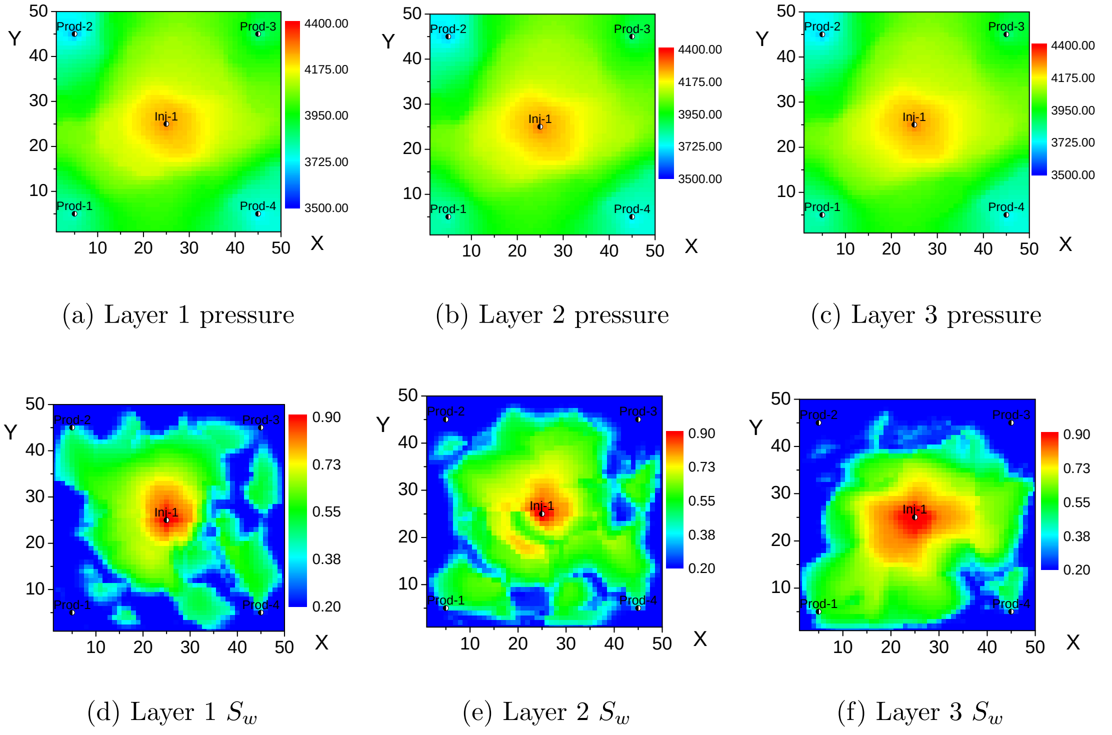
Figure 2.29: True and observed acoustic impedance time 0 (channel example).



(a) True acoustic impedance

(b) Measured acoustic impedance

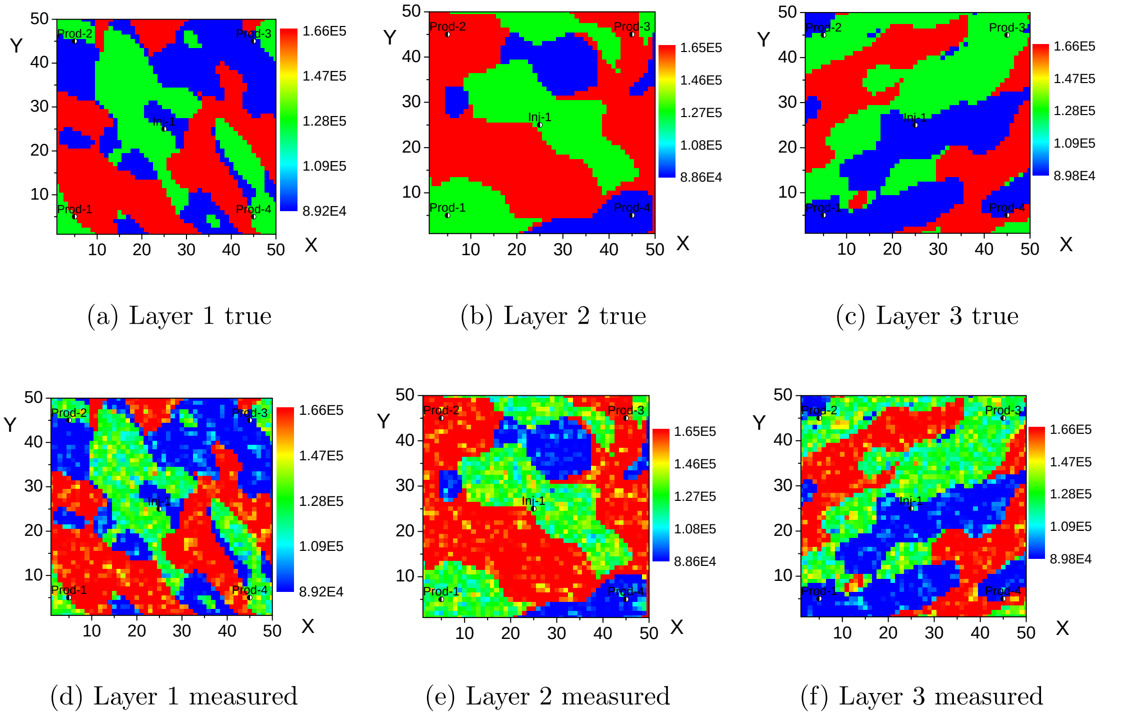
Figure 2.30: True and observed acoustic impedance at 300 days (channel example).



(a) Layer 1 pressure (b) Layer 2 pressure (c) Layer 3 pressure

(d) Layer 1 S_w (e) Layer 2 S_w (f) Layer 3 S_w

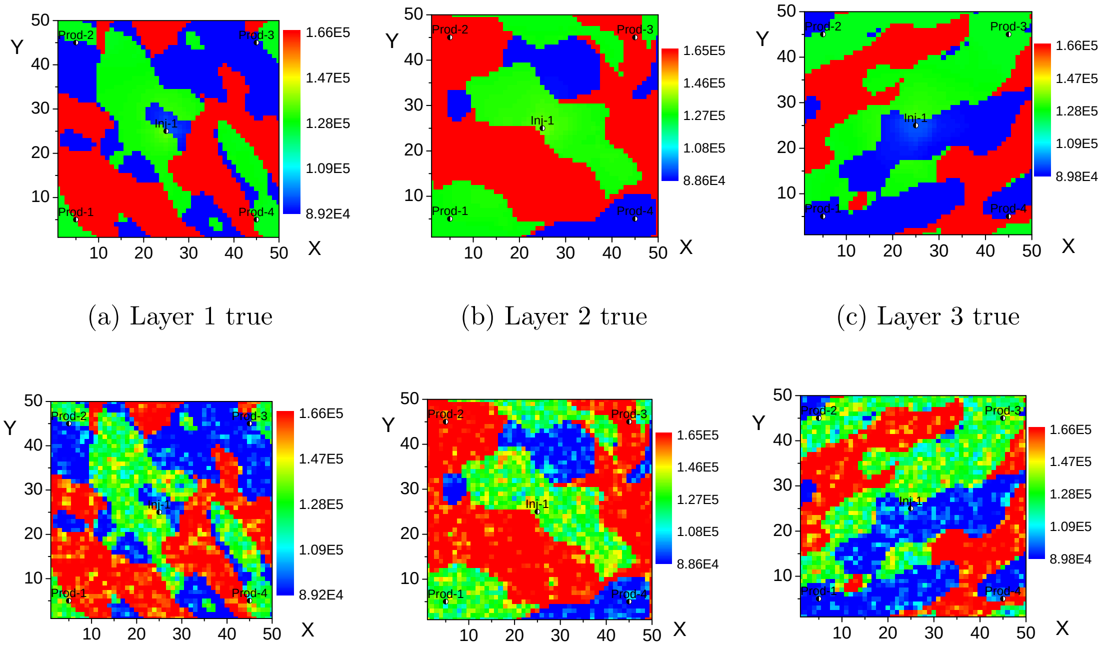
Figure 2.31: Pressure and water saturation at 300 days (three facies example).



(a) Layer 1 true (b) Layer 2 true (c) Layer 3 true

(d) Layer 1 measured (e) Layer 2 measured (f) Layer 3 measured

Figure 2.32: True and observed acoustic impedance at time 0 (three facies example).



(a) Layer 1 true (b) Layer 2 true (c) Layer 3 true

(d) Layer 1 measured (e) Layer 2 measured (f) Layer 3 measured

Figure 2.33: True and observed acoustic impedance at 300 days (three facies example).

data, and 5% of the true value in the water cut data. The rate schedule and data of the three facies example are shown in Fig. 2.27. The schedule is the same as the channel example except the high injection rate is 1000 STB/day. Measurement error definition is also the same as for the channel example.

For the two pluri-Gaussian examples, there are two seismic surveys: prior to production, and at 300 days. Noise is added to the true inverted seismic data. The noise has the same variance, type of variogram and spatial correlation range as the PUNQS3 model introduced earlier. Figs. 2.28 shows the pressure and water saturations for the channel example at the seismic survey of 300 days. The initial pressure is homogeneous 3500.88 psi, and the initial water saturation is 0.2. The gas saturation is zero throughout the life of reservoir. Figs. 2.29 and 2.30, respectively, show the corresponding true and measured acoustic impedances at the two survey times for the channel example. Figs. 2.31, 2.32 and 2.33, respectively, show the pressure and water saturation, true and measured acoustic impedances for the three facies example. The initial pressure in the top layer is 3500.35 psi.

2.3 Summary

This chapter introduces one Gaussian geological model (PUNQS3 model), and two pluri-Gaussian geological model (three facies model and channel model). A new truncation scheme with two perpendicular ellipses is used to generate facies distribution of a channelized reservoir. The sensitivity of the facies distribution to the configuration of these two ellipses was also studied. The channel facies is not very geologically realistic, but gives a reasonable model for studying whether production and seismic data can resolve the connectivity between wells. For the cases in Chapter 6, the facies porosity and permeability are assumed to be uncertain (follow Gaussian distribution) and are put into the model part of the state vector of EnKF. For the cases in Chapter 3, 4 and 5 we use true values of porosity and permeability.

CHAPTER 3

ENSEMBLE KALMAN FILTER FOR PRODUCTION DATA

The ensemble Kalman filter (EnKF) was introduced by Evensen [22] as a sequential data assimilation method. This method starts from an initial ensemble of state vectors, which can represent the prior probability distribution of model parameters and the initial condition of the reservoir, from which we can predict the reservoir performance in the future by simulating each ensemble member forward. The state vector y is defined to contain uncertain model parameters and primary variables in the simulation:

$$y = \left[m^T, p^T\right]^T, \tag{3.1}$$

where m is a random column vector of model parameters (porosity and log-permeability fields for PUNQS3 case, facies properties and Gaussian random fields for pluri-Gaussian cases); p is a column vector of reservoir simulator primary variables such as pressure, water saturation, gas saturation, R_s and R_v if applicable. Parameters such as relative permeability, fluid contact depth etc., are assumed to be known in the experiments considered here and are not included in the state vector.

In EnKF, at each data assimilation step, the prior pdf for the random vector y is modeled as Gaussian distribution with $y \sim \mathsf{N}\{\bar{y}, C_y\}$, where $\bar{y}$ is the prior mean vector and C_y is the prior covariance matrix. The essence of the state vector is that a realization of y together with known model parameters represents a possible state of the reservoir at a specific time, and the set of realizations of y can be used to predict the future performance of the reservoir and characterize its uncertainty. The initial ensemble is a set of N_e realizations of the model parameters and the initial state of the reservoir, which is defined as a matrix

$$Y_0 = \left[y_{0,1}, y_{0,2}, \cdots, y_{0,N_e}\right], \tag{3.2}$$

where $y_{0,j} = \left[m_{0,j}^T, p_{0,j}^T\right]^T, j = 1, 2, \cdots, N_e$ represent N_e realizations of the state vector generated from its initial probability distribution. Each realization in the matrix Y_0 is constructed to honor the hard data observed at well locations. Starting from Y_0 we can predict reservoir performance at any time in the future, but these predictions are not conditioned to the dynamic data.

Suppose we have measurements of production data and/or inverted seismic data at N time steps $\{t_i, i = 1, N\}$:

$$\{d_{obs,i}, C_{d,i}, i = 1, 2, \cdots, N\}, \tag{3.3}$$

where $d_{obs,i}$ is the observed column data vector at t_i, with its dimension represented by $N_{d,i} \times 1$ ($N_{d,i}$ is the number of observed data at time t_i). Here we assume the measurement error vector follows a Gaussian distribution with mean equal to the $N_{d,i} \times 1$ zero vector and covariance matrix $C_{d,i}$, which is an $N_{d,i} \times N_{d,i}$ matrix. In EnKF, an ensemble of noisy measurements is used as conditioning data:

$$d_{uc,i,j} = d_{obs,i,j} + \epsilon_{i,j}, j = 1, 2, \cdots, N_e. \tag{3.4}$$

where each $\epsilon_{i,j}$ vector is a sample of the measurement error at time t_i generated from the Gaussian distribution $\mathsf{N}(0, C_{d,i})$.

The predicted state vector at the time we first have measurements (t_1) can be obtained by running N_e simulations starting from Y_0, and the ensemble of predictions are stored as the columns of an $N_y \times N_e$ matrix $Y_1^p = \left[y_{1,1}^p, y_{1,2}^p, \cdots, y_{1,N_e}^p\right]$, where N_y is the dimension of y, and the superscript p indicates prediction. In this state vector, the predicted vectors of model parameters are identical to the initial ensemble of model vectors (i.e. $m^p_{1,j} = m_{0,j}$), and the primary variables are obtained by running the reservoir simulator from time zero to t_1. In the EnKF framework, the predictions contained in Y^p_1 are adjusted (analyzed or updated) to match the vectors of measurements, the $d_{uc,1,j}$'s. The analyzed state matrix Y^a_1 has the same dimensions as Y^p_1. The realizations in the analyzed state matrix are used to predict the prior states at the second data assimilation time (t_2). The predicted state matrix is Y^p_2, i.e., the columns of Y^p_2 ($y^p_{2,j}$) are updated (analyzed) by updating $y^p_{2,j}$ to $y^a_{2,j}$ by matching $d_{uc,2,j}$, $j = 1, 2, \cdots, N_e$. This process is repeated each time we have measurements until all the data are assimilated, and then we can predict the future performance of the reservoir which is conditional to all the measurements. This sequential assimilation scheme is faster than RML method [45, 63, 69], because predictions do not require running the simulator from time zero after each analysis.

At the nth assimilation time, to evaluate the data mismatch, it is convenient to also store the predicted data from the ensemble in an $N_{d,n} \times N_e$ matrix:

$$D^p_n = \left[d^p_{n,1}, d^p_{n,2}, \cdots, d^p_{n,N_e}\right], \tag{3.5}$$

where the jth column ($d^p_{n,j}$) denotes the data predicted at the time t_n from the jth analyzed ensemble member at the previous data assimilation step, and the corresponding data mismatch can be written as:

$$\delta d_{mis,n,j} = d_{uc,n,j} - d^p_{n,j}. \tag{3.6}$$

In the following sections of this chapter, we will show the details of the EnKF algorithm and its implementation on the three examples described in the last chapter, i.e. the PUNQS3 model, the three facies model and the channel model. This chapter will focus on assimilating the production data, and the next chapter will focus on assimilating seismic data.

3.1 Ensemble Kalman Filter (EnKF) Algorithm

The ensemble Kalman filter algorithm is used to update (analyze) the predicted state ensemble Y_n^p to match data at the nth assimilation step, so that the analyzed ensemble of realizations Y_n^a approximately represents a sample of the pdf for y_n conditioned to all data up to and including time t_n (see [24]). From the definition in Eqs. 3.5 and 3.6, $d_{n,j}^p$ is the predicted data, and $\delta d_{mis,n,j}$ is the data mismatch of the jth ensemble member at time t_n. The analysis equation (see [27]) at time t_n for updating the jth ensemble member is given by

$$\begin{aligned} y_{n,j}^a &= y_{n,j}^p + C_{Y_n^p,D_n^p}\left[C_{D_n^p,D_n^p} + C_{D,n}\right]^{-1}\delta d_{mis,n,j} \\ &= y_{n,j}^p + R_n \delta d_{mis,n,j}, \end{aligned} \tag{3.7}$$

where

$$R_n = C_{Y_n^p,D_n^p}\left[C_{D_n^p,D_n^p} + C_{D,n}\right]^{-1}, \tag{3.8}$$

where $C_{Y_n^p,D_n^p}$ is the covariance matrix of the predicted state and predicted data with dimension equal to $N_y \times N_{d,n}$, and $C_{D_n^p,D_n^p}$ is the auto-covariance matrix of predicted data with dimension equal to $N_{d,n} \times N_{d,n}$. Eq. 3.7 gives the analyzed state vector for the jth ensemble member. The matrix R_n is the same for each ensemble member and is called the Kalman gain. As is shown in the analysis equation, R_n multiplied by the data mismatch of the jth ensemble member gives the adjustment term. This analysis equation is similar to performing one Gauss-Newton method using an approximate of average gradient for minimizing

$$\begin{aligned} O_j(y_{n,j}) =& \frac{1}{2}\left[y_{n,j} - y_{n,j}^p\right]^T C_{Y_n^p,Y_n^P}^{-1}\left[y_{n,j} - y_{n,j}^p\right] \\ &+ \frac{1}{2}\left[d_{uc,n,j} - d_{n,j}\right]^T C_{D_n}^{-1}\left[d_{uc,n,j} - d_{n,j}\right]; \end{aligned} \tag{3.9}$$

(see [81]).

In the EnKF algorithm, we do not have to explicitly compute $C_{Y_n^p,D_n^p}$ and $C_{D_n^p,D_n^p}$. The covariances are represented by the ensemble and are approximated by

$$\begin{aligned} C_{Y_n^p,D_n^p} &= \frac{1}{N_e-1}\sum_{j=1}^{N_e}\left[y_{n,j}^p-\overline{y_n^p}\right]\left[d_{n,j}^p-\overline{d_n^p}\right]^T \\ C_{D_n^p,D_n^p} &= \frac{1}{N_e-1}\sum_{j=1}^{N_e}\left[d_{n,j}^p-\overline{d_n^p}\right]\left[d_{n,j}^p-\overline{d_n^p}\right]^T, \end{aligned} \tag{3.10}$$

where $\overline{y_n^p}$ and $\overline{d_n^p}$ are the mean vectors given by

$$\overline{y_n^p} = \frac{1}{N_e}\sum_{j=1}^{N_e} y_{n,j}^p \tag{3.11}$$

and

$$\overline{d_n^p} = \frac{1}{N_e}\sum_{j=1}^{N_e} d_{n,j}^p. \tag{3.12}$$

Let $\overline{Y_n^p}$ be an $N_y \times N_e$ matrix which has all its column equal to $\overline{y_n^p}$, and $\overline{D_n^p}$ be an $N_{d,n} \times N_e$ matrix which has all its column equal to $\overline{d_n^p}$. We define the error matrices:

$$\begin{aligned} \Delta Y_n^p &= Y_n^p - \overline{Y_n^p} \\ \Delta D_n^p &= D_n^p - \overline{D_n^p}. \end{aligned} \tag{3.13}$$

Note the covariance matrix approximations of Eq. 3.10 are equivalent to

$$\begin{aligned} C_{Y_n^p,D_n^p} &= \frac{1}{N_e-1}\Delta Y_n^p(\Delta D_n^p)^T \\ C_{D_n^p,D_n^p} &= \frac{1}{N_e-1}\Delta D_n^p(\Delta D_n^p)^T. \end{aligned} \tag{3.14}$$

Eq. 3.7 can be reformulated using Eqs. 3.13 and 3.14 as

$$\begin{aligned} y^a_{n,j} &= y^p_{n,j} + \Delta Y^p_n (\Delta D^p_n)^T \left[\Delta D^p_n (\Delta D^p_n)^T + (N_e - 1) C_{D,n}\right]^{-1} \delta d_{mis,n,j} \\ &= y^p_{n,j} + \Delta Y^p_n X_{n,j}, \end{aligned} \tag{3.15}$$

where the vector $X_{n,j}$ is given by

$$X_{n,j} = (\Delta D^p_n)^T \left[\Delta D^p_n (\Delta D^p_n)^T + (N_e - 1) C_{D,n}\right]^{-1} \delta d_{mis,n,j}. \tag{3.16}$$

Eq.3.15 can be rewritten as

$$y^a_{n,j} = y^p_{n,j} + \sum_{j'=1}^{N_e} \left[y^p_{n,j'} - \overline{y^p_n}\right] x_{n,j,j'}, \tag{3.17}$$

where, $x_{n,j,j'}$ is the j'th component of the vector $X_{n,j}$. The last equation indicates that the analyzed ensemble member is a linear combination of the predicted ensemble members. Therefore, the analysis has N_e degree of freedom to match the measured data. This might cause problems when we assimilate seismic data because it is difficult to match N_d data with only N_e degree of freedom where $N_d >> N_e$. It is also difficult to store and invert the matrix $C_{D^p_n,D^p_n} + C_{D,n}$ if the number of data is large.

For the pluri-Gaussian geological model, the EnKF state vector contains a pair of (z_1, z_2) values in each gridblock so that the facies boundaries can be moved through analysis to match the observed data. The initial ensemble is guaranteed to honor the hard data as described above. The standard EnKF update, however, cannot guarantee that an analyzed model still honors the hard data. We use the adjustment scheme shown in the following subsection to guarantee that the analyzed model still satisfies the hard data.

3.1.1 Honoring Hard Data When Assimilating Dynamic Data

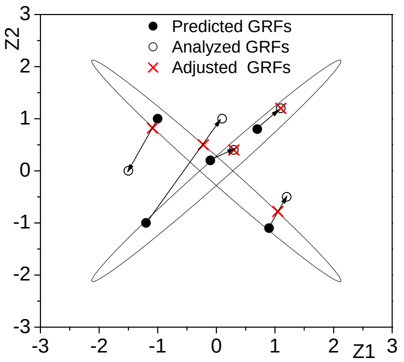


Figure 3.1: Example of Gaussian random fields before and after analysis.

Fig. 3.1 shows an example of the channel model with five wells located in the channel facies according to the hard data. As we can see, the solid symbols shows the Gaussian random field values in the predicted model which satisfies hard data, and the open circles are those in the analyzed model in which three out of five wells moved out of the channel. There are $N_a = N_w \times N_z$ pairs of (z_1, z_2) Gaussian random field values at the well gridblocks, where N_w is the number of wells, and N_z is number of vertical layers (in the channel example, $N_z = 1$). All (z_1, z_2) values need to honor the observed facies after analysis.

A subvector of any column vector $A = [a_1, a_2, \cdots, a_N]^T$ can be defined with a given list of entries, i.e., $I = \{i_1, i_2, \cdots, i_M\}$ ($\{1 \leq i_j \leq N, j = 1, 2, \cdots, M\}$). The subvector has M components: $A_b = [a_{i_1}, a_{i_2}, \cdots, a_{i_M}]^T$. Suppose $y^p_{b,n,j}$ is a subvector of $y^p_{n,j}$, and $y^a_{b,n,j}$ is the subvector of $y^a_{n,j}$, and they share the same list of entries. The entry list in these two subvectors could be (1) one component; (2) all components; (3) the Gaussian random field values at the well locations or (4) the model parameters and primary variables in a specific grid block. Here, we define another formulation of the analysis equation. According to Eq. 3.15, the analysis can be applied to part of the state vector, i.e.

$$y^a_{b,n,j} = y^p_{b,n,j} + \sum_{j'=1}^{N_e} \left[y^p_{b,n,j'} - \overline{y^p_{b,n}} \right] x_{n,j,j'}, \tag{3.18}$$

where $\overline{y^p_{b,n}}$ is the subvector of $\overline{y^p_n}$ with the same entry list as $y^p_{b,n,j}$ and $y^a_{b,n,j}$. The subvector $y^a_{b,n,j}$ obtained from Eq. 3.18 is the same one obtained from $y^a_{n,j}$ which is computed from Eq. 3.15 or Eq. 3.7 because the coefficients of the vector $X_{n,j}$ are used the same way no matter whether we are updating the whole vector or a subvector.

Define $y^p_{b_w,n,j}$ as a vector containing all pairs of predicted Gaussian random field values at the well gridblocks from the jth realization at the nth data assimilation step. This column vector has dimension N_a. Define $y^a_{b_w,n,j}$ as the corresponding vector of the Gaussian random field values at the well gridblocks obtained with the EnKF analysis step. Because the initial ensemble honors the hard data, we assume all the values in $y^p_{b_w,n,j}$ honor the observed facies, and we need to find an adjustment to $y^a_{b_w,n,j}$ so that it also honors the hard data.

If the analyzed model obtained with EnKF does not honor one or more of the hard data (facies observations at wellblocks), we will redo the analysis step completely but when we redo it, we add some pseudo-data to the analysis equation so that the analyzed (updated) Gaussian random fields honor the hard data. The procedure is as follows:

1. The expanded predicted data at the nth data assimilation step for the jth realization is defined as

   $$\tilde{d}^p_{n,j} = \left[(y^p_{b_w,n,j})^T, (d^p_{n,j})^T\right]^T, \tag{3.19}$$

   in which the predicted Gaussian random field values at the well grid blocks ($y^p_{b_w,n,j}$) are added as the predicted pseudo-data. The deviation of the predicted data from the mean prediction is defined as

   $$\Delta\tilde{D}^p_n = \tilde{D}^p_n - \overline{\tilde{D}^p_n}, \tag{3.20}$$

   where $\tilde{D}^p_n = \left[\tilde{d}^p_{n,1}, \tilde{d}^p_{n,2}, \cdots, \tilde{d}^p_{n,N_e}\right]$, and $\overline{\tilde{D}^p_n}$ is a matrix with each column equal to the mean of the column vectors in $\tilde{D}_n^p$.

2. The corresponding data mismatch term for the jth realization is expanded to

$$\delta\tilde{d}_{mis,n,j} = \left[(c_{n,j})^T, (\delta d_{mis,n,j})^T\right]^T, \tag{3.21}$$

where $c_{n,j}$ is an unknown column vector with the same dimension as $y^p_{b_w,n,j}$ and is related to the mismatch of hard data. As shown later, the $c_{n,j}$ vector will be determined so that the analyzed model satisfies the hard data.

3. The measurement error matrix is defined as

$$C_{\tilde{D},n} = \begin{pmatrix} I & \Theta \\ \Theta & C_{D,n} \end{pmatrix}, \tag{3.22}$$

where I is a $N_a \times N_a$ identity matrix.

The covariance matrices related to the expanded data are given by

$$\begin{aligned} C_{Y_n^p,\tilde{D}_n^p} &= \frac{1}{N_e - 1}\Delta Y_n^p \Delta\tilde{D}_n^{p\,T}, \\ C_{\tilde{D}_n^p,\tilde{D}_n^p} &= \frac{1}{N_e - 1}\Delta\tilde{D}_n^p \Delta\tilde{D}_n^{p\,T}. \end{aligned} \tag{3.23}$$

The EnKF analysis equation with expanded data is

$$\begin{aligned} \tilde{y}^a_{n,j} &= y^p_{n,j} + \Delta Y_n^p (\Delta\tilde{D}_n^p)^T \left[\Delta\tilde{D}_n^p (\Delta\tilde{D}_n^p)^T + (N_e - 1)C_{\tilde{D},n}\right]^{-1} \delta\tilde{d}_{mis,n,j} \\ &= y^p_{n,j} + \Delta Y_n^p \tilde{X}_{n,j}, \end{aligned} \tag{3.24}$$

where the vector $\tilde{X}_{n,j}$ is given by

$$\tilde{X}_{n,j} = (\Delta\tilde{D}_n^p)^T \left[\Delta\tilde{D}_n^p (\Delta\tilde{D}_n^p)^T + (N_e - 1)C_{\tilde{D},n}\right]^{-1} \delta\tilde{d}_{mis,n,j}. \tag{3.25}$$

Letting $\tilde{x}_{n,j,j'}$ denote the j'th component of $\tilde{X}_{n,j}$, Eq. 3.24 can be written as

$$\tilde{y}^a_{n,j} = y^p_{n,j} + \sum_{j'=1}^{N_e} \left[y^p_{n,j'} - \overline{y^p_n} \right] \tilde{x}_{n,j,j'}. \tag{3.26}$$

The EnKF update with pseudo-data for the Gaussian random fields in the well gridblocks is included in Eq. 3.26 and is written as

$$\tilde{y}^a_{b_w,n,j} = y^p_{b_w,n,j} + \sum_{j'=1}^{N_e} \left[y^p_{b_w,n,j'} - \overline{y^p_{b_w,n}} \right] \tilde{x}_{n,j,j'}. \tag{3.27}$$

We call the analysis step shown in Eqs. 3.24, 3.26 and 3.27 as analysis with adjustment. We follow three steps to perform such an analysis.

Step 1: Normal Analysis. In this step, $y^a_{n,j}$ (and $y^a_{b_w,n,j}$) are obtained from the standard EnKF analysis without expansion of the data vector. From step 1 to step 3, we consider ensemble members one by one. If the analyzed model of an ensemble member honors the hard data, we will not perform any of the following steps. If any hard data are not honored, adjustment is needed. We define $y^{adj}_{b_w,n,j}$ as the target vector after adjustment. This target vector satisfies the hard data. Our objective is to find pseudo-data ($c_{n,j}$) for the Gaussian random fields at well gridblocks so that the EnKF analysis step will give values of the Gaussian random fields at well gridblocks sufficiently close to this target vector so that hard data will be honored. There are $N_w \times N_z$ pairs of Gaussian random field values in each of the three vectors, $y^p_{b_w,n,j}$, $y^a_{b_w,n,j}$ and $y^{adj}_{b_w,n,j}$, where each pair corresponds to values at well gridblock, where we have an observed facies. Suppose $Z^p = (z^p_1, z^p_2)$ is a pair in $y^p_{b_w,n,j}$, $Z^a = (z^a_1, z^a_2)$ is the corresponding pair in $y^a_{b_w,n,j}$, and $Z^{adj} = (z^{adj}_1, z^{adj}_2)$ is the corresponding pair in $y^{adj}_{b_w,n,j}$. We define

$$Z^{adj} = (1 - \alpha) Z^p + \alpha Z^a, \tag{3.28}$$

where α is a real number between 0 and 1. Fig. 3.2 shows a example of a pluri-Gaussian model (for channels). The predicted pair is shown as a black solid circle and the analyzed pair as a black open circle. Z^{adj} represents the line segment connecting Z^p to Z^a. We wish to pick a value of α so that Z^{adj} is consistent with the facies observed at the specific wellblock under consideration. There are generally an infinite number of α's that satisfy this criterion. As discussed below, our choice is to pick an α that gives a Z^{adj} as close as possible to Z^a subject to the condition that the facies observation is honored.

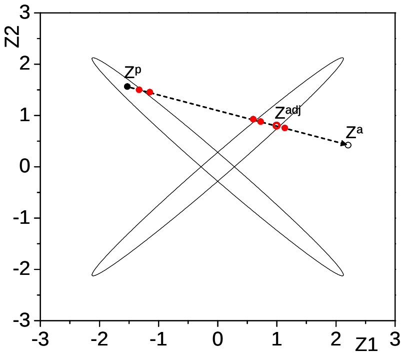


Figure 3.2: Gaussian random fields adjustment of channel case.

For concreteness, we assume that we are using two ellipses to define the truncation map as in the channel example. Also assume that we are considering a particular well gridblock where the hard datum corresponds to the observation of the channel facies. Also assume that the corresponding prediction of the pair of Gaussian random fields, Z^p, is consistent with the hard datum, but the updated pair, Z^a, is not as shown in Fig. 3.2 as an example for the channel model. We first calculate the points where the line segment connecting Z^p to Z^a intersects the truncating ellipses. We denote these points by α_i, $i = 1, 2, \cdots L$ where for the situation depicted in Fig. 3.2, L=3. Instead of using the largest value of these α_is as α in Eq. 3.28, we use a slightly different value for α to ensure that we will be able to generate pseudo-data for the pair of Gaussian random fields at well gridblocks so that the EnKF analysis step will yield an updated state vector consistent with hard data. To do so, we define $2L$ β

values as follows: $\{\beta_1 = 0.9\alpha_1, \beta_2 = \alpha_1 + 0.1(\alpha_2 - \alpha_1), \beta_3 = \alpha_2 - 0.1(\alpha_2 - \alpha_1), \beta_4 = \alpha_2 + 0.1(\alpha_3 - \alpha_2), \cdots, \beta_{2L-1} = \alpha_L - 0.1(\alpha_L - \alpha_{L-1}), \beta_{2L} = \alpha_L + 0.1(1 - \alpha_L)\}$. The red symbols in Fig. 3.2 show these β_j's. For each hard data, we set α equal to the largest β_j so that when $\alpha = \beta_j$ is used in Eq. 3.28, Z^{adj} is consistent with the hard data. Note this choice of β_j corresponds to the one that gives the Z^{adj} closest to Z^a. This procedure is used to determine all pairs in the target vector $y^{adj}_{b_w,n,j}$.

Step 2: Solve for the Adjustment. In this step, the $c_{n,j}$ term in the expanded data mismatch vector of Eq. 3.21 is determined so that the analyzed vector in Eq. 3.27 will be equal to the target vector found in step 1, i.e. $\tilde{y}^a_{b_w,n,j} = y^{adj}_{b_w,n,j}$, which will ensure that the hard data are honored. Rewrite Eq. 3.27 as

$$\tilde{y}^a_{b_w,n,j} = y^p_{b_w,n,j} + A_{n,j}\delta\tilde{d}_{mis,n,j}, \tag{3.29}$$

where $A_{n,j}$ is an $(N_a) \times (N_a + N_{d,n})$ matrix given by

$$A_{n,j} = \Delta Y^p_{b_w,n}(\Delta\tilde{D}^p_n)^T \left[\Delta\tilde{D}^p_n(\Delta\tilde{D}^p_n)^T + (N_e - 1)C_{\tilde{D},n}\right]^{-1},$$

which can be partitioned as

$$A_{n,j} = [A1_{n,j}, A2_{n,j}]\,. \tag{3.30}$$

Here, the sub-matrix $A1_{n,j}$ contains the first N_a columns of $A_{n,j}$, and the sub-matrix $A2_{n,j}$ contains the last $N_{d,n}$ columns. Using Eq. 3.30, Eq. 3.29 can be rewritten as

$$\tilde{y}^a_{b_w,n,j} = y^p_{b_w,n,j} + A1_{n,j}c_{n,j} + A2_{n,j}\delta d_{mis,n,j}, \tag{3.31}$$

where

$$\delta d_{mis,n,j} = d_{uc,n,j} - d^p_{n,j}.$$

Setting $\tilde{y}^{a}_{b_w,n,j} = y^{adj}_{b_w,n,j}$ in Eq. 3.31, we can find $c_{n,j}$ by solving

$$(A1_{n,j})c_{n,j} = (y^{adj}_{b_w,n,j} - y^{p}_{b_w,n,j} - A2_{n,j}\delta d_{mis,n,j}). \tag{3.32}$$

This method assumes that pairs of Gaussian random field values at different well locations are uncorrelated because the pairs in the target vector are obtained independently. This assumption requires that the distance between any two wells exceeds the correlation lengths of the two Gaussian random fields and that the Gaussian random fields are uncorrelated vertically.

Step 3: Redo the Analysis. After the pseudo-data vector $c_{n,j}$ is found, the adjusted state vector can be obtained using Eq. 3.24. Because we redo the analysis with the expanded data, each data point is effectively integrated only once.

3.2 Quantitative Comparisons of Results from Each Cases

Although general qualitative conclusions can be obtained by examining figures which show data matches, future predictions and a comparison between the true reservoirs and realizations obtained at the end of data assimilation, it is useful to also try to define some quantitative means for comparison. These quantitative results will be shown in tables.

Here, we use the two pluri-Gaussian models as examples. For both models, the last data assimilation time is at 360 days, and predictions are made forward from 360 days to 660 days. In addition, the final ensemble is rerun from time zero to 660 days to compare with these procedures as a consistency check [77]. For quantitative comparison, we partition observed data and predictions into nine subsets. Each of the first five subsets, $k = 1, \cdots 5$, pertains to bottomhole pressure at a well, and each of the last four subsets, $k = 6, 7, 8, 9$, represents watercut at one well. For comparison of predictions, we denote the true prediction for the kth subset of data by $d_{k,true,1}, d_{k,true,2}, \cdots, d_{k,true,N_p}$, where these N_p data are uniformly distributed (daily) during the 300 days (so $N_p = 300$) of prediction subsequent to the time period where data are assimilated, i.e., $d_{k,true,l}$ represents the true prediction of subset k at the $360 + l$ days. Because the time steps in the simulator are non-uniform, we obtain these daily predictions by linearly interpolation if necessary. Using the same method, the predictions of the subset k from the jth conditional ensemble member (simulated from time zero) can also be written as $d^p_{k,j,1}, d^p_{k,j,2}, \cdots, d^p_{k,j,N_p}$, for $j = 1, 2, \cdots, N_e$, where $d^p_{k,j,l}$ represents the prediction of subset k at the $360+l$ days for the jth conditional realization. The ensemble average of the predicted data from the conditional ensemble for the subset k is denoted by $d^p_{k,avg,1}, d^p_{k,avg,2}, \cdots, d^p_{k,avg,N_p}$, for $k = 1, 2, \cdots 9$, where

$$d^p_{k,avg,i} = \frac{1}{N_e} \sum_{j=1}^{N_e} d^p_{k,j,i}, \text{ for } i = 1, 2, \cdots, N_p.$$

As the measurement errors for pressure and watercut data are quite different, we put data mismatches on a dimensionless basis by normalizing the predictions of the kth type by $\sigma^2_{k,last}$, which denotes the variance of the measurement error of the kth type at the last data assimilation time (360 days). The total prediction error is then defined by

$$O^p_{TOT} = \sqrt{\frac{1}{9N_p} \sum_{k=1}^{9} \sum_{i=1}^{N_p} \frac{(d_{k,true,i} - d^p_{k,avg,i})^2}{\sigma^2_{k,last}}}. \tag{3.33}$$

The corresponding average prediction error for pressure and watercut data are defined, respectively, by

$$O^p_{BHP} = \sqrt{\frac{1}{5N_p} \sum_{k=1}^{5} \sum_{i=1}^{N_p} \frac{(d_{k,true,i} - d^p_{k,avg,i})^2}{\sigma^2_{k,last}}}, \tag{3.34}$$

and

$$O^p_{WCT} = \sqrt{\frac{1}{4N_p} \sum_{k=6}^{9} \sum_{i=1}^{N_p} \frac{(d_{k,true,i} - d^p_{k,avg,i})^2}{\sigma^2_{k,last}}}. \tag{3.35}$$

We also make comparisons between the true model and the ensemble of conditional realizations. Although the Gaussian random fields are the model parameters, our objective is to generate reasonable facies distributions, and, thus, we make comparisons on the basis of the facies occupying a gridblock in the true case and the ensemble of realizations of facies on the gridblock.

Suppose the true facies distribution is $f_1, f_2, \cdots, f_{N_g}$, where N_g is the number of gridblocks and f_i denotes the facies in the ith gridblock. The facies distribution of the jth ensemble member is denoted by $f_{1,j}, f_{2,j}, \cdots, f_{N_g,j}$, where $f_{i,j}$ is the facies of the ith gridblock for the jth ensemble member. We define the model mismatch at the ith gridblock for the jth ensemble member as $\delta m_{i,j} = 0$ if $f_{i,j} = f_i$, and $\delta m_{i,j} = 1$ otherwise. Then the model mismatch error is defined by

$$O_m = \frac{1}{N_e \times N_g} \sum_{j=1}^{N_e} \sum_{i=1}^{N_g} \delta m_{i,j}. \tag{3.36}$$

We also compare data mismatches obtained from the predictions based on running the simulator from time zero with each ensemble member obtained at the final data assimilation step (360 days). Let N_d denote the total number of data that are assimilated using realizations of the observed data obtained by adding noise to the observations, where these perturbed observations are denoted by $d_{uc,j,1}, d_{uc,j,2}, \cdots, d_{uc,j,N_d}$, for $j = 1, 2, \cdots, N_e$, where N_d is the number of production data assimilated over all the data assimilation steps. The corresponding predicted data generated by running the reservoir simulator from time zero using the jth conditional ensemble member as input is denoted by $d^p_{j,1}, d^p_{j,2}, \cdots, d^p_{j,N_d}$, and $j = 1, 2, \cdots, N_e$. The average data mismatch is then defined by

$$O^d_{TOT} = \sqrt{\frac{1}{N_e \times N_d} \sum_{j=1}^{N_e} \sum_{i=1}^{N_d} \frac{(d_{uc,j,i} - d^p_{j,i})^2}{\sigma_i^2}}, \tag{3.37}$$

where σ_i^2 is the variance of measurement error of the ith datum. The corresponding data mismatches for bottomhole pressure and water cut considered separately are denoted, respectively, by O_{BHP}^d and O_{WCT}^d, and are obtained by limiting the sum to only bottomhole pressure or only water cut data, respectively.

Finally, we consider the prediction errors defined as in Eqs. 3.33, 3.34 and 3.35 but based on simulating each model in the final ensemble from time zero. We calculate $O_{TOT}^{p,EnKF}$, $O_{BHP}^{p,EnKF}$ and $O_{WCT}^{p,EnKF}$, respectively, with the same formulas used to compute O_{TOT}^p, O_{BHP}^p and O_{WCT}^p, respectively, except the predictions are based on continuing simulation with the final analyzed ensemble of state.

For PUNQS3 model, we use same algorithm to compute the quantitative conclusions of data mismatches and prediction errors for bottomhole pressure, GOR and watercut. The model mismatch O_m, however, is redefined as

$$O_m = \sqrt{\frac{1}{N_e \times N_m} \sum_{j=1}^{N_e} \sum_{i=1}^{N_m} \frac{(m_{i,j} - m_{i,true})^2}{\sigma_{m,i}^2}}, \tag{3.38}$$

where $m_{i,true}$ is the true value of the ith model parameter, N_m is the number of model parameters and $\sigma_{m,i}$ is the standard deviation of this model parameter computed from the initial ensemble.

3.3 Case One (Base Case): EnKF for Production Data

As the first case, only production data will be assimilated. The results will be used to identify and demonstrate the problems of EnKF in solving the history matching problems in a general way. Knowing its limitations, we will be more clear about the motivations of the subsequent studies. This is a base case because the results will be used as a reference for the other cases considered later, i.e. assimilating water breakthrough time, normal score transform applied to water and gas saturations, assimilating seismic data as additional data and iterative EnKF.

To show how EnKF works for each example, three types of predictions are presented. We show the set of predictions from the initial set of realizations, predictions forward from the last data assimilation step and predictions from time zero using the final analyzed ensemble of realizations. In all cases, a red curve represents a prediction generated from the true model, and symbols (black or red) represent the observed data. The ensemble predictions are shown as gray curves in all figures.

For some PUNQS3 model cases, the final ensemble of analyzed models are shown in 6 plots, i.e., mean and standard deviation ratio of porosity, $\ln k_h$ and $\ln k_z$. The standard deviation ratio of a property in a gridblock is the standard deviation of this property in the final ensemble divided by the standard deviation of this property in the initial ensemble. The value range of the standard deviation ratio is from 0 to 1, and lower value means smaller posterior uncertainty (larger uncertainty reduction). We will indicate the mean and the standard deviation ratio as “mean” and “STDR”, respectively.

3.3.1 PUNQS3 Example

Fig. 3.3 shows the predictions of the bottom hole pressure and water cut for producer 11. Fig. 3.4 shows the predictions of water cut for producers 1, 4, 5 and 12. Fig. 3.5 shows the predictions of GOR for producers 1, 4 and 15. Fig. 3.6 shows the predictions of field cumulative gas and water; Fig. 3.7 repeats the true model shown in Chapter 2. Fig. 3.8 shows the estimated means and standard deviation ratios of the model fields, and Figs. 3.9 and 3.10 show the analyzed water and gas saturation distributions from two conditional realizations at 2936 days together with the water and gas saturation distributions from the true model at 2936 days.

Using Fig. 3.3 as the example, the predictions from the initial geological model (before EnKF) are shown in Figs. 3.3(a) and 3.3(d). The gray curves are the ensemble predictions obtained from the initial ensemble members. Blue curves always represent the average of these predictions, and the red curves always represent the prediction from the true case. This type of prediction evaluates uncertainty before applying EnKF to assimilate the production data.

Shown in Figs. 3.3(b) and 3.3(e) are predictions during the EnKF assimilation steps. There are some sharp changes at the times of data assimilation. Before and after these sharp changes are the predictions before and after assimilating the data at this step. The predictions after the last assimilation time represents the uncertainty of future performance obtained from the last updated ensemble of state.

Shown in Figs. 3.3(c) and 3.3(f) are predictions from the final ensemble obtained by assimilating all data (rerun from time zero). These predictions are reasonably consistent with those made forward from the last data assimilation step.

From the predicted BHP and water cut, we can see that the posterior predictions from EnKF either made forward from the final data assimilation step or rerun from time zero gives overall better data matches and lower uncertainty in predicted future performance than predictions made with the initial ensemble. The truth still falls inside the ensemble predictions. Note, the biggest difficulty is in assimilating and predicting water production. Also note, however, that producer 11 is the only well with water breakthrough during data assimilation, and even for this well, we only have two data corresponding to a water cut greater than zero. From the EnKF assimilation shown in Fig. 3.3(e), the positive water cut data were not matched well because most of ensemble members did not breakthrough and EnKF can not obtain the correct correlation between the true predictions (red curve) and state vectors. We can see that although true predicted water cut curve (red curve) still falls inside the band of prediction, it is quite close to the upper bound of the predictions. We will see some improvements when the water breakthrough time is assimilated in the later discussions. The bottom hole pressure predictions for the other wells are very similar quantitatively to producer 11; therefore we will not show them here. The water cut predictions for producers 1, 4, 5 and 12 are shown in Fig. 3.4, from which we can see that although we do not have water breakthrough at these wells during the data assimilation period, the predictions from EnKF and the final ensemble bound the prediction for the true model (red curve), and uncertainties are decreased compared to those from the initial ensemble. The GOR predictions for producers 1, 4 and 15 are shown in Fig. 3.5. The GOR for the other three producers are almost constant over the whole period of production, and the data matches are good; therefore they are not shown here.

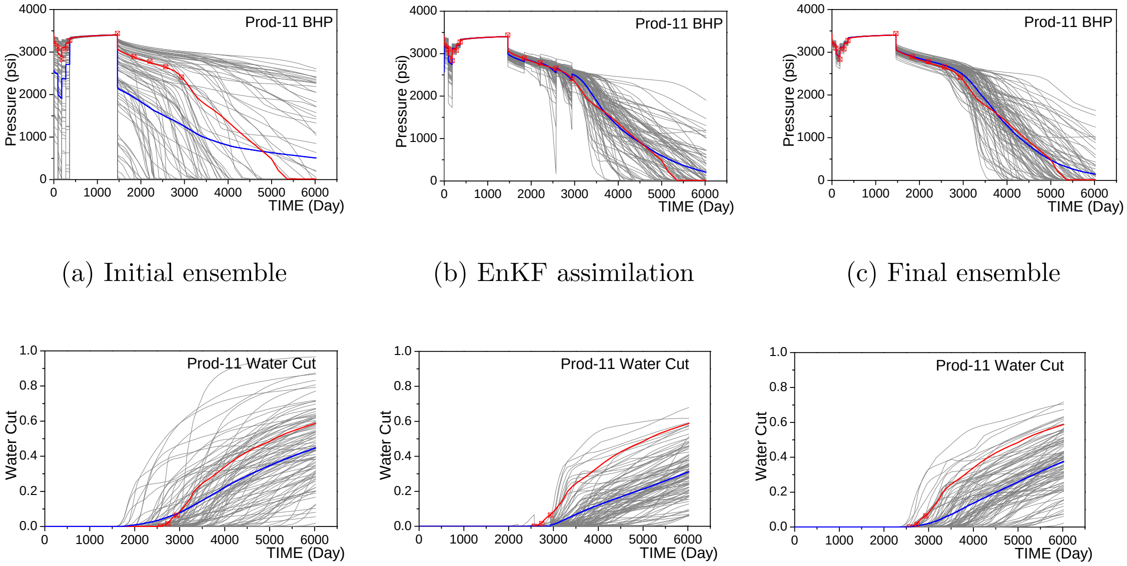
Fig. 3.6 shows the predicted field cumulative gas and water productions. The prediction of cumulative gas production is quite good because the average prediction for each well is close to the truth. The true cumulative water prediction seems quite close to the upper bound of the conditional prediction although the uncertainty is decreased.The production schedule is based on a target oil rate, which can be met for most of the production period so the oil rate and cumulative oil are very close to the truth throughout both the data assimilation and the prediction periods.

Fig. 3.8 shows the mean and standard deviation ratio (STDR) of the posterior models, i.e. porosity, $\ln k_h$ and $\ln k_z$. The following comments pertain to these results:

1. The STDR of the fields are low around the perforated locations of wells (open circles). It is especially obvious for the porosity and $\ln k_h$ fields;

2. Some high porosity and high permeability zones in the true model are recovered in each layers, but there still exist some unexpected "belts", i.e., the left top of the first layer (compare to the true model shown in Fig. 3.7);

3. The wells are not perforated at the first and second layers, but the STDR in these layers also decreased significantly (by half in some area), this might because of spurious correlation between the fields in each layer induced by limited ensemble size.



(a) Initial ensemble (b) EnKF assimilation (c) Final ensemble

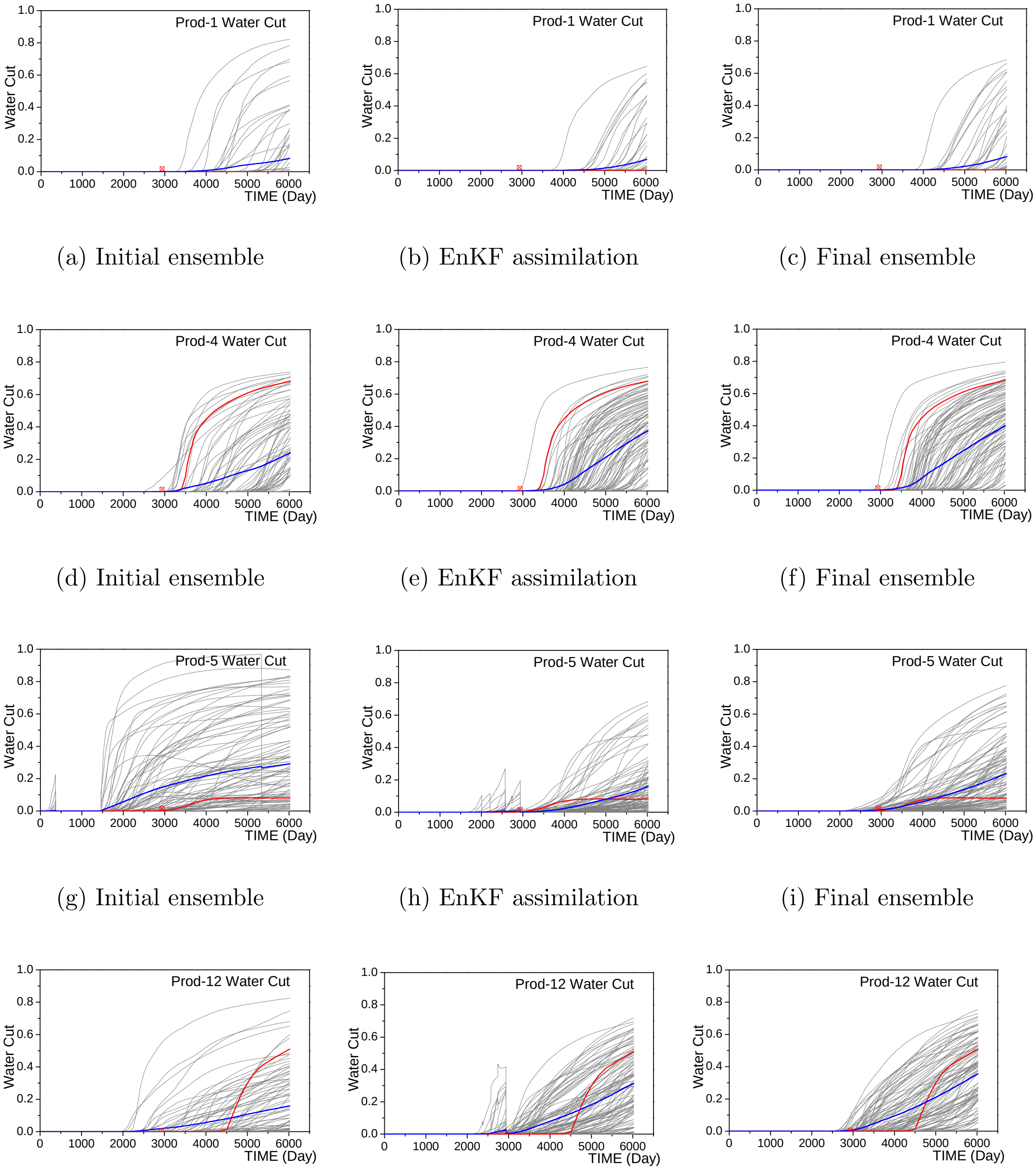
(d) Initial ensemble (e) EnKF assimilation (f) Final ensemble

Figure 3.3: Prod11 BHP and WCT (PUNQS3, EnKF of production data).

Fig. 3.9 shows the comparisons of true and two conditional water saturations distributions at 2936 days. In each of the plots we can clearly see how water influxes into the oil region. Physically plausible values of S_w are between 0.2 (irreducible water saturation) and 0.9 $(1 - S_{or})$. The white color in these analyzed saturations represents non-physical values (undershooting) of water saturations induced by the EnKF analysis, and they are truncated to the closest physical values before the next prediction, i.e., any water saturations above 0.9 are set equal 0.9, and any below 0.2 are set equal 0.2. When we apply a normal score transform to the water saturations during data assimilation, water saturation values will automatically be constrained to the upper and lower physical bounds.

3.3.2 Three Facies Example

Fig. 3.11 shows the three types of predictions for injector and producer 3 bottom hole pressure for the three facies example. For this example, the matching and prediction of pressure of all wells are quite similar; therefore, we will not show



(a) Initial ensemble

(b) EnKF assimilation

(c) Final ensemble

(d) Initial ensemble

(e) EnKF assimilation

(f) Final ensemble

(g) Initial ensemble

(h) EnKF assimilation

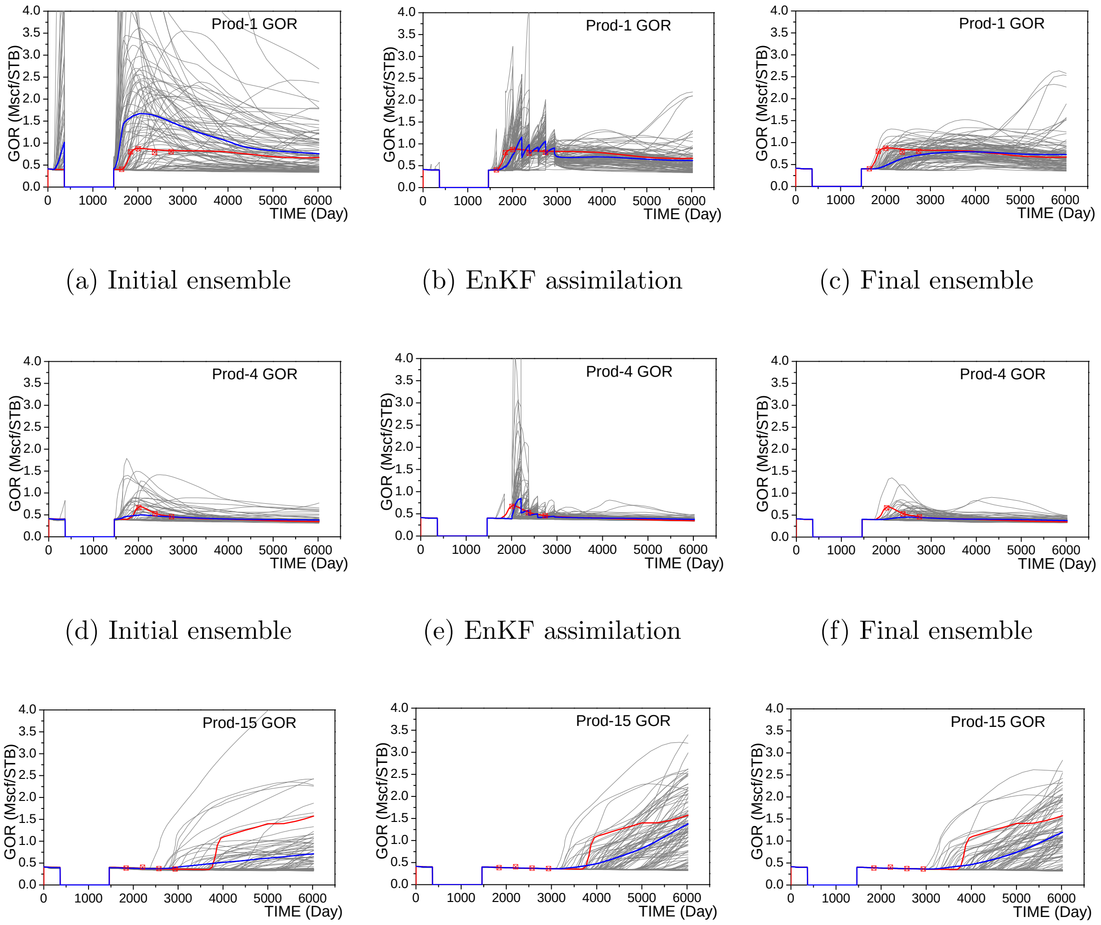
(i) Final ensemble

(j) Initial ensemble

(k) EnKF assimilation

(l) Final ensemble

Figure 3.4: WCT of Prod1, Prod4, Prod5 and Prod12 (PUNQS3, EnKF of production data).



(a) Initial ensemble

(b) EnKF assimilation

(c) Final ensemble

(d) Initial ensemble

(e) EnKF assimilation

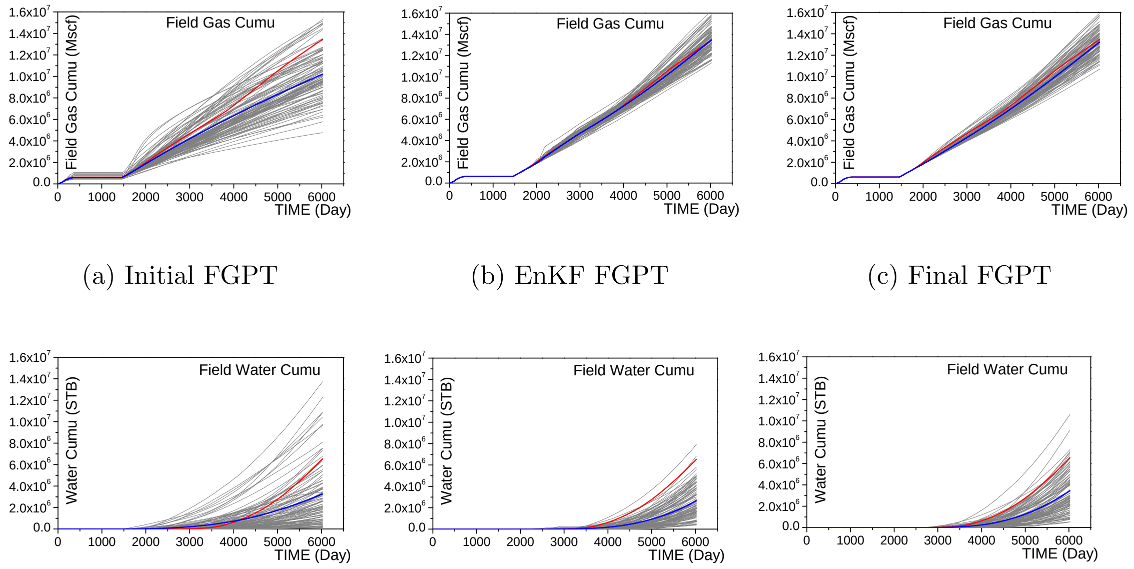
(f) Final ensemble

(g) Initial ensemble

(h) EnKF assimilation

(i) Final ensemble

Figure 3.5: Prod1, prod4 and prod 15 GOR (PUNQS3, EnKF of production data).



(a) Initial FGPT

(b) EnKF FGPT

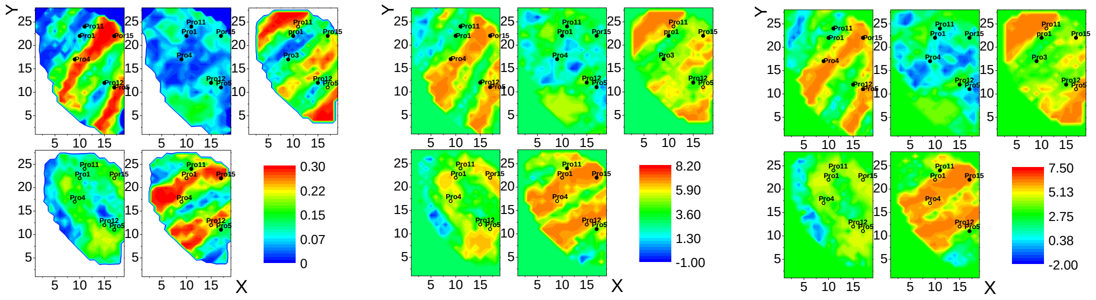
(c) Final FGPT

(d) Initial FWPT

(e) EnKF FWPT

(f) Final FWPT

Figure 3.6: FGPT and FWPT predictions (PUNQS3, EnKF of production data).

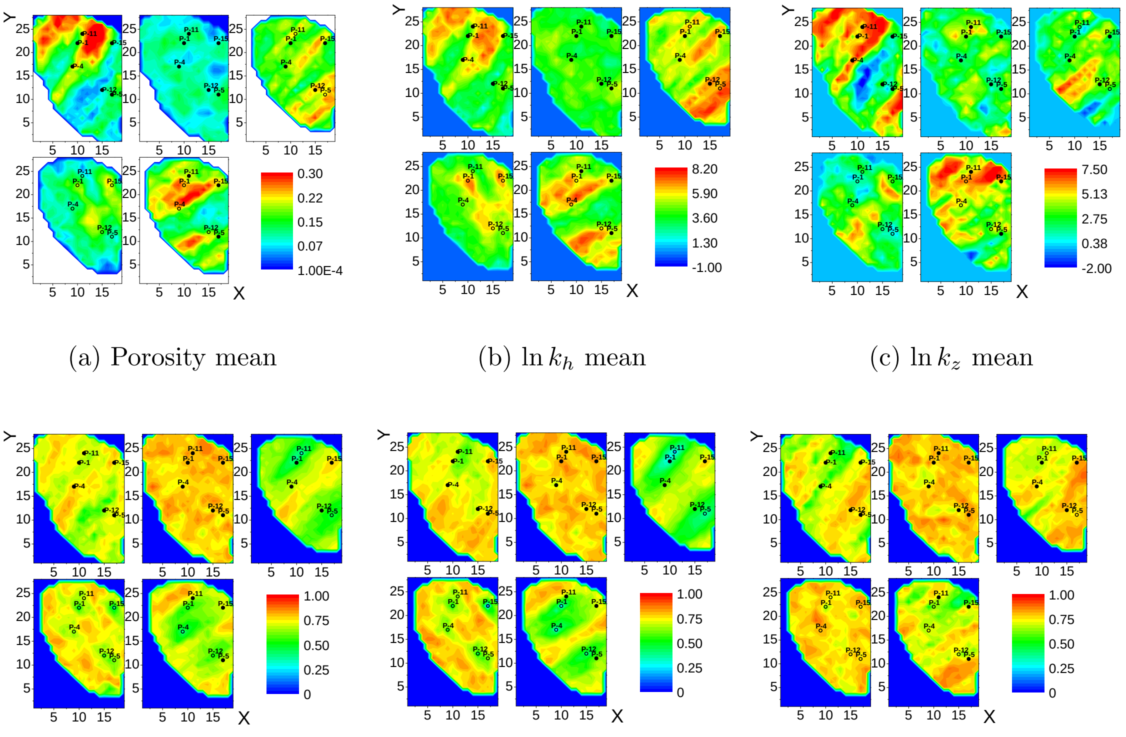


(a) Porosity

(b) $\ln k_h$

(c) $\ln k_z$

Figure 3.7: True model of PUNQS3: porosity, $\ln k_h$, $\ln k_z$.



(a) Porosity mean

(b) $\ln k_h$ mean

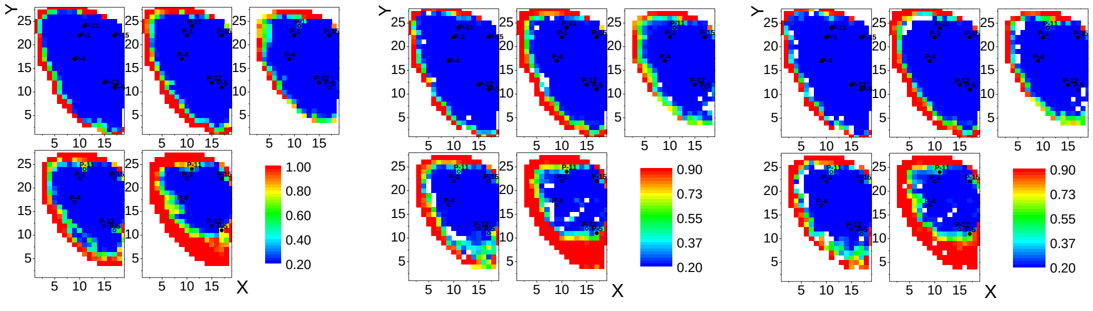
(c) $\ln k_z$ mean

(d) Porosity STDR

(e) $\ln k_h$ STDR

(f) $\ln k_z$ STDR

Figure 3.8: Mean and STDR of estimated fields (PUNQS3, EnKF of production data).

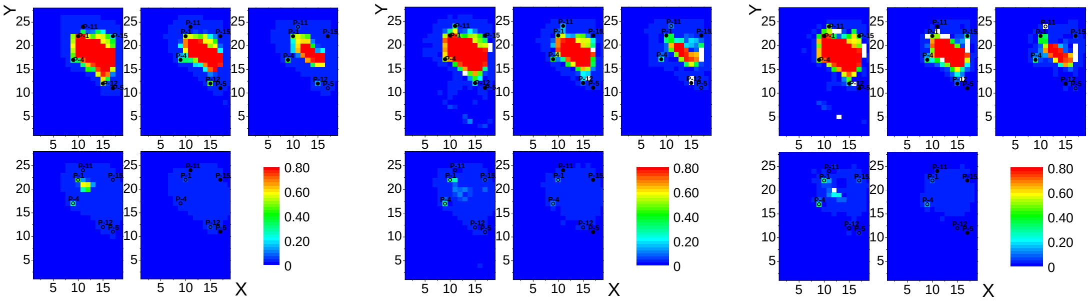


(a) True S_w

(b) Posterior S_w (En40)

(c) Posterior S_w (En60)

Figure 3.9: Water saturation at 2936 days (PUNQS3, EnKF of production data).



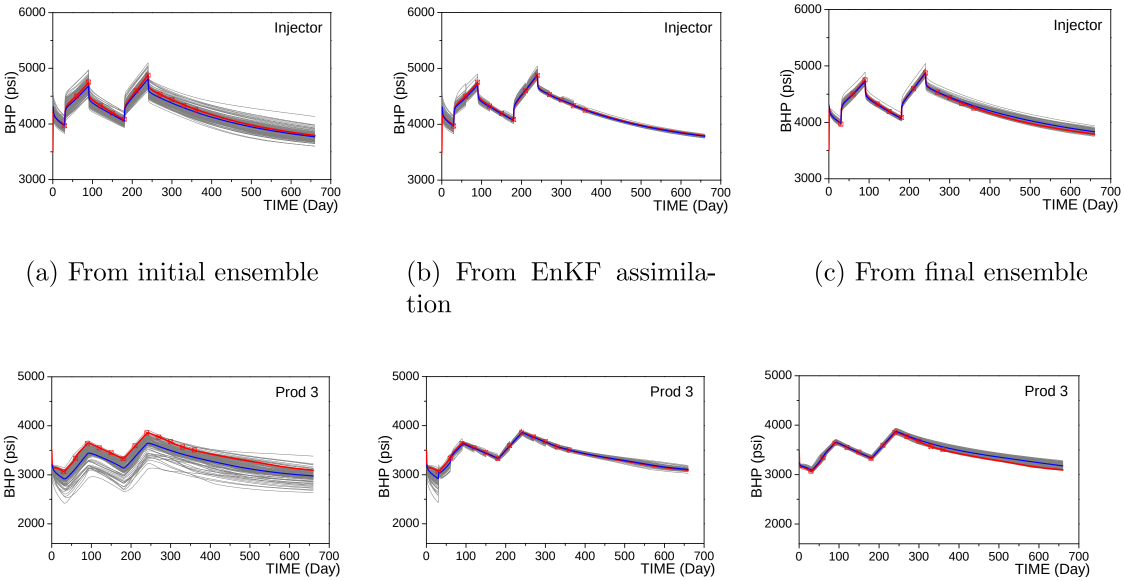
(a) True S_g (b) Estimated S_g (En40) (c) Estimated S_g (En40)

Figure 3.10: Gas saturation at 2936 days (PUNQS3, EnKF of production data).

those of the other wells. Compared to the predictions from the initial ensemble, the predictions from the final ensemble give narrower bands of uncertainty, and the average prediction is close to the truth. Although the truth is still inside the bands, the uncertainty in the prediction is quite small and there is no guarantee that we have actually characterized uncertainty.

Fig. 3.12 show the predictions of water cut for each producer. Predictions from both EnKF assimilation and the final ensemble give smaller uncertainty on the prediction, and the truth falls inside bands. The average of EnKF assimilation predictions are close to the truth, and the water cut for producer 4 is underestimated. Predictions from the final ensemble have quite good match of data except that producer 1 gives a little earlier water breakthrough. Note we obtained much better estimates of predicted water cut in this example than we obtained for PUNQS3. This is because of the fact that more water cut data were assimilated and EnKF was able to successfully assimilate these water cut data.

Fig. 3.13 gives the predicted field cumulative oil. Because the production history is constrained by liquid rates and the constraints are always satisfied during the assimilations, cumulative water production is quantitatively the same as cumulative oil production and is not shown here. The EnKF assimilation and the final ensemble predictions are slightly biased but the average prediction is quite close to the truth.



(a) From initial ensemble

(b) From EnKF assimilation

(c) From final ensemble

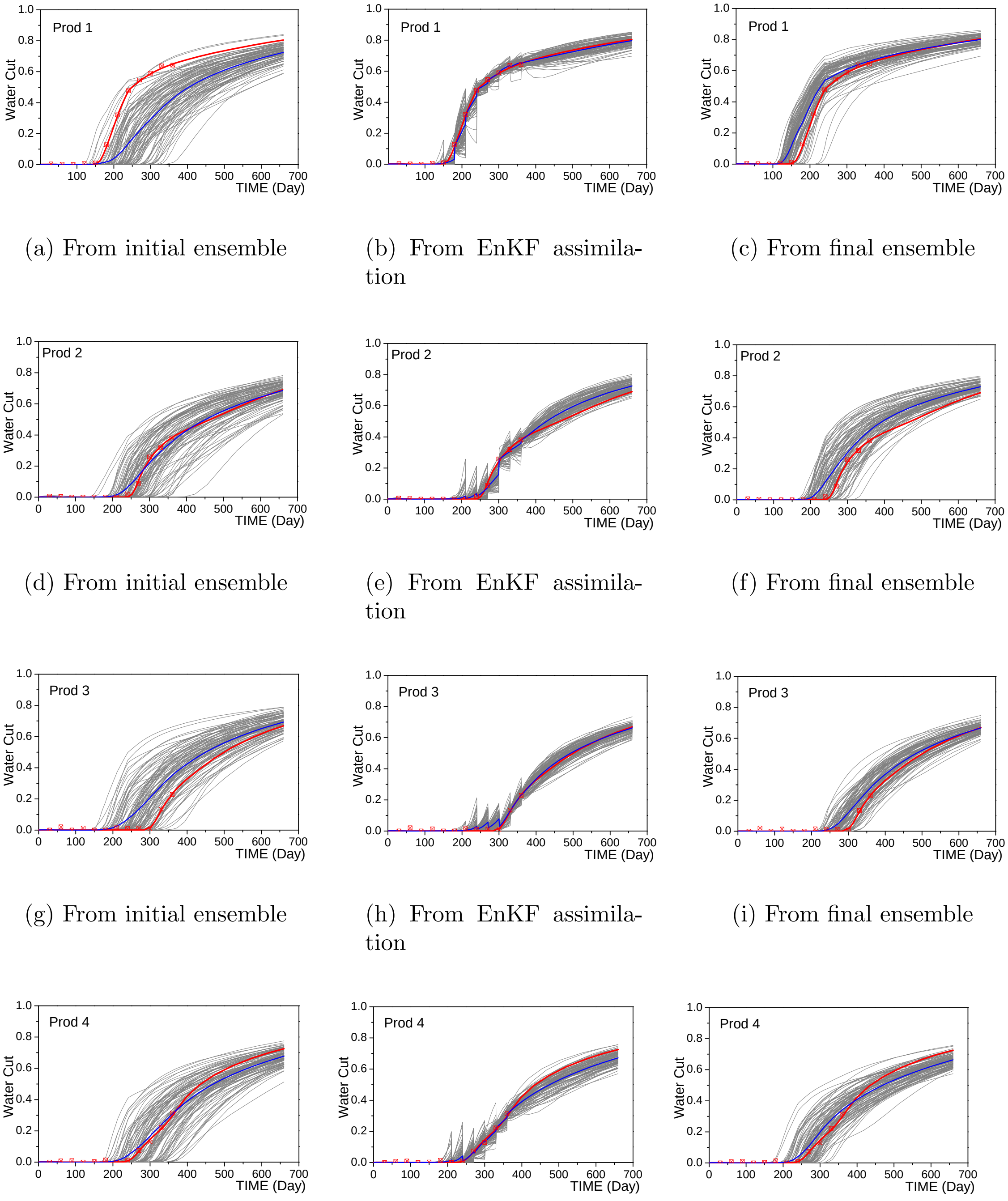
(d) From initial ensemble

(e) From EnKF assimilation

(f) From final ensemble

Figure 3.11: Injector and prod3 BHP (3 facies example, EnKF of production data).

Fig. 3.14 repeats the true facies distribution shown in Chapter 2. Fig. 3.15 shows facies distributions from two initial and conditional ensemble members. After matching the dynamic data, the conditioned facies are closer to the truth (Fig. 3.14) but still vary considerably from realization to realization. We can see that our procedure for incorporating hard data has succeeded in producing realizations which honor the hard data (facies observations at wells), see Fig. 2.22 and Table 2.3. The facies map obtained from EnKF has geological features similar to the unconditional realizations but integrating production data has resulted in a significant change in the distribution of facies. Fig. 3.16 shows the true and conditional water saturation distributions at 360 days. Compared to the true distribution, the realizations of the saturation distribution are smoother, and display less bypassed oil than the true distribution. Producer 2 and producer 3 do not have water breakthrough in the second and third layer in the truth case, but have water breakthrough in all the layers for both posterior ensemble members.



(a) From initial ensemble

(b) From EnKF assimilation

(c) From final ensemble

(d) From initial ensemble

(e) From EnKF assimilation

(f) From final ensemble

(g) From initial ensemble

(h) From EnKF assimilation

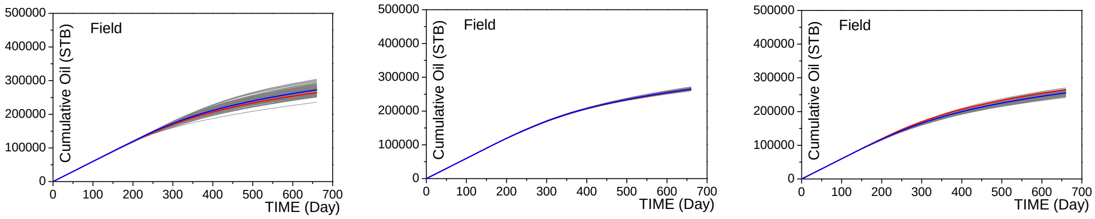
(i) From final ensemble

(j) From initial ensemble

(k) From EnKF assimilation

(l) From final ensemble

Figure 3.12: Producer WCT (3 facies example, EnKF of production data).

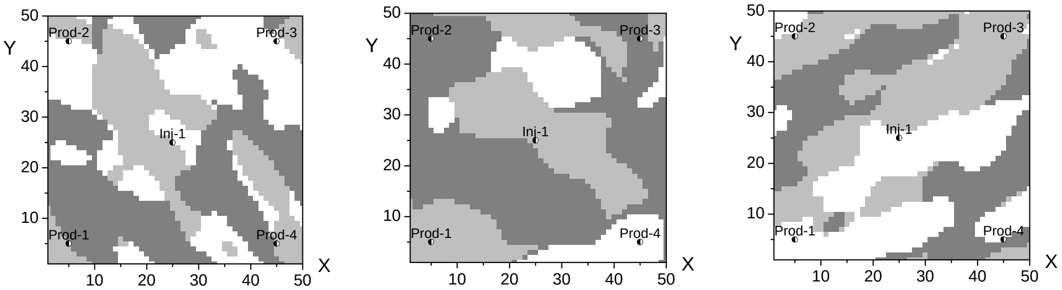


(a) From initial ensemble

(b) From EnKF assimilation

(c) From final ensemble

Figure 3.13: FOPT predictions (3 facies example, EnKF of production data).

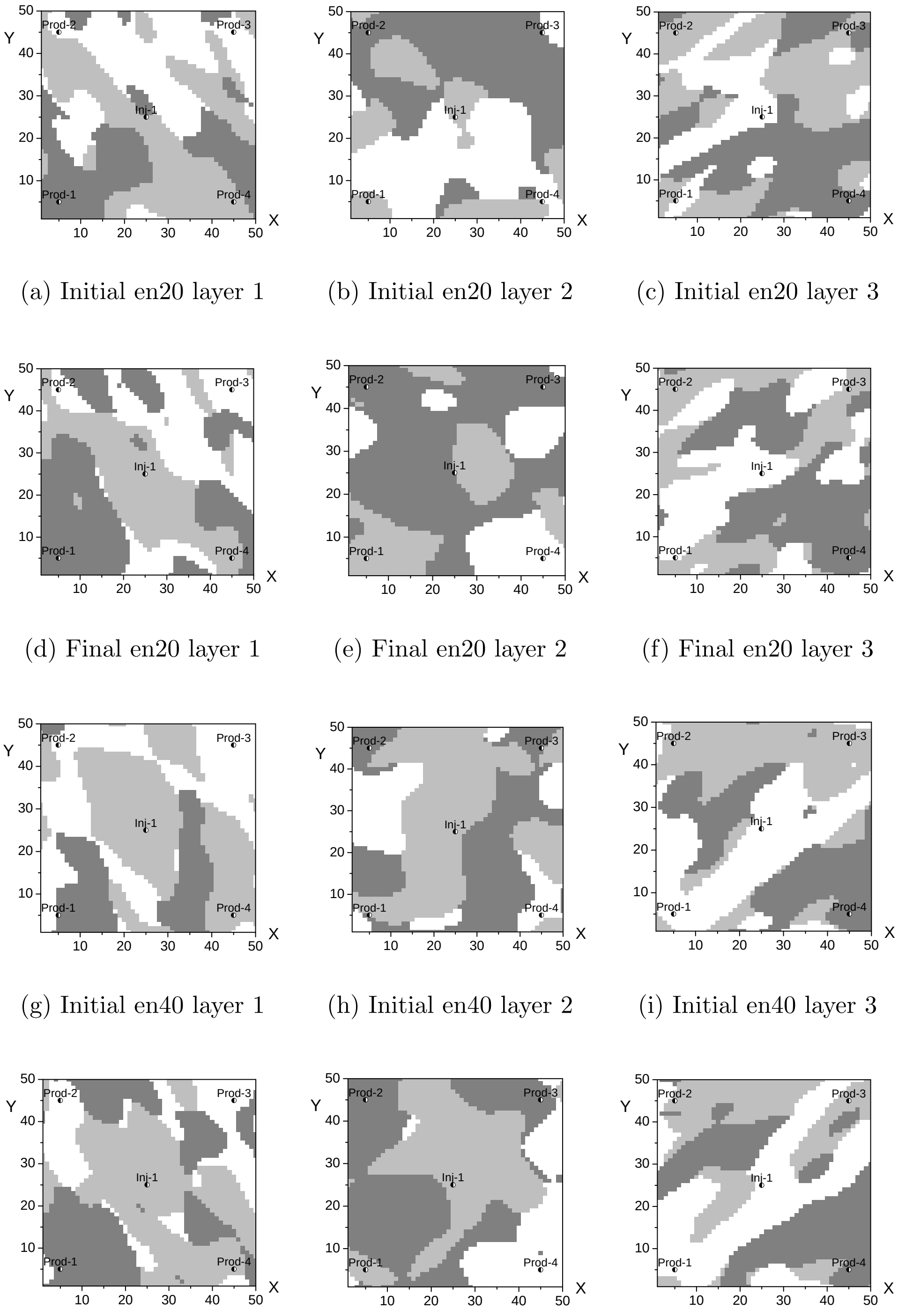


(a) Layer 1

(b) Layer 2

(c) Layer 3

Figure 3.14: Three facies example, true model.



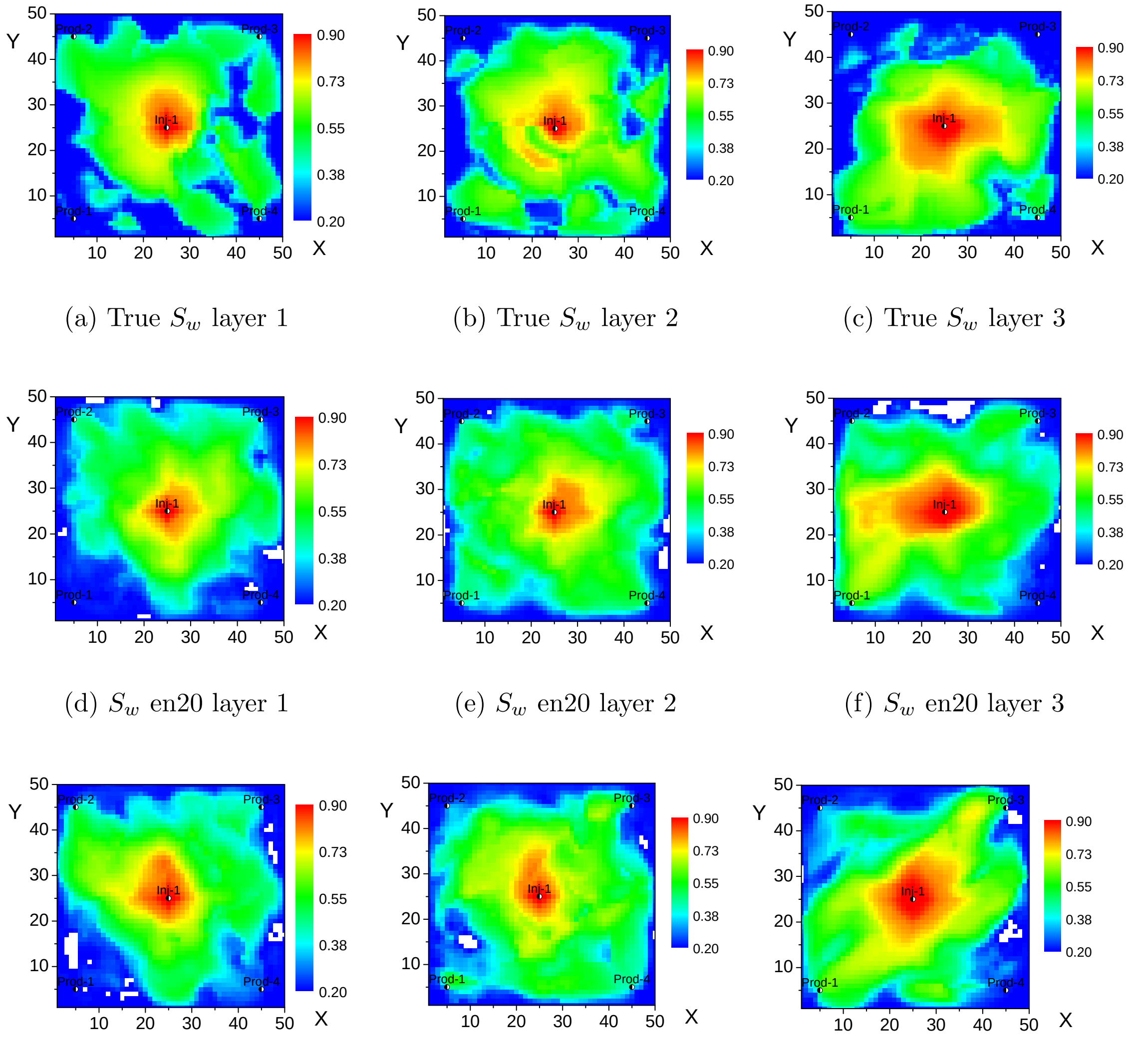
(a) Initial en20 layer 1
(b) Initial en20 layer 2
(c) Initial en20 layer 3

(d) Final en20 layer 1
(e) Final en20 layer 2
(f) Final en20 layer 3

(g) Initial en40 layer 1
(h) Initial en40 layer 2
(i) Initial en40 layer 3

(j) Final en40 layer 1
(k) Final en40 layer 2
(l) Final en40 layer 3

Figure 3.15: Initial and final facies (3 facies example, EnKF of production data).



(a) True S_w layer 1

(b) True S_w layer 2

(c) True S_w layer 3

(d) S_w en20 layer 1

(e) S_w en20 layer 2

(f) S_w en20 layer 3

(g) S_w en40 layer 1

(h) S_w en40 layer 2

(i) S_w en40 layer 3

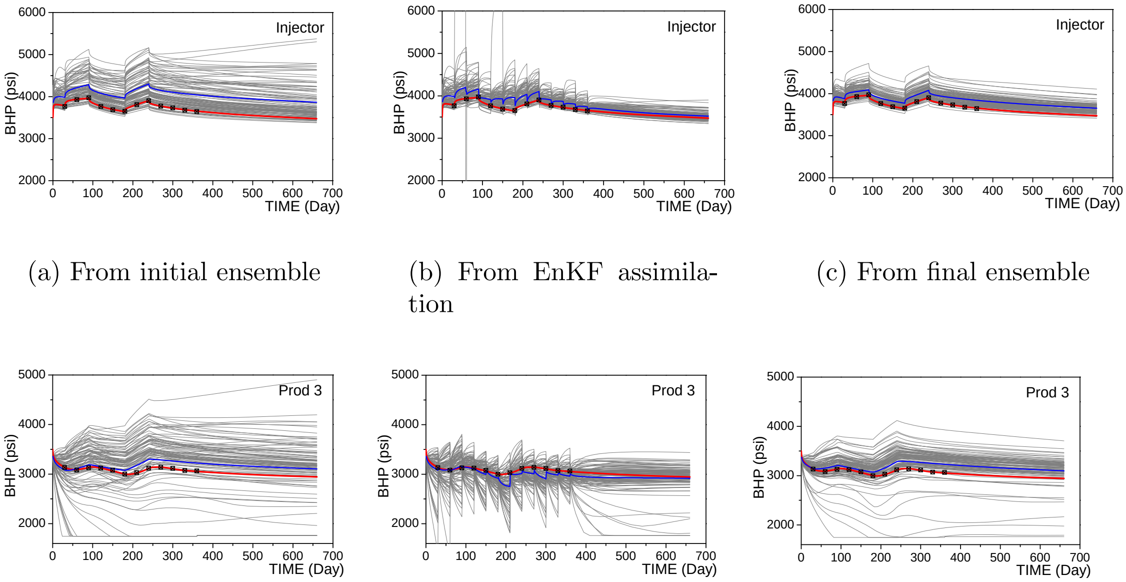
Figure 3.16: True and analyzed S_w (3 facies example, EnKF of production data).

3.3.3 Channel Example

Fig. 3.17 shows the predicted injector and producer 3 bottom hole pressure for the channel example. The predictions from the initial ensemble have much higher uncertainty than those from the three facies case, and after EnKF assimilation of production data the uncertainties are decreased. Fig. 3.18 shows the predictions of water cut data for the four producers. The average water cut predictions (blue curve) for producers 1, 2, and 4 based on running from time zero with the final ensemble gives slightly earlier water breakthrough than the truth (red curve); corresponding predictions for producer 3 give the results far from the truth. Therefore, matching water cut data is clearly more difficult than matching pressure data. Note that the predictions from the final ensemble are quite different from those made forward from the final data assimilation time, which means the basic assumption that EnKF gives results at any time are consistent with the results obtained by rerunning from time zero is not valid. Fig. 3.19 shows the three types of predictions of field cumulative oil, which also indicate inconsistent results between the two types of conditional predictions.

Although we have investigated the usefulness of matching breakthrough time before matching water cut data, intuitively one would not expect that matching water breakthrough times would yield a great improvement in results in this case, as water cut data matches were not extremely poor during data assimilation. One can see, however, that during the data assimilation for producer 1, water saturation in the well gridblock was increased at a data assimilation step, but then at the next prediction, water production dropped significantly because the EnKF analysis step only increased water saturation appropriately in the well gridblock but not in the surrounding gridblocks. Thus, it is possible that even in this case, the matching of breakthrough times may improve results.

Fig. 3.20 repeats the true facies distribution shown in Chapter 2. We will be more clear about the problem if we look at the analyzed model and observed facies. Fig. 3.21 shows, in the channel case, the facies distribution for two ensemble members and the corresponding water saturation distribution at the end of data assimilation, which are compared with the corresponding water saturation distribution generated with the true model. Fig. 3.21(a) shows a probability map of the channel facies obtained by assigning to each gridblock the probability that the gridblock is occupied by the channel facies. For a gridblock, this probability is equal to the number of ensemble members that have the channel facies in that gridblock divided by the number of ensemble members, N_e. In the plot, however, we show at each gridblock the number of ensemble member occupied by the channel facies. This probability map indicates that the connectivity between the water injection well at the center and producer 3 is somewhat lower than that between the injection well and the other three producers. This is consistent with the fact that producer 3 has overall the latest water breakthrough. This probability map also indicates we have the right connectivity but with considerable uncertainty in the location of the channels. The channel proportion is lower and the channel width in the two realizations is smaller than in the truth case (Fig. 3.20). Below the two facies distributions are shown the corresponding distribution of water saturation obtained with EnKF. Note these saturations are far too diffuse since high water saturations should pertain to the channel facies similar to the true water saturation distribution shown in Fig. 3.21(d) which was generated from the true facies map of Fig. 3.20. These results, together with the behavior of water cut data assimilation at producer 1 and 3 mentioned above, indicate that the EnKF update of saturation is not consistent with the update of the distribution of the channel facies. To obtain consistent results, as the locations of channel facies are moved, EnKF should move high values of water saturation in a similar way so that the high water saturation values are in the channel.



(a) From initial ensemble

(b) From EnKF assimilation

(c) From final ensemble

(d) From initial ensemble

(e) From EnKF assimilation

(f) From final ensemble

Figure 3.17: Injector and prod3 BHP (channel example, EnKF of production data).

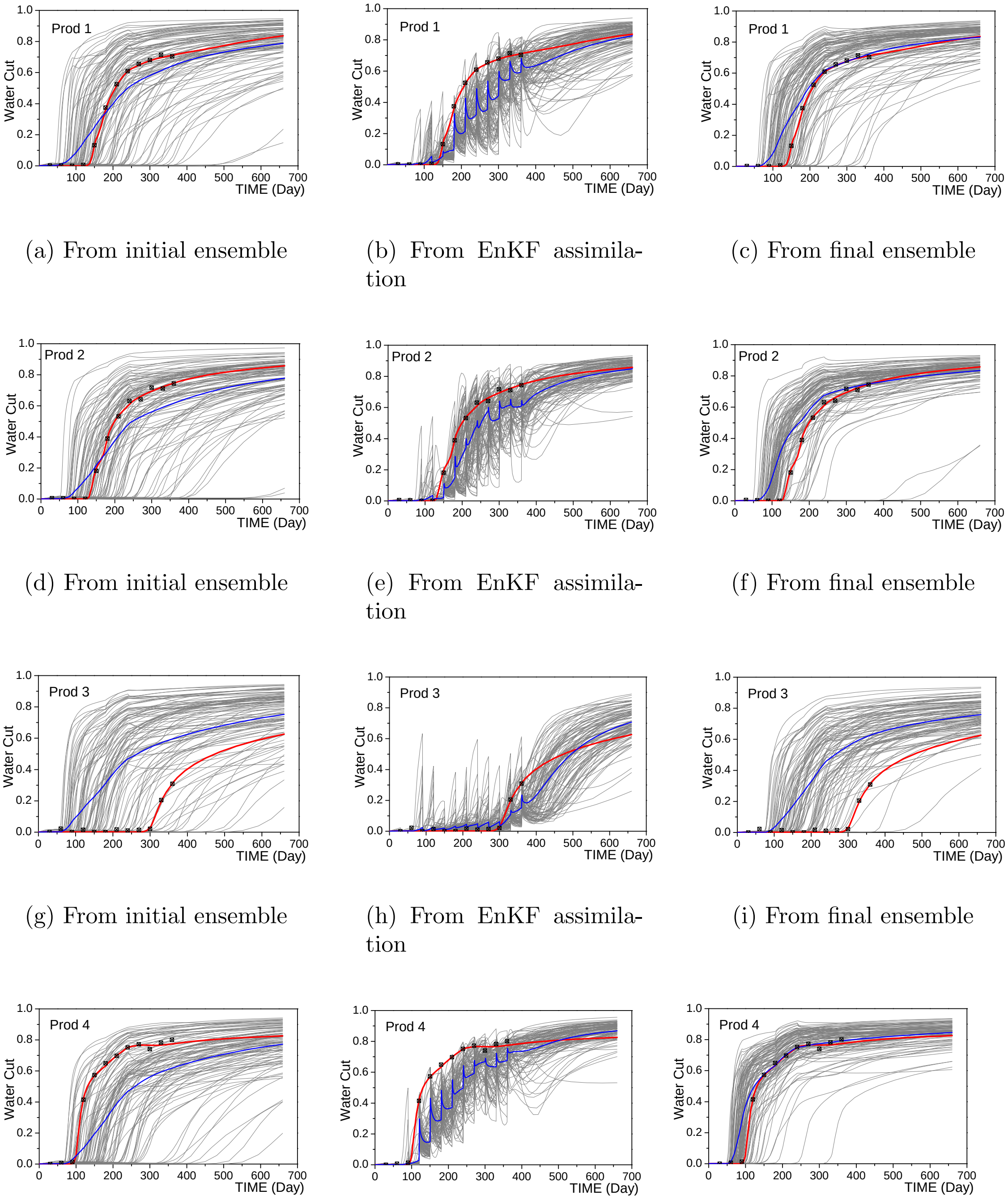
3.4 Matching Water Breakthrough Time

Motivated by results on matching water breakthrough times before matching water cut data when using gradient based methods, we consider assimilating water breakthrough time data before assimilating subsequent water cut data with EnKF.

Well	Observed BT	$\hat{t}_n$	STD
Pro-11	2700.0	2572.0	10.0

Table 3.1: PUNQS3 case water breakthrough time

To integrate water breakthrough times, we first must decide at which time step it should be assimilated. Although we have experimented with various options, in all results presented here, a water breakthrough time is assimilated at or before the earliest water breakthrough time of the ensemble predictions. Specifically, we let $t_{\text{obs},bt}$ be the observed water breakthrough time at a particular well and let $\hat{t}_n$ be the latest data assimilation time prior to $t_{\text{obs},bt}$. If no realization exhibits breakthrough at this well prior to time $\hat{t}_n$, then we assimilate data for the time of breakthrough at time



(a) From initial ensemble

(b) From EnKF assimilation

(c) From final ensemble

(d) From initial ensemble

(e) From EnKF assimilation

(f) From final ensemble

(g) From initial ensemble

(h) From EnKF assimilation

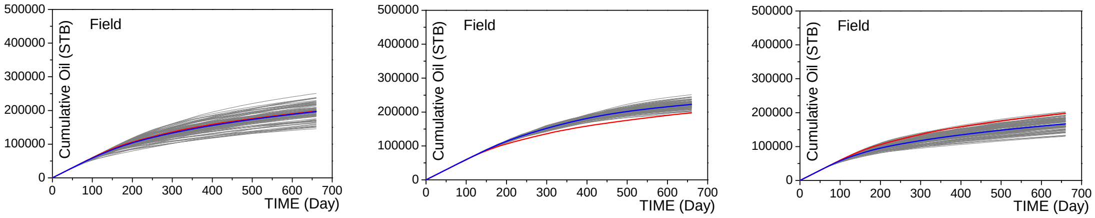
(i) From final ensemble

(j) From initial ensemble

(k) From EnKF assimilation

(l) From final ensemble

Figure 3.18: Producer WCT (channel example, EnKF of production data).



(a) From initial ensemble

(b) From EnKF assimilation

(c) From final ensemble

Figure 3.19: FOPT predictions (channel example, EnKF of production data).

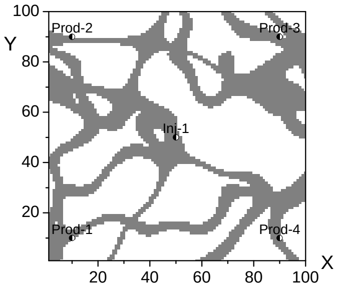
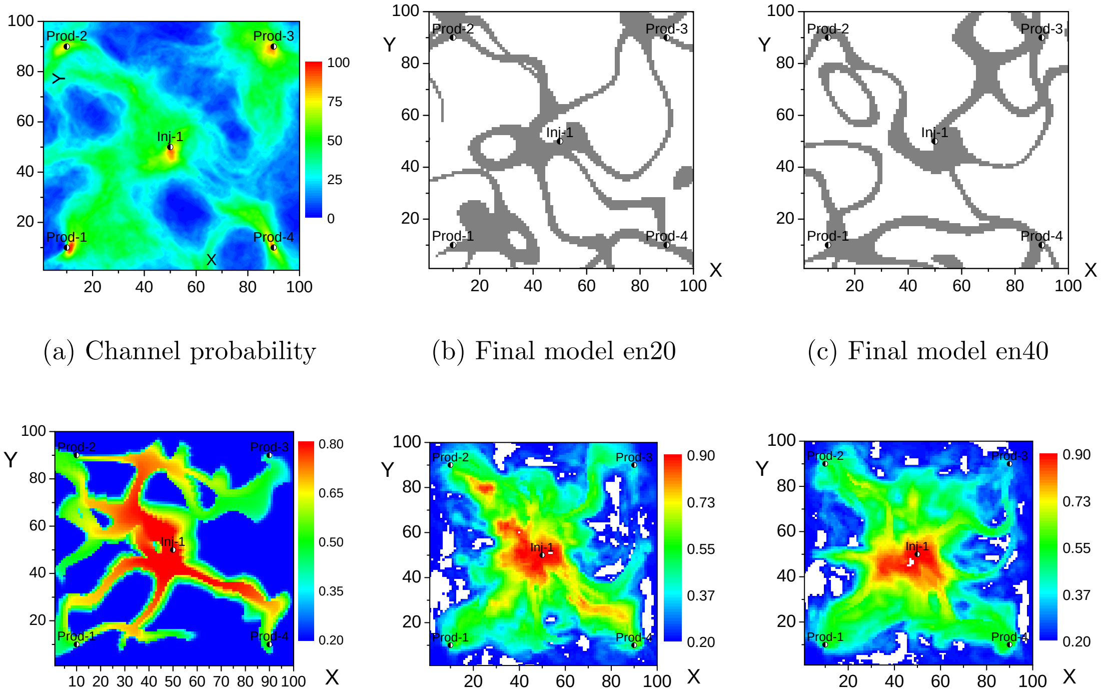


Figure 3.20: Channel example, true facies distribution.



(a) Channel probability

(b) Final model en20

(c) Final model en40

(d) S_w from true model

(e) S_w en20

(f) S_w en40

Figure 3.21: Channel probability, two posterior facies and water saturation distributions at 360 days (channel example, EnKF of production data).

Well	Observed BT	$\hat{t}_n$	STD
Prod1	135.0	120.0	5.0
Prod2	135.0	120.0	5.0
Prod3	290.0	270.0	5.0
Prod4	105.0	90.0	5.0

Table 3.2: Channel case water breakthrough time

Well	Observed BT	$\hat{t}_n$	STD
Prod1	165.0	150.0	5.0
Prod2	255.0	240.0	5.0
Prod3	315.0	300.0	5.0
Prod4	255.0	240.0	5.0

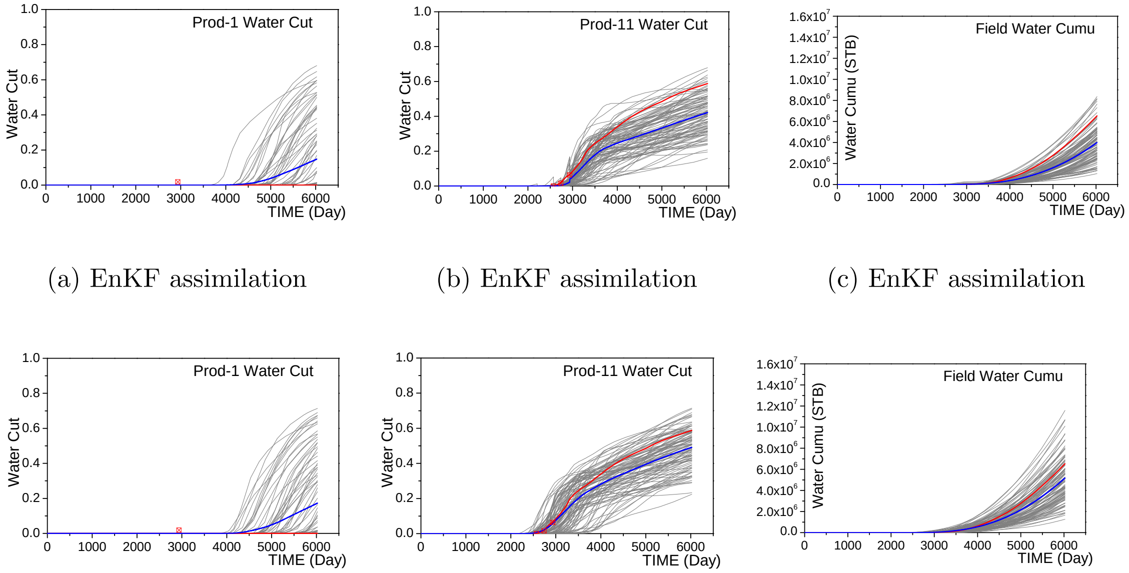
Table 3.3: Three facies case water breakthrough time

$\hat{t}_n$. Otherwise, we assimilate water breakthrough time at the earliest assimilation time at which predicted water cut data indicates that one or more ensemble members exhibits water breakthrough. Each well is dealt with individually.

To get the predicted water breakthrough time of a model from the simulator, if $t_{bt,1}$ is the first reservoir simulator time step with the water cut value wct_1 higher than a critical value (we use $wct_{cr} = 0.01$ as the "critical value"), and $t_{bt,0}$ is the previous reservoir simulator time step with water cut of wct_0, the predicted breakthrough time is computed as

$$t_{bt} = t_{bt,0} + \frac{t_{bt,1} - t_{bt,0}}{wct_1 - wct_0} \times (wct_{cr} - wct_0). \tag{3.39}$$

Tables 3.1, 3.2 and 3.3 give the observed water breakthrough times, the latest data assimilation time ($\hat{t}_n$) prior to the observed breakthrough time and the breakthrough time standard deviation of the measurement error. In the PUNQS3 case, only one well has water breakthrough during the data assimilation period, and hence we can not match water breakthrough time from the other wells. In the two pluri-Gaussian cases, the water breakthrough time is available for all wells. The



(a) EnKF assimilation (b) EnKF assimilation (c) EnKF assimilation

(d) Final ensemble (e) Final ensemble (f) Final ensemble

Figure 3.22: Prod1, 11 WCT and field cumulative water production (PUNQS3, EnKF with BT).

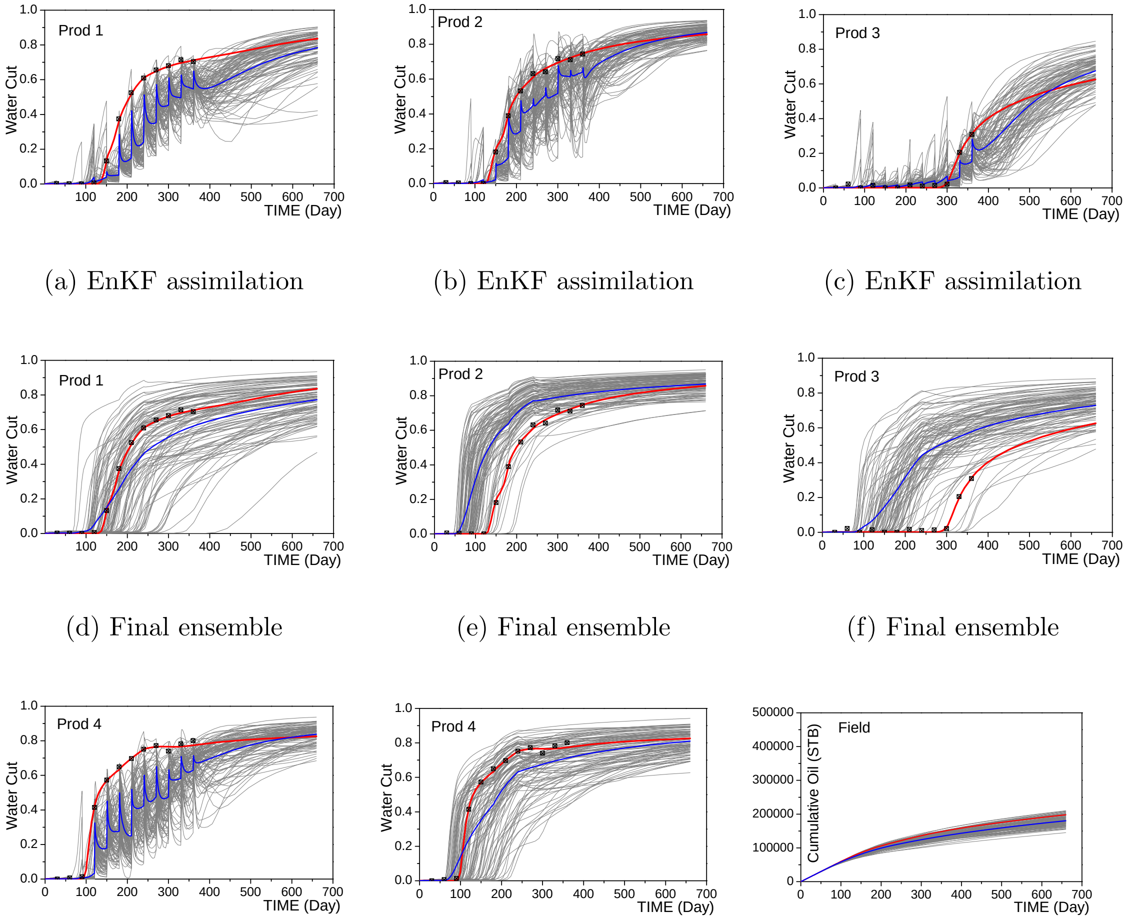
measurement error of breakthrough time is 10 days for the PUNQS3 case, and are 5 days for the two pluri-Gaussian cases.

3.5 Case Two: EnKF with Water Breakthrough Time

The difference between this case and the example considered previously is that the water breakthrough times are additional data to be assimilated. We hope this can improve the match of water cut data. As was discussed in the last case, the water cut data were not matched well in the PUNQS3 and channel examples. Therefore, this study will focus on these two examples.

3.5.1 PUNQS3 Example

In the PUNQS3 example, the water breakthrough time of producer 11 is assimilated. Because the first water breakthrough for any ensemble member occurred during the ensemble prediction period between 2008 days and 2206 days, it is assimilated at 2206 days. The observed water breakthrough time is at 2700 days. The

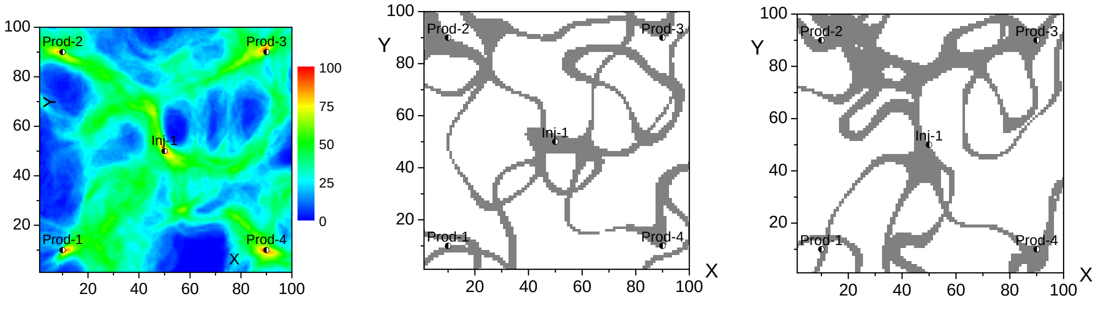


(a) EnKF assimilation (b) EnKF assimilation (c) EnKF assimilation

(d) Final ensemble (e) Final ensemble (f) Final ensemble

(g) EnKF assimilation (h) Final ensemble (i) Final ensemble

Figure 3.23: Producers WCT, and FOPT(channel example, EnKF with BT).



(a) Channel probability (b) Facies En20 (c) Facies En40

Figure 3.24: Channel probability and estimated facies(channel example, EnKF with BT).

additional data of water breakthrough time did not significantly affect predictions except for the water cut for producers 1 and 11, which are shown in Fig. 3.22 together with cumulative water production (Figs. 3.22(c) and 3.22(f)). Compared to the results without assimilating water breakthrough time, there are some considerable improvements:

1. For producer 11, the data match of water cut is better, and the predicted band is narrower, and the average prediction is closer to the truth;

2. For producer 1, even more realizations exhibit water breakthrough than was the case without matching water breakthrough time (Fig. 3.4). In the truth case no water is produced at producer 1. Producer 1 is close to producer 11, so it is not supervising that assimilating water breakthrough time at producer 11 affected predicted water cut at producer 1. It is, however, disappointing that EnKF was unable to correct these erroneous positive water cut predictions back to zero.

3. The cumulative water prediction band is narrower than was obtained form the case without assimilating water breakthrough time (Fig.3.6), and the average prediction is closer to the truth.

3.5.2 Channel Example

In the channel example, each well has predicted water breakthrough from at least one ensemble member of the initial ensemble before the first data assimilation time (30 days), and hence the four water breakthrough times are assimilated together at the 30 days. Plots in Fig. 3.23 show the predictions of water cut data from the final state of EnKF assimilation and the final ensemble for the four producers and field cumulative oil obtained by matching four additional water breakthrough times. Fig. 3.24 shows the channel PDF and two estimated facies model. Comparing these results with those obtained in the base case, we can see that the improvements are quite limited:

1. For the predictions from EnKF assimilation, no improvements can be found;
2. For the predictions from final ensemble, only the water cut of producer 1 and 4 are improved because more ensemble members get later water breakthrough. The field cumulative oil prediction from the final ensemble is also better because the uncertainty band is decreased, and the average prediction is closer to the truth;
3. The estimated channel PDF shows similar connectivity between injector and each producer for the two cases. The conditioned facies maps show up with more channels compared with the results shown in Fig. 3.21.

3.6 Normal Score Transform of Saturations

In the predicted ensemble, water saturation in each gridblock does not generally follow a Gaussian distribution. In particular, for a gridblock which contains the true water front, the ensemble of saturation values is dominated by values near either irreducible water saturation or near $1 - S_{or}$ if no gas is present; the proportion of values near the middle of this saturation range may be relatively small. Thus, the predicted ensemble of water saturation for this gridblock may be bi-modal which is inconsistent with the basic EnKF assumption that the ensemble of predictions is Gaussian. Some realizations in the analyzed (updated) ensemble may have a gridblock water saturation below S_{iw} or greater than $1 - S_{or}$ which requires truncation back to physically reasonable values. Analyzed water saturation obtained from Eqs. 3.17 and 3.18 is a linear combination of the ensemble of predicted water saturation and can result in more medium water saturations between the two peaks.

One way to attempt to eliminate non-physical values and deal with non-Gaussian problem is to transform the original variables to Gaussian variables [8].

Gu and Oliver [35] tried this approach for a one-dimensional water-flooding problem by applying a global normal score transform to saturations before applying the EnKF analysis step so that the analysis is made on variables which roughly follow a Gaussian distribution. After analysis, the corresponding updated variable are transformed back to the saturation domain. The forward and backward transform are based on same cumulative distribution function (cdf). In the work of Gu and Oliver [35], water saturations in all gridblocks share the same empirical cdf constructed from the entirety of predicted saturation values, i.e., they apply a global normal score transform. Here, we consider this same procedure but also consider a local score transform which refers to using a different cdf to define the transform on each gridblock. One potential problem with the local tranform is that the highest predicted gridblock saturation value corresponds to the highest value of the transformed variable, and hence no analyzed transform variable can transform back to a value of saturation higher than this predicted value. However, our experiments with methods for alleviating this potential problem did not improve results.

The following procedure shows the complete transform-analysis-backtransform algorithm for water saturation in a gridblock for the local normal score transform, which can be applied to gas saturation in the same way.

1. **Forward transform.** Denote water saturations of a grid block in the predicted ensemble at the nth data assimilation time as $S^p_{w,n} = \left[S^p_{w,n,1}, S^p_{w,n,2}, \cdots, S^p_{w,n,N_e}\right]^T$. The predicted cdf curve can be constructed by sorting from smallest to largest all entries in $S^p_{w,n}$ as $V = [v_1, v_2, \cdots, v_{N_e}]^T$. The predicted cdf evaluated at the ith entry of V is

$$CDF(v_i) = \frac{i}{N_e}. \tag{3.40}$$

A Gaussian cdf can be constructed from a zero mean and unit variance pdf. In the implementation, this cdf is represented by two arrays:

$$\begin{aligned} V_g &= \left[v_{g,1}, v_{g,2}, \cdots, v_{g,N_g}\right]^T \\ F_g &= \left[cdf_{g,1}, cdf_{g,2}, \cdots, cdf_{g,N_g}\right]^T, \end{aligned} \tag{3.41}$$

where V_g contains the discrete values evenly distributed in the region $[-3, 3]$, and F_g contains the corresponding Gaussian cdf values for each entries of V_g, i.e. $cdf_{g,i} = CDF_g(v_{g,i})$, where CDF_g is the Gaussian cdf. We use $N_g = 2000$ in this study. In the forward transform, the transformed data $\hat{S}^p_{w,n} = \left[\hat{S}^p_{w,n,1}, \hat{S}^p_{w,n,2}, \cdots, \hat{S}^p_{w,n,N_e}\right]$ can be found by solving:

$$CDF(S^p_{w,n,i}) = CDF_g(\hat{S}^p_{w,n,i})$$

numerically through linear interpolation of the two cdf.

2. **Analysis.** The transformed water saturation prediction in the current gridblock $\hat{S}^p_{w,n}$ can be updated using

$$\hat{S}^a_{w,n,j} = \hat{S}^p_{w,n,j} + \sum_{j'=1}^{N_e} (\hat{S}^p_{w,n,j'} - \hat{\bar{S}}^{p}_{w,n}) x_{n,j,j'}, \tag{3.42}$$

in which $x_{n,j,j'}$'s are the same as those in Eq. 3.18, and $\hat{\bar{S}}^{p}_{w,n}$ equals to the average of all columns in $\hat{S}^p_{w,n}$.

3. **Backward transform.** The updated water saturation in the current gridblock can be obtained by backtransforming $\hat{S}^a_{w,n}$ using the cdfs used in the forward normal score transform, i.e. CDF and CDF_g. Specifically, we find $S^a_{w,n,i}$ by solving for each i,

$$CDF_g(\hat{S}^a_{w,n,i}) = CDF(S^a_{w,n,i})$$

numerically with linear interpolation with the cdfs represented in Eqs. 3.40 and

3.41.

This process is applied to each gridblock.

We call the process described above as local normal score transform because the predicted cdf curves are constructed locally for each gridblock. Gu and Oliver [35] use a single predicted cdf curve for all gridblocks. Therefore, we call their approach global normal score transform. In this study, we construct this global cdf curve using the predicted water saturations in all gridblock of the N_e ensemble members. These water saturation values are sorted for smallest to largest and defined as $V^G = \left[v_1^G, v_2^G, \cdots, v_{N^G}^G\right]^T$, where N^G is N_e times the number of active gridblocks. The global cdf can be written as

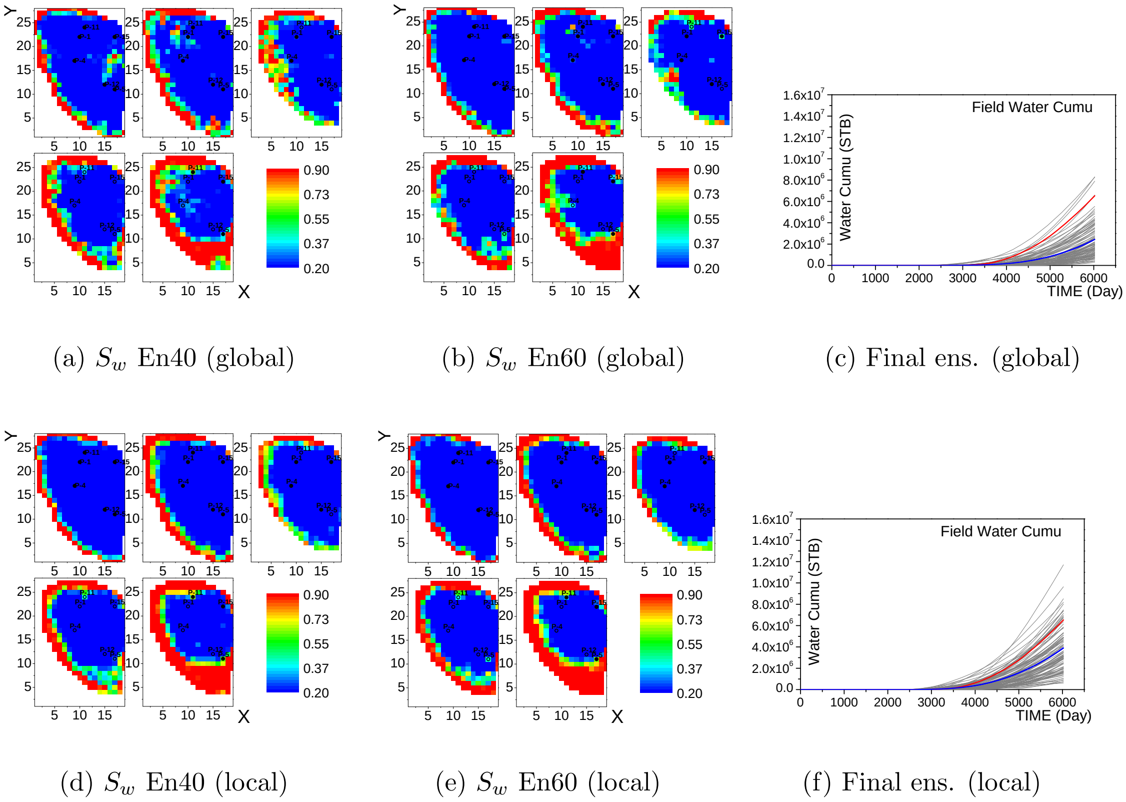
$$CDF^G(v_i^G) = \frac{i}{N^G}. \tag{3.43}$$

In the global normal score transform, same process as for local normal score transform is used except the cdf values in Eq. 3.40 for each gridblock is replaced by Eq. 3.43.

The analyzed state vectors through normal score transforms can be guaranteed to be physically feasible because the upper-bound and lower-bound of the analyzed state have to fall in those from the predicted state vector. This, however, can cause difficulty if we use the local cdf, because it will be impossible for the analyzed (updated) values of saturations to be outside the bounds of the predicted state.

3.7 Case Three: EnKF with Normal Score Transform of Saturations

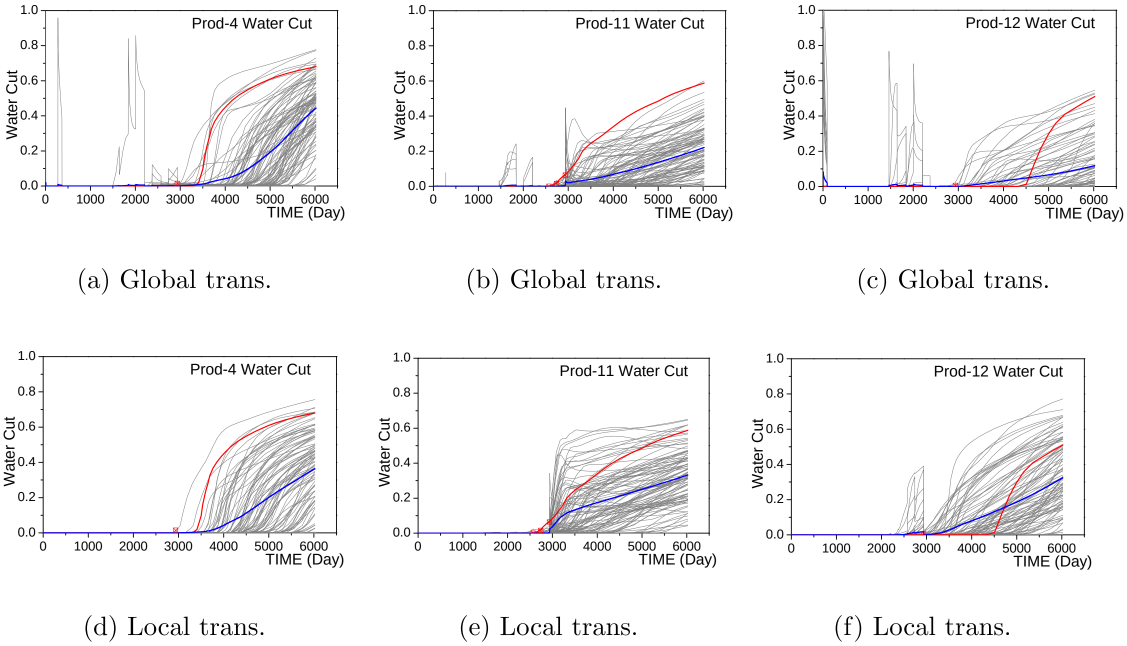
The major purpose of this case is to compare the impact of global and local normal score transform methods to the EnKF assimilation. The PUNQS3 and channel examples are studied by analyzing the water and gas saturations using the global and local normal score transform. For the channel example, an additional case of



(a) S_w En40 (global) (b) S_w En60 (global) (c) Final ens. (global)

(d) S_w En40 (local) (e) S_w En60 (local) (f) Final ens. (local)

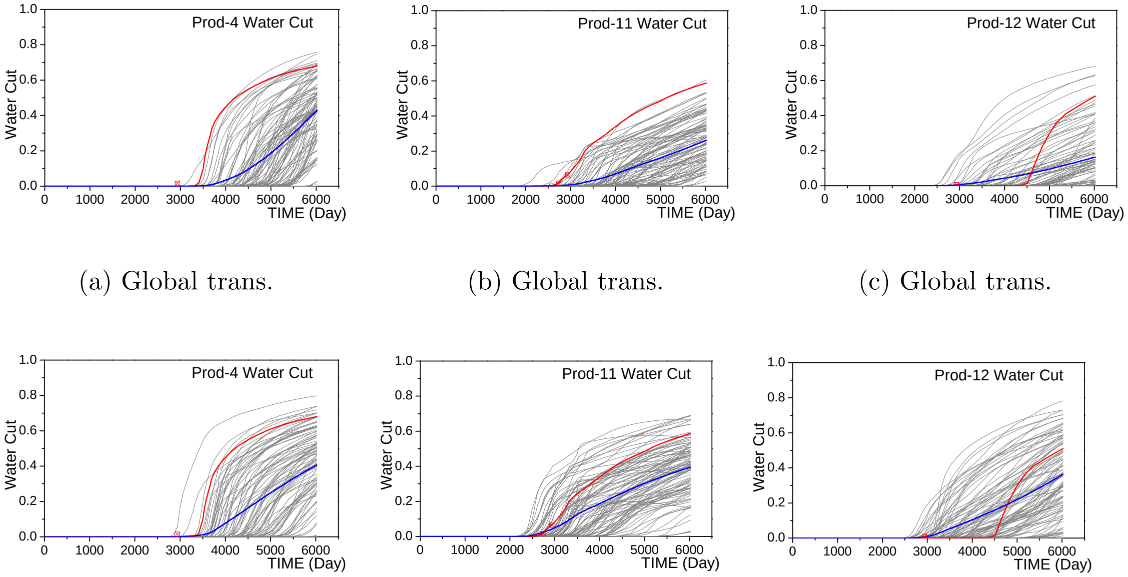
Figure 3.25: S_w at 2936 days and posterior cumulative water (PUNQS3, EnKF with normal score transform).



(a) Global trans. (b) Global trans. (c) Global trans.

(d) Local trans. (e) Local trans. (f) Local trans.

Figure 3.26: Prod4, 11, 12 WCT, EnKF assimilation predictions, comparison between local and global normal score transform (PUNQS3, assimilating production data).



(a) Global trans. (b) Global trans. (c) Global trans.

(d) Local trans. (e) Local trans. (f) Local trans.

Figure 3.27: Prod4, 11, 12 WCT, predictions from final ensemble, comparison between local and global normal score transform (PUNQS3, assimilating production data).

assimilating water breakthrough times plus local normal score transform of water saturations is also shown.

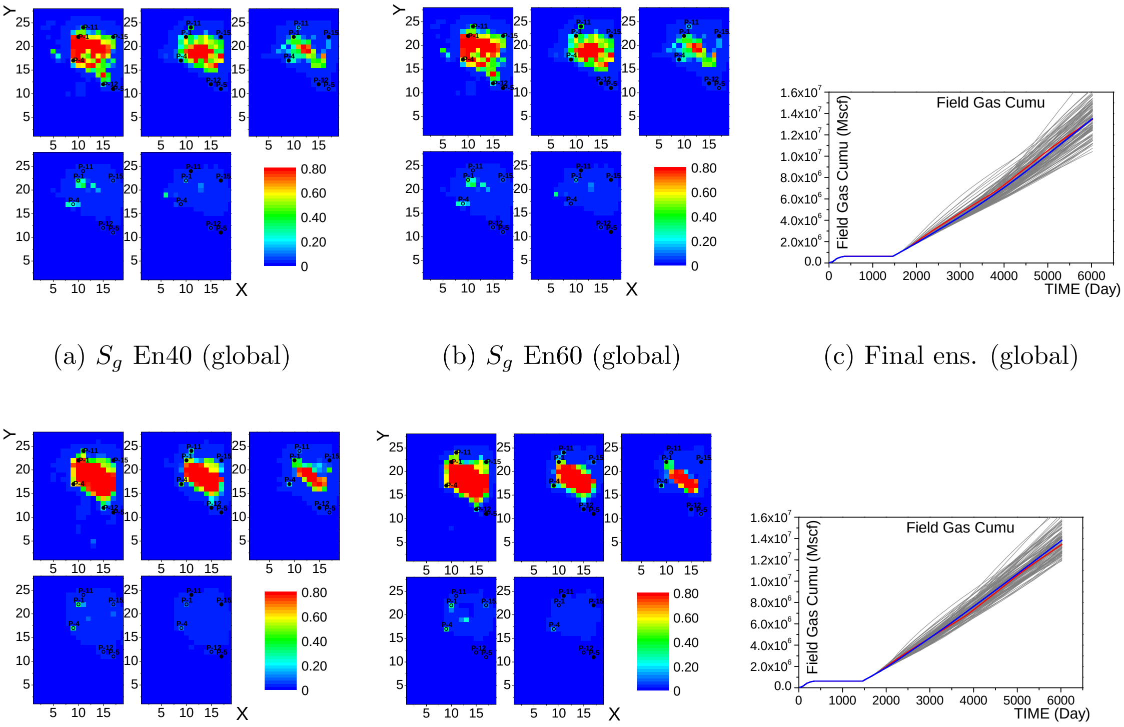
3.7.1 PUNQS3 Example

Fig. 3.25 shows the analyzed water saturation distributions for two posterior realizations and the posterior field cumulative water production when the global and local normal score transforms are applied. The analyzed water saturation from the global normal score transform shows some abnormal features:

1. The analyzed S_w distribution oscillates near the WOC, especially in the third layer for the both posterior realizations;

2. Unexpected high water saturation blocks occurs in the oil region, which is obvious in the first and second layer in the realization 40.

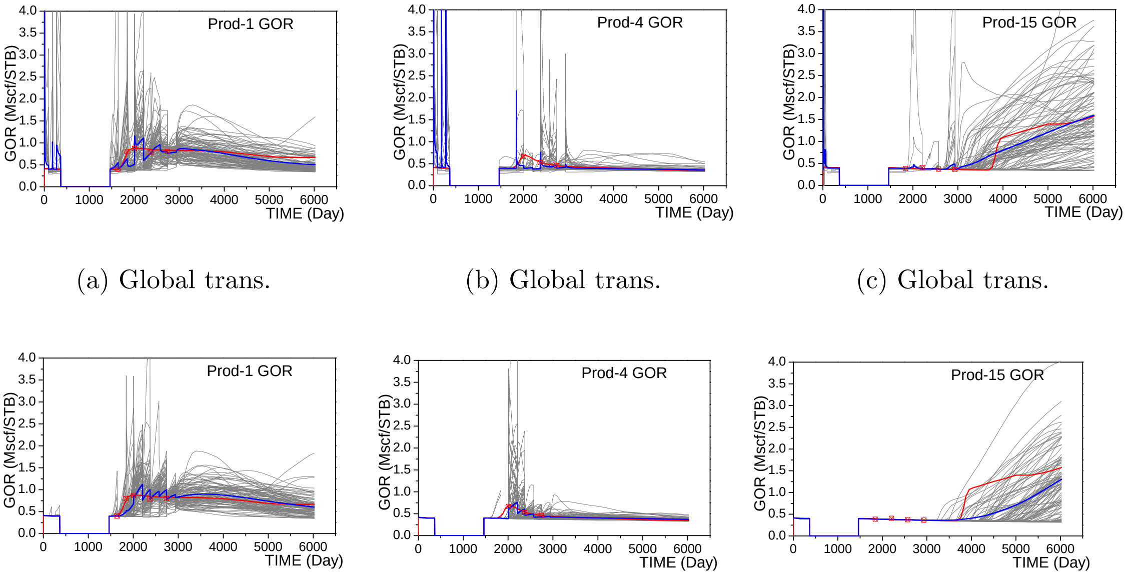
In contrast, the local normal score transform gives very stable water saturation dis-



(a) S_g En40 (global) (b) S_g En60 (global) (c) Final ens. (global)

(d) S_g En40 (local) (e) S_g En60 (local) (f) Final ens. (local)

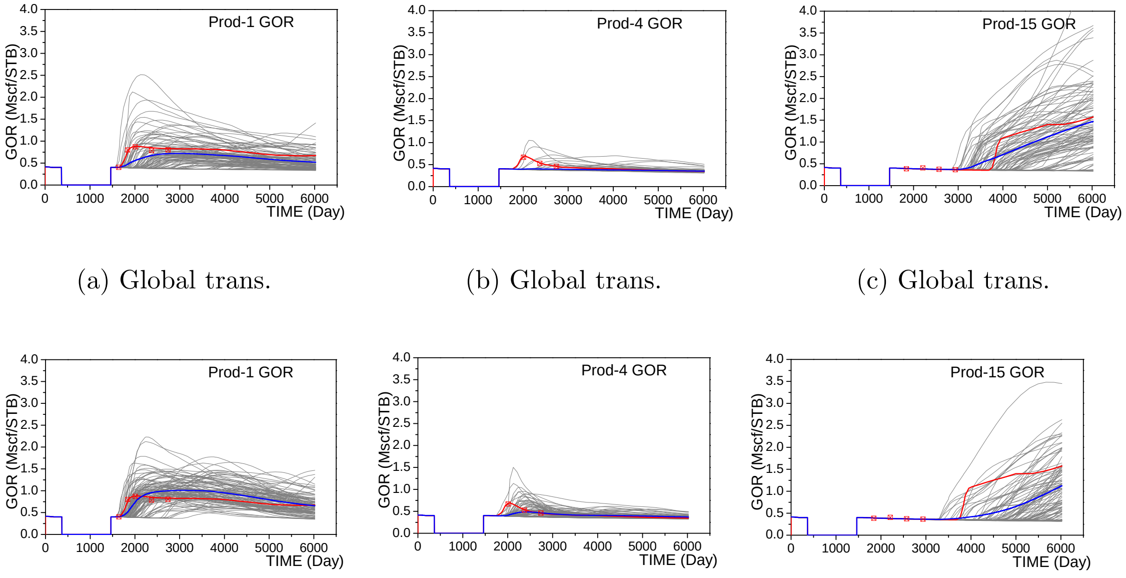
Figure 3.28: S_g at 2936 days and posterior cumulative gas (PUNQS3, EnKF with normal score transform).



(a) Global trans. (b) Global trans. (c) Global trans.

(d) Local trans. (e) Local trans. (f) Local trans.

Figure 3.29: Prod1, 4, 15 GOR, EnKF assimilation predictions, comparison between local and global normal score transform (PUNQS3, assimilating production data).



(a) Global trans.

(b) Global trans.

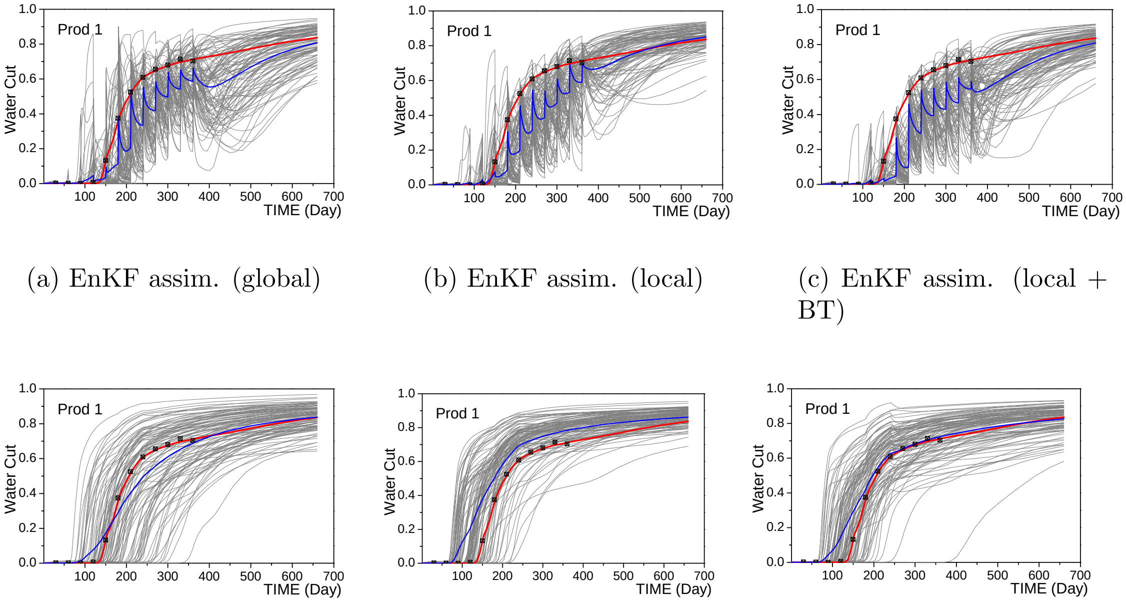
(c) Global trans.

(d) Local trans.

(e) Local trans.

(f) Local trans.

Figure 3.30: Prod1, 4, 15 GOR, predictions from final ensemble, comparison between local and global normal score transform (PUNQS3, assimilating production data).



(a) EnKF assim. (global)

(b) EnKF assim. (local)

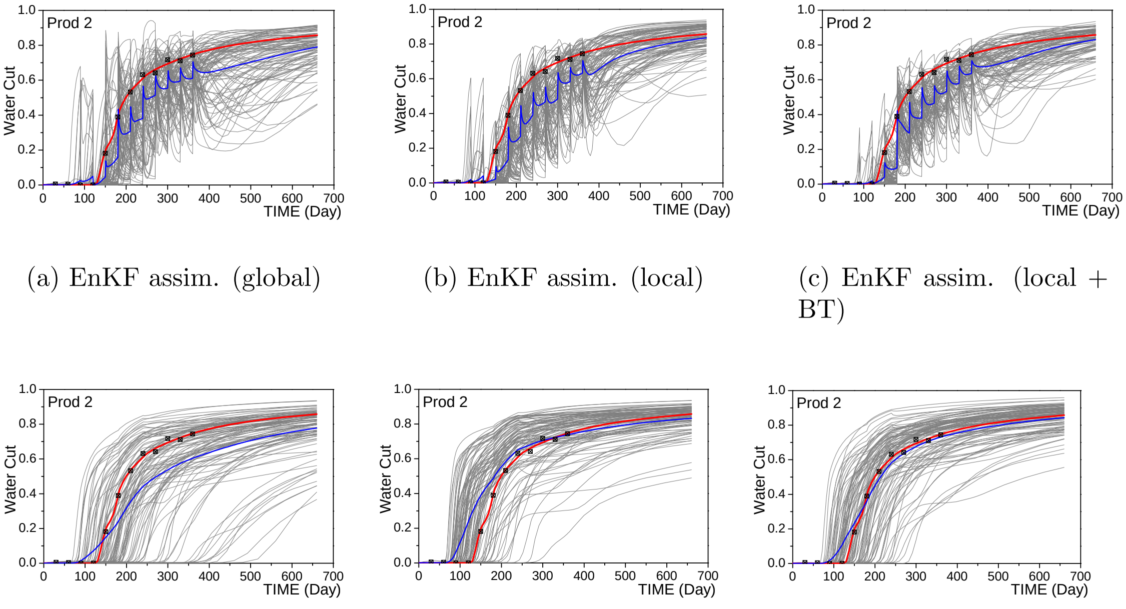
(c) EnKF assim. (local + BT)

(d) Final ens. (global)

(e) Final ens. (local)

(f) Final ens. (local + BT)

Figure 3.31: Producer 1 WCT (channel case, EnKF with normal score transform).



(a) EnKF assim. (global)

(b) EnKF assim. (local)

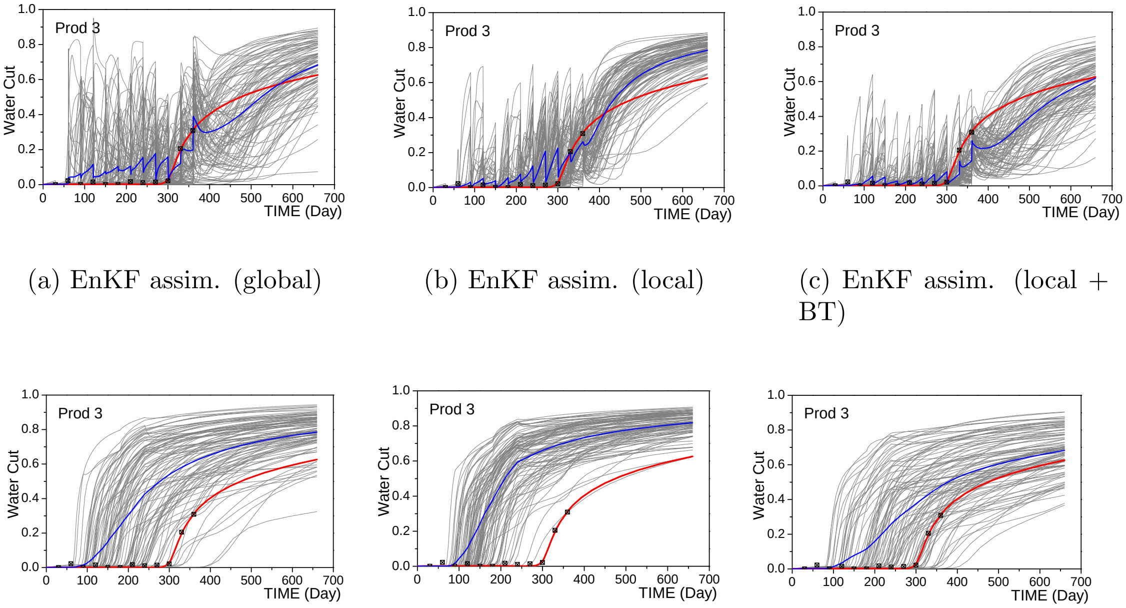
(c) EnKF assim. (local + BT)

(d) Final ens. (global)

(e) Final ens. (local)

(f) Final ens. (local + BT)

Figure 3.32: Producer 2 WCT (channel case, EnKF with normal score transform).



(a) EnKF assim. (global)

(b) EnKF assim. (local)

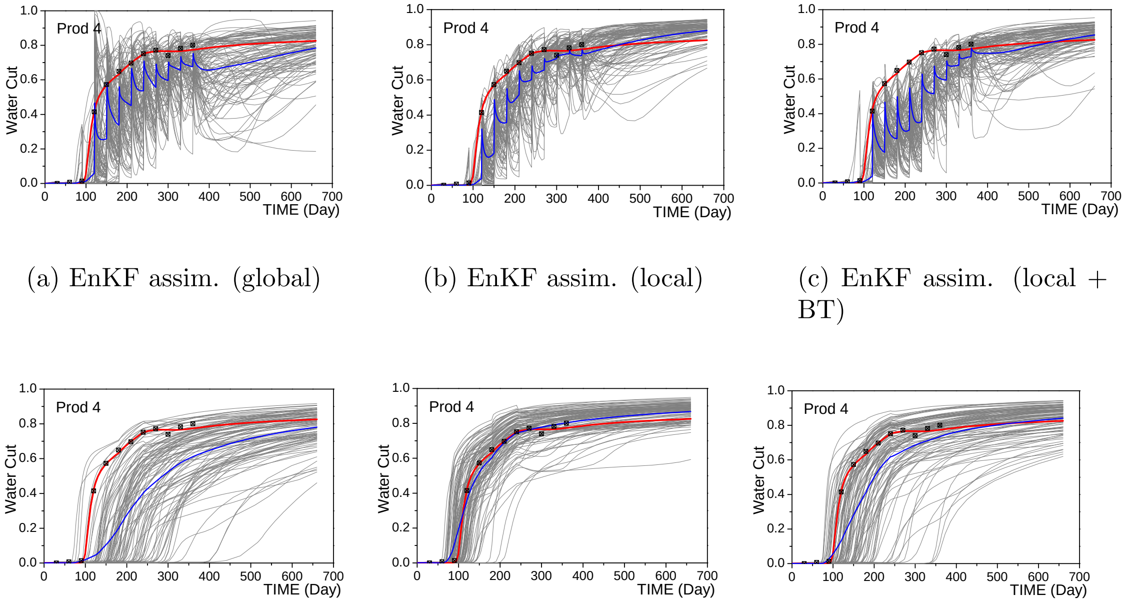
(c) EnKF assim. (local + BT)

(d) Final ens. (global)

(e) Final ens. (local)

(f) Final ens. (local + BT)

Figure 3.33: Producer 3 WCT (channel case, EnKF with normal score transform).



(a) EnKF assim. (global)

(b) EnKF assim. (local)

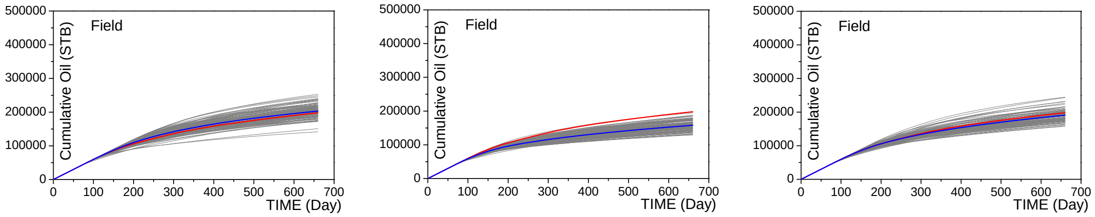
(c) EnKF assim. (local + BT)

(d) Final ens. (global)

(e) Final ens. (local)

(f) Final ens. (local + BT)

Figure 3.34: Producer 4 WCT (channel case, EnKF with normal score transform).

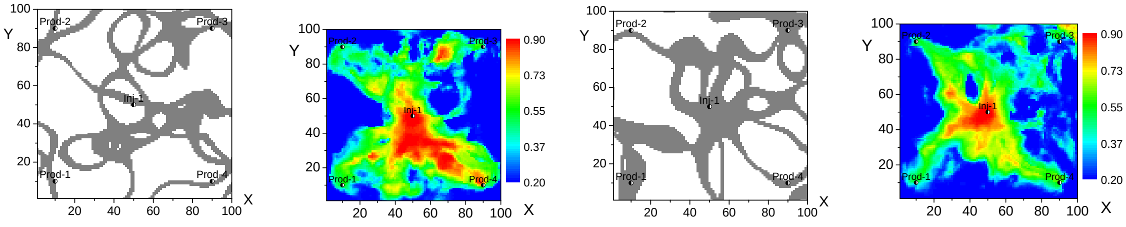


(a) Global

(b) Local

(c) Local + BT

Figure 3.35: Field cumulative oil production (channel case, EnKF with normal score transform).



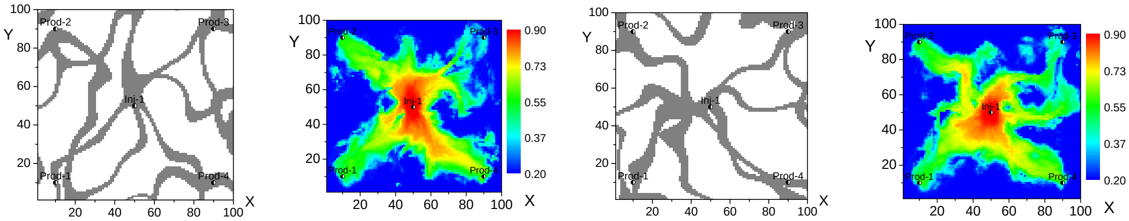
(a) Facies en20

(b) S_w en20

(c) Facies en40

(d) S_w en40

Figure 3.36: Two realizations of conditional facies and water saturations at 360 days (channel model, EnKF of production data, global normal score transform).



(a) Facies en20 (b) S_w en20 (c) Facies en40 (d) S_w en40

Figure 3.37: Two realizations of conditional facies and water saturations at 360 days (channel model, EnKF of production data, local normal score transform).

tributions. The predicted cumulative water from final ensemble is slightly better than that from the global normal score transform, and is also better than that from the case when the transforms are not used (Fig. 3.9), i.e., the average prediction is closer to the true prediction.

Figs. 3.26 shows the predictions forward from the data assimilation and rerun from time zero for producers 4, 11 and 12 water cut data from the case of global normal score transform, and Fig. 3.27 shows the corresponding predictions from the case of local normal score transform. The local normal score transform gives smoother predictions and better estimate of truth than does the global normal score transform. The predictions from the local normal score transform are also better than those from the base case (Fig. 3.4), especially for producers 11 and 12 in both EnKF assimilation and predictions from final ensemble.

Fig. 3.28 shows the corresponding results of the conditioned gas saturation distributions and field cumulative gas production from global and local normal score transforms. We can see that there are also more oscillations in the gas saturations from the global normal score transform. The global and local normal score transforms give similar predicted cumulative gas production.

Figs. 3.29 and 3.30 show the prediction of GOR for producers 1, 4 and 15 from EnKF assimilation and final ensemble for both cases. Here, the global normal score transform results in more accurate predictions of GOR for producer 15 than does the local normal score transform. The two methods give similar predicted GORs for the other wells.

3.7.2 Channel example

Figs. 3.31 to 3.34 are for the channel example, which show the prediction of water cut in each well for three cases: 1) global normal score transform, 2) local normal score transform, and 3) local normal score transform with water breakthrough time as additional data. In the third case, all the four water breakthrough times are assimilated at 30 days. It is difficult to say which transform works here, but over all the best results are obtained when we use both local normal score transform and match water breakthrough time. This can be seen clearly by considering results from cumulative oil production shown in Fig. 3.35.

Fig. 3.36 shows two realizations of the facies distribution conditioned to all production data and the corresponding analyzed water saturations at 360 days for the global normal score transform case. Fig. 3.37 shows corresponding results obtained using the local normal score transform case. Note that all the water saturations are within the correct physical range, which is enforced by the normal score transforms. However, the saturation distributions are still too diffuse although slightly improved in this regard compared to the saturation map obtained with standard EnKF (Fig. 3.21). The failure to obtain a significant improvement in the saturation map may be because the normal score transform is designed to account for the non-Gaussian distribution of water saturation rather than the effect of nonlinearity. In Chapter 6, we will try an alternative scheme to obtain improved results.

3.8 Summary

Tables 3.4 and 3.5 show the quantitative comparison of all PUNQS3 cases considered in this chapter. Recall that $O^{p,EnKF}_{BHP}$, $O^{p,EnKF}_{GOR}$, $O^{p,EnKF}_{WCT}$ and $O^{p,EnKF}_{TOT}$ are prediction errors obtained by simply continuing to run the simulator forward

Cases	$O^{p,EnKF}_{BHP}$	$O^{p,EnKF}_{GOR}$	$O^{p,EnKF}_{WCT}$	$O^{p,EnKF}_{TOT}$
Prior	22.935	14.604	19.643	19.366
Prod	7.901	12.002	17.669	13.149
ScGlobal	7.866	8.567	20.35	13.533
ScLocal	8.228	15.167	17.149	14.045
BT	7.955	13.638	13.433	11.968
ScLocalBT	8.623	14.785	13.547	12.602

Table 3.4: Prediction errors, PUNQS3 example, predicting forward from last data assimilation, EnKF of production data.

Cases	O_m	O^d_{BHP}	O^d_{GOR}	O^d_{WCT}	O^d_{TOT}	O^p_{BHP}	O^p_{GOR}	O^p_{WCT}	O^p_{TOT}
Prior	1.482	29.928	29.405	11.839	28.668	22.935	14.604	19.643	19.366
Prod	1.457	2.505	4.285	3.164	3.123	7.893	13.323	16.004	12.857
ScGlobal	1.393	2.358	5.099	3.43	3.381	8.975	8.684	19.951	13.589
ScLocal	1.49	2.767	4.819	4.553	3.585	8.049	16.079	15.647	13.761
BT	1.458	2.596	3.989	3.707	3.121	8.326	14.6	12.414	12.063
ScLocalBT	1.48	2.899	4.482	4.74	3.56	8.522	15.795	11.038	12.165

Table 3.5: Model mismatch, data mismatch, prediction error, PUNQS3 example, predicting final ensemble from time zero, EnKF of production data.

Cases	$O^{p,EnKF}_{BHP}$	$O^{p,EnKF}_{WCT}$	$O^{p,EnKF}_{TOT}$
Prior	20.26	10.097	16.533
Prod	11.134	5.465	9.063
ScGlobal	10.204	7.058	8.943
ScLocal	9.425	6.602	8.29
BT	24.335	6.337	18.624
ScLocalBT	11.135	7.707	9.761

Table 3.6: Prediction errors, channel example, predicting forward from last data assimilation, EnKF of production data.

Cases	O_m	O^d_{BHP}	O^d_{WCT}	O^d_{TOT}	O^p_{BHP}	O^p_{WCT}	O^p_{TOT}
Prior	0.444	57.603	28.656	46.993	20.26	10.097	16.533
Prod	0.423	25.205	24.386	24.844	11.323	9.89	10.71
ScGlobal	0.4623	31.27	26.698	29.326	11.144	11.479	11.294
ScLocal	0.445	22.382	25.996	24.056	14.601	13.5	14.122
BT	0.444	41.413	24.28	34.853	21.381	8.478	16.909
ScLocalBT	0.445	24.245	20.052	22.478	5.812	4.647	5.326

Table 3.7: Model mismatch, data mismatch, prediction error, channel example, predicting final ensemble from time zero, EnKF of production data.

from the last data assimilation time; O^p_{BHP}, O^p_{GOR}, O^p_{WCT} and O^p_{TOT} are prediction errors obtained by running the simulator from time zero using the final ensemble conditioned to all data; O^d_{BHP}, O^d_{GOR}, O^d_{WCT} and O^d_{TOT} are data mismatches, and O_m is the model mismatch. The first row of numbers in both tables represent the predictions and data matches generated from the initial ensemble. The second row is the base case obtained with standard EnKF. The last four rows are the results from the global normal score transform (ScGlobal), local normal score transform (ScLocal), matching water breakthrough time (BT) and combining local normal score transform and matching water breakthrough time (ScLocalBT). Comparing the first row and the second row of both tables, as we expected, posterior predictions gives much smaller data mismatches and prediction errors. The prediction errors from the final ensemble (Table 3.5 rerun from time zero) are smaller than those from EnKF prediction (Table 3.4), which is consistent with our observation in the figure of the basic case. It is quite difficult to draw any firm conclusion from the tables to compare the global normal score transform and local normal score transform. When water breakthrough times are assimilated, the data mismatches and prediction errors of water cut are smaller, but those of pressure and GOR are slightly higher, but the overall data mismatches and prediction errors are smaller.

Tables 3.6 and 3.7 show the quantitative comparisons for all channel cases considered in this chapter. The base case (second row) gives better data matches and predictions compared to the prior model. It is still difficult to draw any firm conclusion from the table to distinguish between the performance of the global and local normal score transforms. Assimilating water breakthrough times (row 5) shows the data mismatches and prediction errors are higher than for the base case. When the local normal score transform is combined with assimilating water breakthrough times (the last case), better data match and predictions are obtained from final ensemble. We believe this does not indicate a more robust scheme because the same combination failed to improve the results in the PUNQS3 model.

In summary, (1) The adjustment scheme for ensuring that the observations of facies type at a well is honored works properly; (2) Although the normal score transforms maintains saturation values within physical bounds, they yield no significant improvement in terms of data matches and the estimates of future predictions. Results for the local and global normal score transforms are quantitatively similar except for the fact that the global normal score transform yields non-physical oscillations in the water saturation distributions. Overall, it appears that assimilating water breakthrough times in conjunction with the local normal score transform gave better results, but none of the modifications give very significant improvement over results obtained with standard EnKF.

CHAPTER 4

ENSEMBLE KALMAN FILTER FOR SEISMIC DATA

Inverted seismic data are normally available in a few discrete time steps during production history. In this study, we consider only acoustic impedance data. Similar to the definition of observed production data (Eq. 3.3), the observed acoustic impedance data can be defined as $S_{obs,n} = [s_{obs,n,1}, s_{obs,n,2}, \cdots, s_{obs,n,N_s}]$, where n is the index for data assimilation step (t_n), and N_s is the number of seismic data which is equal to the number of gridblocks because we effectively have measurements in each gridblock. The seismic data could be the inverted acoustic impedance and/or Poisson's ratio etc.

In assimilating these inverted seismic data, the number of measurements is normally much larger than the number of ensemble members. This causes two problems: (i) it is not computationally feasible to compute or "invert" the matrix $C_{D_n^p,D_n^p} + C_{D,n}$ that appears in Eq. 3.7 (now we use $S_{obs,n}$ as seismic data); (ii) at any data assimilation step, the vector of model parameters, m, in each updated ensemble member must be a linear combination of the corresponding models in the initial ensemble so there are not sufficient degrees of freedom to properly assimilate data when the number of independent data is far greater than the number of ensemble members.

In order to solve this problem, there are two approaches in the literature: (i) global analysis [73] compresses the number of data to $N_e - 1$ using singular value decomposition; (ii) local analysis [59, 25, 75] updates the state vector components in a gridblock using the seismic data located at its neighbor gridblocks to obtain better match of data, which shares similar idea with covariance localization method used in the references [41, 15]. Both of these two methods have limitations: after the global analysis of seismic data, all ensemble members become very close to each other and it is difficult to match subsequent data unless the ensemble is very large; the local analysis can introduce more freedom to match the data but may induce loss of spatial smoothness in the conditional models.

During the global analysis of seismic data, to guarantee the hard data is honored for the pluri-Gaussian model, we use the same adjustment scheme as was used in the last chapter for assimilation of production data. If the local analysis procedure fails to give updated Gaussian random fields which honor the hard data (observed facies type at any well gridblock), we will also modify the update by introducing pseudo-hard data. However, this becomes somewhat more complicated than the adjustment for the global analysis, because it is necessary to use pseudo-data not only when updating the Gaussian random fields at a well gridblock with EnKF, but also when updating the Gaussian-random fields in nearby gridblocks if these gridblocks are updated using seismic data in the well gridblock.

For a pluri-Gaussian geological model, the Gaussian random fields must be smooth in order to generate continuous facies distributions. Local analysis of seismic data offer results in facies distributions that are inconsistent with the desired smoothness. To recover the geological smoothness, we project the difference between each updated realization from EnKF analysis and its prior on to a space spanned by an ensemble of realizations of the initial model of Gaussian random fields where each of these realizations has been conditioned to the hard data. This initial ensemble is obtained by starting with an ensemble of 1200 members and then using singular value decomposition to reduce it to 200 members corresponding to the 200 largest singular values. The updated model from this projection procedure may also violate the hard data. An adjustment scheme for the projected model is also introduced to honor the hard data, in which the projection coefficients are adjusted to ensure that hard data are honored.

The examples within this chapter compare the application of global and local analysis methods for assimilating inverted seismic data (acoustic impedance data). For each cases, the production data and seismic data observed at the same time step are assimilated separately: seismic data are assimilated using the ensemble conditioned to the production data as the prior. For the PUNQS3 case, the first set of seismic data are observed at time zero, and the second set is observed at 2008 days. For both the three-facies and two facies cases, the first set of seismic data are also obtained at time zero, and the second data set is obtained at 300 days. The same production data used previously are also assimilated. Note that the two acoustic impedance data sets are assimilated at the time they are "measured," i.e., we assimilate 3D seismic data, not time-lapse seismic data. It is worthwhile to mention that for a small 2D synthetic problem, Skjervheim et al. [74] found that better estimates of the permeability field model were obtained by assimilating inverted seismic data at the time they were measured instead of using 4D data. Although the authors give no explanation of why assimilating difference data gives worse results than assimilating seismic data directly, we believe there may be two reasons for this. First, the variance of the difference data is twice the variance of the individual data sets. Secondly, using difference data requires updating of the state vector at the time of the first seismic survey several times using the ensemble Kalman smoother (EnKS, [24]) as production data measured between the times of the two surveys are assimilated. This could allow errors due to strong nonlinearities or non-Gaussianity to accumulate.

Noisy seismic measurements and predicted seismic data are denoted, respectively, by $S_{uc,n}$ and S_n^p (similar to Eqs. 3.4 and 3.5) and are given by $N_s \times N_e$ matrices:

$$\begin{aligned} S_{uc,n} &= \left[s_{uc,n,1}, s_{uc,n,2}, \cdots, s_{uc,n,N_e}\right] \\ S_n^p &= \left[s_{n,1}^p, s_{n,2}^p, \cdots, s_{n,N_e}^p\right], \end{aligned} \tag{4.1}$$

where $s_{uc,n,j}$ is obtained by summing up the observed seismic data at the nth data assimilation step ($S_{obs,n}$) and the jth realization of the measurement error in $S_{obs,n}$, and $s_{n,j}^p$ is the predicted seismic data for the jth ensemble member at the nth data assimilation time step. If the seismic data are assimilated directly, the analysis equation is similar to Eq. 3.7 except that the data are replaced by the seismic data. Before showing the analysis equation for the seismic data at the nth data assimilation step, we need to define the following terms. The error matrix of predicted seismic data is

$$\Delta S_n^p = S_n^p - \overline{S_n^p} \tag{4.2}$$

where $\overline{S_n^p}$ represents a matrix with each column equals to the average of the columns of S_n^p. The data mismatch of the jth ensemble member is

$$\delta s_{mis,n,j} = s_{uc,n,j} - s_{n,j}^p. \tag{4.3}$$

Two approximations of covariance matrices can be written in ensemble form as

$$\begin{aligned} C_{Y_n^p, S_n^p} &= \frac{1}{N_e - 1} \Delta Y_n^p (\Delta S_n^p)^T \\ C_{S_n^p, S_n^p} &= \frac{1}{N_e - 1} \Delta S_n^p (\Delta S_n^p)^T. \end{aligned} \tag{4.4}$$

Similar to Eq. 3.15, the standard EnKF analysis for the seismic data can be written as

$$y_{n,j}^a = y_{n,j}^p + \Delta Y_n^p (\Delta S_n^p)^T \left[\Delta S_n^p (\Delta S_n^p)^T + (N_e - 1) C_{S,n}\right]^{-1} \delta s_{mis,n,j}, \tag{4.5}$$

where $C_{S,n}$ represents the measurement error matrix of the seismic data. If $C_{S,n}$ is the true full-rank covariance matrix, then this $N_s \times N_s$ matrix is nonsingular and there is no loss of rank when updating the covariance matrix [43]. If $C_{S,n}$ is also represented by a sample of size N_e, then loss-of-rank issues can become significant [43, 28]. Specifically, suppose we have an ensemble of seismic data measurement error denoted by the matrix $E_{S,n} = [e_{S,n,1}, e_{S,n,2}, \cdots, e_{S,n,N_e}]$ with each column as a realization of seismic data measurement error, then we can approximate $C_{S,n}$ as:

$$C_{S,n} = \frac{1}{N_e - 1} E_{S,n} E_{S,n}^T. \tag{4.6}$$

As discussed above, it is not feasible to assimilate the seismic data using Eq. 4.5. In the following sections, we will discuss implementations of the global analysis and local analysis to assimilate the seismic data.

4.1 Global Analysis of Seismic Data

The singular value decomposition of the matrix ΔS_n^p can be computed as $\Delta S_n^p = U \Lambda V^T$, where U is an $N_s \times N_s$ matrix, Λ is an $N_s \times N_e$ matrix, and V is a $N_e \times N_s$ matrix.

We can rewrite Eq. 4.5 using $UU^T = I$ (or $U^{-1} = U^T$), as

$$\begin{aligned} y_{n,j}^a &= y_{n,j}^p + \Delta Y_n^p (\Delta S_n^p)^T U U^T \left[\Delta S_n^p (\Delta S_n^p)^T + (N_e - 1) C_{S,n} \right]^{-1} U U^T \delta s_{mis,n,j} \\ &= y_{n,j}^p + \Delta Y_n^p (U^T \Delta S_n^p)^T \left[U^T \Delta S_n^p (U^T \Delta S_n^p)^T + (N_e - 1) U^T C_{S,n} U \right]^{-1} U^T \delta s_{mis,n,j}. \end{aligned} \tag{4.7}$$

Now we define transformed predicted data as

$$\hat{S}_n^p = U^T S_n^p \tag{4.8}$$

with the jth column as

$$\hat{s}_{n,j}^p = U^T s_{n,j}^p. \tag{4.9}$$

The corresponding predicted error matrix is

$$\Delta \hat{S}_n^p = U^T \Delta S_n^p, \tag{4.10}$$

and the transformed data mismatch for the jth ensemble member is

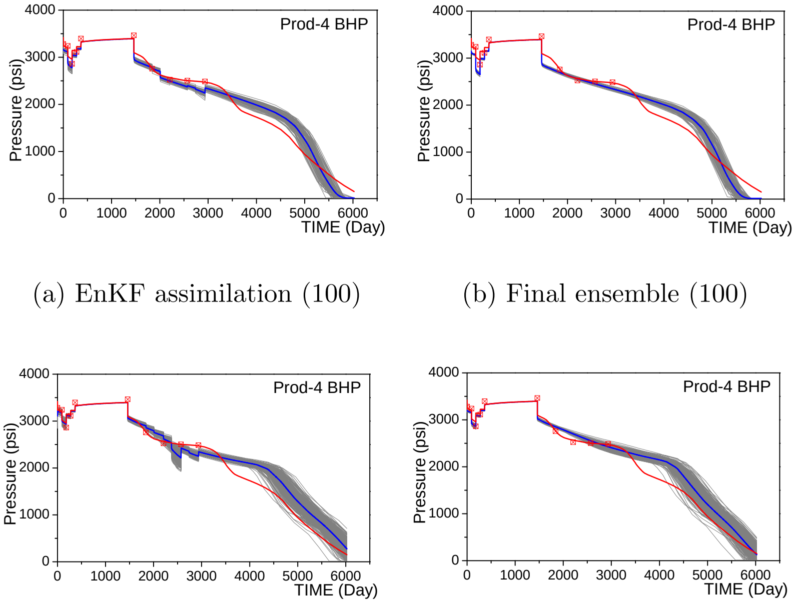
$$\delta \hat{s}_{mis,n,j} = U^T \delta s_{mis,n,j}. \tag{4.11}$$

Here, the transformed data terms are denoted with hat. Using Eqs. 4.8 to 4.11 and Eq.4.6, Eq. 4.7 can be written as:

$$y_{n,j}^a = y_{n,j}^p + \Delta Y_n^p (\Delta \hat{S}_n^p)^T \left[\Delta \hat{S}_n^p (\Delta \hat{S}_n^p)^T + \hat{E}_{S,n} \hat{E}_{S,n}^T \right]^{-1} \delta \hat{s}_{mis,n,j}, \tag{4.12}$$

where $\hat{E}_{S,n} = U^T E_{S,n}$, and $\hat{E}_{S,n}$ represents the transformed measurement error matrix. If U is a full matrix of left singular vectors with N_s columns, Eq. 4.7 and 4.12 are equivalent to Eq. 4.5.

However, there are only maximum $N_e - 1$ non-zero eigenvalues in Λ. This is because ΔS_n^p has only N_e columns and is a deviation matrix with each column equals to the corresponding column in S_n^p subtract the average of all columns in S_n^p (Eq. 4.2), which will induce loss of one dimension. In the global analysis scheme, we use only $N_e - 1$ eigenvectors corresponding to the $N_e - 1$ non-zero eigenvalues. Redefine U as a $N_s \times N_e - 1$ matrix containing only the first $N_e - 1$ column of the original U. This definition is then applied to the transformed data in Eqs. 4.8 to 4.11. The dimension of the transformed predicted data ($\hat{S}_{n,j}^p$) and transformed data mismatch ($\delta \hat{s}_{mis,n,j}$) are then decreased to $N_e - 1$, and hence the computation of Eq. 4.12 become feasible because we need only to invert a $(N_e - 1) \times (N_e - 1)$ matrix. However, in this case, $UU^T \neq I$, and therefore, Eqs. 4.5 and 4.12 are no longer equivalent. Although Eq. 4.12 is not equivalent to the standard EnKF scheme because of limited ensemble



(a) EnKF assimilation (100) (b) Final ensemble (100)

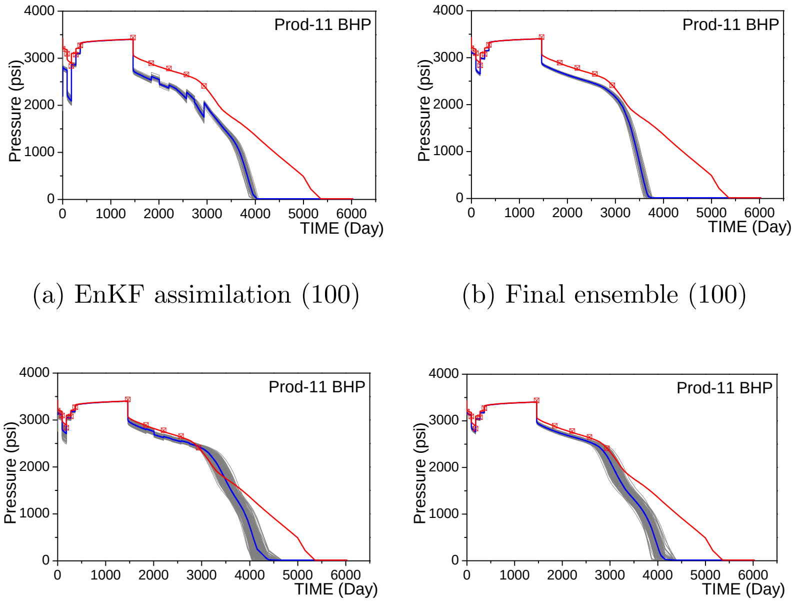
(c) EnKF assimilation (300) (d) Final ensemble (300)

Figure 4.1: Prod4 BHP for ensemble size of 100 and 300 (PUNQS3, global analysis).

size, like Skjervheim et al. [73], we will still use this equation to analyze the seismic data because this solution makes it feasible to assimilate seismic data using EnKF. However, as will be shown, this scheme can suffer from severe filter divergence, and we may need a very large ensemble to get meaningful results. This is the major reason why we need local analysis as the second choice to assimilate the seismic data.

4.2 Global Analysis Results

A global analysis of seismic data for the PUNQS3 model and the channel model will be discussed here. Two ensemble sizes of 100 and 300 are used for both models. Using an ensemble size of 100 gives very small uncertainty of the posterior model, bad data matches and wrong predictions for both examples. This is because the ensemble members get very close to each other after assimilating the first seismic data. Some improvement can be observed when the ensembles are enlarged. At a specific data assimilation step with both production and seismic data, production



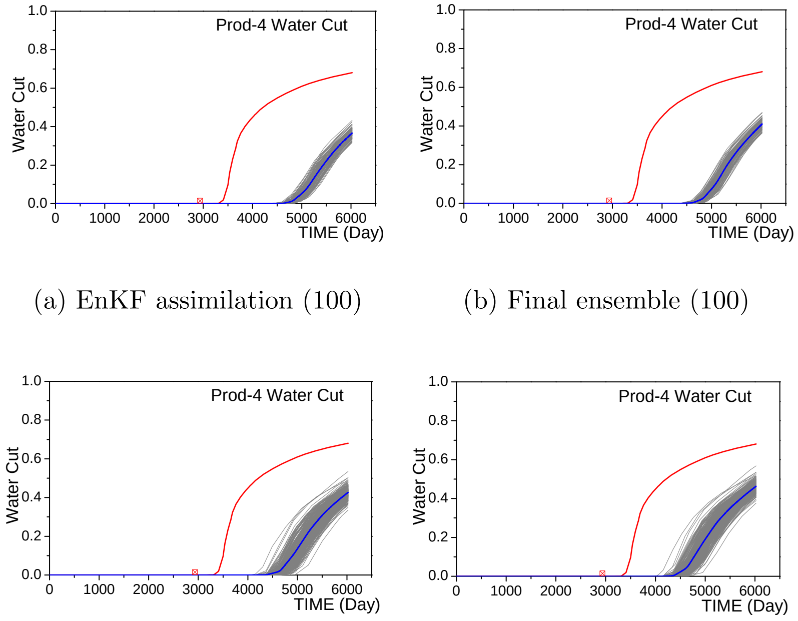
(a) EnKF assimilation (100)

(b) Final ensemble (100)

(c) EnKF assimilation (300)

(d) Final ensemble (300)

Figure 4.2: Prod11 BHP for ensemble size of 100 and 300 (PUNQS3, global analysis).



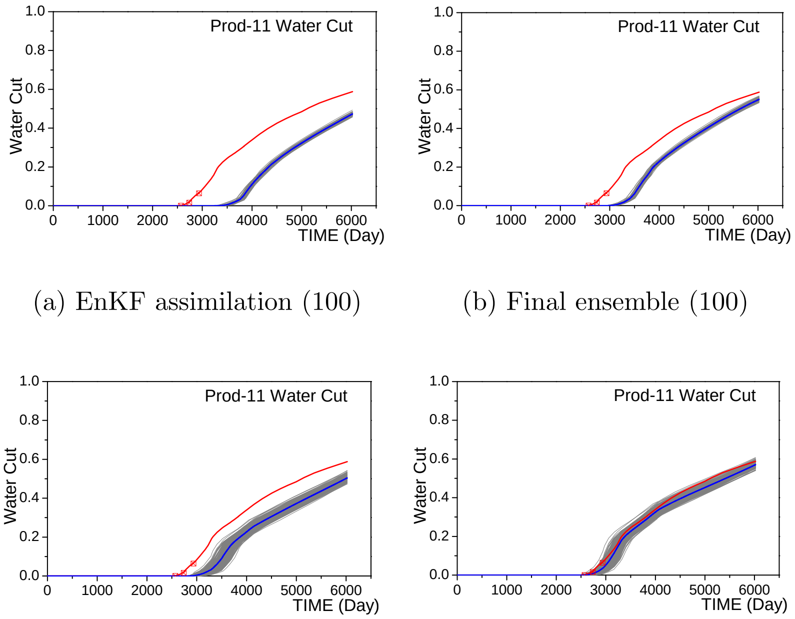
(a) EnKF assimilation (100)

(b) Final ensemble (100)

(c) EnKF assimilation (300)

(d) Final ensemble (300)

Figure 4.3: Prod4 WCT for ensemble size of 100 and 300 (PUNQS3, global analysis).



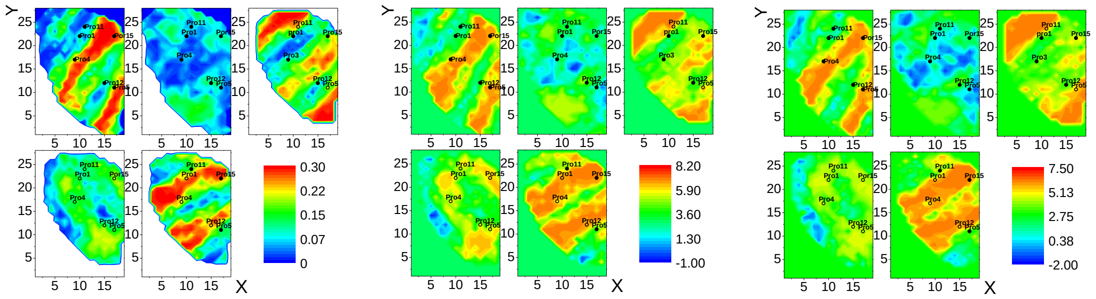
(a) EnKF assimilation (100)

(b) Final ensemble (100)

(c) EnKF assimilation (300)

(d) Final ensemble (300)

Figure 4.4: Prod11 WCT for ensemble size of 100 and 300 (PUNQS3, global analysis).

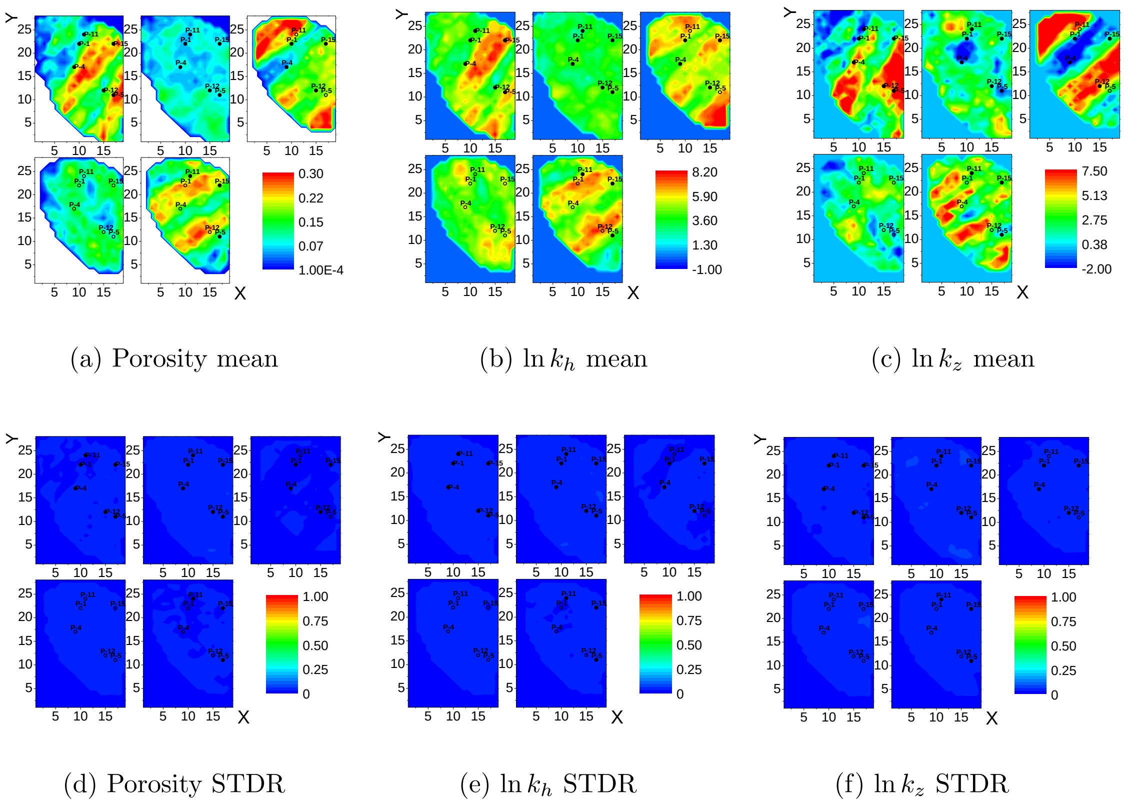


(a) Porosity

(b) $\ln k_h$

(c) $\ln k_z$

Figure 4.5: True model of PUNQS3: porosity, $\ln k_h$, $\ln k_z$.



(a) Porosity mean

(b) $\ln k_h$ mean

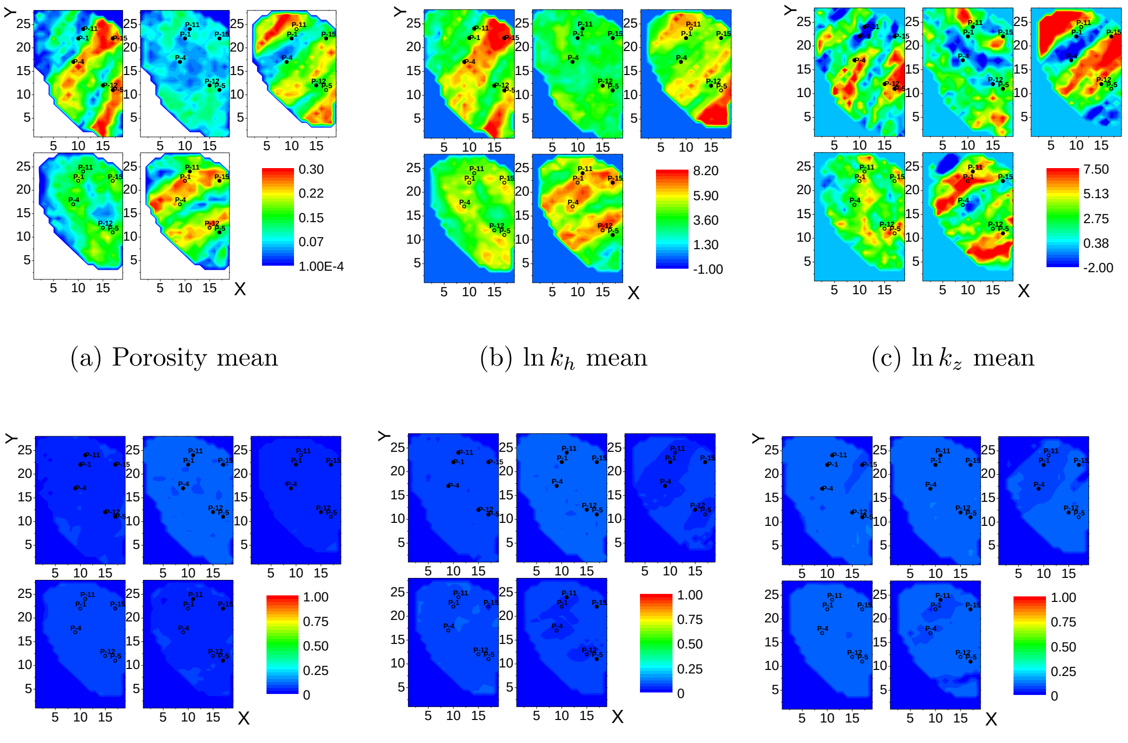
(c) $\ln k_z$ mean

(d) Porosity STDR

(e) $\ln k_h$ STDR

(f) $\ln k_z$ STDR

Figure 4.6: Mean and STDR of conditional fields, ensemble size is 100 (PUNQS3, global analysis).



(a) Porosity mean

(b) $\ln k_h$ mean

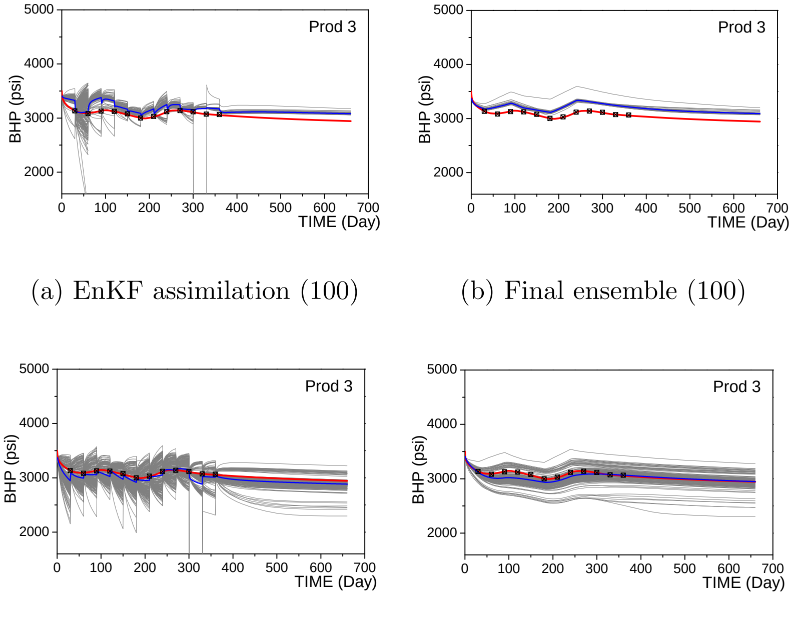
(c) $\ln k_z$ mean

(d) Porosity STDR

(e) $\ln k_h$ STDR

(f) $\ln k_z$ STDR

Figure 4.7: Mean and STDR of conditional fields, ensemble size is 300 (PUNQS3, global analysis).



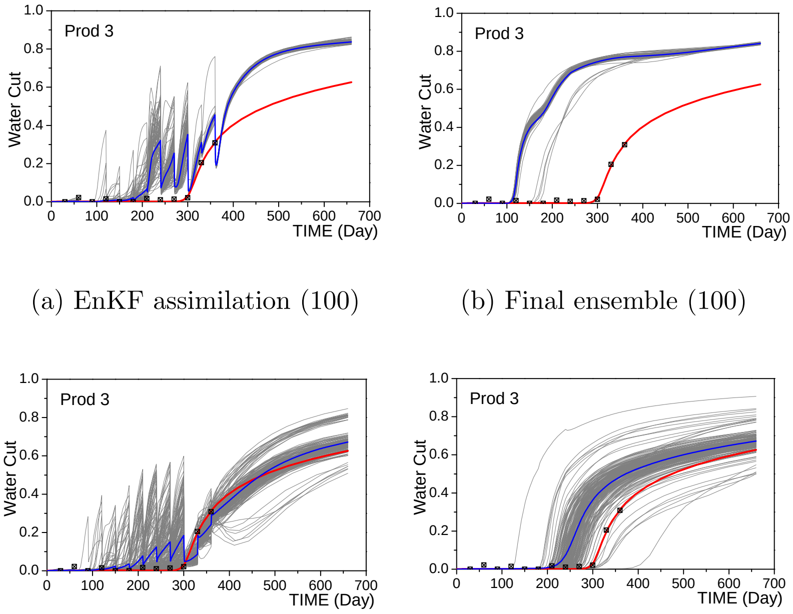
(a) EnKF assimilation (100)

(b) Final ensemble (100)

(c) EnKF assimilation (300)

(d) Final ensemble (300)

Figure 4.8: Prod3 BHP, ensemble size of 100 and 300 (channel case, global analysis).



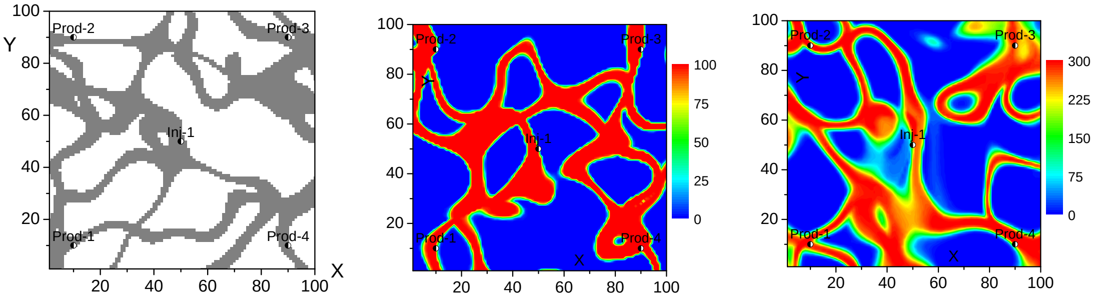
(a) EnKF assimilation (100)

(b) Final ensemble (100)

(c) EnKF assimilation (300)

(d) Final ensemble (300)

Figure 4.9: Prod3 WCT, ensemble size of 100 and 300 (channel case, global analysis).



(a) True facies

(b) Channel PDF (100)

(c) Channel PDF (300)

Figure 4.10: True facies, channel PDF of two ensemble size 100 and 300 (channel case, global analysis).

data are assimilated first to get a conditional ensemble, which is then used as the prior ensemble to assimilate the seismic data, i.e., assimilate the production data and seismic data separately and sequentially.

4.2.1 PUNQS3 Example

Figs. 4.1 to 4.4 show the predicted bottom hole pressure and water cut in producers 3 and 11 with ensemble sizes of 100 and 300. Even with the ensemble size of 300, the predictions still fail to bound the truth except for water cut data for producer 11 when we predict from time zero using the final ensemble. We can say that for both ensemble sizes the conditional predictions as well as the data matches are wrong.

Figs. 4.6 and 4.7 show the means and standard deviation ratios (STDR) of the conditional fields for both ensemble sizes, in which the posterior means of porosity and $\ln k_h$ fields can roughly capture the structure of the truth (Fig. 4.5), but the posterior mean of $\ln k_z$ is much different from the truth with some overshooting and undershooting. From the figures, the standard deviations of the conditional model for the 100 ensemble members case is decreased to under 10% (STDR) of the values in the initial ensemble. For the case with 300 ensemble members, this ratio is higher but is still below 20%. Thus, it seems clear that the global analysis greatly underestimates the posterior uncertainties of models and predictions (severe filter divergence), which make it difficult to assimilate subsequent production data with EnKF.

4.2.2 Channel Example

Figs. 4.8 and 4.9 show the predictions of bottom hole pressure and water cut for producer 3 for the cases with different ensemble sizes of 100 and 300. Note that the predictions are extremely biased for the 100 ensemble member case, but are much more reasonable for the 300 ensemble case although predictions from time zero appear to be somewhat biased. Fig. 4.10 shows the true facies distribution and the channel PDF of these two cases. The small ensemble size case gives a very small uncertainty in the distribution of facies, which is obviously wrong connectivity between producer 4 (near lower right corner) and injector, which corresponds to the fact that the predictions are too narrow and far from the truth. The larger ensemble size of 300 gives higher uncertainty in the channel distribution and better overall results. However, for an ensemble size of 300, the connectivity between the injector and producers 3 and 4 is better. All ensemble members converge to a wrong facies map, see Fig. 4.10.

From the results of both examples, it is clear that global analysis require a large ensemble to give a reasonable result, and even an ensemble size of 300 may be insufficient to correctly characterize the posterior uncertainty.

4.3 Local Analysis of Seismic Data

Local analysis is the second method we will consider to assimilate seismic data. In this scheme, we update variables gridblock by gridblock. $y_{b_i,n}$ contains all entries of state vector associated with the ith gridblock. In this way, the local analysis can be used to update the state vector components located in a certain gridblock using the data in its neighbor gridblocks, and hence the state vector components from different gridblocks are updated using different set of data.

Denote a predicted state matrix composed of N_e predicted state vectors with all their components located in the ith gridblock as $Y^p_{b_i,n} = \left[y^p_{b_i,n,1}, y^p_{b_i,n,2}, \cdots, y^p_{b_i,n,N_e}\right]^T$, where $y^p_{b_i,n,j}$ represents all entries in the predicted state vector located in the ith gridblock, and hence it is the subvector of $y^p_{n,j}$ with the entry list indicating the components in the ith gridblock, i.e. porosity, permeability, Gaussian random field values, pressure, and saturations etc. The dimension of matrix $Y^p_{b_i,n}$ is $N_{y,b_i} \times N_e$, where N_{y,b_i} is the number of entries in the ith subvector, for $i = 1, 2, \cdots, N$, and

N is the total number of gridblocks. Local analysis will find the updated states in the ith gridblock ($Y^a_{b_i,n}$) by matching the seismic data located in the neighbor of this gridblock. $s_{obs,b_i,n}$ is a vector which contains all observed seismic data in the neighborhood of gridblock i that will be used to update all entries of the state vector corresponds to gridblock i. We then use $s_{uc,b_i,n,j}$ to denote a perturbation of $s_{obs,b_i,n}$ obtained by adding noise. Here, $S^p_{uc,b_i,n}$ denotes a matrix containing the perturbed seismic data for the ensemble, i.e., $S_{uc,b_i,n} = [s_{uc,b_i,n,1}, s_{uc,b_i,n,2}, \cdots, s_{uc,b_i,n,N_e}]$. In this study, the size of the neighborhood used is 5 × 5, so the number of data included when performing the EnKF analysis for each gridblock is 25.

The predicted data matrix for the ith gridblock is written as

$$S^p_{b_i,n} = \left[s^p_{b_i,n,1}, s^p_{b_i,n,2}, \cdots, s^p_{b_i,n,N_e}\right],$$

where $s^p_{b_i,n,j}$ denotes the predicted seismic data from the jth realization in the neighbor gridblocks of the ith gridblock, and hence $S^p_{b_i,n}$ has the same dimension as $S_{uc,b_i,n}$ ($N_{d,b_i,n} \times N_e$). To construct a local analysis equation, we define error matrices of the predicted state vector and the predicted data localized for the ith gridblock:

$$\begin{aligned} \Delta Y^p_{b_i,n} &= Y^p_{b_i,n} - \overline{Y^p_{b_i,n}} \\ \Delta S^p_{b_i,n} &= S^p_{b_i,n} - \overline{S^p_{b_i,n}}, \end{aligned} \tag{4.13}$$

where $\overline{Y^p_{b_i,n}}$ and $\overline{S^p_{b_i,n}}$, respectively, are the matrices with each column equal to the average of the columns in $Y^p_{b_i,n}$ and $S^p_{b_i,n}$, respectively. The corresponding approximation of covariance matrices can be written as

$$\begin{aligned} C_{Y^p_{b_i,n},S^p_{b_i,n}} &= \frac{1}{N_e-1}\Delta Y^p_{b_i,n}(\Delta S^p_{b_i,n})^T \\ C_{S^p_{b_i,n},S^p_{b_i,n}} &= \frac{1}{N_e-1}\Delta S^p_{b_i,n}(\Delta S^p_{b_i,n})^T, \end{aligned} \tag{4.14}$$

and the data mismatches for the jth realization is

$$\delta s_{mis,b_i,n,j} = s_{uc,b_i,n,j} - s^p_{b_i,n,j}. \tag{4.15}$$

Using the definitions in Eqs. 4.13 to 4.15, each column of the analyzed ensemble matrix $Y^a_{b_i,n} = \left[y^a_{b_i,n,1}, y^a_{b_i,n,2}, \cdots, y^a_{b_i,n,N_e}\right]$ can be obtained using

$$y^a_{b_i,n,j} = y^p_{b_i,n,j} + \Delta Y^p_{b_i,n}(\Delta S^p_{b_i,n})^T \left[\Delta S^p_{b_i,n}(\Delta S^p_{b_i,n})^T + (N_e - 1)C_{S_{b_i,n}}\right]^{-1} \delta s_{mis,b_i,n,j}, \tag{4.16}$$

where $C_{S_{b_i,n}}$ is the measurement error covariance matrix for the measurements entries of $s_{obs,b_i,n}$.

The analyzed ensemble from Eq. 4.16 is no longer confined to the subspace spanned by the predicted ensemble and thus, for local analysis it is easier to find an updated ensemble that matches a large data set because the components in different gridblocks are updated using different linear combination of the predictions. However, the smoothness of the updated models may be less than the smoothness inherent in the covariance of the predicted ensemble because the state vector components with more than 5 gridblocks apart are updated using totally different data sets. The degradation of the spatial smoothness in the conditional models can be a significant problem for the pluri-Gaussian case because the analyzed model may not have smooth boundaries between facies. In this study, a projection method is used to eliminate this problem.

For the pluri-Gaussian geological model, an adjustment algorithm is needed to honor the hard data whenever the models are updated. In the rest of this section, adjustment schemes for both the local analysis and the projection process will be discussed.

4.3.1 Honoring Hard Data in Local Analysis

For the pluri-Gaussian geological model, the analyzed model obtained using Eq. 4.16 may violate the hard data (observed facies) if the ith gridblock is a well gridblock. To honor the hard data, we will follow three steps similar to those shown in Chapter 3. Here, if the hard data at a well gridblock is not honored, we augment the seismic data in this gridblock with pseudo data.

To define the expanded data vector, we need to consider the following complications for the local analysis: 1) we assume the size of neighborhood is small enough so that the local analysis for a well gridblock does not include the data in another well gridblock; 2) for a well gridblock, if an adjustment is needed to honor the hard data, the number of the pseudo data is 2 because only one pair of Gaussian random field values are included in this gridblock; 3) for a gridblock without a well, if a neighbor gridblock includes a pseudo data (because of violation of the hard data at that well), the pseudo data will also be included in the local analysis. Therefore, we can define the expanded data as follows:

1. The expanded predicted data for the jth ensemble member is defined as

$$\tilde{s}^p_{b_i,n,j} = \left[(y^p_{b_{w,k},n,j})^T, (s^p_{b_i,n,j})^T \right]^T, \tag{4.17}$$

   where $y^p_{b_{w,k},n,j}$ represents the predicted Gaussian random field values (one pair) of the kth well gridblock for the jth ensemble member if two conditions are satisfied: (1) the kth well gridblock is in the neighborhood of the ith gridblock and (2) the hard data in the kth gridblock is not honored (or this gridblock contains pseudo data to enforce honoring the hard data). Here, i could be equal to k. The corresponding expanded predicted data matrix can be denoted as $\tilde{S}^p_{b_i,n}$ with $\tilde{s}^p_{b_i,n,j}$ as its jth column, and the corresponding error matrix is

$$\Delta\tilde{S}^p_{b_i,n} = \tilde{S}^p_{b_i,n} - \overline{\tilde{S}^p_{b_i,n}}, \tag{4.18}$$

where $\overline{\tilde{S}^p_{b_i,n}}$ is a matrix with each column equals to the average of the columns of $\tilde{S}^p_{b_i,n}$. Note that if either one of the above two conditions is not satisfied, we will use Eq. 4.16 as the analysis equation for the ith gridblock (without data expansion) and the adjustment scheme described here will not be used.

2. If the predicted data is augmented, the corresponding data mismatch for the jth ensemble member is defined as

$$\delta\tilde{s}_{mis,b_i,n,j} = \left[(c_{b_{w,k},n,j})^T, (\delta s_{mis,b_i,n,j})^T\right]^T, \tag{4.19}$$

   where $c_{b_{w,k},n,j}$ is the pseudo data included to honor the hard data in the kth well gridblock. $c_{b_{w,k},n,j}$ has the same dimension as $y^p_{b_{w,k},n,j}$.

3. Similar to Eq. 3.22, we define $C_{\tilde{S}_{b_i,n}}$ as the covariance matrix of the measurement error expanded by an identity matrix.

With these definitions, the corresponding covariance matrix for the predicted state and the predicted data and the auto covariance matrix of the predicted data can be written as

$$\begin{aligned} C_{Y^p_{b_i,n},\tilde{S}^p_{b_i,n}} &= \frac{1}{N_e - 1}\Delta Y^p_{b_i,n}(\Delta\tilde{S}^p_{b_i,n})^T \\ C_{\tilde{S}^p_{b_i,n},\tilde{S}^p_{b_i,n}} &= \frac{1}{N_e - 1}\Delta\tilde{S}^p_{b_i,n}(\Delta\tilde{S}^p_{b_i,n})^T. \end{aligned} \tag{4.20}$$

Similar to Eq. 3.24, the local analysis equation with expanded data can be written as

$$\tilde{y}^a_{b_i,n,j} = y^p_{b_i,n,j} + \Delta Y^p_{b_i,n}(\Delta \tilde{S}^p_{b_i,n})^T \left[\Delta \tilde{S}^p_{b_i,n}(\Delta \tilde{S}^p_{b_i,n})^T + (N_e - 1)C_{\tilde{S}_{b_i,n}}\right]^{-1} \delta \tilde{s}_{mis,b_i,n,j}, \tag{4.21}$$

where the tilde terms represent the expanded predicted data and expanded data mismatches.

Using a similar scheme to the one in the last chapter, we use the following three steps to find the values of the $c_{b_{w,k},n,j}$'s to be used as pseudo-data to ensure that the facies observations at well gridblocks will be honored.

1. **Normal Analysis.** The three steps are applied to the ensemble members one by one. For the jth member, we use Eq. 4.16 (without pseudo data) to obtain the Gaussian random fields at the N_w well gridblocks:

   $$Y^a_{b_w,n,j} = \left[y^a_{b_{w,1},n,j}, y^a_{b_{w,2},n,j}, \cdots, y^a_{b_{w,N_w},n,j}\right]^T. \tag{4.22}$$

   If the analyzed state vector in the kth well gridblock $y^a_{b_{w,k},n,j}$ is consistent with the hard data, then no pseudo data will be introduced into this gridblock, but if not, the adjusted Gaussian random field pairs for this gridblock ($y^{adj}_{b_{w,k},n,j}$) is found using the method shown in the last chapter (Eq. 3.28).

2. **Calculate Adjustment.** Each of the adjusted pair in a well gridblock contains two values, and the corresponding pseudo data $c_{b_{w,k},n,j}$'s can be found by solving Eq. 4.21 (with $\tilde{y}^a_{b_i,n,j}$ replaced by $y^{adj}_{b_{w,k},n,j}$) for $c_{b_{w,k},n,j}$. Similar to Eq. 3.29 in the last chapter we solve

   $$y^{adj}_{b_{w,k},n,j} = y^p_{b_k,n,j} + A_{b_k,n,j}\delta \tilde{s}_{mis,b_k,n,j}, \tag{4.23}$$

where

$$A_{b_k,n,j} = \Delta Y^p_{b_k,n}(\Delta \tilde{D}^p_{b_k,n})^T \left[\Delta \tilde{D}^p_{b_k,n}(\Delta \tilde{D}^p_{b_k,n})^T + (N_e - 1)C_{\tilde{D},b_k,n}\right]^{-1} \tag{4.24}$$

is a $2 \times (2 + N_{s,b_k,n})$ matrix. Eq. 4.23 is solved only for the well gridblock to find the pseudo data. Here, the same algorithm as shown in Eqs. 3.30, 3.31 and 3.32 is used.

3. **Redo Analysis.** Using the solved pseudo data obtained in step 2, Eq. 4.21 is applied to perform the local analysis of the state vector for the gridblocks affected by the pseudo data, and the $\tilde{y}^a_{b_i,n,j}$ will be obtained as the locally analyzed state vector at the ith gridblock after adjustment. For the gridblocks not affected by the pseudo data, the local analysis will be performed using Eq. 4.16.

In the following discussion, we will always use $y^a_{n,j}$ to denote the analyzed state vector whether or not its components are obtained using the adjustment scheme.

4.3.2 Local Analysis of Seismic Data with Projection

As mentioned in the last section, a drawback of the local analysis scheme is that some non-smooth components can be introduced into the analyzed models so that the continuity of the fields can be severely degraded. Assume the predicted model of the jth ensemble member before analysis is written as $m^p_{n,j} = \left[m^p_{n,j,1}, m^p_{n,j,2}, \cdots, m^p_{n,j,N_m}\right]^T$, and $m^a_{n,j} = \left[m^a_{n,j,1}, m^a_{n,j,2}, \cdots, m^a_{n,j,N_m}\right]^T$ is the model obtained through local analysis, where N_m is the number of the model parameters. The correction term is $\delta m^a_{n,j} = m^a_{n,j} - m^p_{n,j}$. Here, we will use a projection method to remove the unwanted non-smooth components in $\delta m^a_{n,j}$ while keeping the smooth components.

Assume we have a large ensemble of N^b_e realizations of the model conditioned only to the hard data (without assimilating the dynamic data). This ensemble can be written as a matrix:

$$M^0 = \left[m_1^0, m_2^0, \cdots, m_{N_e^b}^0 \right], \tag{4.25}$$

which is totally different from the initial ensemble. The correction term is projected onto a subspace of the space spanned by the columns of M^0. To project each $\delta m_j^{n,a}$ onto the subspace spanned by the m_j^0's, we simply minimize the objective function of Eq. 4.26 to obtain the vector of coefficients, $\alpha_j = [\alpha_{j,1}, \alpha_{j,2}, \cdots, \alpha_{j,N_e^b}]^T$.

$$O(\alpha_j) = (\delta m_{n,j}^a - M^0 \alpha_j)^T (\delta m_{n,j}^a - M^0 \alpha_j); \tag{4.26}$$

then we set $\delta \hat{m}_{n,j}^a = M^0 \alpha_j$. Setting the gradient equal to zero and solving for α_j gives

$$\begin{aligned} \nabla_{\alpha_j} O(\alpha_j) &= -(M^0)^T (\delta m_{n,j}^a - M^0 \alpha_j) = 0 \\ (M^0)^T M^0 \alpha_j &= (M^0)^T \delta m_{n,j}^a. \end{aligned} \tag{4.27}$$

The singular value decomposition of M^0 is given by

$$M^0 = U_p \Lambda_p V_p^T, \tag{4.28}$$

where p is the number of positive singular values that are retained in the approximation, Λ_p is a $p \times p$ matrix with the largest p singular values on its diagonal, the jth column of U_p is the jth left singular vector and the jth column of V_p is the jth right singular vector. Using this singular value decomposition in the second equation of Eq. 4.27 and using standard procedures gives

$$\alpha_j = V_p \Lambda_p^{-1} U_p^T \delta m_{n,j}^a. \tag{4.29}$$

Using this result in the definition of $\delta\hat{m}^a_{n,j}$ gives

$$\begin{aligned}\delta\hat{m}^a_{n,j} &= M^0 V_p(\Lambda_p)^{-1}U_p^T\delta m^a_{n,j} = U_p\Lambda_p V_p^T V_p(\Lambda_p)^{-1}U_p^T\delta m^a_{n,j} \\ &= U_p U_p^T \delta m^a_{n,j}.\end{aligned} \tag{4.30}$$

For the examples presented in this work, we use $N_e^b = 1200$ for all cases and use 200 left singular vectors corresponding to the 200 largest singular values in Eq. 4.30. We have experimented with choosing the number of singular values retained based on the ratio of largest to smallest singular value retained, but for all examples we have tried with an ensemble size of 100 (for EnKF), the choice of 200 has worked well.

The corresponding modified analyzed model is given by

$$\hat{m}^a_{n,j} = m^p_{n,j} + \delta\hat{m}^a_{n,j}. \tag{4.31}$$

The projected models $\hat{m}^a_{n,j}$'s are then used to replace the $m^a_{n,j}$'s that were obtained by local analysis and adjustment. However, this modification of the model is not guaranteed to honor the hard data for the pluri-Gaussian model.

If we let $\beta_j = U_p^T\delta m^a_{n,j}$, we can rewrite Eq. 4.30 as

$$\delta\hat{m}^a_{n,j} = U_p\beta_j, \tag{4.32}$$

and hence Eq. 4.31 become

$$\hat{m}^a_{n,j} = m^p_{n,j} + U_p\beta_j. \tag{4.33}$$

The state vector with the ensemble of models obtained by the projection method is denoted by $\hat{y}^a_{n,j}$, $j = 1, 2, \cdots N_e$. Let $\hat{y}^a_{b_w,n,j}$ be the subvector of $\hat{y}^a_{n,j}$ that corresponds to pairs of the Gaussian random fields at well gridblocks. Note this means that $\hat{y}^a_{b_w,n,j}$ is actually a subvector of $\hat{m}^a_{n,j}$. So from Eqs. 4.31 and 4.32, it follows that

$$\hat{y}^{a}_{b_w,n,j} = y^{p}_{b_w,n,j} + U_{p,b_w}\beta_j, \tag{4.34}$$

where U_{p,b_w} is the submatrix of U_p corresponding to well data, i.e.,

$$U_{p,b_w}\beta_j = \hat{y}^{a}_{b_w,n,j} - y^{p}_{b_w,n,j}. \tag{4.35}$$

If $\hat{y}^{a}_{b_w,n,j}$ is not consistent with observed facies at one or more well gridblocks, we compute $\hat{y}^{adj}_{b_w,n,j}$ using the same method described in Chapter 3 (Eq. 3.28) and then we can obtain a modified $\hat{\beta}_j$ by solving

$$\hat{y}^{adj}_{b_w,n,j} = y^{p}_{b_w,n,j} + U_{b_w}\hat{\beta}_j \tag{4.36}$$

to obtain $\hat{\beta}_j$ by solving

$$U_{b_w}\hat{\beta}_j = \left(\hat{y}^{adj}_{b_w,n,j} - y^{p}_{b_w,n,j}\right) \tag{4.37}$$

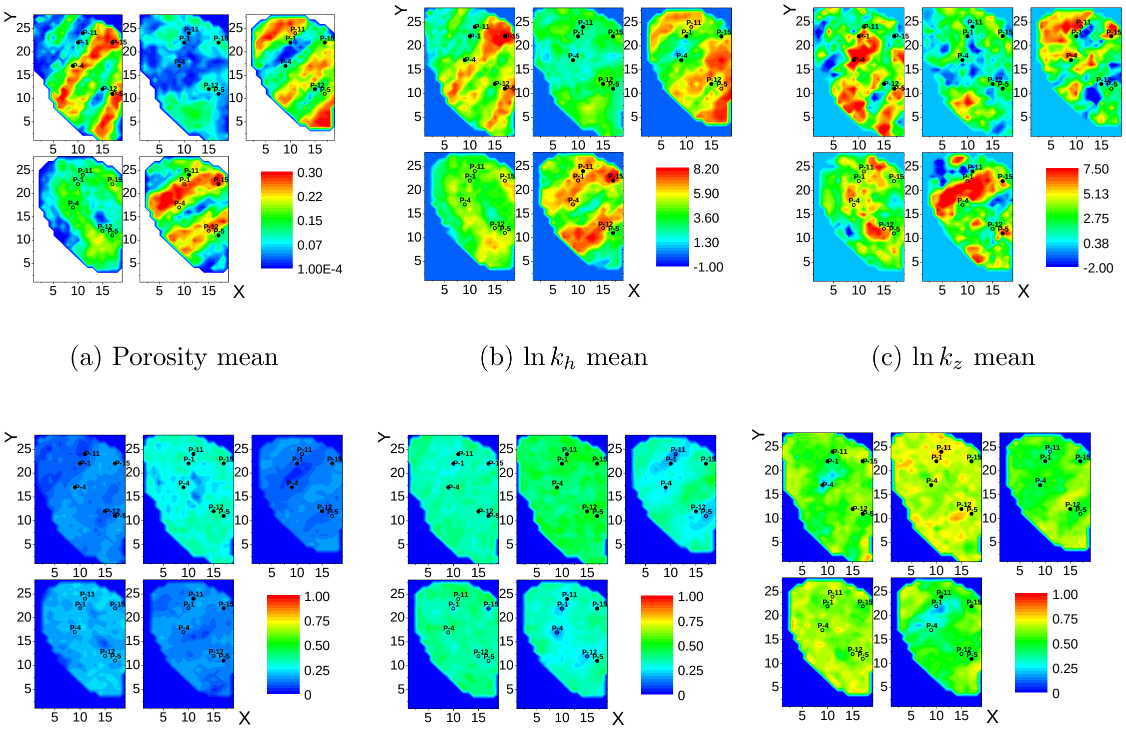
using SVD method. Replacing β_j in Eq. 4.33 with $\hat{\beta}_j$ gives

$$\check{m}^{a}_{n,j} = m^{p}_{n,j} + U_p\hat{\beta}_j, \tag{4.38}$$

where $\check{m}^{a}_{n,j}$ is the projected and adjusted model that honors the hard data. This scheme is applied to each ensemble member.

4.4 Local Analysis Example

As was noted, the local analysis provides more degrees of freedom for matching inverted seismic data. The purpose of this section is to: (1) evaluate the the local analysis scheme in assimilating seismic data in both the Gaussian and pluri-Gaussian geological models; (2) to demonstrate that the projection scheme can preserve smoothness in the conditional models in both the Gaussian and pluri-Gaussian cases.



(a) Porosity mean

(b) $\ln k_h$ mean

(c) $\ln k_z$ mean

(d) Porosity STDR

(e) $\ln k_h$ STDR

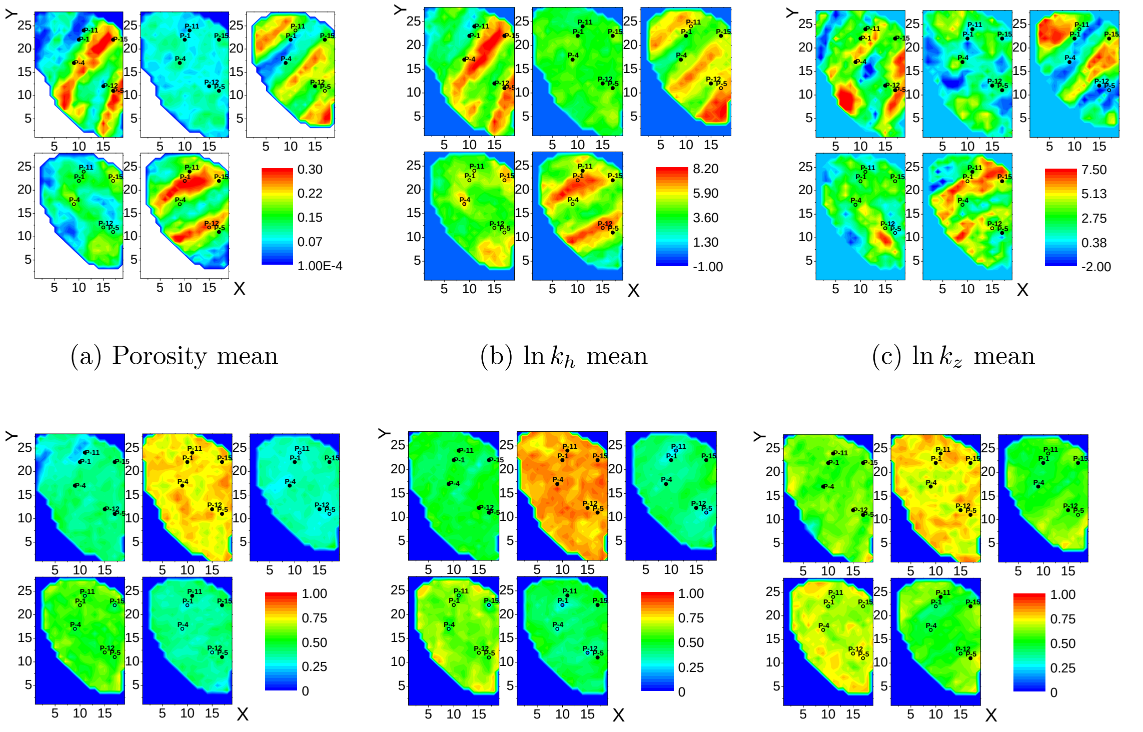
(f) $\ln k_z$ STDR

Figure 4.11: Mean and STDR of conditional fields (PUNQS3, EnKF local analysis, no projection).

We consider all three examples introduced in Chapter 2. For each example, two sets of seismic data as well as production data are assimilated. Local analysis with and without projection will be compared. For the channel model, the results from the case of local analysis with projection will be shown, but the local analysis without projection failed to give continuous channel facies; as a result the reservoir simulator could not be run forward with most ensemble members.

4.4.1 PUNQS3 Example

The conditional models obtained with EnKF for the PUNQS3 example are shown in Figs. 4.11 (without projection) and 4.12 (with projection). The posterior means of porosity and $\ln k_h$ fields for both cases capture the basic structure of the true model (Fig. 4.5), but the ones from the case with projection are much smoother.



(a) Porosity mean

(b) $\ln k_h$ mean

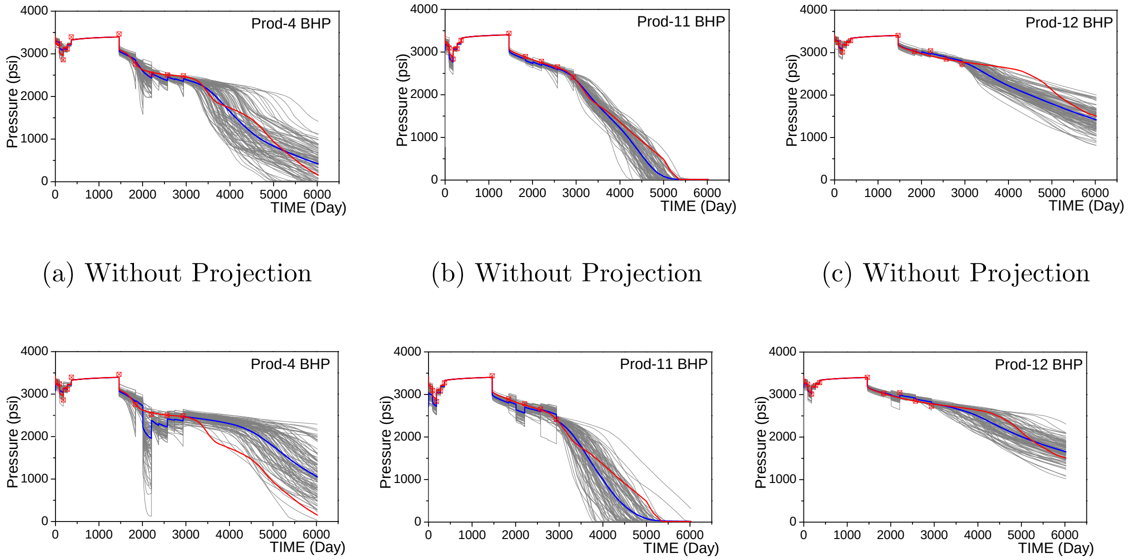
(c) $\ln k_z$ mean

(d) Porosity STDR

(e) $\ln k_h$ STDR

(f) $\ln k_z$ STDR

Figure 4.12: Mean and STDR of conditional fields (PUNQS3, EnKF local analysis, with projection).



(a) Without Projection

(b) Without Projection

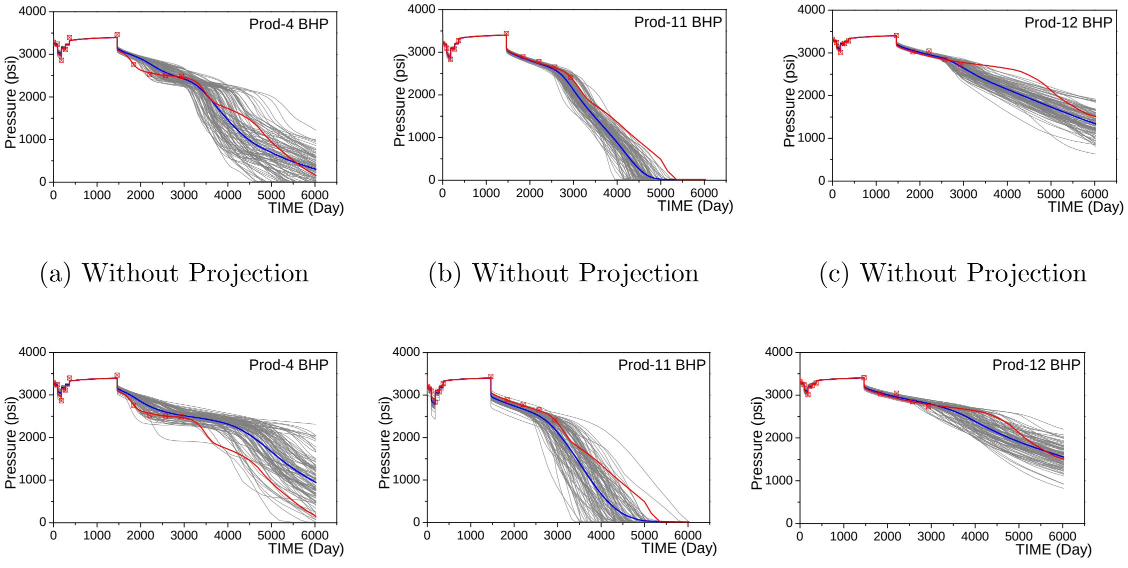
(c) Without Projection

(d) With Projection

(e) With Projection

(f) With Projection

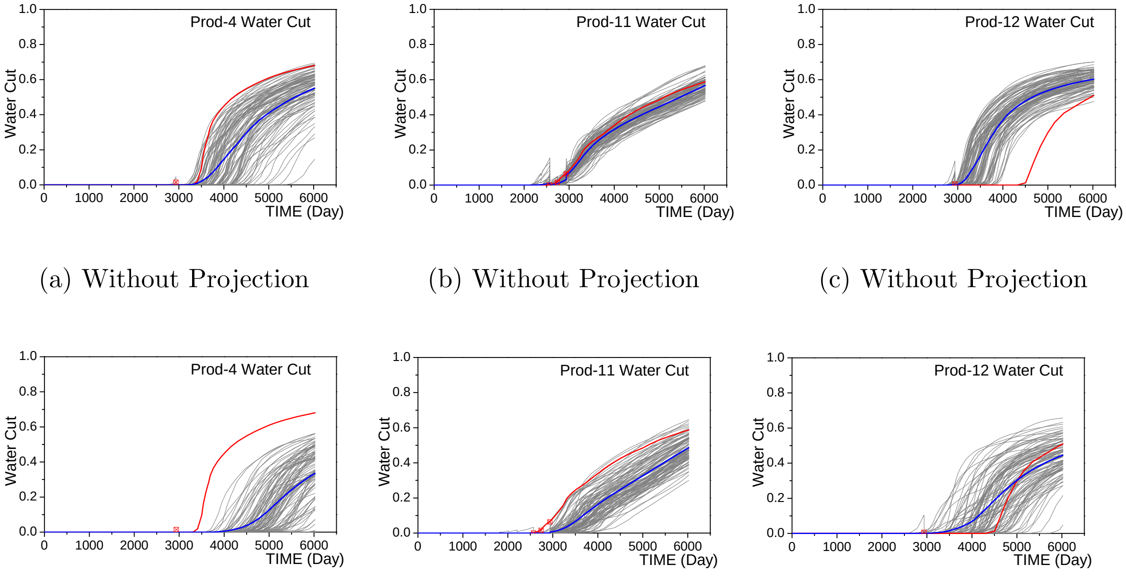
Figure 4.13: Prod4, 11, 12 BHP, assimilation predictions, comparison between with and without projection (PUNQS3, EnKF local analysis).



(a) Without Projection (b) Without Projection (c) Without Projection

(d) With Projection (e) With Projection (f) With Projection

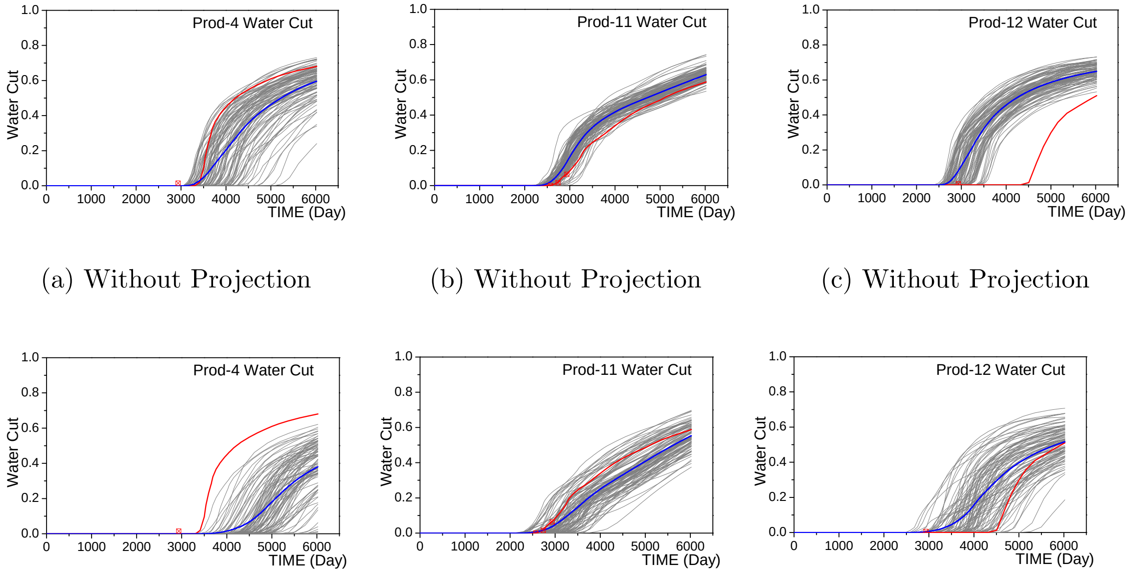
Figure 4.14: Prod4, 11, 12 BHP, predictions from the final ensemble, comparison between with and without projection (PUNQS3, EnKF local analysis).



(a) Without Projection (b) Without Projection (c) Without Projection

(d) With Projection (e) With Projection (f) With Projection

Figure 4.15: Prod4, 11, 12 WCT, assimilation predictions, comparison between with and without projection (PUNQS3, EnKF local analysis).



(a) Without Projection

(b) Without Projection

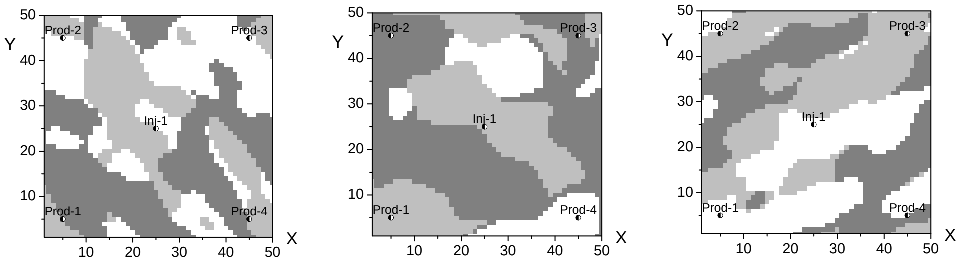
(c) Without Projection

(d) With Projection

(e) With Projection

(f) With Projection

Figure 4.16: Prod4, 11, 12 WCT, predictions from the final ensemble, comparison between with and without projection (PUNQS3, EnKF local analysis).

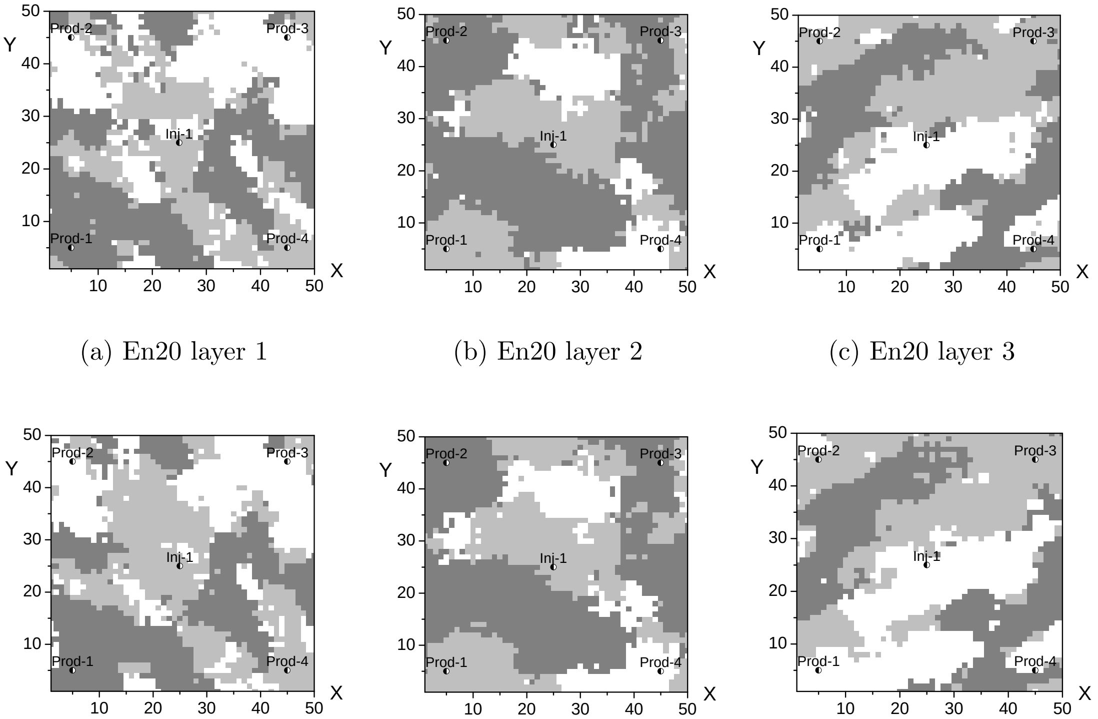


(a) Layer 1

(b) Layer 2

(c) Layer 3

Figure 4.17: Three facies example, true model.



(a) En20 layer 1

(b) En20 layer 2

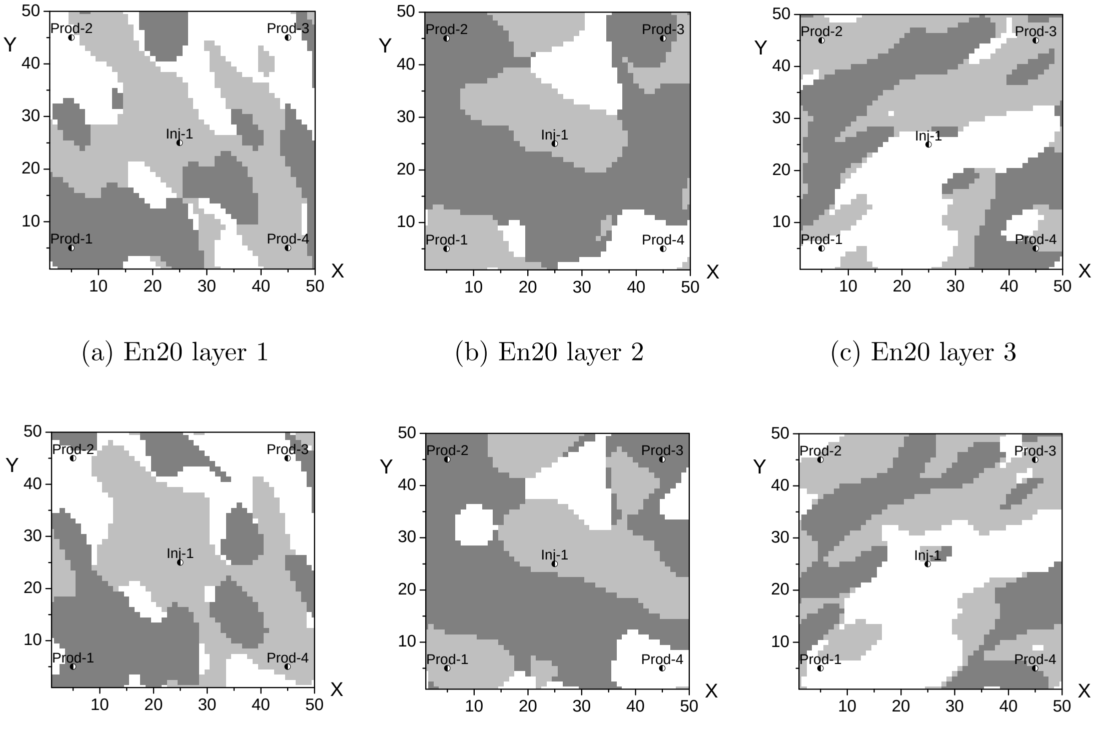
(c) En20 layer 3

(d) En40 layer 1

(e) En40 layer 2

(f) En40 layer 3

Figure 4.18: Two realizations of conditional facies for the case without projection (three facies model, EnKF local analysis).



(a) En20 layer 1

(b) En20 layer 2

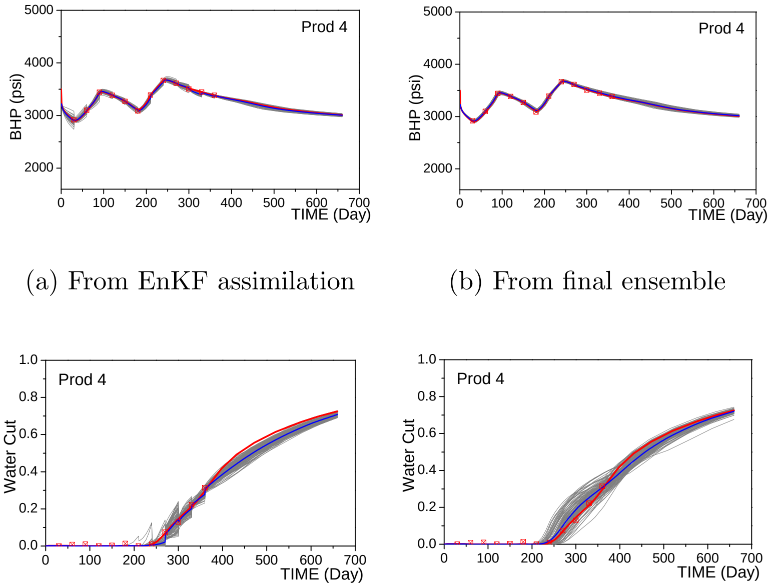
(c) En20 layer 3

(d) En40 layer 1

(e) En40 layer 2

(f) En40 layer 3

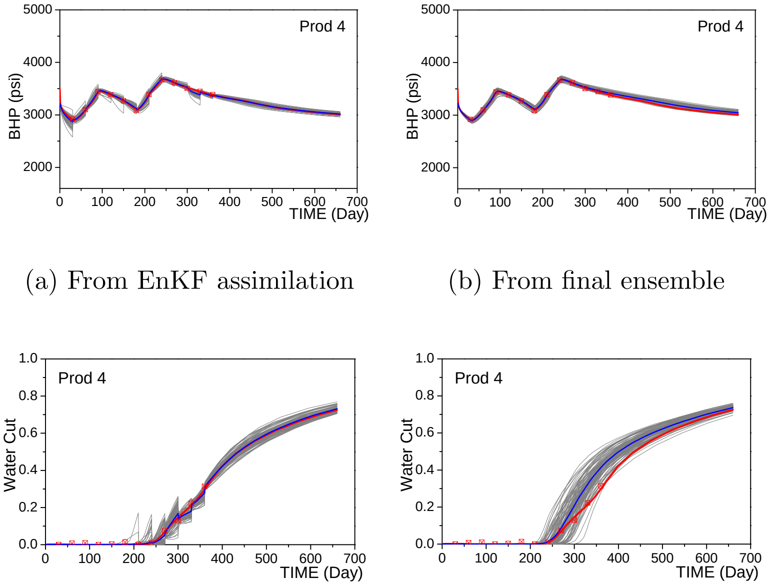
Figure 4.19: Two realizations of conditional facies for the case with projection (three facies model, EnKF local analysis).



(a) From EnKF assimilation (b) From final ensemble

(c) From EnKF assimilation (d) From final ensemble

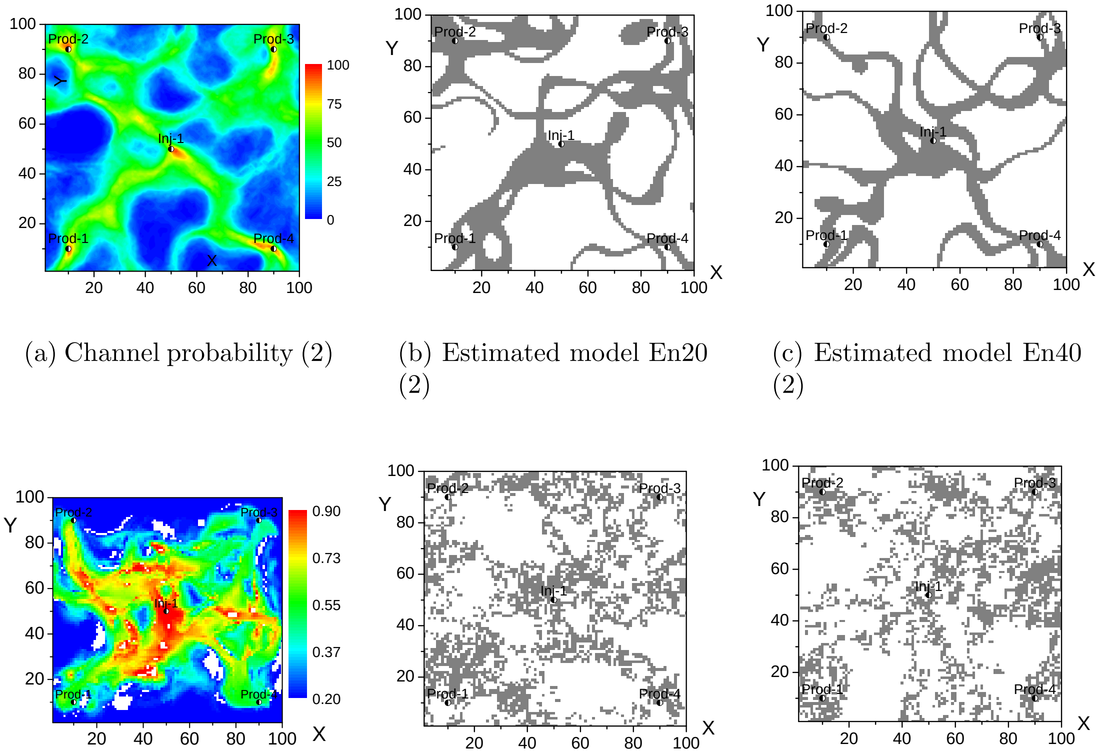
Figure 4.20: Producer 4 BHP and WCT, without projection, (3 facies case, EnKF local analysis).



(a) From EnKF assimilation (b) From final ensemble

(c) From EnKF assimilation (d) From final ensemble

Figure 4.21: Producer 4 BHP and WCT, with projection, (3 facies case, EnKF local analysis).



(a) Channel probability (2)

(b) Estimated model En20 (2)

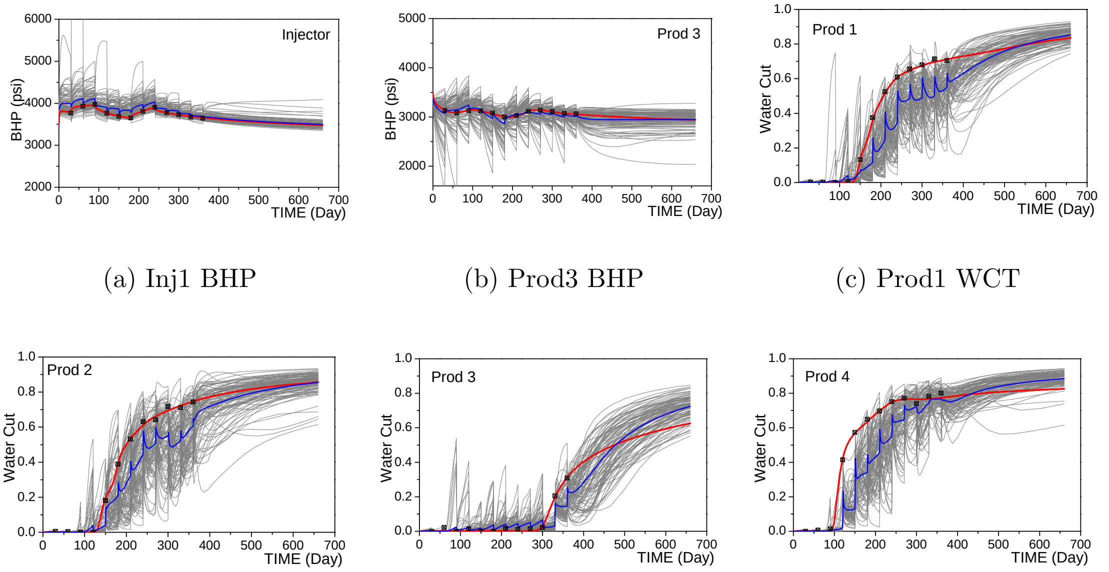
(c) Estimated model En40 (2)

(d) Posterior S_w En20 (2)

(e) Estimated model En20 (1)

(f) Estimated model En40 (1)

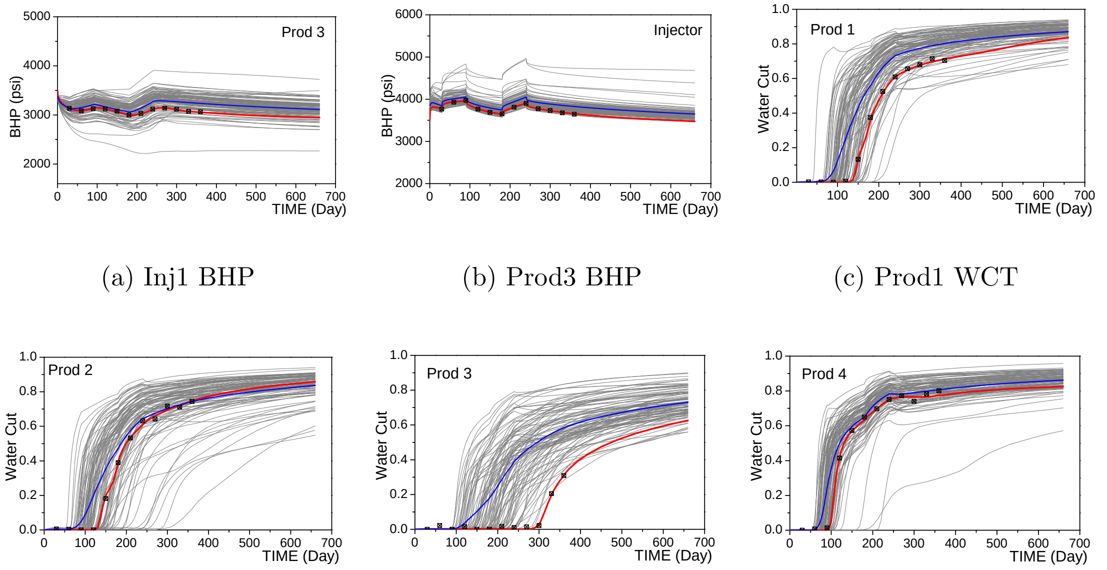
Figure 4.22: Channel probability, two posterior models and water saturations (channel case, EnKF local analysis, (1): without projection (2): with projection).



(a) Inj1 BHP (b) Prod3 BHP (c) Prod1 WCT

(d) Prod2 WCT (e) Prod3 WCT (f) Prod4 WCT

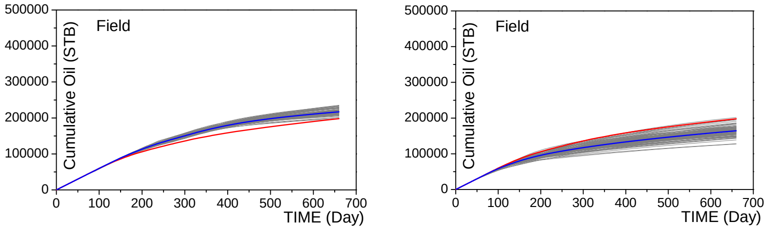
Figure 4.23: Predictions from EnKF assimilation (channel case, EnKF local analysis with projection).



(a) Inj1 BHP (b) Prod3 BHP (c) Prod1 WCT

(d) Prod2 WCT (e) Prod3 WCT (f) Prod4 WCT

Figure 4.24: Predictions from EnKF assimilation (channel case, EnKF local analysis with projection).



(a) EnKF FOPT (b) Posterior FOPT

Figure 4.25: FOPT predictions (channel case, EnKF local analysis with projection).

For the case without projection, the continuity of $\ln k_z$ field is much worse than if projection is used, and additionally more overshooting and undershooting occur compared to the true model (Fig. 4.5). The standard deviation ratios from the case without projection are significantly lower than those from the case with projection, especially for the porosity field. We can see that the local analysis of the seismic data can introduce non-smooth components into the models, and the projection can help to remove this roughness.

The predictions of bottom hole pressure and water cut for producers 4, 11 and 12 from the two cases with and without projection are shown in Figs. 4.13 to 4.16. Unfortunately, overall, no improvement in predictions were obtained using projection.

Comparing the results in the base case of the PUNQS3 example, the local analysis of seismic data with projection give a better estimate of model fields but not significantly more dependable predictions.

4.4.2 Three Facies Example

Fig. 4.18 shows two conditional realizations of the facies distributions for the three facies case without using projection in the local analysis of seismic data. Most structures in the truth case (Fig. 4.17) are captured, but some undesired roughness is introduced into the facies distribution by local analysis. If we use the projection method in the local analysis, the facies maps (shown in Figs. 4.19) are much smoother and also capture most features of the true model.

Figs. 4.20 and 4.21 show the predictions of bottom hole pressure and water cut for producer 4 for both cases. The data matches and predictions are good for both cases, but the uncertainty band for the case with projection is wider because the conditional models are more varied.

4.4.3 Channel Example

In the channel case, local analysis without projection failed to assimilate dynamic data after assimilating the first seismic data set because the channels became discontinuous and for most of the resulting models the simulation would not run. Fig. 4.22 shows the conditional facies distribution of both cases with and without using projection. The plots numbered with a 1 are the conditional facies distributions (seismic data assimilation at time zero) for the case without using projection. We can see that both conditional realizations of the channel facies are highly discontinuous. The plots numbered with a 2 are from the case with projection, in which the channel PDF, two conditional realizations of facies distributions and a realization of estimated water saturation distribution are shown. We can see that the projection results in the channels that are continuous. The conditional models can correctly reflect the connectivity of the reservoir and are more similar to the true model (Fig. 4.10(a)) than what is obtained in the case assimilating only the production data (Fig. 3.21).

Figs. 4.23 and 4.24 show the bottom hole pressure predictions for producers 1 and 3 and water cut predictions for all producers. Compared to the predictions from the base case (Figs. 3.17 and 3.18) there are no significant improvements.

4.5 Summary

Tables 4.1 and 4.2 show the quantitative comparison of the four PUNQS3 cases discussed in this chapter. From the tables, the case with a larger ensemble

Cases	$O^{p,EnKF}_{BHP}$	$O^{p,EnKF}_{GOR}$	$O^{p,EnKF}_{WCT}$	$O^{p,EnKF}_{TOT}$
Global	10.826	23.222	19.222	18.492
Global 300	9.856	16.568	18.028	15.239
Local No Proj.	4.334	5.301	14.065	9.031
Local + Proj.	8.032	23.319	18.418	17.772

Table 4.1: Prediction errors, PUNQS3 example, predicting forward from last data assimilation, EnKF of the production data and seismic data.

Cases	O_m	O^d_{BHP}	O^d_{GOR}	O^d_{WCT}	O^d_{TOT}	O^p_{BHP}	O^p_{GOR}	O^p_{WCT}	O^p_{TOT}
Global	1.036	3.196	4.847	2.806	3.671	11.353	21.471	17.407	17.251
Global 300	0.929	2.207	4.249	1.807	2.858	10.053	14.013	16.169	13.649
Local No Proj.	0.83	1.947	3.216	6.242	2.926	5.61	5.238	15.622	10.049
Local + Proj.	1.164	2.501	4.396	2.998	3.147	8.368	20.35	17.375	16.187

Table 4.2: Model mismatch, data mismatch, prediction error, PUNQS3 example, predicting final ensemble from time zero, EnKF of the production data and seismic data.

Cases	$O^{p,EnKF}_{BHP}$	$O^{p,EnKF}_{WCT}$	$O^{p,EnKF}_{TOT}$
Prior	7.674	6.709	7.261
Prod. Only	1.508	3.506	2.594
Local No Proj.	1.122	2.313	1.754
Local With Proj	1.017	0.904	0.969

Table 4.3: Prediction errors, three facies example, predicting forward from last data assimilation, EnKF of production data and seismic data.

Cases	O_m	O^d_{BHP}	O^d_{WCT}	O^d_{TOT}	O^p_{BHP}	O^p_{WCT}	O^p_{TOT}
Prior	0.605	18.624	13.944	16.707	7.674	6.709	7.261
Prod. Only	0.537	5.967	8.259	7.077	6.508	4.701	5.775
Local No Proj.	0.217	2.682	3.452	3.048	1.596	1.931	1.753
Local With Proj	0.327	3.743	5.454	4.583	4.138	2.665	3.559

Table 4.4: Model mismatch, data mismatch, prediction error, three facies example, predicting final ensemble from time zero, EnKF of production data and seismic data.

Cases	$O^{p,EnKF}_{BHP}$	$O^{p,EnKF}_{WCT}$	$O^{p,EnKF}_{TOT}$
Global	6.091	11.97	9.181
Global 300	6.336	3.455	5.255
Local with Proj.	12.34	4.726	9.723

Table 4.5: Prediction errors, channel example, predicting forward from last data assimilation, EnKF of production data and seismic data.

Cases	O_m	O^d_{BHP}	O^d_{WCT}	O^d_{TOT}	O^p_{BHP}	O^p_{WCT}	O^p_{TOT}
Global	0.417	11.045	32.691	23.297	7.91	16.702	12.599
Global 300	0.437	11.161	13.829	12.418	7.907	5.824	7.058
Local with Proj.	0.41	28.421	21.142	25.444	12.694	8.541	11.043

Table 4.6: Model mismatch, data mismatch, prediction error, channel example, predicting final ensemble from time zero, EnKF of production data and seismic data.

size of 300 gives smaller data mismatches and prediction errors, but as was discussed above, the global analysis with 300 realizations still leads to filter divergence, i.e., too narrow and wrong predictions. Compared to the global analysis of seismic data, the local analysis gives better data matches and predictions. The model mismatch, data mismatches and prediction errors from the local analysis case with projection are higher than those from the case without projection, but the conditional models are smoother. The model mismatches from these four cases are all much lower than those from the cases in Chapter 3 (Table 3.5) indicating better estimation of models.

Tables 4.3 and 4.4 show the quantitative comparison of the three facies cases discussed up to now including the base case discussed in Chapter 3. We can see that although the local analysis case with projection gives a more reasonable model (good continuity), the model mismatch, data mismatches and prediction errors are all bigger than those from the case without projection (except the predictions from the final EnKF assimilation state). Compared to the case assimilating only the production data, assimilating additional seismic data gives a better model, better data match and better predictions.

Tables 4.5 and 4.6 show the quantitative comparison of the channel cases discussed in this chapter. We can see that global analysis with 300 realizations gave model mismatch, data mismatches and prediction errors significantly smaller than obtained in the other cases including those from the local analysis cases, but the conditional models from the global analysis are obviously too close to each other and converged to a wrong model. The local analysis with projection gives smaller model mismatch than the other cases, but worse data matches and predictions.

CHAPTER 5

ENSEMBLE RANDOMIZED MAXIMUM LIKELIHOOD METHOD

The ensemble Kalman filter (EnKF) assimilates production and seismic data sequentially in time, and in each data assimilation step model parameters and simulation primary variables are adjusted to match the data measured at the assimilation step [24]. In this way, the predictions for the next data assimilation step can be obtained by restarting the simulator using the analyzed models and simulation primary variables without running from time 0. This is why the process is fast. However, EnKF may give inaccurate predictions because primary variables and data are nonlinear functions of the model parameters, and the predictions may not satisfy a Gaussian distribution which is a fundamental assumption used to define the EnKF analysis equation. Each EnKF analysis step is similar to one Gauss-Newton update [68], which gives the proper minimum of the objective function related only if the system is linear [77]. As a result, the conditional primary variables may be inconsistent with the corresponding conditional model parameters. As is shown in Chapter 3, this is especially obvious in the channel cases where the conditional saturations and models are inconsistent. The normal score transform did not fix this problem because it was designed to deal with non-Gaussian distribution of the state variables, not non-linearity. An iterative scheme [49, 86] can be an option to solve this problem because (i) the predictions are obtained by simulating the ensemble members from time zero rather than from the analyzed state, and hence the predictions will always be consistent with the corresponding models; (ii) iterative schemes can handle non-linearity.

In this study, an iterative scheme introduced by Gu and Oliver [34] (we call

Gu-Oliver scheme or ensemble randomized maximum likelihood (EnRML)) is tested, which also assimilate data sequentially in time. This scheme is a modification of an iterative method presented by Zupanski [86]. This method is effectively the Gauss-Newton method except each realization is updated using an approximation to the average sensitivity matrix, rather than the sensitivity matrix pertaining to the particular model being updated. This method offers advantages for this study, because the gradient of the objective function does not exist on the facies boundary for the pluri-Gaussian model. The problem that we have sometimes encountered with the Gu-Oliver procedure [34] is that when the same step size is used for all ensemble members, some accepted updated models give a worse data match than is obtained with the corresponding prior models. Therefore, in this study, the EnRML scheme is modified by using individual line search for each ensemble member to guarantee that the updated model gives a better match of the observed data at each iteration. We call this scheme as modified Gu-Oliver scheme with individual line search.

5.1 Ensemble Randomized Maximum Likelihood (EnRML)

At the nth data assimilation time, the matrix of predicted models is denoted by $M_n^p = \left[m_{n,1}^p, m_{n,2}^p, \cdots, m_{n,N_e}^p\right]$ with $m_{n,j}^p$ denoting N_m-dimensional column vectors of the predicted model parameters for the jth ensemble member. Noisy data $(d_{uc,n,j})$ and covariance matrices are the same as defined in Eqs. 3.4 and 3.14. We want to obtain the set of analyzed models $M_n^a = [m_{n,1}^a, m_{n,2}^a, \cdots, m_{n,N_e}^a]$ by minimizing the objective function

$$\begin{aligned} O_j(m_{n,j}) =& \frac{1}{2}\left(m_{n,j} - m_{n,j}^p\right)^T C_{M_n}^{-1}\left(m_{n,j} - m_{n,j}^p\right) \\ &+ \frac{1}{2}\left(g(m_{n,j}) - d_{uc,n,j}\right)^T C_{D,n}^{-1}\left(g(m_{n,j}) - d_{uc,n,j}\right), \end{aligned} \tag{5.1}$$

for $j = 1, 2, \cdots, N_e$. We let $m_{n,j}^a$ denote the minimizing model, i.e., $m_{n,j}^a = \text{Argmin}[O_j(m_{n,j})]$ for $j = 1, 2, \cdots, N_e$. Here, the function g is the system equation for the predicted data, i.e. $d_{n,j} = g(m_{n,j})$, and $C_{M,n}$ is the covariance matrix of the prior model at the nth data assimilation step. Note we have replaced the parameter-state estimation problem used in EnKF by a problem where we only care about the estimates of realizations of the model.

By viewing EnKF as an optimization algorithm, Reynolds et al. [68] were able to derive an iterative EnKF method. This iterative method or equivalently the first iterative EnKF method of Li and Reynolds [49] is based on the following equation for assimilating data at the nth data assimilation time (time t_n):

$$\delta m_{n,j}^{\ell+1} = -\Big[C_{M_n} + C_{M_n} G_{n,j,\ell}^T \big(C_{D_n} + G_{n,j,\ell} C_{M_n} G_{n,j,\ell}^T\big)^{-1} G_{n,j,\ell} C_{M_n}\Big] \\ \big\{C_{M_n}^{-1}(m_{n,j}^{\ell} - m_{n,j}^{p}) + G_{n,j,\ell}^T C_{D_n}^{-1}(d_n(m_{n,j}^{\ell}) - d_{uc,n,j})\big\}, \tag{5.2}$$

which represents the ℓth ($\ell = 0, 1, 2, \cdots$) Gauss-Newton iteration for the randomized-maximum-likelihood method (RML) for generating samples of the pdf for the model conditional to observed data. Eq. 5.2 gives the search direction for Gauss-Newton iteration. The new updated model of parameters is obtained from

$$m_{n,j}^{\ell+1} = m_{n,j}^{\ell} + \alpha_j^{\ell+1} \delta m_{n,j}^{\ell+1}, \tag{5.3}$$

where $\alpha_j^{\ell+1}$ represents the step-size which can be determined by a line search procedure. At this point, $G_{n,j,\ell}$ denotes the gradient of predicted data d_n with respect to m evaluated at $m_{n,j}^{\ell}$, and $m_{n,j}^{p}$ denotes the jth realization obtained by conditioning to data up to and including the $(n-1)$st data assimilation time. Note $G_{n,j,\ell}$ is simply a sensitivity matrix related to the predicted data at the nth assimilation time.

The term in square brackets in Eq. 5.2 represents the inverse of the Gauss-Newton Hessian, and the term in braces represents the gradient of the objective function that is being minimized; in the iterative method of Reynolds et al. [68], the term $G^T_{n,j,\ell}C^{-1}_{D_n}(d_n(m^\ell_{n,j})-d_{uc,n,j})$ is computed by an adjoint method, as implemented by Li et al. [51] and Zhang and Reynolds [82] and the Hessian inverse is evaluated in the same way as the corresponding terms in the EnKF procedure.

As in EnKF, the iterative scheme is a sequential data assimilation method. Eq. 5.2 is for the assimilation of data at the nth data assimilation step (time t_n). Each iteration, however, requires a simulation run from time zero to time t_n and a corresponding adjoint solution using the adjoint implementation of Li et al. [51]. Li and Reynolds [49] suggest a much more computationally efficient iterative scheme that requires only rerunning the simulator from the previous data assimilation step at each iteration.

As mentioned previously, however, for the truncated pluri-Gaussian model the gradient is undefined on facies boundaries (Chapter 1). One possibility for using the basic algorithm of Eq. 5.2 is to replace all individual sensitivity matrices by an average sensitivity matrix $\overline{G}_{n,\ell}$, where $\overline{G}_{n,\ell}$ can be estimated as suggested by Zupanski [86] and Gu and Oliver [34]. Define the deviation matrix of models and predictions in the ℓth iteration as

$$\begin{aligned}\Delta M^\ell_n =& M^\ell_n - \overline{M^\ell_n}\\ \Delta D^\ell_n =& D^\ell_n - \overline{D^\ell_n},\end{aligned} \tag{5.4}$$

where $D^\ell_n = \left[d^\ell_{n,1}, d^\ell_{n,2}, \cdots, d^\ell_{n,N_e}\right]$ is the predicted data matrix with $d^\ell_{n,j} = g\left(m^\ell_{n,j}\right)$, and $\overline{M^\ell_n}$ and $\overline{D^\ell_n}$ respectively, are the matrices with each column equal to the average of the columns in M^ℓ_n and D^ℓ_n, respectively. An approximation to the average sensitivity matrix can be found by solving

$$(\Delta M^\ell_n)^T \overline{G}^T_{n,\ell} = (\Delta D^\ell_n)^T \tag{5.5}$$

using singular value decomposition. We can apply standard matrix inversion lemma

[76] to rewrite Eq. 5.2 as

$$\delta m_{n,j}^{\ell+1} = m_{n,j}^{p} - m_{n,j}^{\ell} + C_{M_n}\overline{G}_{n,\ell}^{T}\big(C_{D_n} + \overline{G}_{n,\ell}C_{M_n}\overline{G}_{n,\ell}^{T}\big)^{-1} \\ \big(d_{uc,n,j} - d_n(m_{n,j}^{\ell}) + \overline{G}_{n,\ell}(m_{n,j}^{\ell} - m_{n,j}^{p})\big). \tag{5.6}$$

Thus, we can compute a search direction with Eq. 5.6 and then use the same step size to update every model for $j = 1, 2, \cdots N_e$, i.e., we replace Eq. 5.3 by

$$m_{n,j}^{\ell+1} = m_{n,j}^{\ell} + \alpha^{\ell+1}\delta m_{n,j}^{\ell+1}. \tag{5.7}$$

This is the procedure used by Gu and Oliver [34]. After each $m_{n,j}^{\ell+1}$ is computed, they evaluate a term which effectively represents the average data mismatch. To evaluate this average data mismatch requires rerunning the simulator from time zero with each proposed updated ensemble member, $m_{n,j}^{\ell+1}$, $j = 1, 2, \cdots N_e$. If the resulting average data mismatch is less than the one obtained with the ensemble $m_{n,j}^{\ell}$, $j = 1, 2, \cdots N_e$, then the set $m_{n,j}^{\ell+1}$, $j = 1, 2, \cdots N_e$ is accepted as the updated ensemble and $\alpha^{\ell+1}$ is increased by some undefined factor as the initial guess for the step size at the next iteration. If the new proposed ensemble results in an increase, then $\alpha^{\ell+1}$ is decreased by some undefined factor and, using this smaller step size, Eq. 5.3 is reapplied to all ensemble members.

At each iteration, some of the final m_j's are equal to the normal EnKF update. In this iterative scheme, the "prior" model covariance C_{M_n} is based on the ensemble of models obtained at the previous data assimilation step and does not change during the iterations. As in EnKF, this covariance matrix is never formally generated and is represented by the ensemble $m_{n,j}^{p}$, $j = 1, 2, \cdots N_e$, i.e.

$$C_{M,n} = \frac{1}{N_e - 1}\Delta M_n^p \left(\Delta M_n^p\right)^T, \tag{5.8}$$

where we use deviation matrix of the prior model for the nth data assimilation step $\Delta M_n^p = M_n^p - \overline{M_n^p}$, where $\overline{M_n^p}$ is a matrix with each column equal to the average of the columns of M_n^p. Combining Eqs. 5.6 and 5.7, and applying Eq. 5.8 we obtain

$$
\begin{aligned}
m_{n,j}^{\ell+1} =& \alpha^{\ell+1} m_{n,j}^p + (1-\alpha^{\ell+1}) m_{n,j}^\ell \\
& + \alpha^{\ell+1} C_{M_n} \overline{G}_{n,\ell}^T \left(\overline{G}_{n,\ell} C_{M_n} \overline{G}_{n,\ell}^T + C_{D,n} \right)^{-1} \left[d_{uc,n,j} - g(m_{n,j}^\ell) + \overline{G}_{n,\ell} \left(m_{n,j}^\ell - m_{n,j}^p \right) \right] \\
=& \alpha^{\ell+1} m_{n,j}^p + (1-\alpha^{\ell+1}) m_{n,j}^\ell \\
& + \alpha^{\ell+1} \Delta M_n^p \left(\overline{G}_{n,\ell} \Delta M_n^p \right)^T \left[(\overline{G}_{n,\ell} \Delta M_n^p)(\overline{G}_{n,\ell} \Delta M_n^p)^T + (N_e - 1) C_{D,n} \right]^{-1} \\
& \left[d_{uc,n,j} - g(m_{n,j}^\ell) + \overline{G}_{n,\ell} \left(m_{n,j}^\ell - m_{n,j}^p \right) \right]
\end{aligned}
\tag{5.9}
$$

with initial guess of model $m_{n,j}^0 = m_{n,j}^p$. The analysis Eq. 5.9 can be written as:

$$
\begin{aligned}
m_{n,j}^{\ell+1} =& \alpha^{\ell+1} m_{n,j}^p + (1-\alpha^{\ell+1}) m_{n,j}^\ell \\
& + \alpha^{\ell+1} \Delta M_n^p \left(\Delta D_n^{\ell-} \right)^T \left[\left(\Delta D_n^{\ell-} \right) \left(\Delta D_n^{\ell-} \right)^T + (N_e - 1) C_{D,n} \right]^{-1} \left[d_{uc,n,j} - \Delta d_{n,j}^{\ell+} \right],
\end{aligned}
\tag{5.10}
$$

where the definitions of $\Delta D_n^{\ell-}$ and $\Delta d_{n,j}^{\ell+}$ are

$$
\begin{aligned}
\Delta D_n^{\ell-} =& G_{n,\ell} \Delta M_n^p \\
\Delta d_{n,j}^{\ell+} =& g(m_{n,j}^\ell) - \overline{G_{n,\ell}} \left(m_{n,j}^\ell - m_{n,j}^p \right).
\end{aligned}
\tag{5.11}
$$

We can rewrite Eq. 5.10 as

$$
m_{n,j}^{\ell+1} = \alpha^{\ell+1} m_{n,j}^p + (1-\alpha^{\ell+1}) m_{n,j}^\ell + \alpha^{\ell+1} \overline{\delta m}_{n,j}^{\ell+1}, \tag{5.12}
$$

where

$$\overline{\delta m}_{n,j}^{\ell+1} = \Delta M_n^p \left(\Delta D_n^{\ell-}\right)^T \left[\left(\Delta D_n^{\ell-}\right)\left(\Delta D_n^{\ell-}\right)^T + (N_e - 1)C_{D,n}\right]^{-1} \left[d_{uc,n,j} - \Delta d_{n,j}^{\ell+}\right]. \tag{5.13}$$

The vector $\overline{\delta m}_{n,j}^{\ell+1}$ could be computed using the same subroutine as the standard EnKF (correction term in Eq. 3.15). Note that in the first iteration, $G_{n,0}$ need not to be computed because if we use the initial given $m_{n,j}^0 = m_{n,j}^p$, then $\Delta D_n^{0-} = \Delta D_n^p$ (Eq. 5.5) and $\Delta d_{n,j}^{0+} = g(m_{n,j}^p)$.

In general, there is no reason to believe that Eq. 5.6, which uses an approximate average sensitivity matrix, gives a downhill direction from all $m_{n,j}^{\ell}$, and for the problems considered here, it does not. Thus, we implemented an individual line search for each ensemble member update, i.e., we use Eq. 5.3 in place of Eq. 5.7, and Eq. 5.10 for individual line search can be written as

$$\begin{aligned} m_{n,j}^{\ell+1} =& \alpha_j^{\ell+1} m_{n,j}^p + (1-\alpha_j^{\ell+1}) m_{n,j}^{\ell} \\ &+ \alpha_j^{\ell+1} \Delta M_n^p \left(\Delta D_n^{\ell-}\right)^T \left[\left(\Delta D_n^{\ell-}\right)\left(\Delta D_n^{\ell-}\right)^T + (N_e - 1)C_{D,n}\right]^{-1} \left[d_{uc,n,j} - \Delta d_{n,j}^{\ell+}\right], \end{aligned} \tag{5.14}$$

where $\alpha_j^{\ell+1}$ is the step size for the jth ensemble member at the nth data assimilation time. At the first iteration, we always attempt a unit step, and the update always corresponds to the standard EnKF. At an iteration, if we do not obtain a decrease in the objective function for a specific ensemble member, then we halve the step size for line search. We will refer to this procedure as modified Gu-Oliver or Gu-Oliver with individual line search.

For the pluri-Gaussian geological model, if a full step size is used ($\alpha_j^{\ell+1} = 1$), exactly the same procedure of adjustment described in Chapter 3 can be used to guarantee the updated model $m_{n,j}^{\ell+1}$ honors the hard data. In applying the adjustment scheme of Chapter 3 for the EnRML scheme, the predicted model corresponds to

$m_{n,j}^p$, and the analyzed model corresponds to $m_{n,j}^{\ell+1}$ where $\alpha_j^{\ell+1} = 1$. If during the line search, the step size is decreased, then $m_{n,j}^{\ell+1}$ might violate the hard data. If this should happen to a specific ensemble member, the proposed model will be rejected and a smaller step size will be tried as line search. If the smallest step size still fails to honor the hard data or fails to give smaller the objective function, the most recently accepted model for this ensemble member will be used, i.e., $m_{n,j}^{\ell+1} = m_{n,j}^{\ell}$.

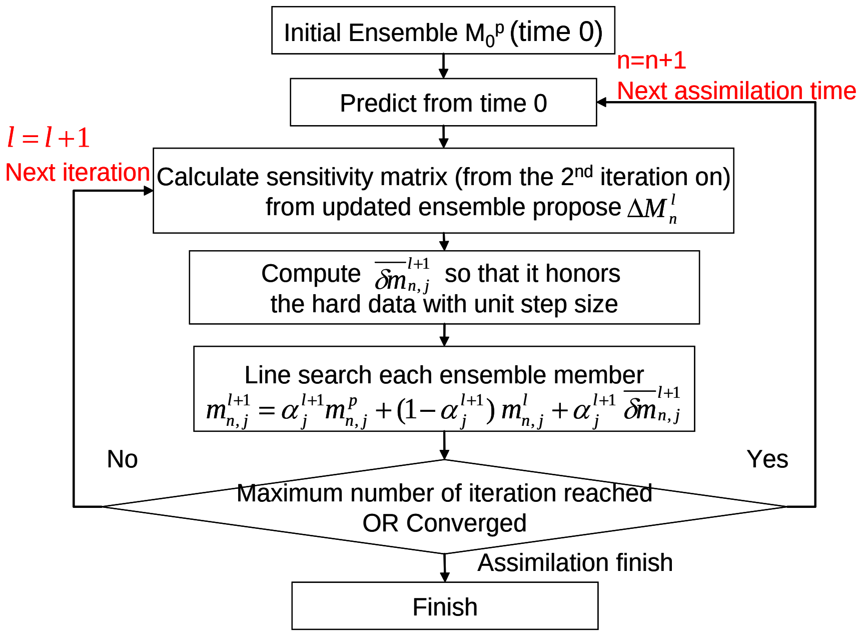

Initial Ensemble M_0^p (time 0)
n=n+1
Next assimilation time
Predict from time 0
$l = l + 1$
Next iteration
Calculate sensitivity matrix (from the 2nd iteration on)
from updated ensemble propose ΔM_n^l
Compute $\overline{\delta m}_{n,j}^{l+1}$ so that it honors
the hard data with unit step size
Line search each ensemble member
$m_{n,j}^{l+1} = \alpha_j^{l+1} m_{n,j}^p + (1-\alpha_j^{l+1})\, m_{n,j}^l + \alpha_j^{l+1}\, \overline{\delta m}_{n,j}^{l+1}$
No
Yes
Maximum number of iteration reached
OR Converged
Assimilation finish
Finish

Figure 5.1: The scheme of EnRML with line search.

Fig. 5.1 shows the EnRML scheme used in this study. Each analysis is composed of multiple iterations, and each iteration is a line search process for each ensemble member. In this figure, the outer loop index of n is for the data assimilation time steps, and the inner loop index of ℓ is for the iterations of each ensemble member which is repeated for $j = 1, 2, \cdots, N_e$. The line search is performed in the inner loop, which is based on the objective function constructed from the data mismatches for each ensemble member, i.e.

$$O_d(m_{n,j}) = \frac{1}{N_{d,n}} \left(g(m_{n,j}) - d_{uc,n,j}\right)^T C_{D,n}^{-1} \left(g(m_{n,j}) - d_{uc,n,j}\right). \qquad (5.15)$$

The line search at the ℓth iteration follows the following rules:

1. If the objective function is smaller than 1.1 for an ensemble member, i.e. $O_d(m_{n,j}^{\ell+1}) < 1.1$, then we set $m_{n,j}^a = m_{n,j}^{\ell+1}$ and no further iterations are done on this ensemble member. However, this realization will still be used to compute the average sensitivity matrix for the followed iterations;

2. The initial step sizes for each data assimilation time step and each ensemble member are all set equal to 1.0. From the second iteration on, the step sizes are same as those obtained at the previous iteration;

3. If the proposed model $(m_{n,j}^{\ell+1})$ fails to give lower objective function value or is invalid (does not honor the hard data in pluri-Gaussian case), we will sequentially try smaller step sizes as $\frac{1}{2}$, $\frac{1}{4}$, and $\frac{1}{8}$ until we get a model with a lower objective function value. If the line search fails for a step size of $\frac{1}{8}$, i.e. do not decrease the objective function, the analyzed model is set equal to the most recently accepted model, i.e. $m_{n,j}^{\ell+1} = m_{n,j}^{\ell}$;

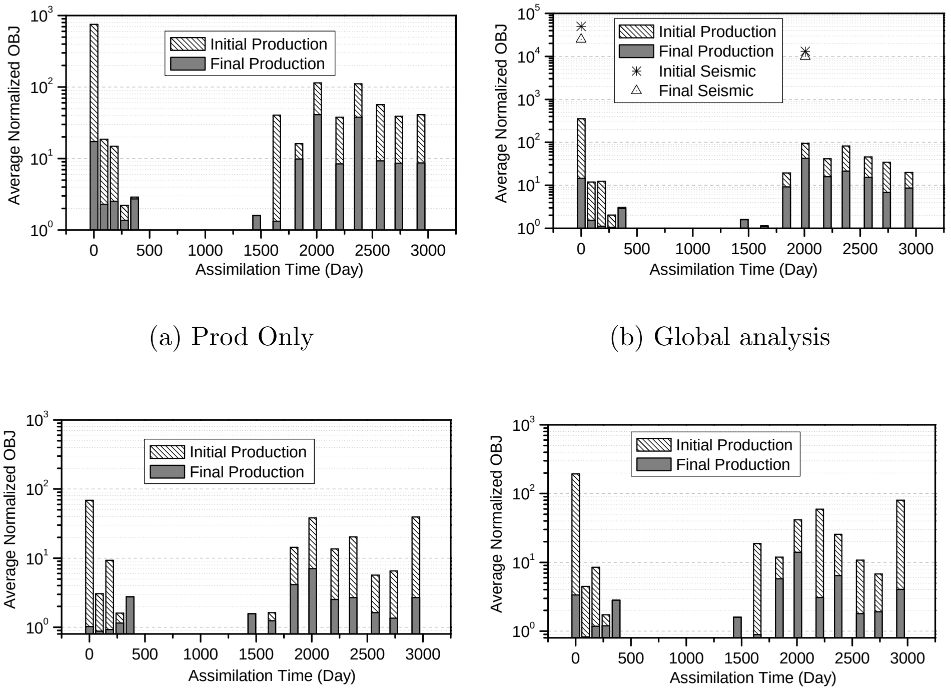
4. If the line search is successful, the step size is multiplied by 2, and we set the maximum step size equals to 1.0.

The iteration procedure will stop if one of the following two conditions are satisfied:

1. All the ensemble members converge $(O_d(m_{n,j}^{\ell+1}) < 1.1$ for all $j)$;

2. The maximum number of iteration (2 iterations) is reached and the decrease in average normalized objective function (Eq. 5.16) is less than 10 percent of that obtained in the previous iteration, i.e., if $\frac{\bar{O}(m_n^{\ell})-\bar{O}(m_n^{\ell+1})}{\bar{O}(m_n^{\ell})} \geq 0.1$, the iteration will continue even if the maximum number of iteration is reached, here

$$\bar{O}_d(m_n) = \frac{1}{N_e}\sum_{j=1}^{N_e} O_d(m_{n,j}). \tag{5.16}$$



(a) Prod Only

(b) Global analysis

(c) Local without projection

(d) Local with projection

Figure 5.2: Comparison of objective functions for the four cases (PUNQS3 case, iterative).

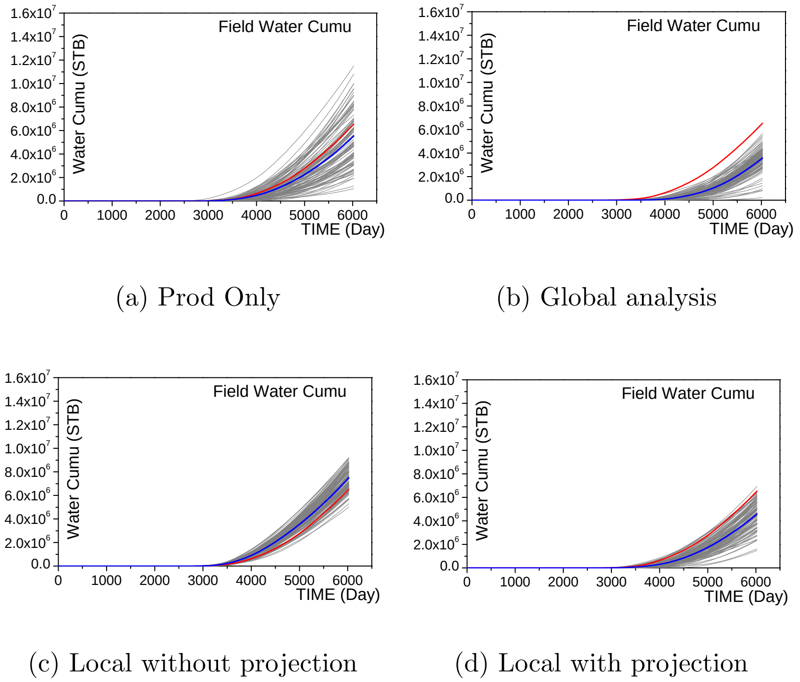
In the following cases, when the seismic data are assimilated with global analysis we use the iterative scheme, and if the local analysis is used, we use the EnKF analysis scheme, and the predictions for the next data assimilation time are obtained by simulating the updated models from time zero.

5.2 EnRML Results for PUNQS3 Example

This example pertains to results from the PUNQS3 example obtained us-

Cases	O_m	O^d_{BHP}	O^d_{GOR}	O^d_{WCT}	O^d_{TOT}	O^p_{BHP}	O^p_{GOR}	O^p_{WCT}	O^p_{TOT}
Prod. Only	1.457	2.505	4.285	3.164	3.123	7.893	13.323	16.004	12.857
Global	1.036	3.196	4.847	2.806	3.671	11.353	21.471	17.407	17.251
Local No Proj.	0.83	1.947	3.216	6.242	2.926	5.61	5.238	15.622	10.049
Local + Proj.	1.164	2.501	4.396	2.998	3.147	8.368	20.35	17.375	16.187

Table 5.1: Model mismatch, data mismatch, prediction error, PUNQS3 example, predicting final ensemble from time zero, EnKF copied from the previous chapters



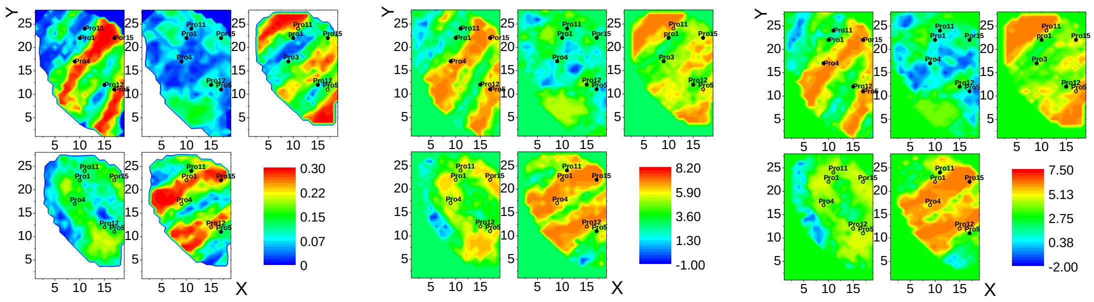
(a) Prod Only

(b) Global analysis

(c) Local without projection

(d) Local with projection

Figure 5.3: Comparison of field cumulative water for the four cases (PUNQS3 case, iterative).



(a) Porosity

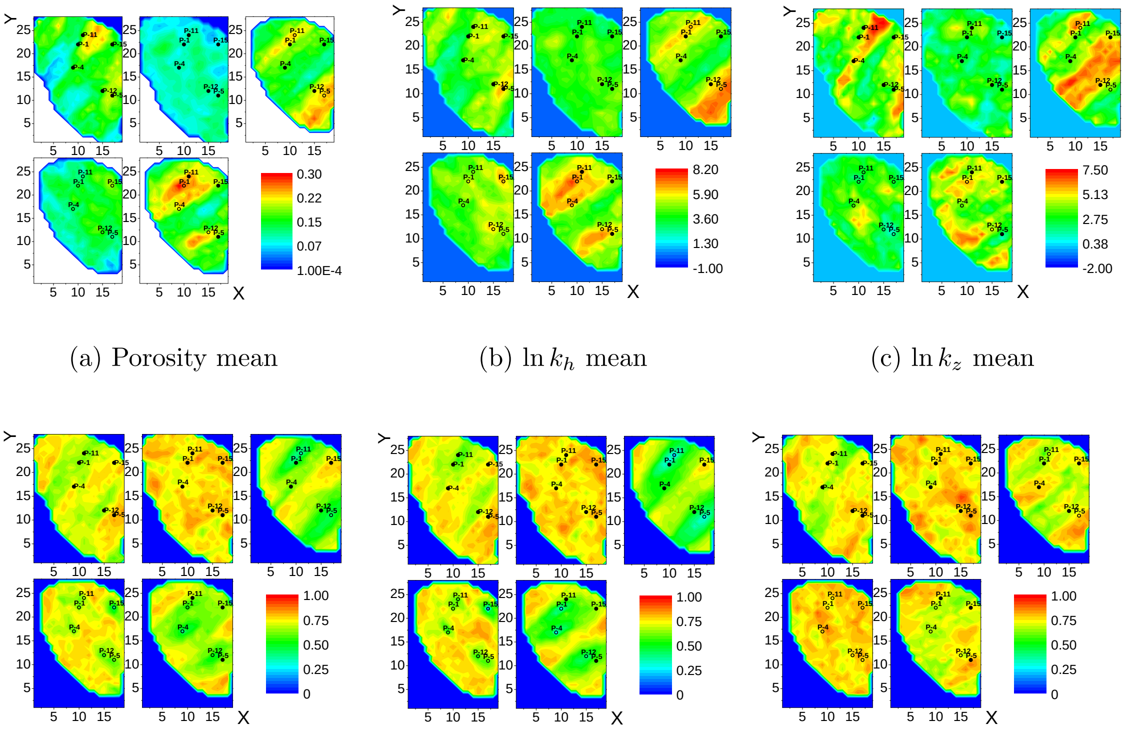
(b) $\ln k_h$

(c) $\ln k_z$

Figure 5.4: True model of PUNQS3: porosity, $\ln k_h$, $\ln k_z$.

Cases	O_m	O^d_{BHP}	O^d_{GOR}	O^d_{WCT}	O^d_{TOT}	O^p_{BHP}	O^p_{GOR}	O^p_{WCT}	O^p_{TOT}
Prod. Only	1.37	2.333	3.994	2.756	2.894	6.275	12.636	9.815	9.922
Global	1.285	2.311	4.321	2.783	3.005	7.671	14.434	18.353	14.189
Local No Proj.	0.79	1.588	3.286	1.66	2.167	7.771	12.539	11.152	10.152
Local With Proj.	1.156	1.828	3.276	2.279	2.33	5.437	16.37	13.22	12.547

Table 5.2: Model mismatch, data mismatch, prediction error, PUNQS3 example, predicting final ensemble from time zero, EnRML



(a) Porosity mean

(b) $\ln k_h$ mean

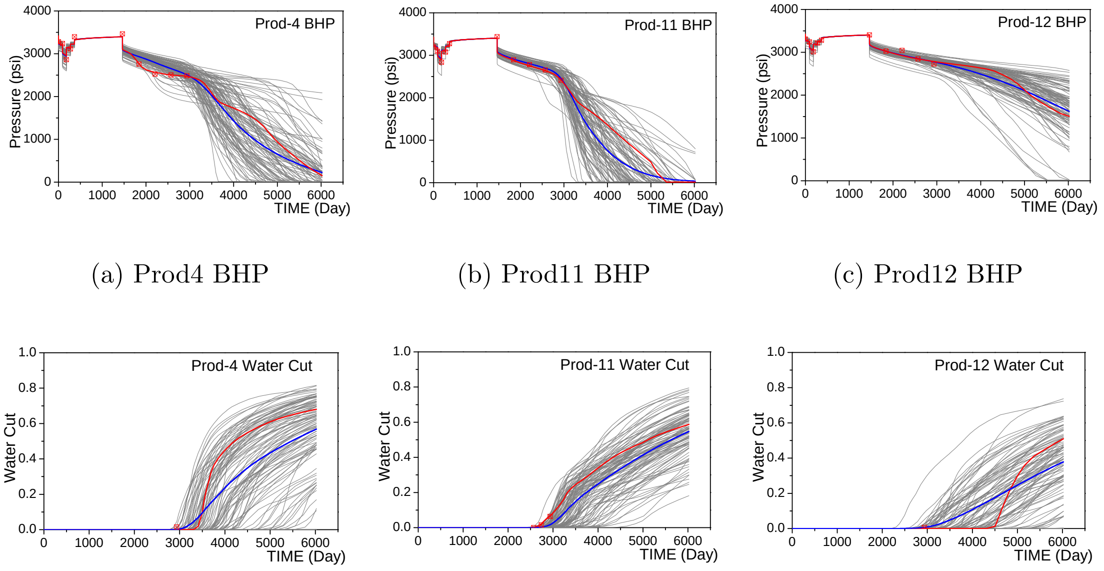
(c) $\ln k_z$ mean

(d) Porosity STDR

(e) $\ln k_h$ STDR

(f) $\ln k_z$ STDR

Figure 5.5: Mean and STDR of conditional fields (PUNQS3, assimilating production data, iterative).



(a) Prod4 BHP

(b) Prod11 BHP

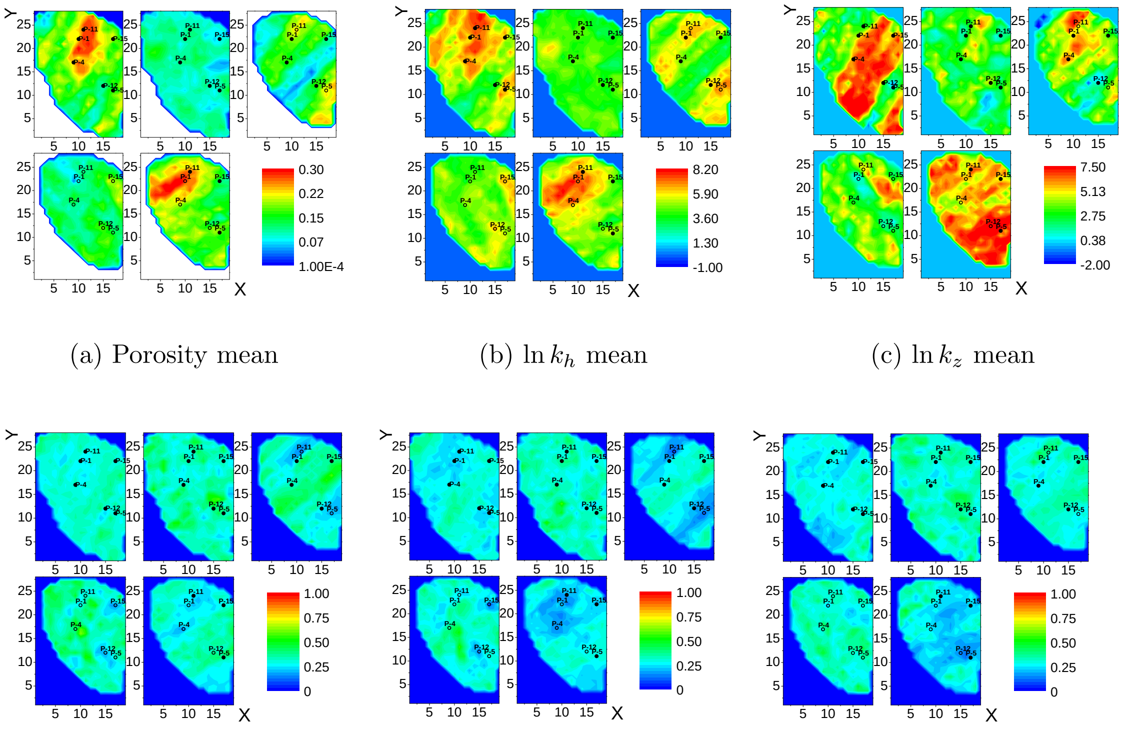
(c) Prod12 BHP

(d) Prod4 WCT

(e) Prod11 WCT

(f) Prod12 WCT

Figure 5.6: Predictions from final model (PUNQS3 case, assimilating production data, iterative).



(a) Porosity mean

(b) $\ln k_h$ mean

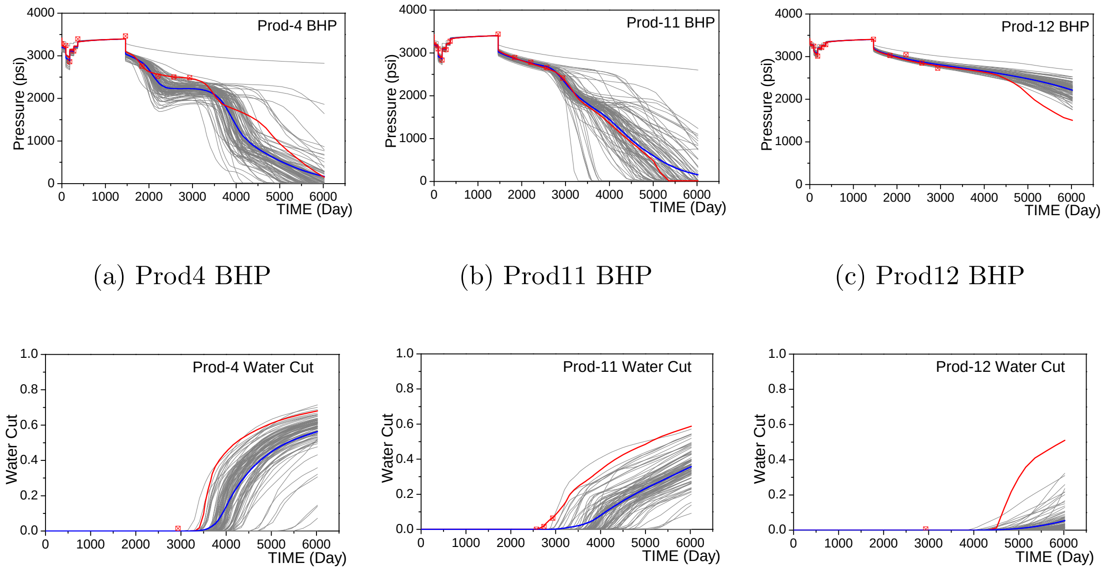
(c) $\ln k_z$ mean

(d) Porosity STDR

(e) $\ln k_h$ STDR

(f) $\ln k_z$ STDR

Figure 5.7: Mean and STDR of conditional fields (PUNQS3, assimilating production data + global analysis of seismic data, iterative).



(a) Prod4 BHP

(b) Prod11 BHP

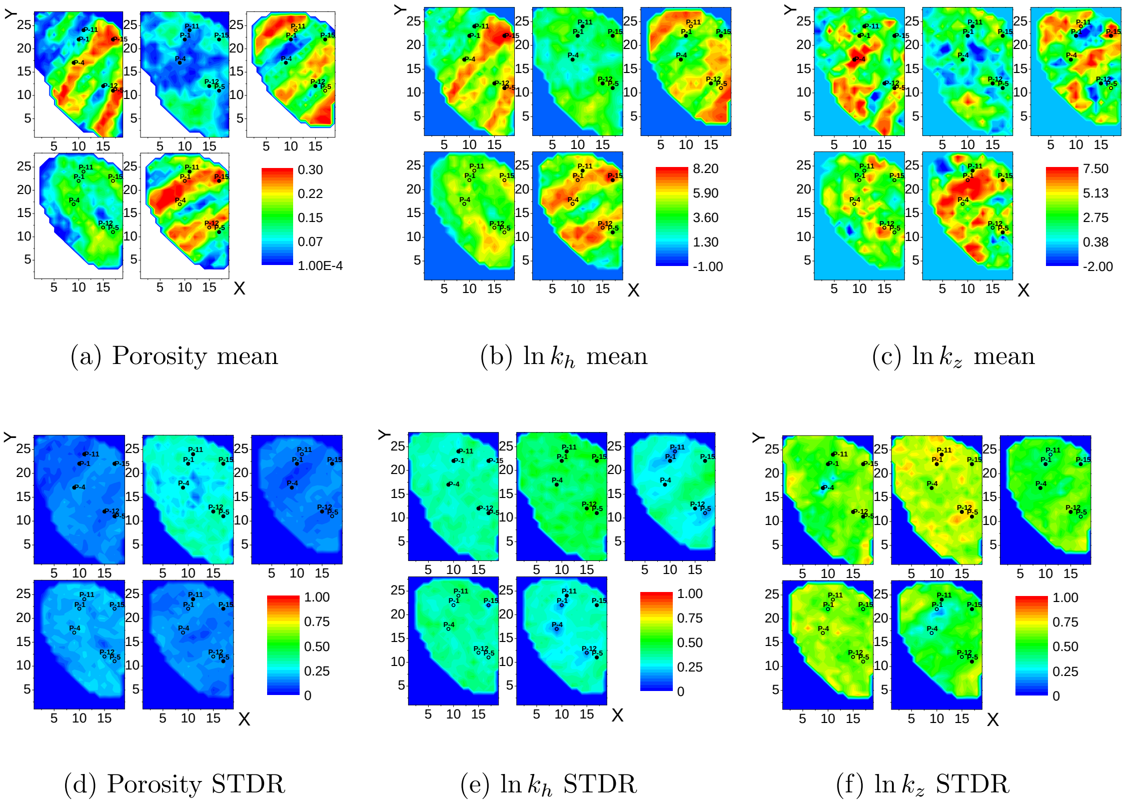
(c) Prod12 BHP

(d) Prod4 WCT

(e) Prod11 WCT

(f) Prod12 WCT

Figure 5.8: Predictions from final model (PUNQS3 case, assimilating production data + global analysis of seismic data, iterative).



(a) Porosity mean

(b) $\ln k_h$ mean

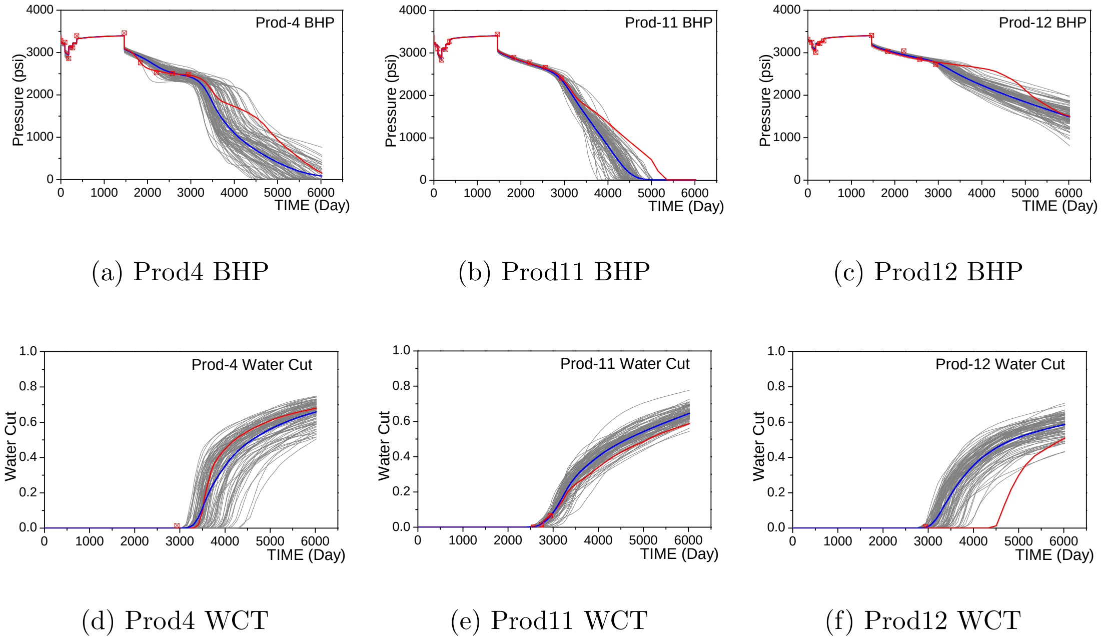
(c) $\ln k_z$ mean

(d) Porosity STDR

(e) $\ln k_h$ STDR

(f) $\ln k_z$ STDR

Figure 5.9: Mean and STDR of conditional fields (PUNQS3, assimilating production data + local analysis of seismic data without projection, iterative).



(a) Prod4 BHP

(b) Prod11 BHP

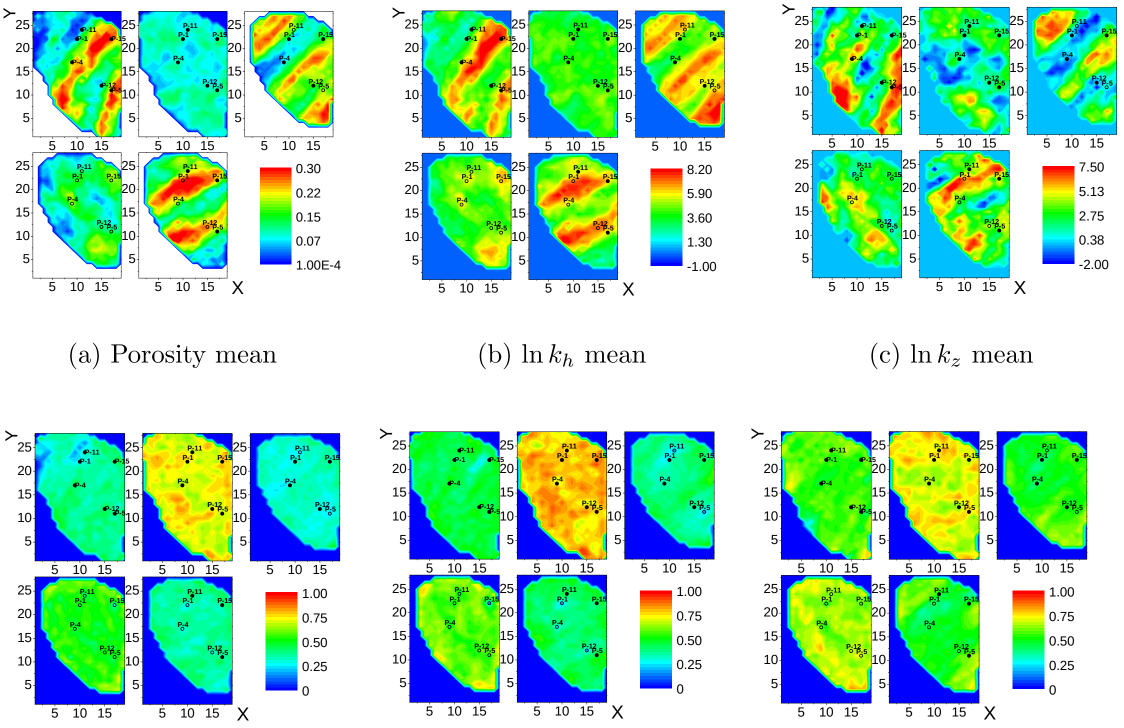
(c) Prod12 BHP

(d) Prod4 WCT

(e) Prod11 WCT

(f) Prod12 WCT

Figure 5.10: Predictions from final model (PUNQS3 case, assimilating production data + local analysis of seismic data without projection, iterative).



(a) Porosity mean

(b) $\ln k_h$ mean

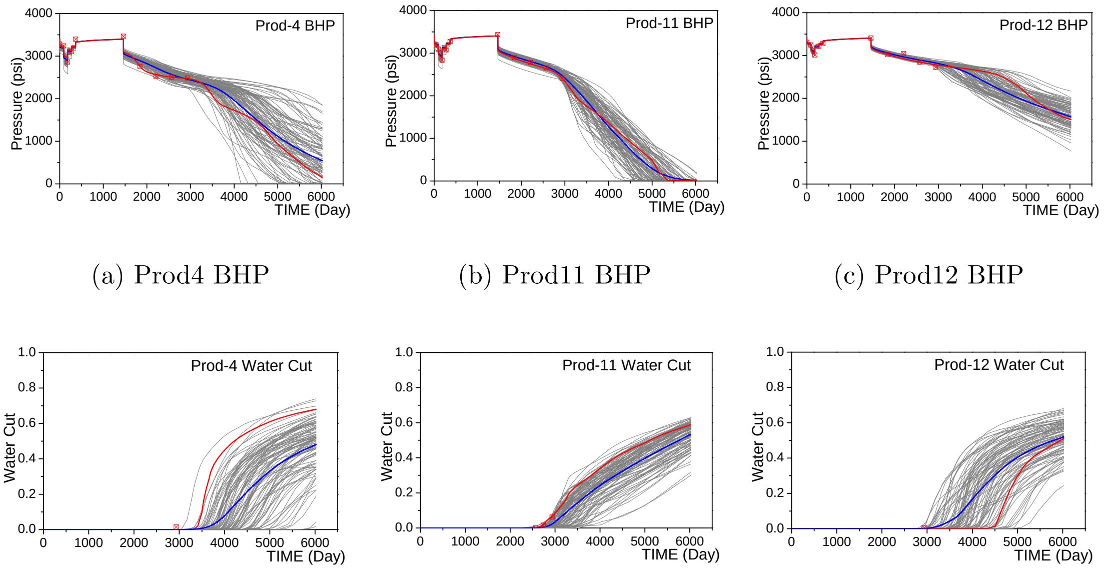
(c) $\ln k_z$ mean

(d) Porosity STDR

(e) $\ln k_h$ STDR

(f) $\ln k_z$ STDR

Figure 5.11: Mean and STDR of conditional fields (PUNQS3, assimilating production data + local analysis of seismic data with projection, iterative).



(a) Prod4 BHP

(b) Prod11 BHP

(c) Prod12 BHP

(d) Prod4 WCT

(e) Prod11 WCT

(f) Prod12 WCT

Figure 5.12: Predictions from final model (PUNQS3 case, assimilating production data + local analysis of seismic data with projection, iterative).

ing the EnRML scheme. Four cases will be considered: A) matching only production data; B) matching production data plus seismic data using global analysis; C) matching production data plus seismic data using local analysis without projection; D) matching production data plus seismic data using local analysis with projection. We focus on the following three aspects: 1) Comparison of conditional predictions and models from EnKF and EnRML; 2) Comparison of global and local analysis of seismic data in the EnRML scheme; 3) Find the impact of the projection method for local analysis of seismic data in the EnRML scheme. For all cases here, the ensemble size is 100. Table 5.2 shows the quantitative comparison of these four cases. Comparing the results to the corresponding EnKF cases (Tables 3.5 and 4.2), which are repeated here in Table 5.1, both the data mismatches and prediction errors are smaller in the EnRML results, except that the global analysis case gives slightly higher prediction errors. In Table 5.2, compared to the case assimilating only the production data using EnRML, the two cases of assimilating seismic data (using local analysis with and without projection) give smaller data mismatches, but gives higher prediction errors in general. To compare the results in detail, we need to look at the conditional models and predictions in figures.

Fig. 5.2 show the average and normalized objective function (averaged over the ensemble and then normalized over the data, Eq. 5.16) of the four cases at each data assimilation time. The shaded columns show the values of the objective function before analysis at each data assimilation step, and the solid columns are those after analysis. In the global analysis case, the stars and triangles represent the objective function values of seismic data before and after global analysis, respectively. The final objective function values of seismic data remain too high after global analysis, which indicates that the global analysis failed to give a good match of the seismic data. From this plot, we can also see that the global analysis does not give improved matches of the production data. After local analysis of the seismic data, a much better match of production data is obtained than after global analysis of seismic data. For both cases with and without projection, the final objective function values in most assimilation steps are below 5. The objective function values from the case with projection is slightly higher than those from the case without projection.

Fig. 5.3 shows the predicted field cumulative water production from the final ensemble for the four cases. Compared to the results from the EnKF case assimilating only production data for the corresponding case (Fig. 3.6), the EnRML scheme gives much better predictions (Fig. 5.3(a)). Although the uncertainty band is a little wider, the average prediction is much closer to the truth. The predictions obtained from assimilating seismic data have much narrower bands, and global analysis gives biased predictions. Before further discussion of these results, we compare the predictions of water cut for each producer.

For each case, the predictions shown are obtained by running the final updated ensemble from time zero. The bottom hole pressure and water cut for producers 4, 11 and 12 will be compared. The conditioned models will also be shown as the ensemble mean and STDR for the three rock property fields.

Figs. 5.5 and 5.6 show the conditional models and predictions for the case assimilating only production data. Compared to the model obtained from EnKF in the base case (Fig. 3.8), the structure obtained from the iterative scheme is much closer to the truth (Fig. 5.4) in that there are fewer incorrect structures, i.e. the high values on the left top of the reservoir in all the three property fields in Fig. 3.8. The STDR from EnRML are similar to those from the base EnKF case. The predictions are also better compared to those obtained in Figs. 3.3 and 3.4, i.e., the predicted bottom hole pressure and water cut for the three producers are all much narrower and the average predictions are all closer to the truth in general.

Figs. 5.7 and 5.8 give the conditional models and predictions for the case where se assimilate production data and seismic data using the global analysis scheme. We can see that the means of the models are are not similar to the truth, the STDR of the three rock property fields are quite low (lower than 0.3), and many structures in the true model are not captured even for the porosity. This is why the final objective function values of seismic data shown in Fig. 5.2(b) are still high, above 10000, where as they should be around 1 for a correct match. The predicted bottom hole pressure and water cut for producer 12 are also worse than those from the case assimilating only production data because the predictions are too narrow and the truth is outside the bands. The water cut data for producer 11 are not matched, and the prediction is wrong. This means the global analysis in EnRML not only has difficulty in matching the seismic data, but also results in worse matches and predictions of the production data.

Figs. 5.9 and 5.10 show the conditional models and predictions for the case assimilating additional seismic data using the local analysis scheme without projection, and Figs. 5.11 and 5.12 show those from the local analysis scheme with projection. The comparison of these results is quite similar to the corresponding comparison in the EnKF scheme: (1) Both cases give good structure of porosity and $\ln k_h$, but without projection the estimated $\ln k_z$ has worse continuity and contains more overshooting and undershooting; (2) The STDR of the conditional fields from the projection case is higher than those from the case without projection; (3) The predictions from the case without projection is narrower, and is wrong for the water cut of the producer 12. Therefore, although the data matches in the case without projection is better (lower objective function compared in Fig. 5.2(b) and Table 5.2), the predictions are actually worse because of each of continuity of $\ln k_z$ field and they under-estimate of the uncertainties in the model parameters. Comparing the results from EnKF and EnRML schemes for the local analysis with projection (Figs. 4.12, 4.14 and 4.16), the conditional models are quite similar (means and STDR), buf the matches and predictions of bottom hole pressures for each producers and water cut

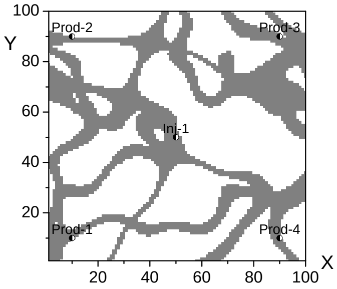
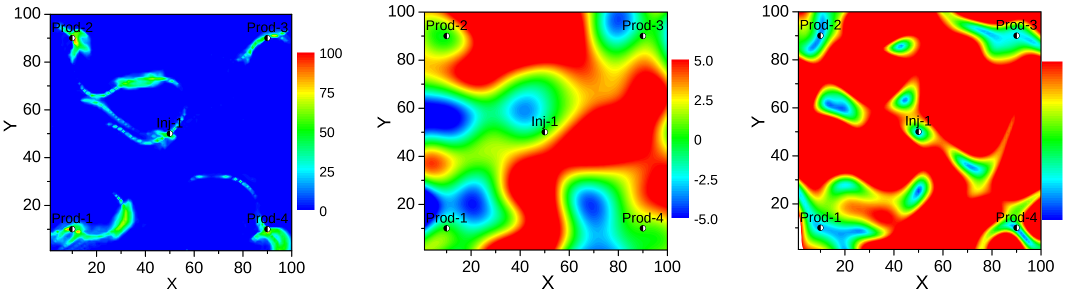


Figure 5.13: Channel example, true facies distribution.



(a) Channel PDF

(b) Z_1 mean

(c) Z_1 STDR

Figure 5.14: Estimated channel PDF, posterior mean and STDR of Z_1 after assimilating the first seismic data using global analysis(channel case, EnRML).

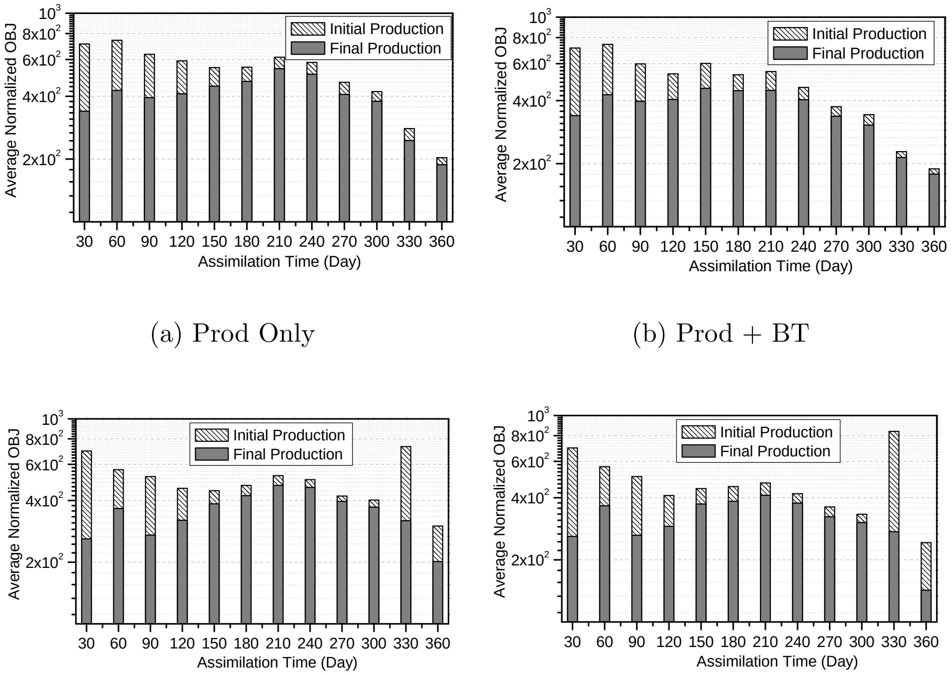
for producer 4 are improved in the EnRML scheme.

5.3 EnRML Results for Channel Example

Global analysis of the seismic data in EnRML scheme can not give reasonable results for the channel example because after the analysis of the seismic data set, although the average objective function value is decreased from 2444 to 230, most gridblocks become non-channel facies except some at and around the wells.

Cases	O_m	O^d_{BHP}	O^d_{WCT}	O^d_{TOT}	O^p_{BHP}	O^p_{WCT}	O^p_{TOT}
Prod. Only	0.444	14.599	18.756	16.576	4.349	5.995	5.146
Prod. + BT	0.458	14.831	17.498	16.071	3.216	3.743	3.46
Local	0.446	15.921	19.71	17.706	3.303	5.88	4.629
Local + BT	0.442	13.318	17.749	15.445	3.496	3.757	3.615

Table 5.3: Model mismatch, data mismatch, prediction error, channel example, predicting final ensemble from time zero, EnRML



(a) Prod Only

(b) Prod + BT

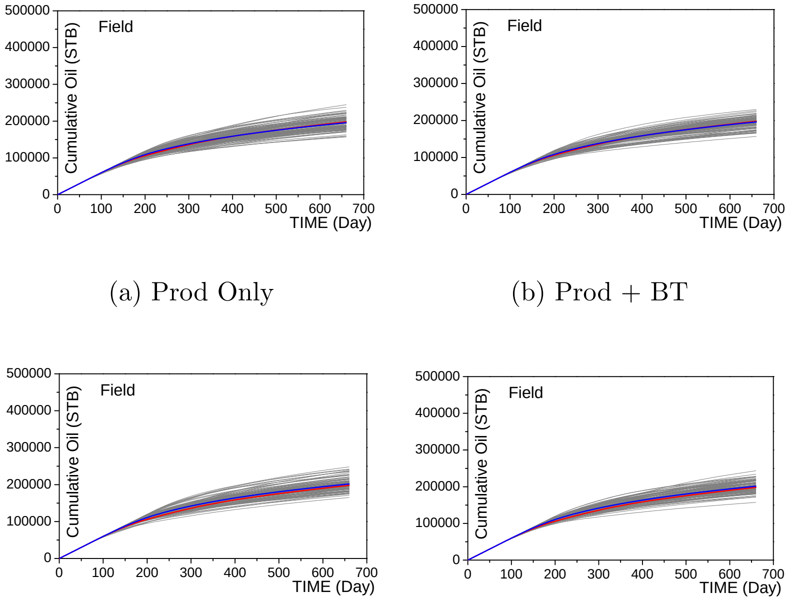
(c) Local with projection

(d) Local with projection + BT

Figure 5.15: Comparison of objective functions for the four cases (channel case, iterative).

This make it impossible to match the production data. The channel pdf and the mean/STDR of the Gaussian random field Z_1 are shown in Fig. 5.14. The channel pdf shows the four wells are completely disconnected, and the conditional Z_1 field is unreasonable as the mean and STD are far too large except the gridblocks close to the wells. A similar phenomenon is observed in the conditional Z_2 field, which is not shown.

In this case, for the channel example, we show four cases: A) assimilation of production data only; B) assimilation of water breakthrough times as additional data; C) assimilation of seismic data using local analysis with projection; D) assimilation of additional water breakthrough times and seismic data using local analysis with projection. The purpose of this case is mainly in two aspects: (1) compare EnRML and EnKF in assimilating production data and seismic data in local analysis for the channel model; (2) check how much improvement can be obtained by assimilating



(a) Prod Only

(b) Prod + BT

(c) Local with projection

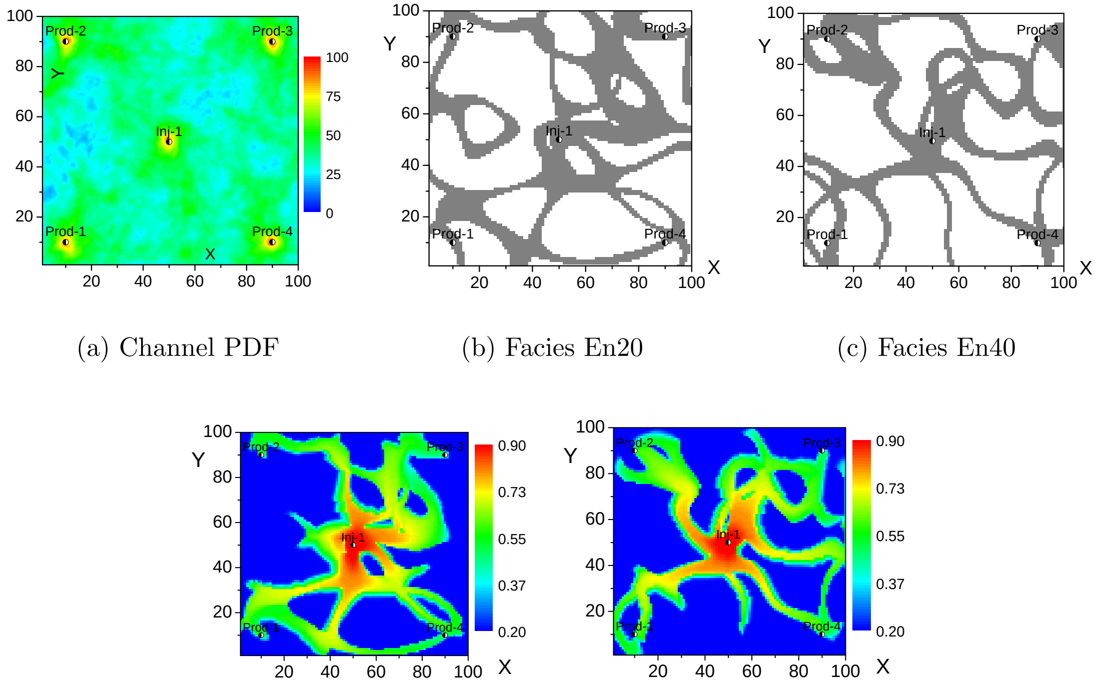
(d) Local with projection + BT

Figure 5.16: Comparison of field cumulative oil for the four cases (channel case, iterative).

water breakthrough times using EnRML scheme. The water breakthrough times for each producers are assimilated at the time $\hat{t}_n$ listed in Table 3.2.

Table 5.3 shows the quantitative comparison of the four cases. Compared to those from the corresponding EnKF cases (Tables 3.7 and 4.6), both data mismatches and prediction errors are much smaller for the EnRML results. Comparing the four cases from the EnRML scheme, assimilating seismic data using local analysis makes data mismatches slightly larger than if we assimilate the production data only, but makes the prediction errors slightly smaller.

Figs. 5.15 and 5.16 show the objective function values of production data and the field cumulative oil predictions for the four cases. Compared to the case assimilating only the production data, matching water breakthrough times gives a slightly lower objective function values than if we do not match them, and assimilating seismic data gives a little higher objective function values than if we do not



(a) Channel PDF

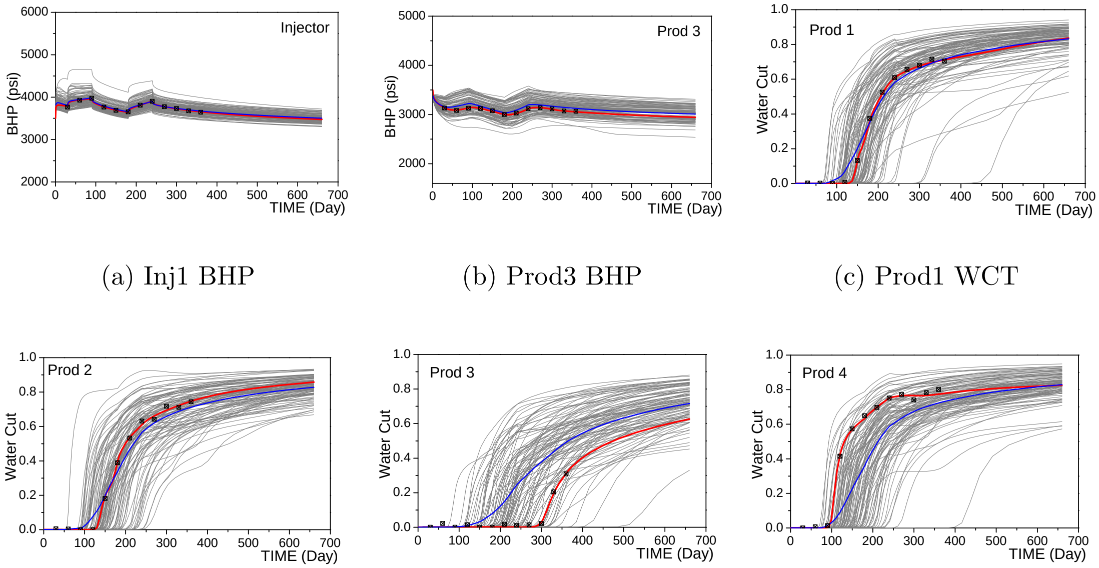
(b) Facies En20

(c) Facies En40

(d) S_w en20

(e) S_w en40

Figure 5.17: Estimated channel PDF and two conditional realizations of facies and water saturation distributions at 360 days (channel case, assimilating production data, iterative).



(a) Inj1 BHP

(b) Prod3 BHP

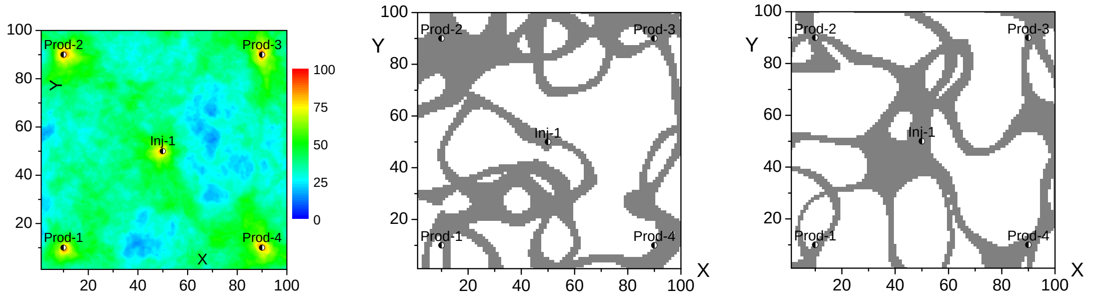
(c) Prod1 WCT

(d) Prod2 WCT

(e) Prod3 WCT

(f) Prod4 WCT

Figure 5.18: Predictions from final model (channel case, assimilating production data, iterative).

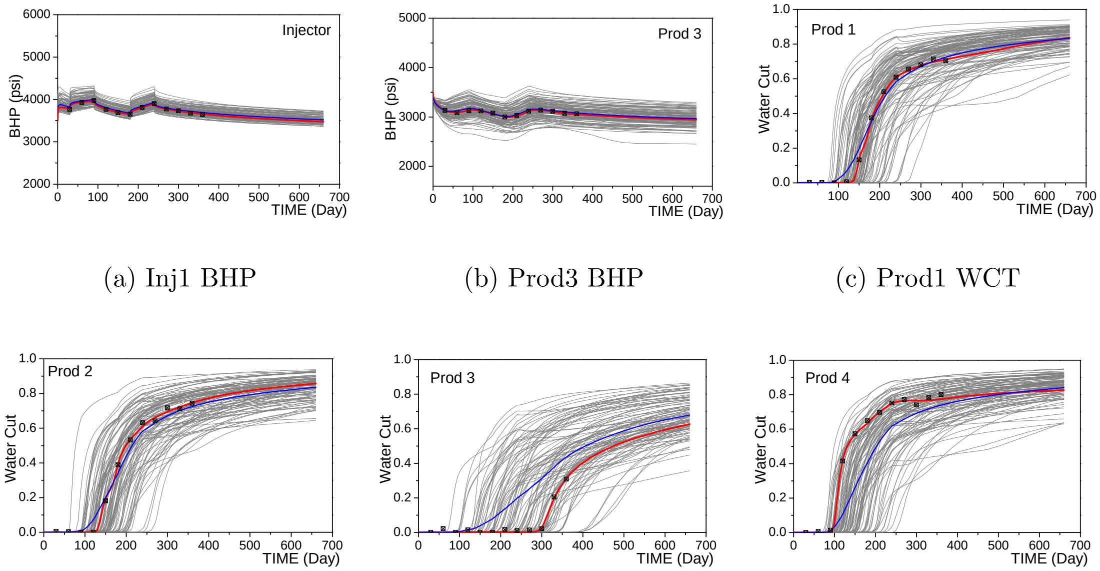


(a) Channel PDF

(b) Facies En20

(c) Facies En40

Figure 5.19: Estimated channel PDF and two conditional realizations of facies distributions (channel case, assimilating production data + water break through time, iterative).



(a) Inj1 BHP

(b) Prod3 BHP

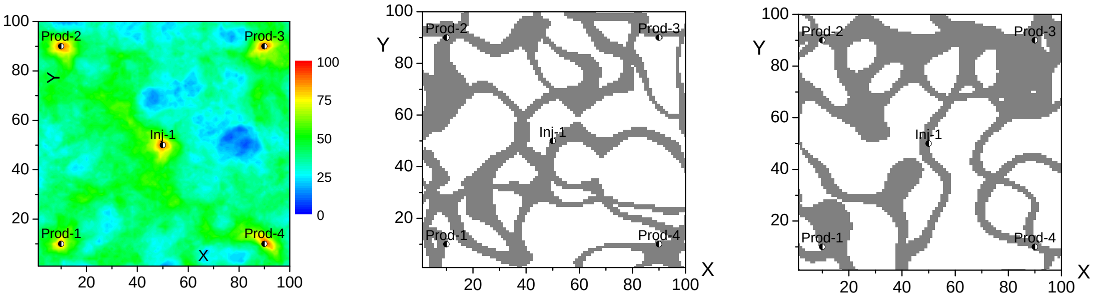
(c) Prod1 WCT

(d) Prod2 WCT

(e) Prod3 WCT

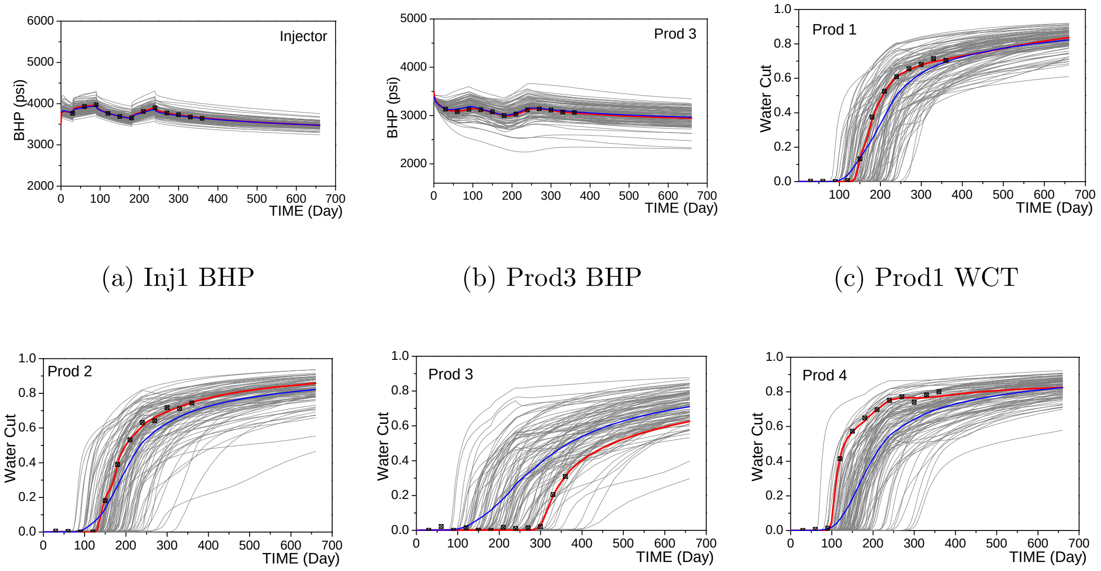
(f) Prod4 WCT

Figure 5.20: Predictions from final model (channel case, assimilating production data + water breakthrough time, iterative).



(a) Channel PDF (b) Facies En20 (c) Facies En40

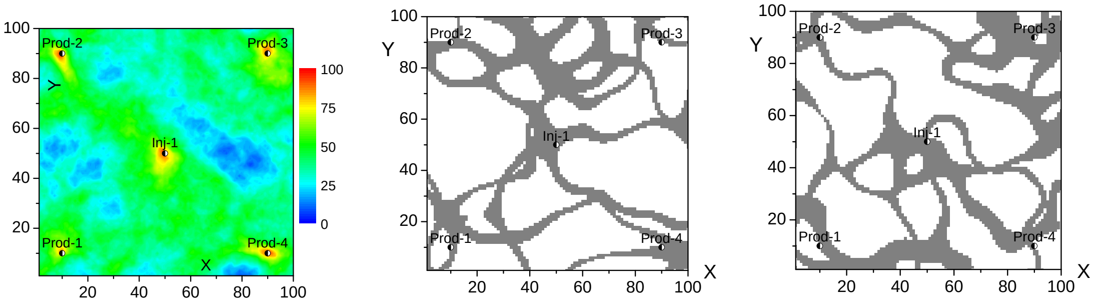
Figure 5.21: Estimated channel PDF and two realizations of facies distributions (channel case, assimilating production data + local analysis of seismic data with projection, iterative).



(a) Inj1 BHP (b) Prod3 BHP (c) Prod1 WCT

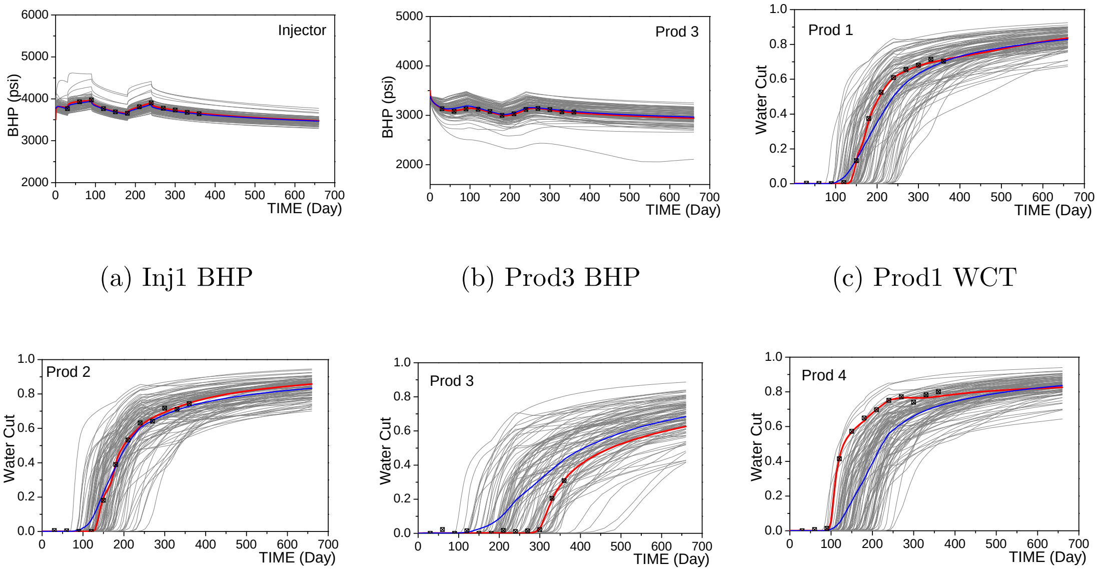
(d) Prod2 WCT (e) Prod3 WCT (f) Prod4 WCT

Figure 5.22: Predictions from final model (channel case, assimilating production data + local analysis of seismic data with projection, iterative).



(a) Channel PDF (b) Facies En20 (c) Facies En40

Figure 5.23: Estimated channel PDF and two realizations of facies distributions (channel case, assimilating production data + water breakthrough time + local analysis of seismic data with projection, iterative).



(a) Inj1 BHP (b) Prod3 BHP (c) Prod1 WCT

(d) Prod2 WCT (e) Prod3 WCT (f) Prod4 WCT

Figure 5.24: Predictions from final model (channel case, assimilating production data + water breakthrough time + local analysis of seismic data with projection, iterative).

match them, which is consistent with our comments for Table 5.3. Assimilating both water breakthrough times and seismic data gives the best data match, but the improvement seem quite limited. The predicted field cumulative oil for the four cases are all similar trend.

Fig. 5.17 shows the channel PDF and two conditional realizations of facies and water saturation distributions at 360 days from the case where we assimilate only production data. Fig. 5.18 shows the predictions of bottom hole pressures from injector and producer 1 and water cut from each producers for the case assimilating only the production data. Comparing results from EnRML with those from EnKF (Figs. 3.17, 3.18 and 3.21), we can make the following comments:

1. The channel PDF from EnRML shows much less clearly the low channel PDF regions between the injector and producer 3, because the conditional facies distributions from the EnRML are more varied (compared to 3.21).

2. The conditional water saturation distributions are consistent with the corresponding facies distributions because they are obtained by simulating the conditional models from time zero.

3. Bottom hole pressure matches from EnRML scheme are much better than those from EnKF scheme (from final ensemble), and predictions of water cut are also improved (except the producer 4) in that the water breakthrough time matches are better and the average predictions are closer to the truth. These can be reflected in the comparison of field cumulative oil (compare Figs. 3.19 and 5.16) because the average is very close to the truth in the EnRML prediction, while it is biased in those from the final EnKF ensemble.

Figs. 5.19 and 5.20 show the conditional models and the predictions of bottom hole pressure of injector and producer 3 and water cut of the four producers from the case assimilating both production data and water breakthrough times. Compared to the previous case (assimilating only production data), there are some improvements:

1. The channel PDF indicates more blue colors (poor connectivity) between the injector and producer 3, which is also reflected in the comparisons of the two conditional realizations of facies distributions;

2. In the predictions of water cut, some realizations with very late breakthrough time in the previous case (assimilating only production data) are corrected towards the truth. We believe this is the major reason why we get smaller water cut data mismatch and prediction errors than the case assimilating only the production data (Table 5.3, row 1 and 2).

Figs. 5.21 to 5.24 show the conditional models and predictions obtained by assimilating seismic data using the local analysis scheme with and without assimilating water breakthrough times. These two cases give quite limited improvements on the production data predictions compared to the previous two cases, but the channel PDF plots indicates a clearer recovery of connectivity between the injector and the producer 3.

5.4 Summary

In the modified Gu-Oliver scheme with individual line search, the predictions are obtained by simulating from time zero, and hence are consistent with the analyzed models. In addition, it is an iterative scheme and hence is more adapted to the nonlinear problem. Therefore, the data matches and predictions obtained from EnRML are somewhat better than those obtained from EnKF. EnRML uses an approximation to an average sensitivity computed from the predicted ensemble. This makes it easy to implement and more suitable for the pluri-Gaussian geological model, in which the true sensitivity of data to the Gaussian random field values does not exist at facies boundaries. However, because it uses an average sensitivity matrix, it is not guaranteed to give a downhill search direction for each realization and thus usually fails to fit data as well as expected.

For the iterative scheme used in this study, global analysis still has difficulty in assimilating the seismic data: (i) for the PUNQS3 model, the final objective function values are still very high and the conditional models can only capture part of the structures in the true model; (ii) for the channel model, the abnormally high and low values of Z_1 and Z_2 are introduced to most gridblocks so that the conditional facies distributions are non-physical, i.e., most gridblocks are non-channel facies and the wells are not connected with channel facies. Therefore, global analysis is not a good candidate for assimilating seismic data in the EnRML scheme. This is due to the deficiency in EnRML mentioned above as we were able to apply global analysis to assimilate seismic data when using EnKF. The local analysis scheme implemented in this study, although it does not assimilate seismic data iteratively, is also efficient when combined with the EnRML scheme used to assimilate the production data. Compared to EnKF, EnRML gives improved results for both the PUNQS3 model and the channel example. Matching water breakthrough times in EnRML scheme gives very slight improved matches and predictions of water cut.

CHAPTER 6

CASES OF PLURI-GAUSSIAN MODEL WITH UNCERTAIN PARAMETERS

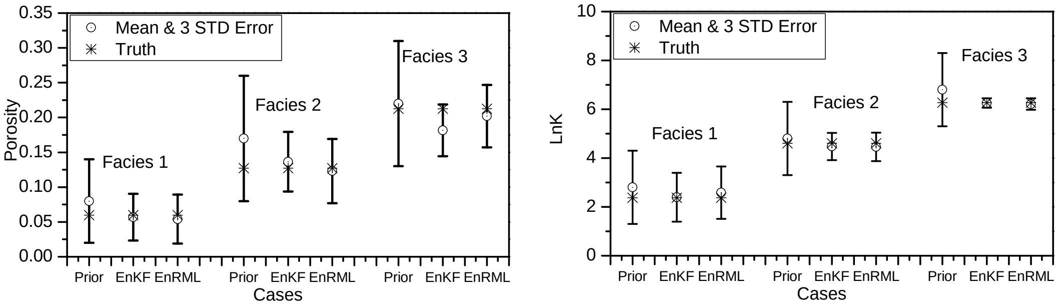
For the two pluri-Gaussian geological models discussed in the previous chapters, the facies porosity and permeability were assumed to be known. In this chapter, we will consider examples where these facies properties are uncertain. Tables 6.1 and 6.2 shows the true porosity and $\ln k$, and their prior means and standard deviations for the examples considered in this chapter. For both models, the correlation coefficients between porosity and $\ln k$ for each facies are 0.5. For the three facies case, the vertical permeability is 10% of horizontal permeability, and hence is not a model parameter.

Property	Truth	Mean	STD
Channel ϕ	0.212	0.15	0.05
Nonchannel ϕ	0.162	0.15	0.05
Channel $\ln k$	7.259	7.20	0.50
Nonchannel $\ln k$	2.441	2.00	1.00

Table 6.1: Channel example porosity and log-permeability.

Property	Truth	Mean	STD
Facies 1 ϕ	0.060	0.08	0.02
Facies 2 ϕ	0.127	0.17	0.03
Facies 3 ϕ	0.213	0.22	0.03
Facies 1 $\ln k$	2.372	2.80	0.50
Facies 2 $\ln k$	4.604	4.80	0.50
Facies 3 $\ln k$	6.269	6.80	0.50

Table 6.2: Three facies example porosity and log-permeability.



(a) Estimated porosity

(b) Estimated $\ln k$

Figure 6.1: Objective function and comparison of conditional facies properties (three facies example).

Cases	$O_{BHP}^{p,EnKF}$	$O_{WCT}^{p,EnKF}$	$O_{TOT}^{p,EnKF}$
Prior	7.674	6.709	7.261
Prod EnKF	2.534	4.954	3.805

Table 6.3: Prediction errors, three facies example, predicting forward from last data assimilation, uncertain facies properties

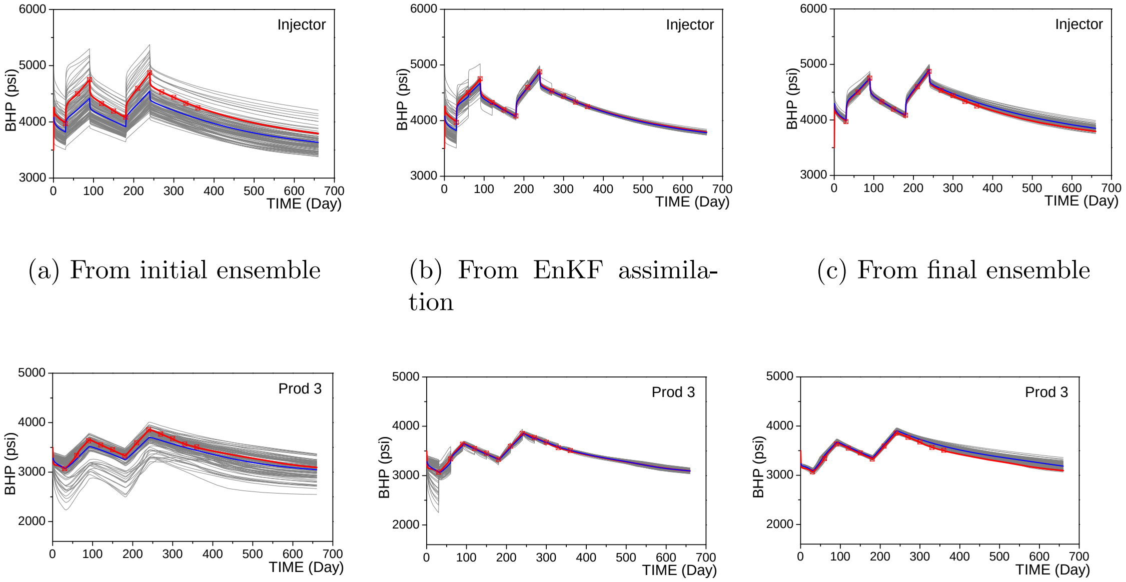
This chapter considers only the assimilation of production data. Here, the porosity and log-permeability ($\ln k$) of each facies are included in the state vector so that they can be conditioned to the production data. For both models, production data are assimilated in both the EnKF and EnRML framework. For all cases, the initial ensemble of Gaussian random fields is the same as those in the previous chapters.

6.1 Three Facies Example

Figs. 6.1(a) and 6.1(b) show the prior and posterior distributions of porosity

Cases	O_m	O_{BHP}^{d}	O_{WCT}^{d}	O_{TOT}^{d}	O_{BHP}^{p}	O_{WCT}^{p}	O_{TOT}^{p}
Prior	0.605	25.203	15.502	21.441	9.634	10.171	9.877
Prod EnKF	0.546	7.395	9.404	8.348	7.293	4.517	6.214
Prod EnRML	0.553	4.176	4.329	4.245	1.437	3.233	2.407

Table 6.4: Model mismatch, data mismatch, prediction error, three facies example, predicting final ensemble from time zero, uncertain facies properties



(a) From initial ensemble

(b) From EnKF assimilation

(c) From final ensemble

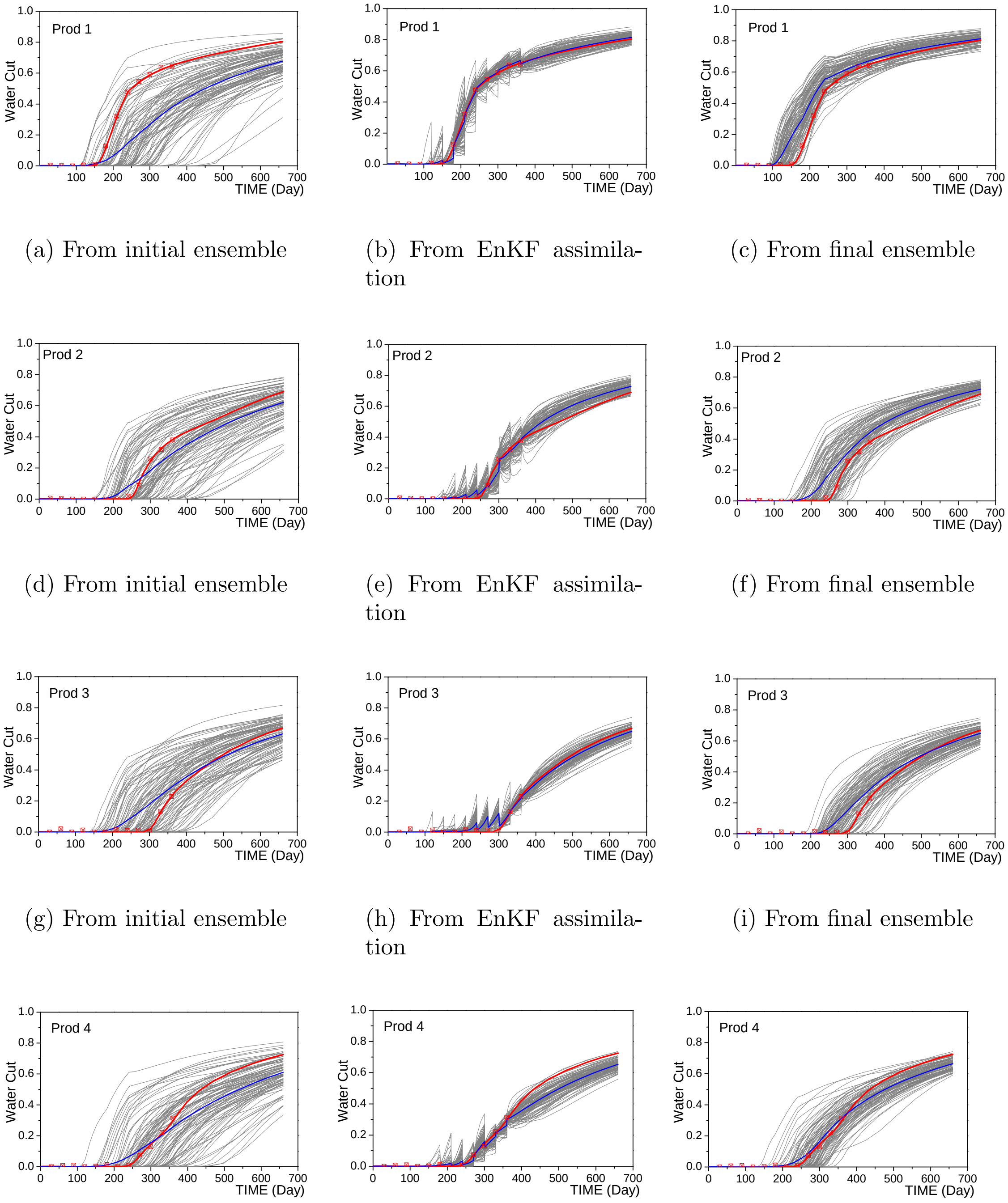
(d) From initial ensemble

(e) From EnKF assimilation

(f) From final ensemble

Figure 6.2: Injector and prod3 BHP (three facies example, EnKF of production data).

and $\ln k$ for the three facies model cases. In each plot, three error bars are given for the properties of the three facies with the width corresponding to plus and minus three standard deviations from the mean. The first bar shows the uncertainty of the prior, and the following two show the posterior uncertainty from EnKF and EnRML. After assimilating data, each of the 6 model parameters has much smaller uncertainty and the truth falls inside the bands. EnRML gives a better estimate of facies 3 porosity than EnKF but the estimates of the other parameters are quite close for the two methods. Fig. 6.2 shows the three types of predictions of bottom hole pressure for the injector and producer 3, and Fig. 6.3 shows the predictions of water cut for all producers. The predictions from the initial ensemble are broader than those from the base case in the Chapter 3, because of the uncertainty in the facies porosity and permeability. Fig. 6.4 shows the predictions from the final ensemble from EnRML. Fig. 6.5 shows field cumulative oil from both EnKF and EnRML cases. We can see good data matches and predictions are obtained from both the EnKF



(a) From initial ensemble

(b) From EnKF assimilation

(c) From final ensemble

(d) From initial ensemble

(e) From EnKF assimilation

(f) From final ensemble

(g) From initial ensemble

(h) From EnKF assimilation

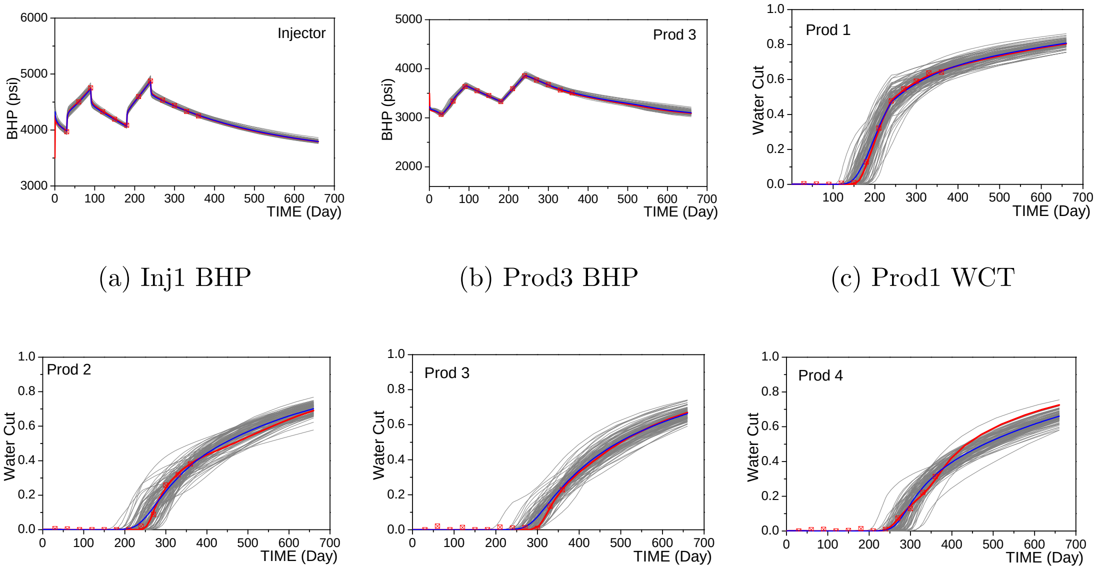
(i) From final ensemble

(j) From initial ensemble

(k) From EnKF assimilation

(l) From final ensemble

Figure 6.3: Producer WCT (three facies example, EnKF of production data).



(a) Inj1 BHP

(b) Prod3 BHP

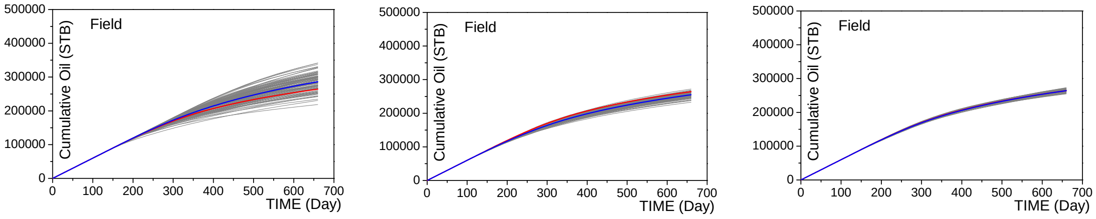
(c) Prod1 WCT

(d) Prod2 WCT

(e) Prod3 WCT

(f) Prod4 WCT

Figure 6.4: Predictions from conditional models (three facies case, EnRML of production data).

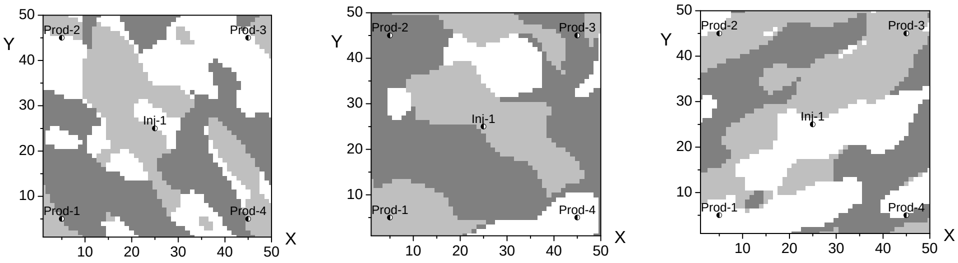


(a) From initial ensemble

(b) EnKF ensemble

(c) EnRML ensemble

Figure 6.5: FOPT predictions (three facies example, EnKF and EnRML of production data).

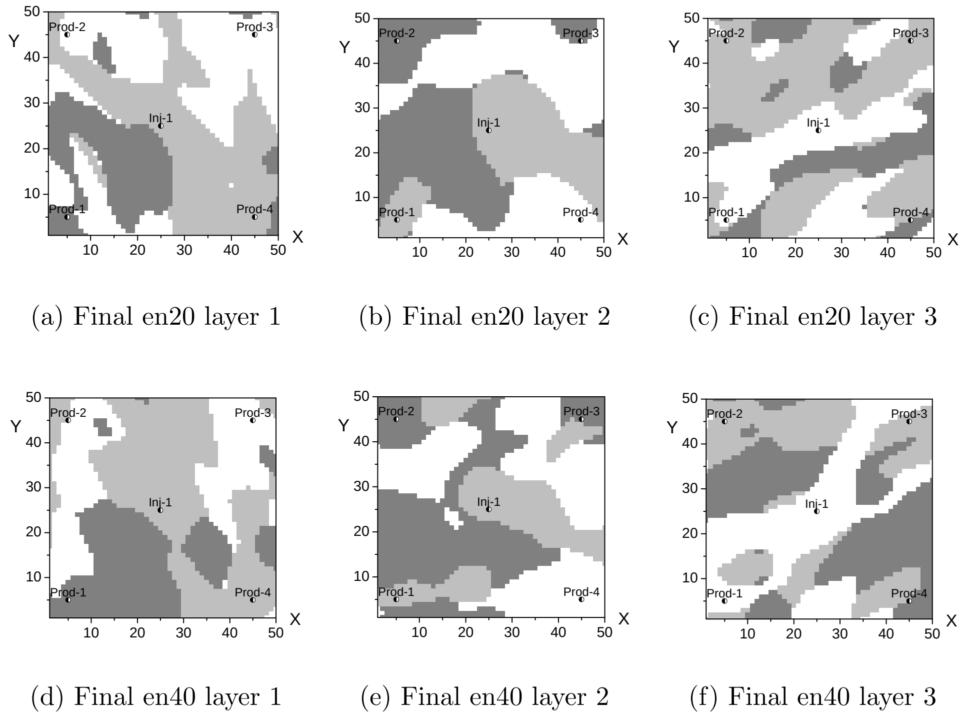


(a) Layer 1

(b) Layer 2

(c) Layer 3

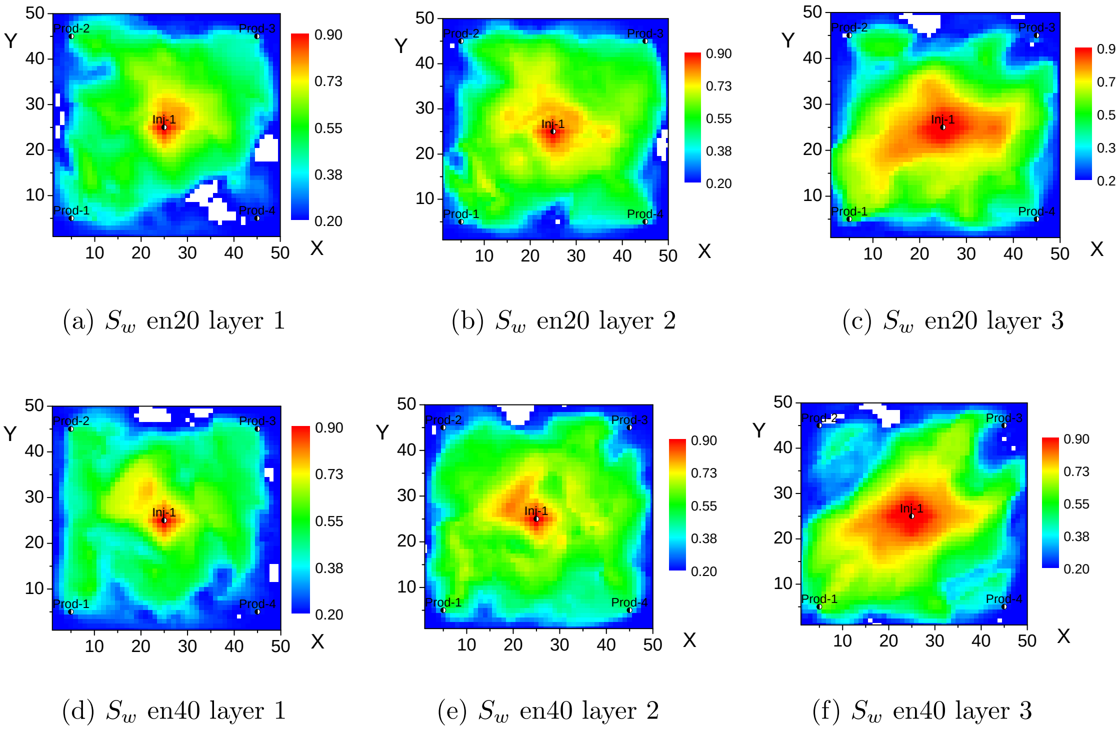
Figure 6.6: Three facies example, true model.



(a) Final en20 layer 1 (b) Final en20 layer 2 (c) Final en20 layer 3

(d) Final en40 layer 1 (e) Final en40 layer 2 (f) Final en40 layer 3

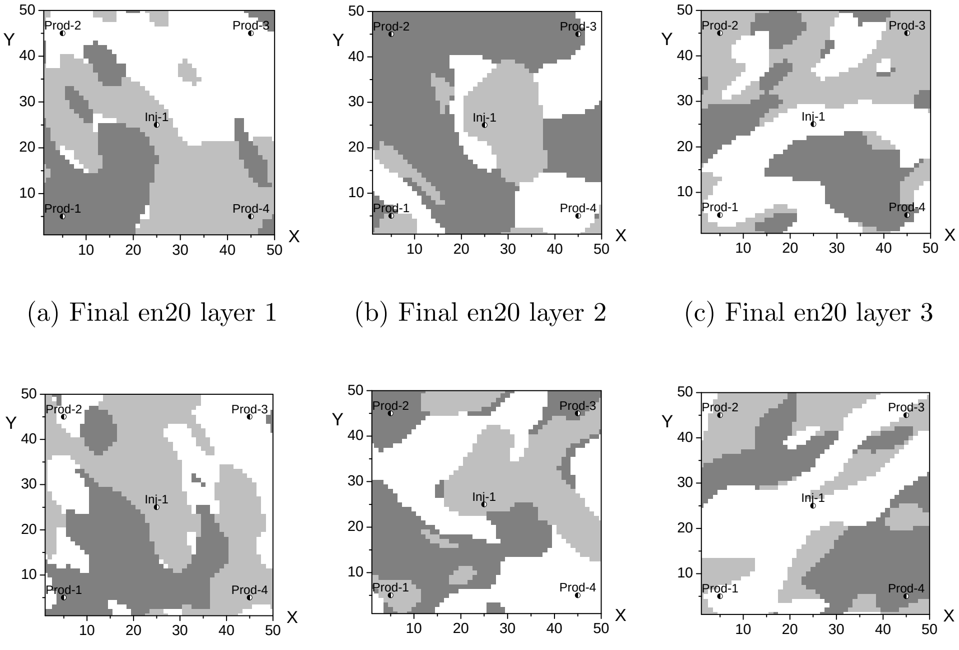
Figure 6.7: Conditional facies distributions (three facies example, EnKF of production data).



(a) S_w en20 layer 1 (b) S_w en20 layer 2 (c) S_w en20 layer 3

(d) S_w en40 layer 1 (e) S_w en40 layer 2 (f) S_w en40 layer 3

Figure 6.8: Conditional S_w (three facies example, EnKF of production data).



(a) Final en20 layer 1

(b) Final en20 layer 2

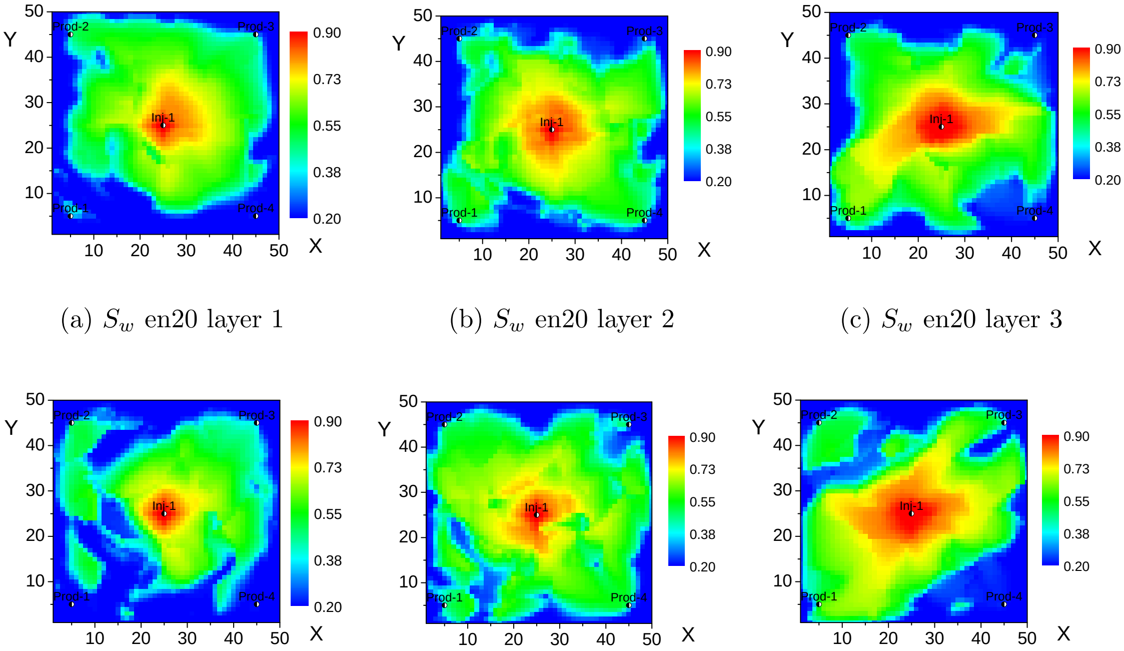
(c) Final en20 layer 3

(d) Final en40 layer 1

(e) Final en40 layer 2

(f) Final en40 layer 3

Figure 6.9: Conditional facies distributions (three facies example, EnRML of production data).



(a) S_w en20 layer 1

(b) S_w en20 layer 2

(c) S_w en20 layer 3

(d) S_w en40 layer 1

(e) S_w en40 layer 2

(f) S_w en40 layer 3

Figure 6.10: Conditional S_w (three facies example, EnRML of production data).

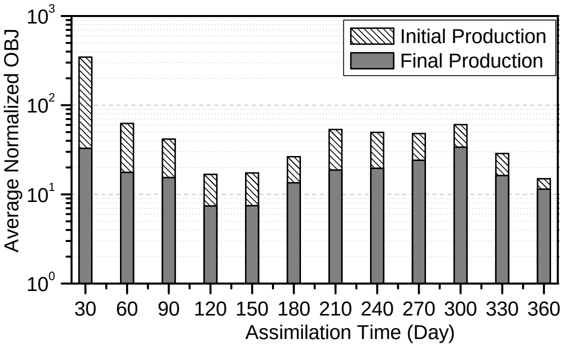


Figure 6.11: Objective functions (three facies example, EnRML of production data).

Cases	$O_{BHP}^{p,EnKF}$	$O_{WCT}^{p,EnKF}$	$O_{TOT}^{p,EnKF}$
Prior	7.674	6.709	7.261
Prod EnKF	4.231	6.626	5.428

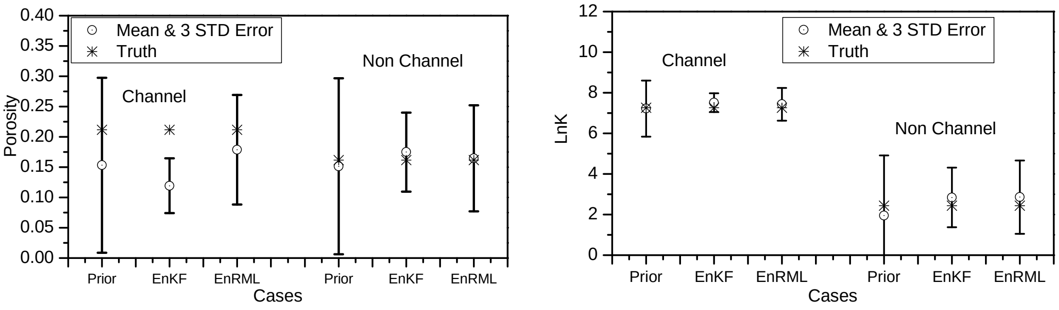
Table 6.5: Prediction errors, channel example, predicting forward from last data assimilation, uncertain facies properties

and EnRML algorithm. The predictions from EnRML give narrower uncertainty bands, better data matches and average predictions closer to the truth. This point is even more clear in the quantitative comparison in the Table 6.4, where the data mismatches and prediction errors are all much smaller in the EnRML case. Two posterior realizations of conditional facies distributions and the corresponding water saturations at 360 days are shown in Figs. 6.7 and 6.8.

Figs. 6.9 and 6.10 gives the two posterior realizations of facies distribution and the corresponding water saturations at the time 360 days. Fig. 6.11 shows the objective functions (average and normalized) before and after analysis at each assimilation time, we can see that most assimilation steps has final objective functions less than 30.

6.2 Channel Example

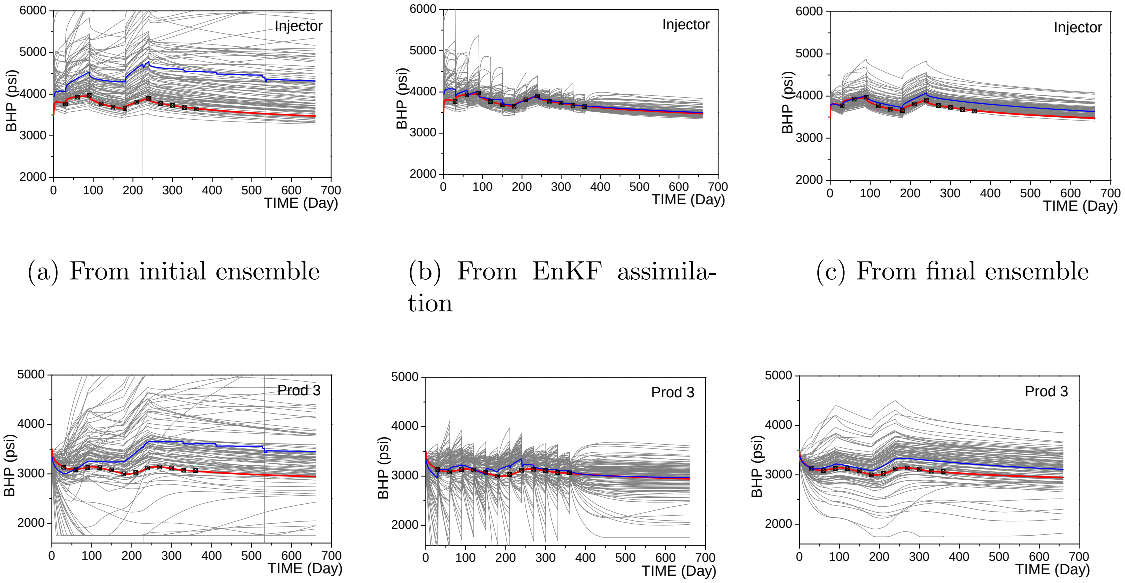
Fig. 6.12 shows the estimates of the facies porosity and $\ln k$ for the channel model. The three bars for each of the four model parameters show prior uncertainty and the posterior uncertainty obtained from EnKF and EnRML, respectively.



(a) Estimated porosity

(b) Estimated $\ln k$

Figure 6.12: Objective function and comparison of conditional facies properties (channel example).



(a) From initial ensemble

(b) From EnKF assimilation

(c) From final ensemble

(d) From initial ensemble

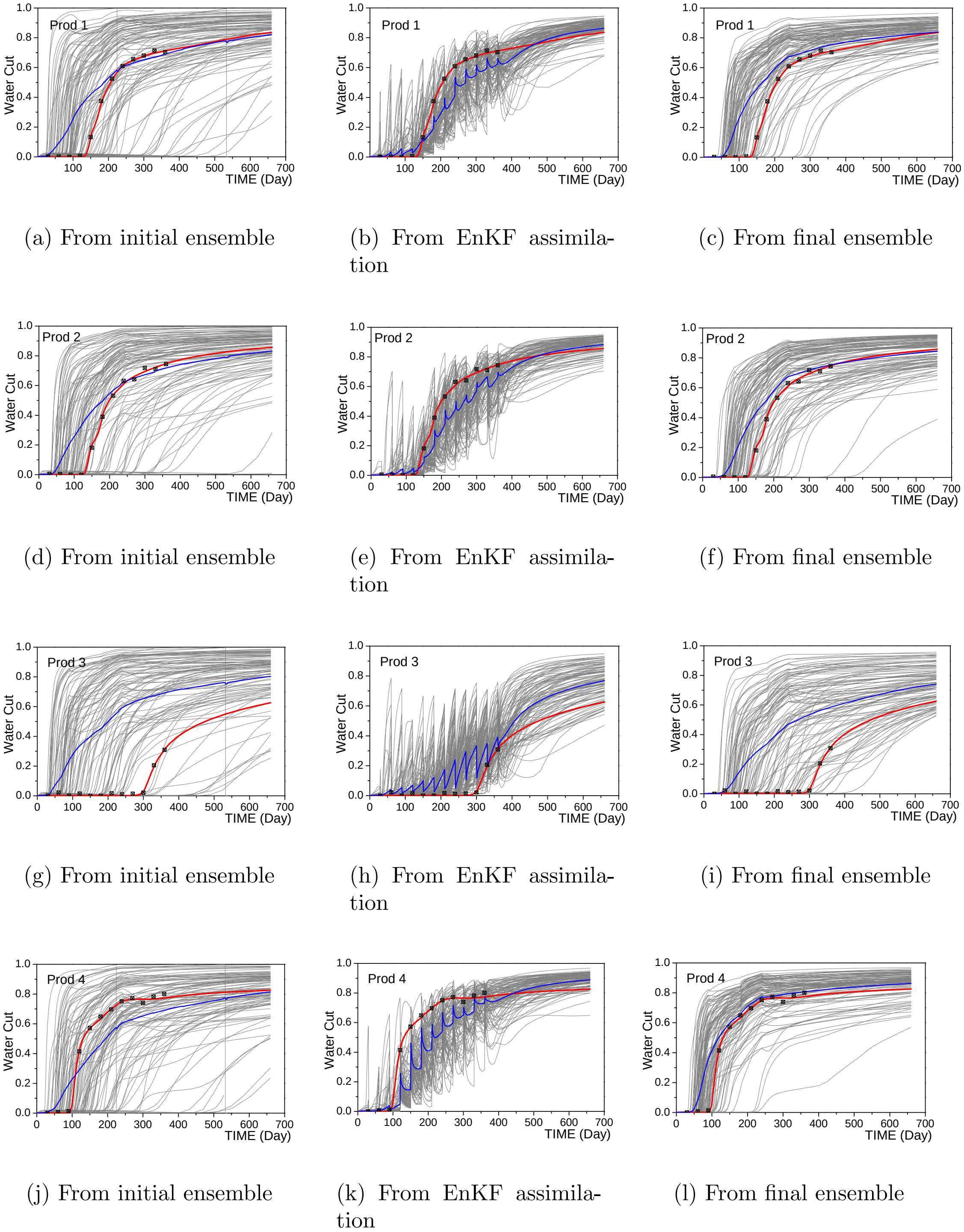
(e) From EnKF assimilation

(f) From final ensemble

Figure 6.13: Injector and prod3 BHP (channel example, EnKF of production data).

Cases	O_m	O^d_{BHP}	O^d_{WCT}	O^d_{TOT}	O^p_{BHP}	O^p_{WCT}	O^p_{TOT}
Prior	0.444	97.719	35.801	76.646	51.363	12.398	39.165
Prod EnKF	0.44	30.777	28.289	29.697	16.44	8.324	13.452
Prod EnRML	0.461	14.142	18.187	16.066	2.845	4.391	3.615

Table 6.6: Model mismatch, data mismatch, prediction error, channel example, predicting final ensemble from time zero, uncertain facies properties



(a) From initial ensemble

(b) From EnKF assimilation

(c) From final ensemble

(d) From initial ensemble

(e) From EnKF assimilation

(f) From final ensemble

(g) From initial ensemble

(h) From EnKF assimilation

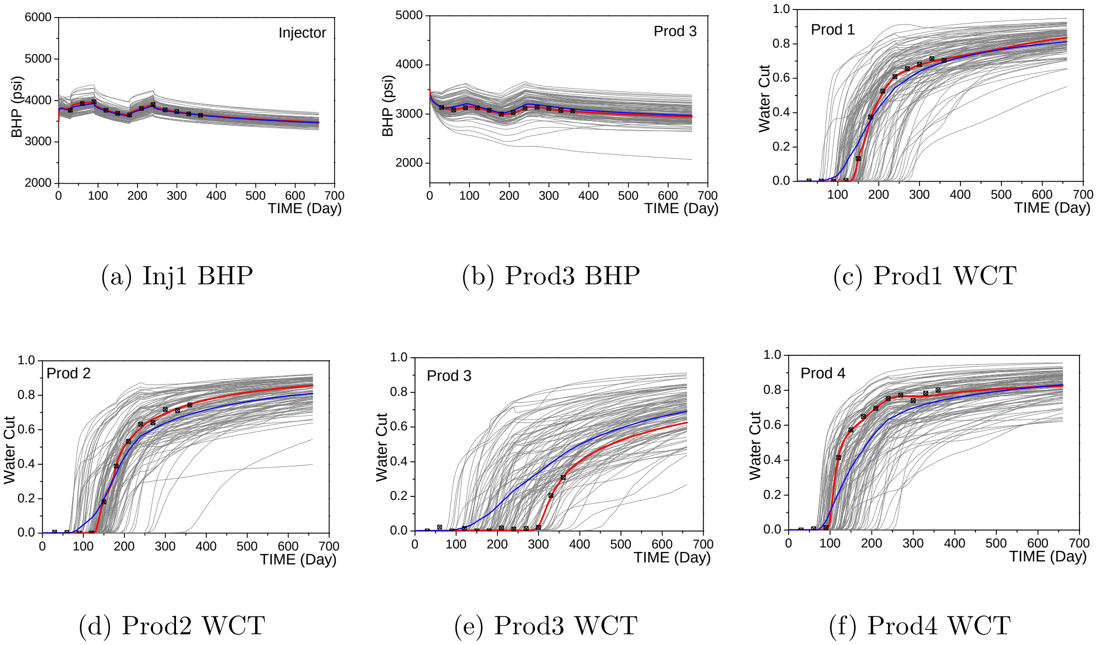
(i) From final ensemble

(j) From initial ensemble

(k) From EnKF assimilation

(l) From final ensemble

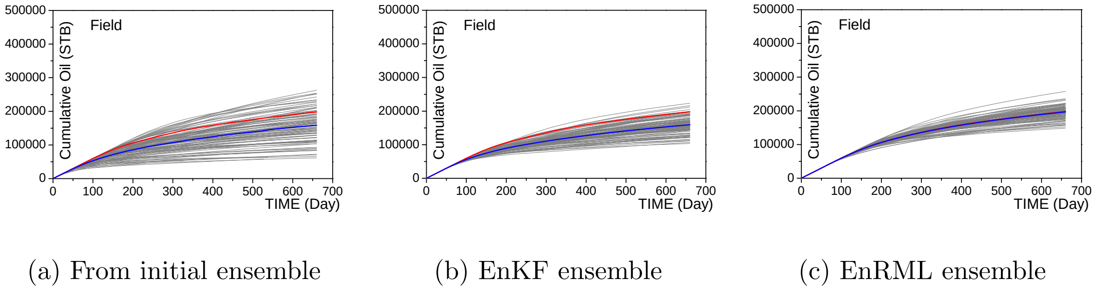
Figure 6.14: Producer WCT (channel example, EnKF of production data).



(a) Inj1 BHP (b) Prod3 BHP (c) Prod1 WCT

(d) Prod2 WCT (e) Prod3 WCT (f) Prod4 WCT

Figure 6.15: Predictions from final model (channel example, EnRML of production data).



(a) From initial ensemble (b) EnKF ensemble (c) EnRML ensemble

Figure 6.16: FOPT predictions (channel example, EnKF and EnRML).

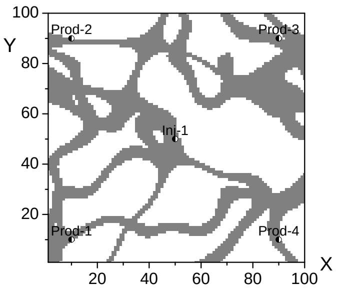
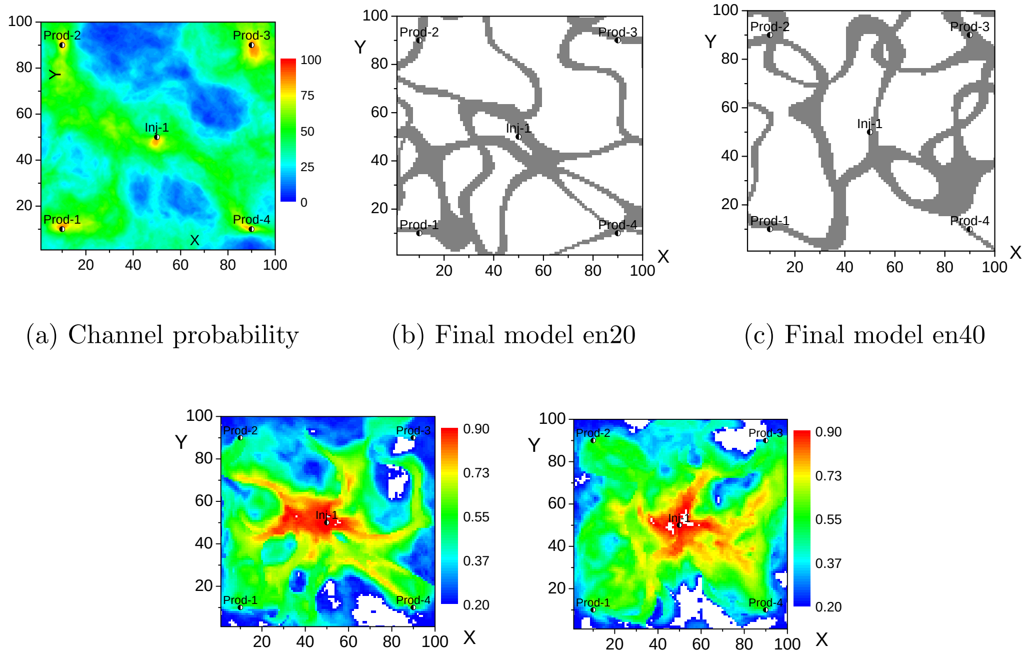


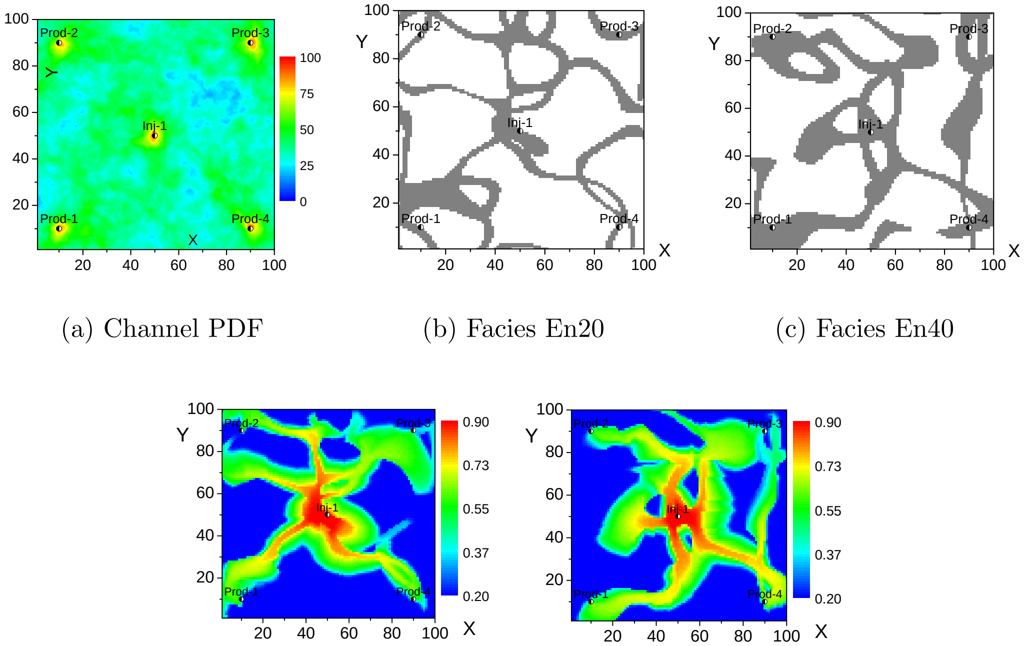
Figure 6.17: Channel example, true facies distribution.



(a) Channel probability (b) Final model en20 (c) Final model en40

(d) S_w en20 (e) S_w en40

Figure 6.18: Estimated channel PDF and two conditional realizations of facies and water saturations at 360 days (channel example, EnKF of production data).



(a) Channel PDF (b) Facies En20 (c) Facies En40

(d) S_w en20 (e) S_w en40

Figure 6.19: Estimated channel PDF and two conditional realizations of facies and water saturations at 360 days (channel example, EnRML of production data).

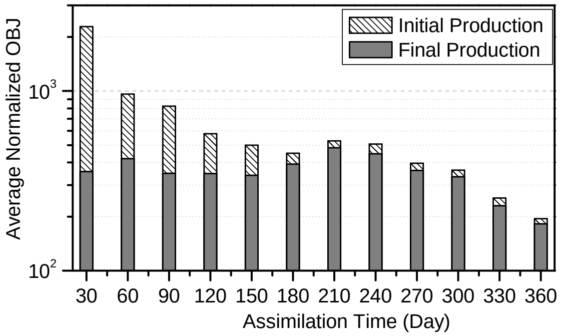


Figure 6.20: Objective functions (channel example, EnRML of production data).

The posterior uncertainty bands are much smaller than those from the prior. The posterior uncertainty band of the channel porosity from EnKF failed to include the true porosity, and the uncertainty bands from EnKF are all smaller than those from EnRML. This may be due to inconsistency between analyzed model parameters and water saturations. Fig. 6.13 shows the predictions of injector and producer 3 pressure from EnKF, and Fig. 6.14 shows the predictions of water cut for each producer from EnKF. Similar to the predictions from the base case in Chapter 3, the average prediction of water cut is farther from the truth than is the average prediction of bottom hole pressure, and the predictions of water cut from producer 3 appear to be biased. Fig. 6.15 shows the predictions of pressure and water cut from EnRML. Comparing these results from EnKF and EnRML, EnRML gives narrower uncertainty bands, better data match and better predictions (the average is closer to the truth), which is obvious in the quantitative comparison of Table 6.6, i.e., the data mismatches and prediction errors are significantly smaller in the EnRML case. Fig. 6.16 compares the field cumulative water productions from EnKF and EnRML. EnRML gives much narrower prediction, and the average prediction is much closer to the truth compared to that from EnKF.

Figs. 6.18 and 6.19 show the posterior channel PDF, facies distributions and corresponding water saturation distributions from both EnKF and EnRML. We can see that the water distributions are consistent with the analyzed models in EnRML.

EnRML gives a more blurred channel PDF map indicating that the facies distribution from EnRML are more varied. Fig. 6.20 shows the average objective function values at each data assimilation step.

6.3 Summary

The uncertainty of the facies properties in the pluri-Gaussian model can be greatly decreased by assimilating the production data using either EnKF or EnRML. However, EnRML gives more dependable estimates of facies porosity and permeability for the method considered. This is because EnRML guarantees that the predictions are always consistent with the updated model.

CHAPTER 7
MEASUREMENT ERROR ESTIMATION METHOD FOR PRODUCTION DATA

Characteristics of measurement error in data is a very important issue in automatic history matching. This study on automatic history matching is focused on the larger issues of data integration and characterization of uncertainty in reservoir description and predicted performance rather than the problem of simply constructing reservoir models that honor production data. In order to characterize uncertainty, our approach to data integration has been based on a Bayesian formulation. Although many of the computational difficulties of the Bayesian formulation have been addressed, until now we have ignored the important problem of characterizing the measurement errors. Our approach to the integration of static and dynamic data and the characterization of uncertainty involves constructing multiple reservoir models (multiple realizations) using the randomized maximum likelihood (RML) method [63, 45, 69] or the ensemble Kalman filter (EnKF) method [22, 62, 60, 24]. In the RML method, construction of an individual realization requires minimization of an objective function which includes a prior model mismatch term, production data mismatch terms and a seismic data mismatch term. As the data covariance matrices for measurement errors determine the relative weighting of different types of data in the objective function, correct integration of data and evaluation of uncertainty require an accurate characterization of these measurement errors. Intuitively, we expect that accurate characterization is more important when multiple data types (e.g., wellbore pressure, GOR, WOR and time lapse seismic) with widely different data covariances are integrated. In the EnKF method, as was discussed in Chapter

3, at each data assimilation time step we also need to sample the noisy measurements for each realization using the covariance matrix of the measurement error, which has impact on the conditional ensemble.

We assume any measurement is the summation of true signal and measurement error; We also assume that the measurement error is Gaussian with zero mean, and the true signal is much smoother than the measurement error. Our approach is to smooth the data properly so that the smoothed signal can approach the truth, and we get measurement error by subtracting it from the measurements. Wavelet transform and polynomial smoothing are often used smoothers. However, the separation of true signal and measurement error by directly smoothing the production data can be very dangerous since the production schedule often changes frequently, so that the underlying signal (or true signal) contains discontinuity (or sharp changes). Applying smoothing algorithms without first partitioning the data may destroy the structure of the signal and the covariance structure of the measurement error.

In order to overcome this difficulty, a procedure for detecting and removing the outliers and edges is necessary before smoothing. After smoothing the edge and outlier free signal, the true signal can be reconstructed by restoring the edges and some necessary outliers into the smoothed signal. At the same time we can obtain a much better estimation of measurement error. In this procedure, median filtering is used to eliminate the outliers, and second derivatives combined with a wavelet transform and median filtering are used to detect the edges.

7.1 Smoothing Algorithms

In this study, we consider two techniques for estimation of the true signal and the noise, local polynomial regression and the denoising with wavelet transforms. Although some results are promising, it is clear that our current implementation of the methods are less reliable if (i) the correlation range for noise is large; (ii) the underlying true signal changes rapidly over short time intervals and/or (iii) the noise is non-stationary.

As our starting point, we assume that the measurement error for a particular data type can be modeled as a stationary random function, and that the underlying "true signal" is relatively smooth. Throughout the true signal refers to the one that would be measured (observed) if there were no measurement, processing or modeling errors. The difference between the unknown true signal and the observed signal represents a realization of the random function for noise or measurement error. Here we simply refer to the difference as noise or measurement error, even though part of this difference may be due to modeling errors. The objective is to separate our measured signal into the true signal and the noise. Once the noise has been estimated, we can estimate its covariance. Although the data can be a function of space (seismic data) as well as time, here we consider only the case where data is measured as a function of time which is the case for production data.

7.1.1 Savitzky-Golay Polynomial Smoothing Algorithm

The Savitzky-Golay smoothing algorithm is a popular technique for smoothing data in the time domain. A good introductory discussion of this method can be found in the reference [65]. More detailed information on smoothing algorithms can be found in the reference [38]. Here, we outline the Savitzky-Golay procedure. We assume that we are given data, $d_i \approx d(t_i)$ where the t_i's represent equally spaced time values, i.e., for some fixed t_0, $t_i = t_0 + i\Delta t$ for $i = 1, 2, \cdots$, where the time increment, Δt is constant. Here, we let $d(t)$ denote the true signal and d_i denote a measured or observed value of $d(t)$ at $t = t_i$. As d_i is corrupted by measurement error, d_i is a noisy approximation of $d(t_i)$. Smoothing algorithms are designed to smooth the observed data in an attempt to find a better estimate of the true $d(t_i)$'s. The d_i data can be filtered (or averaged) by replacing each d_i value by a linear combination of itself and nearby data points. Denoting the linear combination by $\hat{d}_i$, we have

$$\hat{d}_i = \sum_{n=-N_L}^{n=N_R} c_n d_{i+n}, \tag{7.1}$$

where the c_n's are the filter coefficients (or weights) and are specified by the type of filter used. Note N_L is the number of data used to the left of d_i and N_R is the number of points used to the right of d_i when constructing $\hat{d}_i$. Here, we refer to N_L as the left window length, N_R as the right window length and $N_L + N_R + 1$ as the window length. Note that if $c_n = 1/(N_R + N_L + 1)$, then $\hat{d}_i$ represents the average of the $N_R + N_L + 1$ data, d_j, $j = i - N_L, i - N_L + 1, \cdots, N_R$. In this case, the filter defined by Eq. 7.1 is referred to as moving window average or simply a moving average. This type of filter (if N_R=N_L) gives unbiased results if the underlying noise free signal (data) is a linear function of time, but introduces bias when the second derivative of the underlying signal is nonzero. For example, if $d(t)$ has a maximum at $t = t_i$, the moving window average will tend to give a $\hat{d}_i < d(t_i)$ and the magnitude of this underestimation will tend to increase as N_L and N_R increases.

If we use the simple moving average, we are effectively approximating the underlying function in each window as a constant and using the average of the data within the window as the estimate of that constant. In Savitzky-Golay (SG) smoothing, we approximate $d(t)$ by a polynomial in each window. To obtain the approximating polynomial $p(t)$, we simply do a least-squares fit of all the data within the window and then set

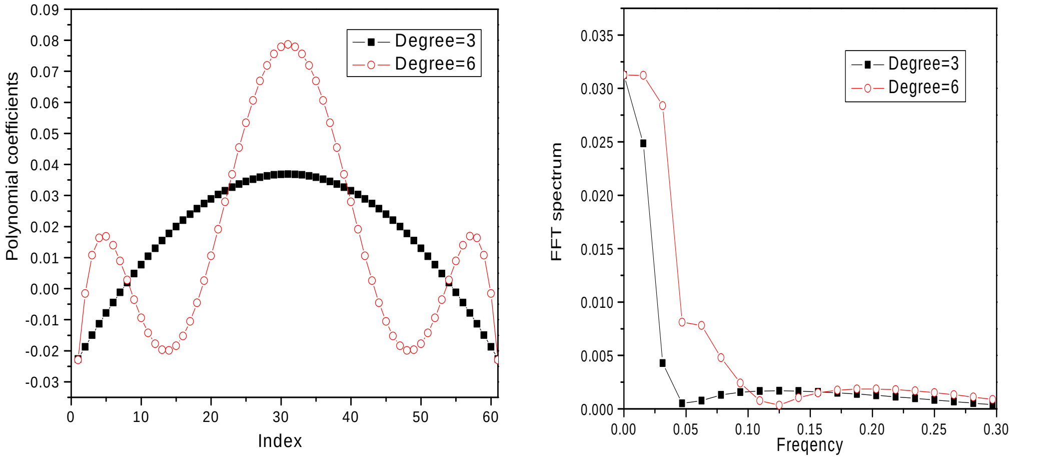
$$\hat{d}_i = p(t_i). \tag{7.2}$$

The key point is that the filter coefficients (c_n's) in Eq. 7.1 are determined a priori independent of the actual data. The specific filter coefficients in SG smoothing depend on specifying three parameters:

1. the left window length, N_L;

2. the right window length, N_R;

3. the degree of the least squares smoothing polynomial (quadratic and quartic are popular choices).



(a) Polynomial coefficients in time domain(diffrent degree).

(b) Frequency response of polynomial coefficients(different degree).

Figure 7.1: Savitzky-Golay smoothing coefficients and the FFT amplitude response (degree of 3 and 6, length of 60).

The left plot in Fig. 7.1 represents a plot of the c_n (SG filter coefficients) versus n for third and sixth degree local smoothing polynomials for the case of a centered window of length 61. Note Eq. 7.1 represents a discrete convolution. Convolution in the time domain is equivalent to multiplication in the frequency domain. The right plot in Fig. 7.1 represents the amplitude versus frequency obtained by applying the fast Fourier transform to $\{c_n\}$. Amplitudes of low frequencies are high but amplitudes corresponding to frequencies above 0.1 are small in both cases. The results illustrate that SG smoothing acts like a low-pass filter with low frequency components preserved and higher frequency components severely damped.

7.1.2 Wavelet With Soft-thresholds Smoothing Algorithm

Wavelets ([12, 19, 2]) have been used in a variety of applications for data compression and signal processing. In the petroleum engineering literature, the wavelet transform has been applied to permanent pressure gauge data to remove noise (denoising) and identify events [44, 6]. Kikani and He [44] recommend a translation invariant wavelet transform using the Haar wavelet with soft thresholding for denoising and use the modulus maximus principal to identify rate changes and maintain its position in time. Athichanagorn and Horne [6] indicate that a spline wavelet is more suitable for event detection (e.g., identification of rate changes) and introduce a hybrid thresholding procedure for denoising. Specifically, they use soft thresholding in regions where data is continuous and hard thresholding near discontinuities. For the permanent pressure gauge data, it may be reasonable to assume measurement errors are uncorrelated, but this assumption can not be expected to apply for production and time-lapse seismic data. Thus, to apply wavelet transforms to separate the noise from the true signal to estimate the covariance functions for production and seismic data measurement errors will require the implementation of procedures that are robust when measurement errors are correlated in time (production data) or space (4D seismic data). The work of Johnstone and Silverman [42] indicates that it is possible to apply denoising even when noise is correlated.

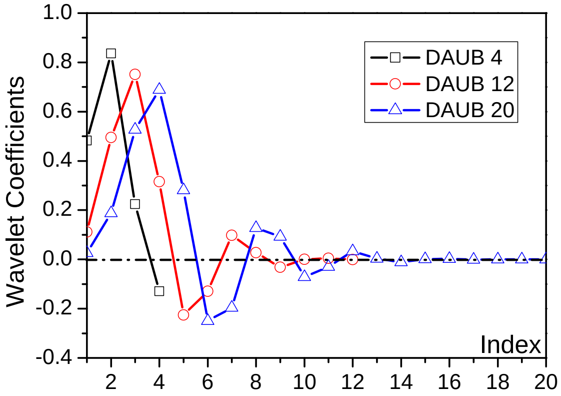


Figure 7.2: Daubechies wavelet coefficients.

A wavelet transform can be applied to decompose a measured signal into several scales, ranging from the smoothest scale to the most detailed scale. In this study, Daubechies wavelet with 20 coefficients (numerical recipe [65]) is used. The

FORTRAN code in this reference is also used. Fig. 7.2 shows the wavelet coefficients if we use the wavelets with 4 coefficients (DAUB4), 12 coefficients (DAUB12) or 20 coefficients (DAUB20). We use the DAUB20 because it is the smoothest, and has better separation of high and low frequency components compared to the other.

Suppose we have sequentially N measurements:

$$\{d_i, i = 1, N\}$$

The discrete wavelet transform technically assumes the number of data is $N = 2^M$ for some integer M. If the length of the data is not a power of 2, we have simply added zeros at the end of the data series to satisfy this requirement. After wavelet transform of the data, we get M scales of wavelet coefficients. The first scale has 2 data, i.e. $A_1 = \{a_{1,1}, a_{1,2}\}$, and from the second scale on they can be written as:

$$A_j = \{a_{j,k}, k = 1, 2, \cdots, 2^{j-1}\}, j = 2, \cdots, M$$

with the j^{th} scale having 2^{j-1} coefficients, and the total number of coefficients is N.

Each scale gives us a unique view on the data being analyzed. It is also a linear transform, and its inverse transform can be applied very efficiently.

Decomposition of the signal into different scales by applying the wavelet transform provides a basis for denoising. Effectively, denoising assumes that the true underlying signal is smooth and has only low frequency components whereas the noise in the measured signal is non-smooth, of high frequency and low energy relative to the true signal.

Under these conditions, after wavelet transform, most coefficients on the most detailed scales represent noise and most coefficients on the smoothest scales represent the true smooth underlying signal. More importantly, the coefficients on the most detailed scale that represent noise are relatively small. Thus, by zeroing these coefficients we can eliminate the components of noise. By taking the inverse wavelet transform of the modified representation of the wavelet transform, we obtain a smoother signal (the denoised signal) which provides an approximation to the true underlying signal.

Soft-thresholds: In our denoising procedure, we use a soft thresholding technique as done by Donoho & Johnstone([19]). In this process, we select a threshold level, δ_j, for scale j, and modify the wavelet coefficients of the detail components at scale j (A_j) as follows:

$$a_{j,n} = \begin{cases} \text{sign}\big(a_{j,n}\big)\big(|a_{j,n}| - \delta_j\big) & \text{for } |a_{j,n}| > \delta_j, \\ 0, & \text{for } |a_{j,n}| \leq \delta_j. \end{cases} \tag{7.3}$$

Note the threshold, δ_j can be different on each scale. Unfortunately, the true signal can have some high frequency components (detail components) and noise, particularly correlated noise, can have some low frequency (smooth) components. Thus, at many scales in the wavelet domain, the true signal and the noise overlap and denoising is much more difficult in this situation. In our approach, we follow ideas of Donoho and Johnstone [20] to denoise data. In this approach, we apply a different threshold on each scale. The threshold, δ_j for the jth scale is computed as follows:

$$\delta_{\text{m}ad} = \frac{\text{median}\big\{|a_{j,1}|, |a_{j,1}|, \cdots, |a_{j,n_j}|\big\}}{0.6745}$$

$$\delta_j = \delta_{\text{mad}} \sqrt{\ln(N)} \tag{7.4}$$

where N is the number of data, and n_j is the number of coefficients in the jth scale.

7.1.3 Covariance of Measurement Error

A good estimation of the true data gives a reasonable estimate of the covariance of the error. Letting d_i denote the noisy data and g_i denote the estimate of the true data, the error is estimated by

$$e_i = f_i - g_i, \tag{7.5}$$

and then the covariance is calculated as

$$C_j = \frac{1}{N_j - 1} \sum_{(k_1 - k_2) = j} \big(e_{k_1} - \bar{e}\big)\big(e_{k_2} - \bar{e}\big), \tag{7.6}$$

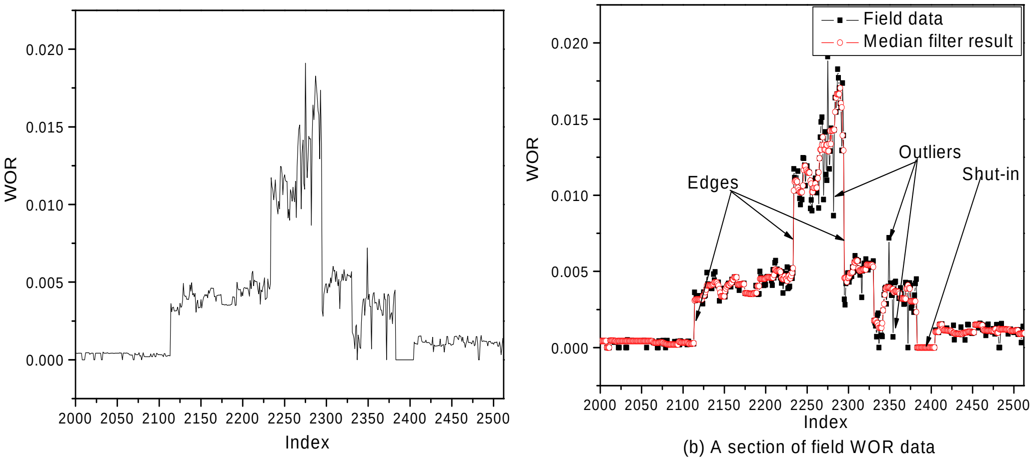
for $j = 1, 2, \cdots N_{max}$ where N_{max} is the maximum correlation distance to be estimated, which is less than $N/2$. N_j is number of pairs in the summation. C_j is the estimated covariance of measurement error at a distance of j times grid size, and $C(0)$ is the estimate of the variance. Here $\bar{e}$ is the mean of the error, i.e.,

$$\bar{e} = \frac{1}{N} \sum_{k=1}^{N} e_k. \tag{7.7}$$

7.2 Effects of outliers and edges

An outlier is a data point (t_i, d_i) (measured at t_i) such that d_i is significantly larger or smaller than the measured values at nearby points. Edges refer to locations in the data set where the underlying data is discontinuous; although the derivative of a function would not exist at a point of discontinuity, in discrete data sets, an approximate derivative computed with a finite difference approximation would give a large value at an edge compared to derivatives at nearby points. In production data, an edge occurs due to a significant change in operating conditions, e.g., shutting in the well. Outliers and edges have an adverse impact on the denoising process so it is important to identify and remove edges prior to denoising.

Outliers and edges are very common in field data; Fig. 7.3 is a segment of WOR data from a North Sea reservoir; it is not difficult to visually pick out some edges and outliers. However, we need an automatic way to do this. As discussed in



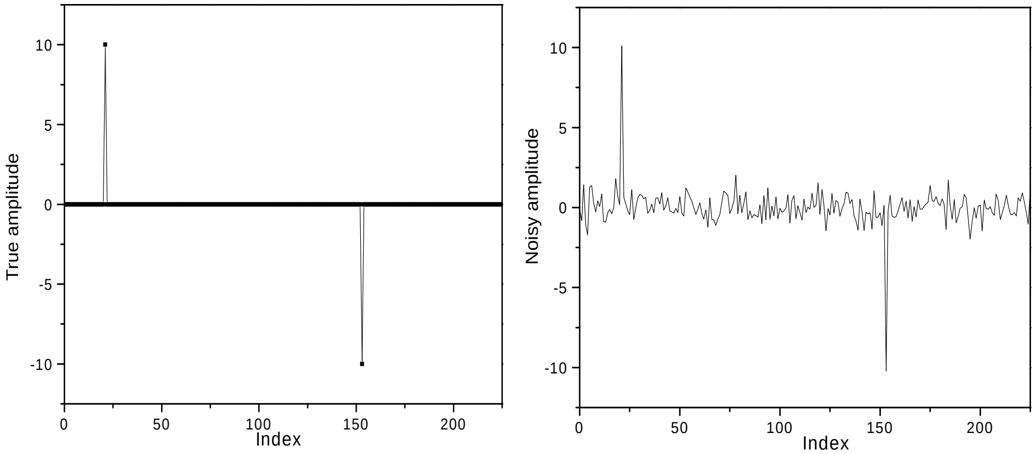
(a) A section of field WOR data (512 samples).

(b) A section of field WOR data and its median filter result (512 samples).

Figure 7.3: A section of field WOR data with outliers and edges

detail later, we use a median filter to identify edges and outliers. The outliers and edges identified by this process are shown in the right panel of Fig. 7.3. The shutin period also has two edges but we treat the shutin period separately as we have no water-oil-ratio (WOR) or gas-oil-ratio (GOR) data recorded during shutin periods.

7.2.1 Adverse Impact of Outliers on Denoising

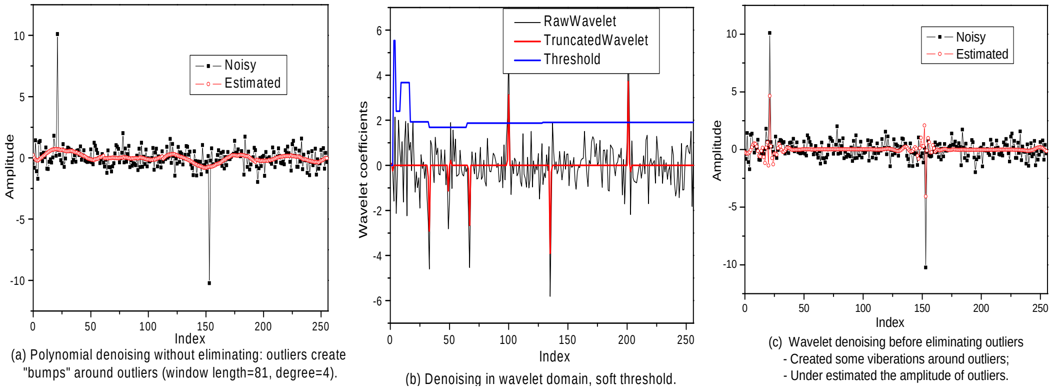


(a) Example of signal with outliers.

(b) Noisy data with two impulse as outliers.

Figure 7.4: An example of outliers, N=256. Two impulses are assumed to be outliers.

In order to illustrate the effect of outliers, we consider a simple example shown in (Fig. 7.4): A true signal was generated by starting with a signal that is identically zero for $i = 1, 2, \cdots N = 256$ and adding two impulses as two outliers. Then an array of uncorrelated Gaussian random numbers were added to the signal to represent noise that we wish to eliminate.



(a) Polynomial denoising without eliminating: outliers create "bumps" around outliers (window length=81, degree=4).

(b) Denoising in wavelet domain, soft threshold.

(c) Wavelet denoising before eliminating outliers
- Created some viberations around outliers;
- Under estimated the amplitude of outliers.

Figure 7.5: Denoising without removing outliers.

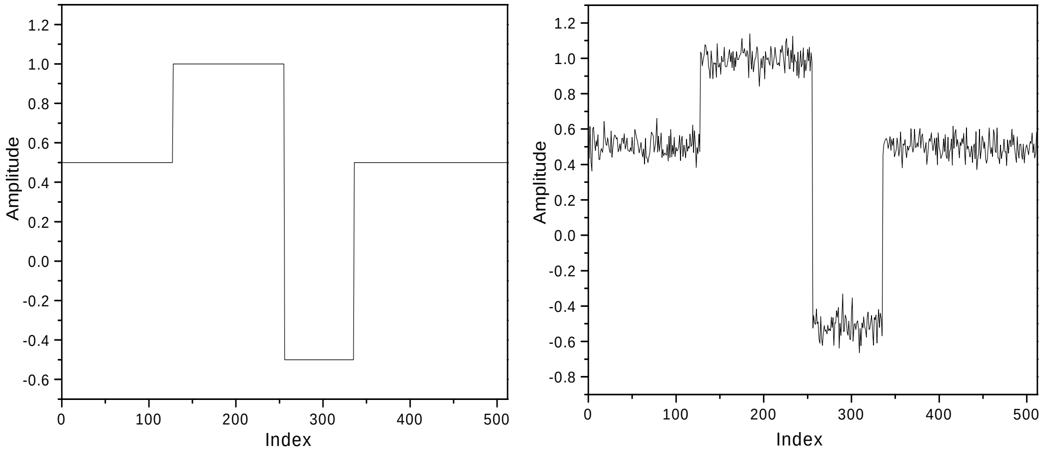
If our algorithms were immune of the existence of outliers, we should be able to recover the true amplitude of impulses and at the same time eliminate or significantly reduce the noise. But as shown in 7.5, we are unable to do this with either SG polynomial smoothing or wavelet denoising.

1. Savitzky-Golay polynomial smoothing fits data within a window about a point with a least squares polynomial. As is well known, outliers have a pronounced effect on least squares estimation. For the example under consideration, the outliers cause SG polynomial smoothing to introduce bumps into the estimated signal at the outlier points even though we have used a very long window length of 81.

2. As suggested by Fig. 7.5(b), in the wavelet domain, outliers tend to have significant components on most scales, so when we apply wavelet thresholding on these scales, we delete some of the energy of the outlier. When back transformed to the time domain (see Fig. 7.5(c)), this can lead to an underestimate

of the amplitude of the outliers and an over estimate of the error at times corresponding to outliers.

7.2.2 Adverse Impact of Edges on Denoising



(a) Signal with 3 edges and 4 steps.

(b) Esamples with 3 edges and uncorrelated noise.

Figure 7.6: An example of data with edges, N=512.

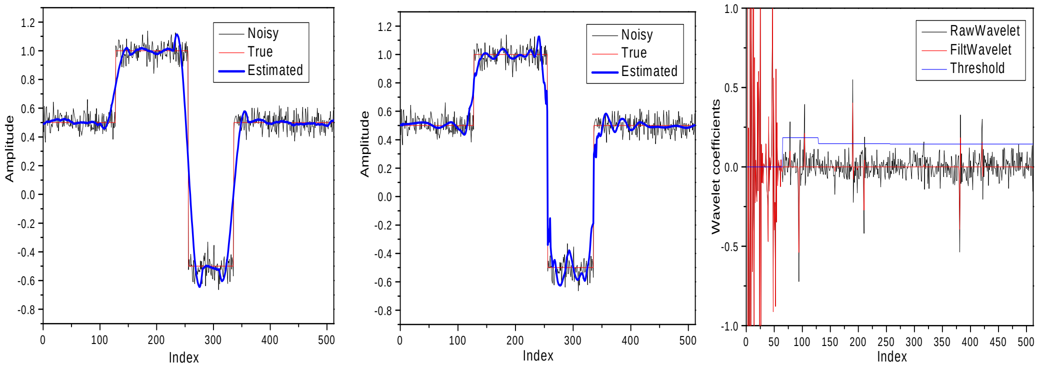
In order to study the effect of edges on polynomial and wavelet denoising, we constructed the simple example shown in Fig. 7.6.

The true signal is a piecewise constant function. The true data consists of 512 uniformly distributed measurements (N=512). Note there are three edges at $N = 128, 256, 329$, respectively. Uncorrelated Gaussian noise was added to the true data to obtain the noisy data shown on the right plot of Fig. 7.6.

Now we directly apply polynomial and wavelet denoising program to the data, and results shown in Fig. 7.7:

1. As shown in Fig. 7.7, edges have a strong effect on both polynomial smoothing and wavelet denoising.

2. In the case of SG polynomial smoothing, if a smoothing window contain an edge, the effect of SG polynomial smoothing (which is a low pass filter) is to



(a) Polynomial denoising: window length=81, degree=4.

(b) Wavelet denoising: keeping 5 scales.

(c) Wavelet denoising (wavelet domain), keeping 5 scales.

Figure 7.7: The effect of edges on denoising.

smooth the edge which introduces errors in the estimated signal especially at points near the edge.

3. When transformed into the wavelet domain (Fig. 7.7(c)), edges have distinctively big coefficients on the detailed scales. The effect is similar to that of outliers; applying soft thresholding we destroy some of the energy corresponding to the edges, so after back transform Fig. 7.7(c), the data are over smoothed and the estimated signal does not accurately depict the edges.

7.2.3 Detect and Remove Outliers

To detect and remove outliers we use median filtering. It is suitable for this purpose because:

1. This procedure can smooth the signal efficiently with a very small window;

2. Most importantly, it can preserve edges very well, so we do not have to worry that the existence of edges will have a detrimental effect on our ability to detect outliers.

In this work, we call this algorithm as **Median Filtering Algorithm (MFA)** for outlier detection. Select a window length (we always use 5 or 7). Here we use 7. For a window length of 7, we replace the data point d_i by the median of the set of values $\{d_{i-3}, d_{i-2}, d_{i-1}, d_i, d_{i+1}, d_{i+2}, d_{i+3}\}$. The resulting data set is referred to as the median smoothed array, $\tilde{D}$, with the ith data point in this array is denoted by $\tilde{d}_i$. Although the new array should be outlier free, it is typically too smooth and in this case, some valuable information on components of the error may be lost and we may even eliminate some of the true signal if the true signal has relatively sharp changes. Thus, we simply use the median smoothed array to detect outliers. Fig. 7.8(a) shows the median smoothed result of a data set containing outliers and edges compared to the true data set. We can see that outliers deviate significantly from the median smoothed data.

Using the median smoothed array $\tilde{D}$, we use following procedures to detect and remove the effect of outliers:

1. Assume the median to be the mean of the data at each sample, and estimate the standard deviation of the error by

$$\tilde{\sigma} = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N} (d_i - \tilde{d}_i)^2}. \tag{7.8}$$

   Here, N is the total number of data points. Any data point d_i that does not satisfy the condition

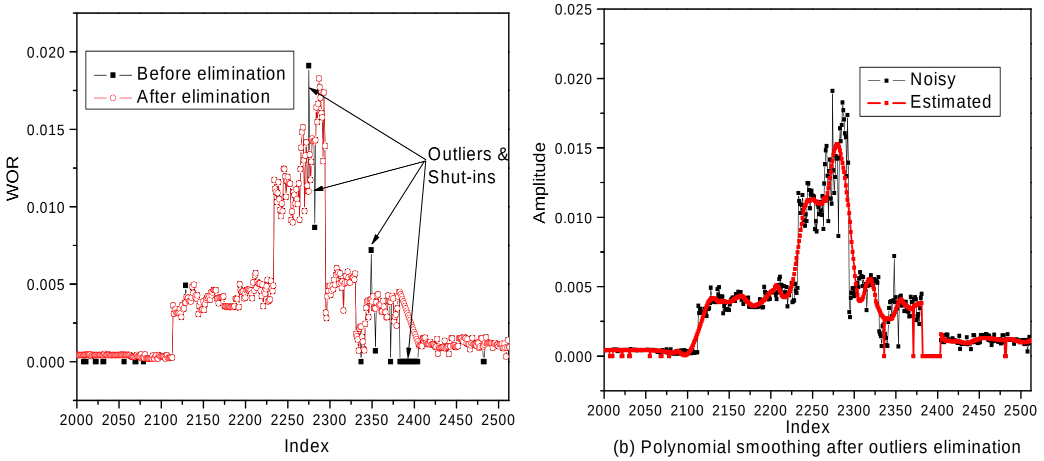
$$\tilde{d}_i - 3\tilde{\sigma} \leq d_i \leq \tilde{d}_i + 3\tilde{\sigma}. \tag{7.9}$$

   are the detected outliers.

2. An outlier free data set $\{\hat{D} = \hat{d}_i, i = 1, 2, \cdots, N\}$ is constructed by removing the outliers from the data set, i.e., $\hat{d}_i = \tilde{d}_i$ if d_i is an outlier, and $\hat{d}_i = d_i$ if it is not an outlier.

3. In the case of production data, we temporarily remove data values corresponding to shutin periods, and we add it back to the record after denoising because it might indicate an event, and we smooth the signal $\hat{D}$ just for the purpose of

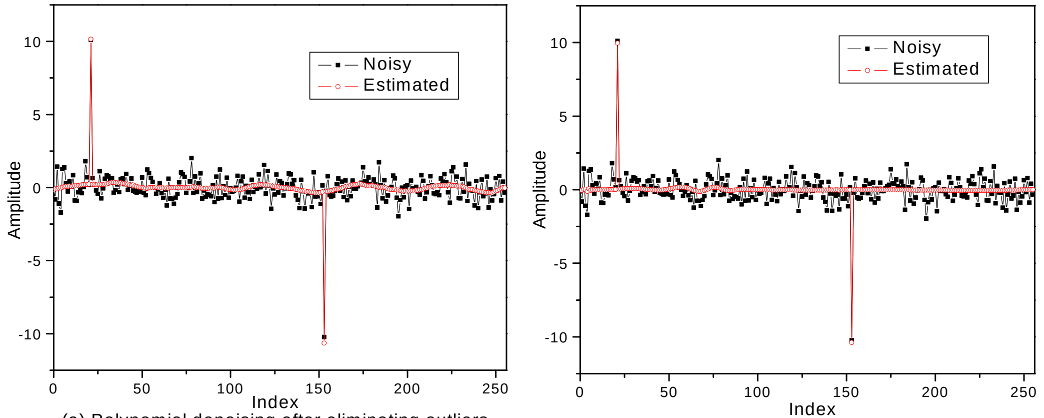
measurement error estimation. Considering the shutin's may be of several continuous samples, we prefer to interpolate linearly between them since median filtered data set is not guaranteed to be continuous in this case.



(a) Noisy signal after outlier detection.

(b) Polynomial smoothing after outliers elimination (degree=4; window length region(5,60)).

Figure 7.8: Outlier detection result from field data



(a) Polynomial denoising after eliminating outliers
Good estimation of outliers; no effect on its neighbors.
(Window length=81, Degree=4).

(b) Wavelet denoising result after eliminating outliers
- No viberations around outliers;
- Good estimation of outliers.

Figure 7.9: Outlier example: denoise with polynomial and wavelet.

Fig. 7.9 reconsiders the example of the smoothing results shown in Fig. 7.5. In this case, we have applied the MFA (median filter algorithm) to remove outliers, smoothed the resulting data, then (unlike our normal procedure of completely discarding outliers) added the values of the outliers back to the estimated signal at the appropriate times. In this case, we have good recovery of the zero signal, in fact almost perfect in the case of wavelet denoising, and good estimates of the amplitudes of the two impulses.

Fig.7.8(b) shows results obtained by applying the same procedure to the WOR field data considered in Fig.7.3. It seems that we have over smoothed around edges, which will have a detrimental effect on the estimation of the error. We should be able to obtain improved results by detecting edges and properly accounting for them.

7.2.4 Detect and Remove Edges

Here, we present a procedure for detecting edges (or boundaries). It is important to note that this procedure should be applied after the removal of outliers, as the process can be adversely affected by the presence of outliers. The edge detection process will effectively segregate the data into periods (flow periods for production data) separated by boundaries so that the underlying trend of the data is relatively smooth within each period.

The following steps are used to detect edges, and we call it as **Edge Detection Algorithm.**

1. First apply the wavelet transform and thresholding on all scales to denoise the data. This process will not preserve the true signal but will make the sharp change in derivatives at an edge very distinct. The wavelet smoothed data (smooth component) is denoted by s_i, $i = 1, 2, \cdots N$ in the time domain.

2. Apply the 5-point median filter to the s_i's to obtain median filtered signal $\tilde{s}_i$, $i = 1, 2, \cdots N$.

3. Calculate approximations to the right and left first-order derivatives of the filtered data, $\tilde{s}_i$'s. The left and right derivatives at t_i, $i = 1, 2 \cdots, N$, are

approximated, respectively, by

$$a_{l,i} = \frac{\tilde{s}_i - \tilde{s}_{i-1}}{\delta t}, \tag{7.10}$$

for $i = 2, \cdots N$ and

$$a_{r,i} = \frac{\tilde{s}_{i+1} - \tilde{s}_i}{\delta t} \tag{7.11}$$

for $i = 1, \cdots N - 1$.

Calculate the absolute difference in these values as

$$b_i = |\Big(|a_{l,i}| - |a_{r,i}|\Big)|, \tag{7.12}$$

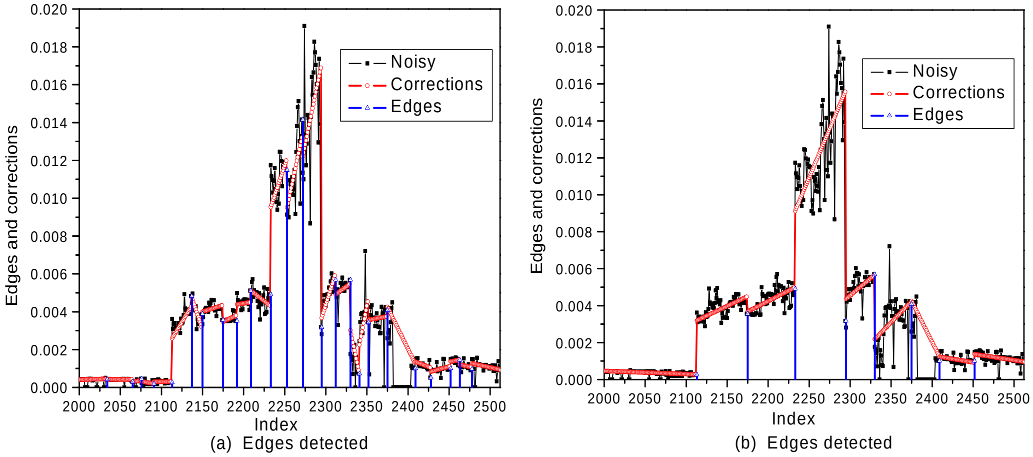
for $i = 2, \cdots N - 1$ and set $b_2 = b_1$ and $b_N = b_{N-1}$.

4. Apply the median filter with a long window length W_e (we define W_e as the edge detect window length and for the field production data W_e equals to 201) to the b_i's to obtain the median filtered results denoted by $\tilde{b}_i$, $i = 1, 2, \cdots N$.

5. If $b_i > 10\tilde{b}_i$, then we tentatively say d_i corresponds to an edge point. It is tentative because with field production data, rapid changes take place over very short periods of time, we may identify too many edges, and some of them might indicate high measurement error values. So we have to remove some of the edges. At this point, we suppose there are K tentative edges occur at i_j, $j = 1, 2, \cdots K$ where $i_1 < i_2 < \cdots < i_K$.

6. For each j, $j = 1, 2, \cdots, K$, consider all potential edges i_l such that $i_j - M \leq i_l \leq i_j + M$. We define M as the minimum edge distance, and we currently use $M = 10$ or $M = 30$. If $b_{i_j} \geq b_{i_l}$ for all i_l in this window, we keep i_j as an edge point. Otherwise we eliminate it. At the end of this process, we have reduced the number of edges. Finally we add the points $i = 1$ and $i = N$ as

edge points.



(a) Edges detected
(detecting window=100; minimum edge distance=10).

(b) Edges detected
(detecting window=100; minimum edge distance=30).

Figure 7.10: Edge detection results

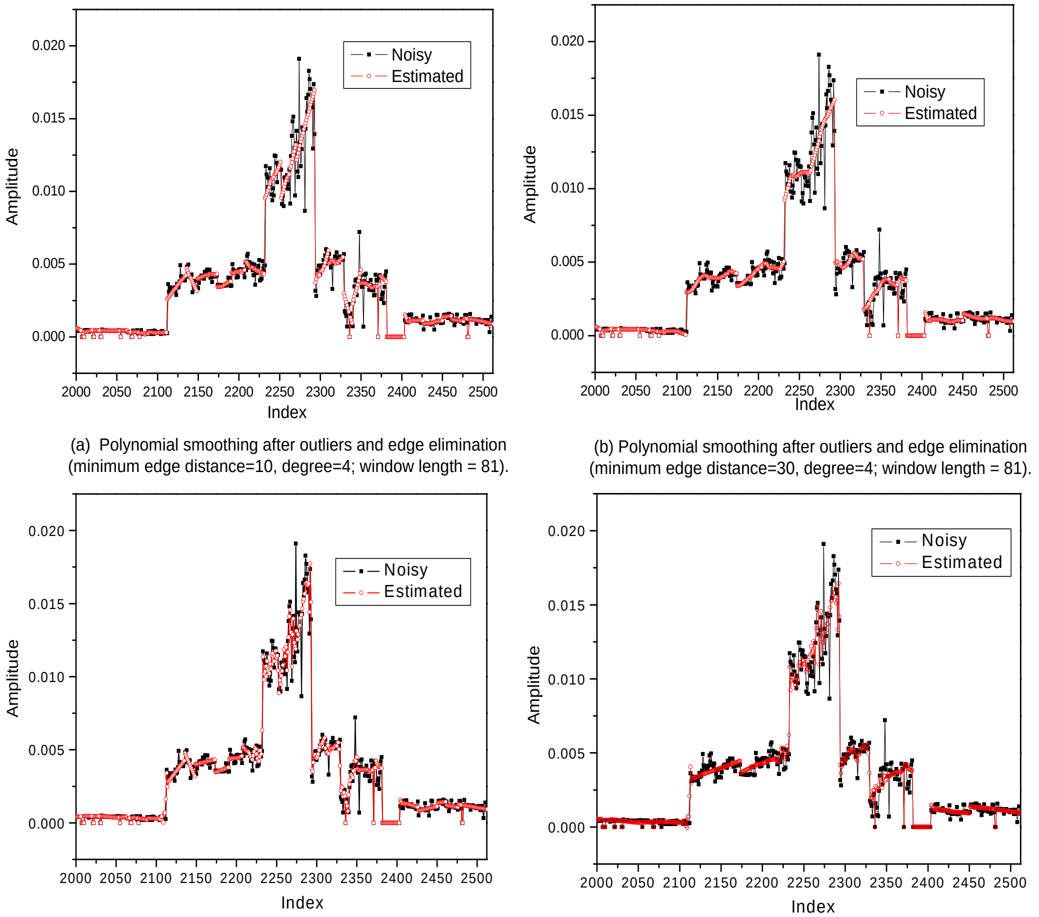
Fig. 7.10 shows the results obtained from the edge detection algorithm applied to the WOR field data of Fig. 7.3, for two different values of M ($M = 10$ and $M = 30$). The red lines are the result obtained by appling a linear least squares fit to the data between each two successive edges, i.e., to each period. By subtracting these lines from the data, we can obtain transformed data which is edge free. We then apply the denoising procedure to estimate the "true signal" for this edge free transformed data, and add the linear parts back to get the final estimate of the signal. The results are shown in Fig. 7.11.

7.2.5 Work-flow of Measurement Error Estimation

The following is the procedure to estimate measurement error considering outliers and edges might exist in the measurements:

1. Detect the outliers and edges from the noisy measurements;

2. Subtract outliers (i.e. shutins) and linear interpolation between edges from the data, and get corrected noisy measurements;



(a) Polynomial smoothing after outliers and edge elimination (minimum edge distance=10, degree=4; window length = 81).

(b) Polynomial smoothing after outliers and edge elimination (minimum edge distance=30, degree=4; window length = 81).

(c) Wavelet smoothing after outliers and edge elimination (minimum edge distance=10; keeping 6 scales).

(d) Wavelet smoothing after outliers and edge elimination (minimum edge distance=30, keeping 6 scales).

Figure 7.11: Smoothing results after edge and outlier elimination

3. Apply wavelet or SG smoothing algorithm to the corrected noisy measurements (outlier and edge free data);

4. Add back necessary outliers and all the edges to the data to reconstruct the estimated true signal;

5. Subtract the estimated true signal from the original noisy measurements and get estimated measurement error;

6. Characterize the measurement error by calculating the covariance function.

7.3 Measurement Error Characterization

After smoothing, we subtract the estimated smooth signal from the original data, to get the estimated error sequence e_i, $i = 1, 2, \ldots N$ and estimate the covariance as described in Eqs. 7.5, 7.6 and 7.7. However, the error in field data may not be stationary. In this case, non-stationarity should be considered in both smoothing procedures and covariance calculation procedures.

For the non-stationary case, we may first compute the expectation (mean) of the error at each point using a moving window average

$$E[e_i] = \frac{1}{N_1 + N_2 + 1} \sum_{j=i-N_1}^{i+N_2} e_j, \tag{7.13}$$

with left and right window size of N_1 and N_2. For the field production data, we have used $N_1 = N_2 = 500$. Then for $i = 101, 201, 301, \cdots N - 100$, we estimate the covariance localized at ith data at a "distance" of k by

$$\mathrm{Cov}(k)_i = \frac{1}{(N_k)_i - 1} \sum_{\substack{k_1,k_2=i-N_1 \\ k_1-k_2=k}}^{i+N_2} [e_{k_1} - E(e_{k_1})][e_{k_2} - E(e_{k_2})] \tag{7.14}$$

where $(N_k)_i$ is the number of pairs in the summation, and C(0) provides the estimate of the variance. Note this procedure will actually provide a set of variances and covariances. If the underlying error function were actually stationary, the sets of covariances should be essentially identical.

7.4 Synthetic Case Study

Here we consider one synthetic example constructed as essentially a smooth version of an actual field data, and a field production GOR data. The noisy data of the synthetic example is shown in Fig. 7.12. There are 512 data points. The data contains 4 internal edges, and the three significant edges can be easily found, but finding the "small edge" at $i = 334$ requires a more careful examination. The synthetic data was generated by adding Gaussian error generated from a Gaussian covariance function with a variance of 0.848e-3 and a correlation length of 10 points. The data contain synthetic shutin periods where no data were recorded.

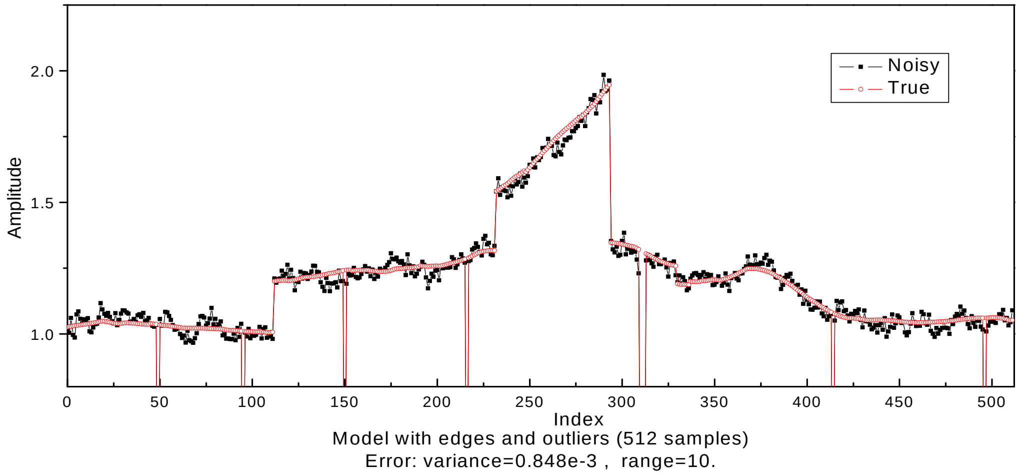


Figure 7.12: Model defined (512 samples)

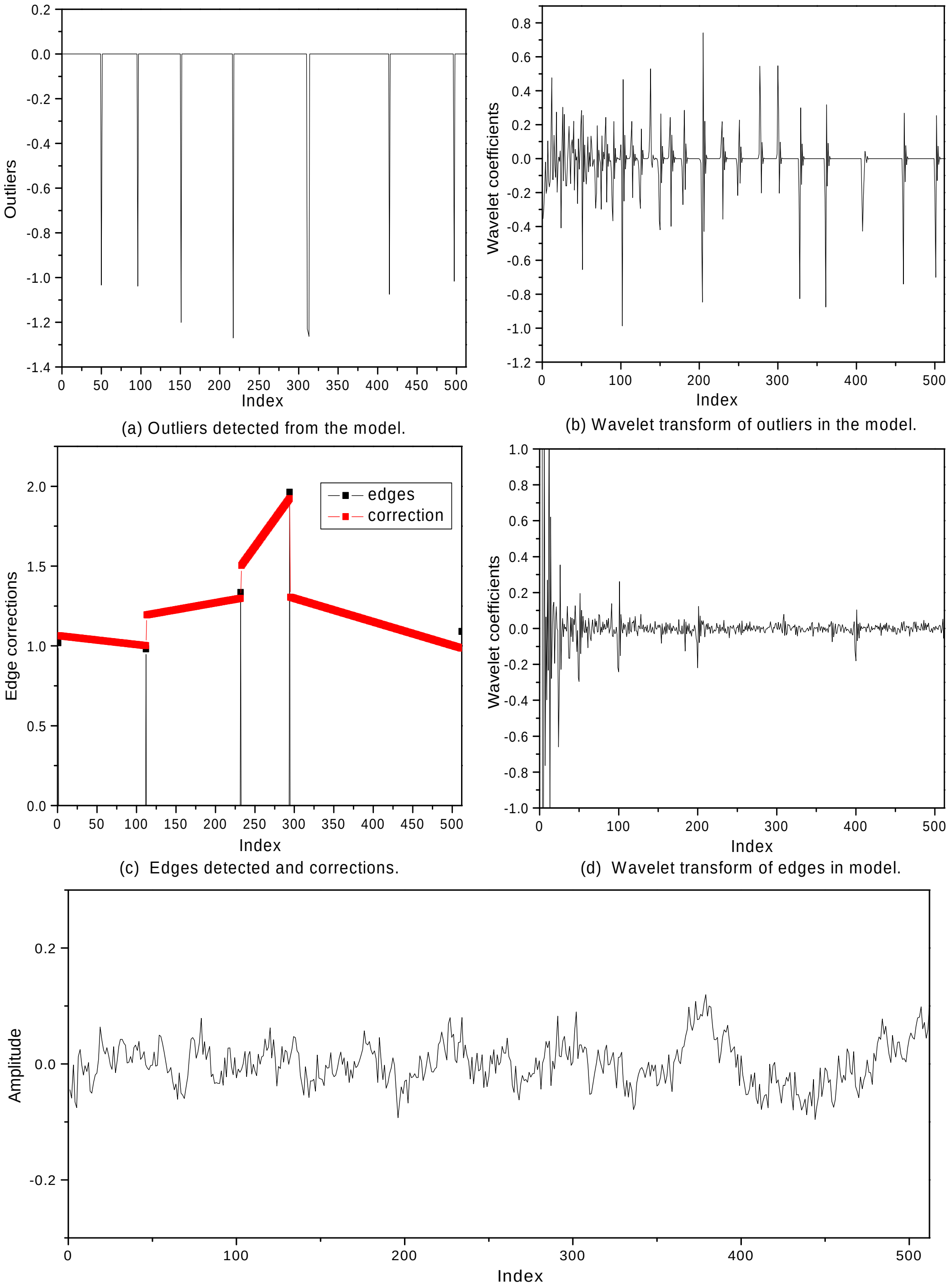
7.4.1 *Detect and Remove Outliers and Edges*

We call this step the data preprocessing step, which corresponds to the first and second step of the work-flow.

1. Apply MFA for outliers detection. Shutin's are replaced by a linear interpolation, and will be preserved in the followed processes. The other outliers will be removed forever.

2. Apply the edge detection algorithm. By subtracting least square fitted lines between edges from the data as described in edge detection section, a transformed edge free data set was obtained for denoising.

Fig. 7.13 is the results of the data preprocessing steps. Fig. 7.13(a) and (b) are detected outliers and their wavelet transform, Fig. 7.13(c) and (d) are detected



(a) Outliers detected from the model.

(b) Wavelet transform of outliers in the model.

(c) Edges detected and corrections.

(d) Wavelet transform of edges in model.

(e) Data set after outlier and edge elimination.

Figure 7.13: Data preprocessing, synthetic data

edges and their wavelet transform. Fig. 7.13(e) shows the data without edges and outliers.

1. All of the shutin's were detected correctly, and 3 out of 4 edges were correctly detected. The small edge in the data set close to $i = 334$ was not detected because the magnitude of the jump in the data is not significantly larger than the amplitude of the noise.

2. Corrections for edges are also effective. Most of the edge components have been removed and the transformed data set is much easier to smooth. This transformed data set is shown as in Fig. 7.13(e).

3. In this case, the specified edge detection window length: $W_e = 100$, minimum edge distance: $M = 10$.

7.4.2 Polynomial Denoising

After data preprocessing, we apply the SG polynomial smoothing to the corrected data (free of outliers and edge effects).

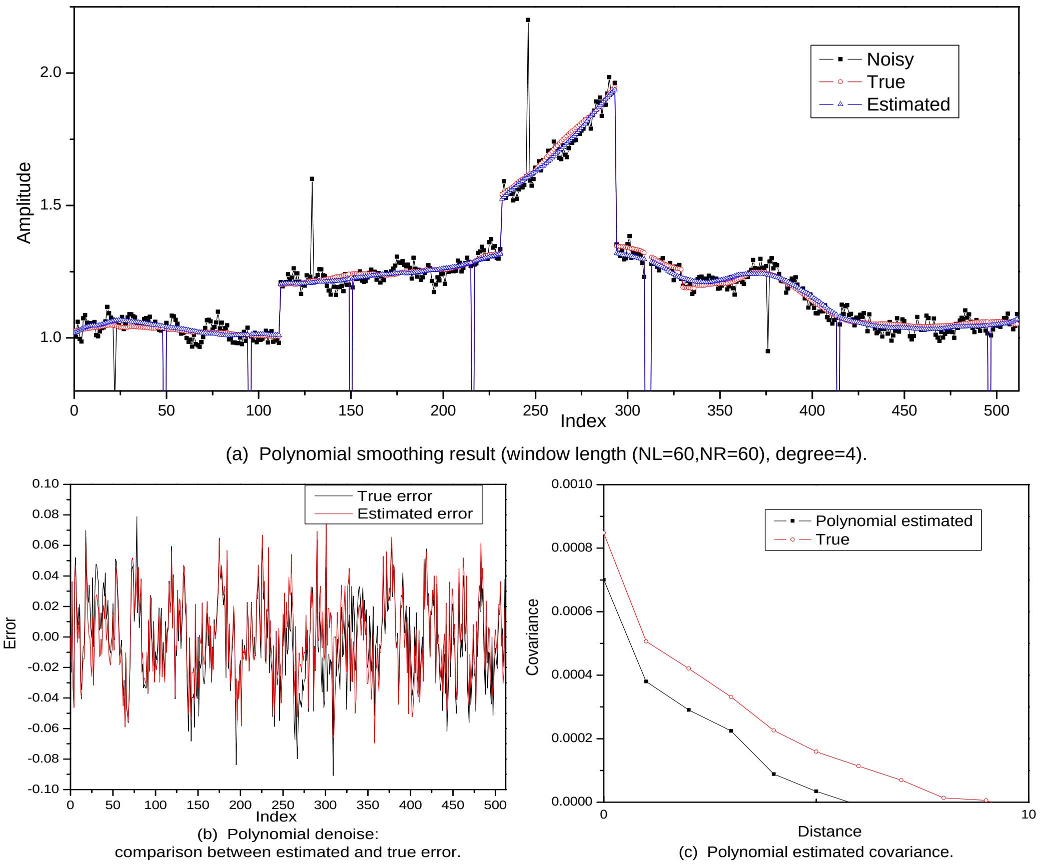
1. The window length is $N_L = N_R = 60$;

2. The degree of the SG smoothing polynomial is fixed as 4;

Fig. 7.14 shows the results obtained for this example.

1. Fig. 7.14(a) shows the true and noisy data and the smoothed data estimated based on the polynomial smoothing algorithm with outerlier and edge elimination. Note the estimated signal has the shutin periods restored and the three major edges are still apparent.

2. Most importantly, the covariance estimated from Eqs. 7.5, 7.6 and 7.7 gives an approximation to the covariance computed directly from the realization of the noise that was added to the true signal. However, the variance, covariance and



(a) Polynomial smoothing result (window length (NL=60,NR=60), degree=4).

(b) Polynomial denoise:
comparison between estimated and true error.

(c) Polynomial estimated covariance.

Figure 7.14: Results from SG polynomial smoothing.

correlation range are somewhat under estimated Fig. 7.14(b,c).

7.4.3 Wavelet Denoising

In wavelet denoising, we just keep 4 smooth scales and apply soft thresholding on the other scales. Fig. 7.15 shows the results of the smoothing algorithm as well as the resulting estimation of the covariance of the measurement error. Fig. 7.15(a) is the wavelet transform of the edge and outlier free data from Fig. 7.13(e), and Fig. 7.15(b) is the summation of the wavelet transform of detected outliers and edges. The edge and outlier components in all scales are significant and are of complicated structures. If we apply soft thresholding directly on the noisy data with edges and outliers, it is quite possible to undermine these structures. To keep the structures, we take the edges and outliers out before smoothing them. Fig. 7.15(c) through (e) shows the smoothing results. The effects of outliers and edges are greatly minimized as shown in the results. The reason why noise and signal are separated so well is because outliers and edges are efficiently eliminated and the remaining signal contains only some low-frequency energy.

7.5 Field Case Study

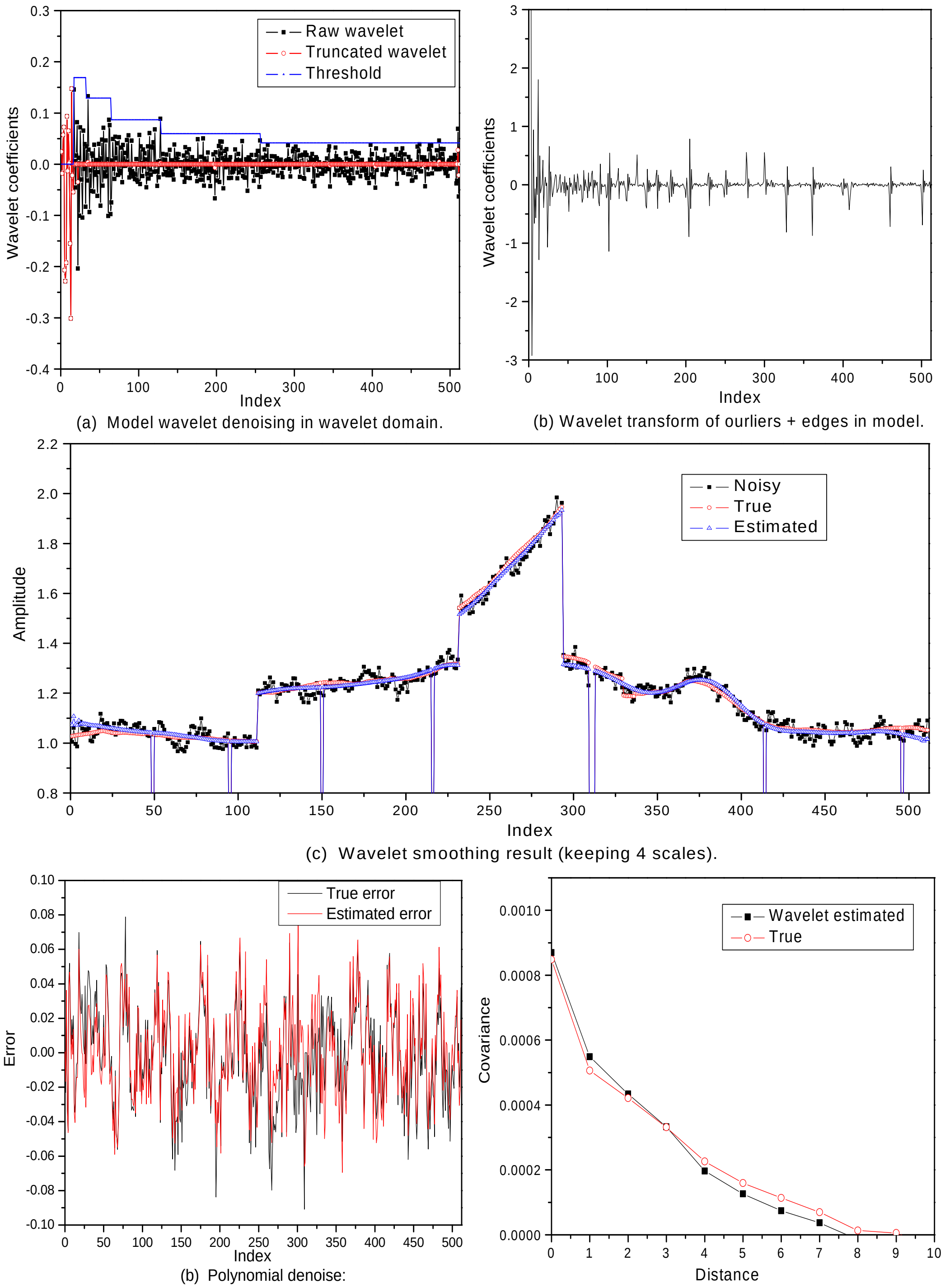
Fig. 7.16 shows the field producing GOR data (std m^3 per m^3). Data points are separated by 1 day so the axis can be considered to be either i, the index of the data point, or time in days. Except for numerous shutin's, the GOR data is essentially constant up to about $i = 2000$. There are 2800 data points. When necessary (in wavelet transform), we append 1296 zeros at the end of the array to make it to be the power of 2, and after denoising, only the first 2800 data points are kept. Therefore, the input of wavelet denoising is 2800 data points and output is also 2800 data points.

7.5.1 Detect and Remove Outliers and Edges

The same data preprocessing procedures and settings are applied as those in the synthetic case. The results are shown in Fig. 7.17(a) and (b) are detected outliers and edges; (c) and (d) are wavelet transform of the detected outliers and edges. Hence, (a) and (b) will be the correction to polynomial smoothing, (c) and (d) will be taken out and preserved in wavelet denoising procedure. (e) shows the results after outliers and edges elimination, which are much more suitable for both polynomial and wavelet denoising methods than the data in Fig. 7.16.

7.5.2 Polynomial Denoising

The polynomial denoising procedure is applied with the parameters as follows:



(a) Model wavelet denoising in wavelet domain.

(b) Wavelet transform of ourliers + edges in model.

(c) Wavelet smoothing result (keeping 4 scales).

(b) Polynomial denoise:
comparison between estimated and true error.

(e) Wavelet estimated covariance.

Figure 7.15: Synthetic Data wavelet smoothing result

1. The window lengths are $N_L = N_R = 50$;

2. The degree of the SG smoothing polynomial is fixed to 4;

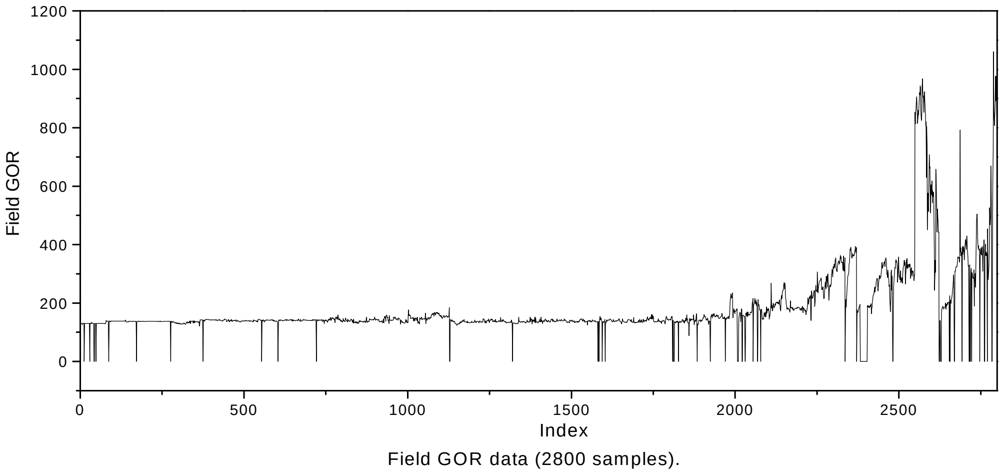


Figure 7.16: Field GOR data (2800 samples)

Fig. 7.18 shows the results.

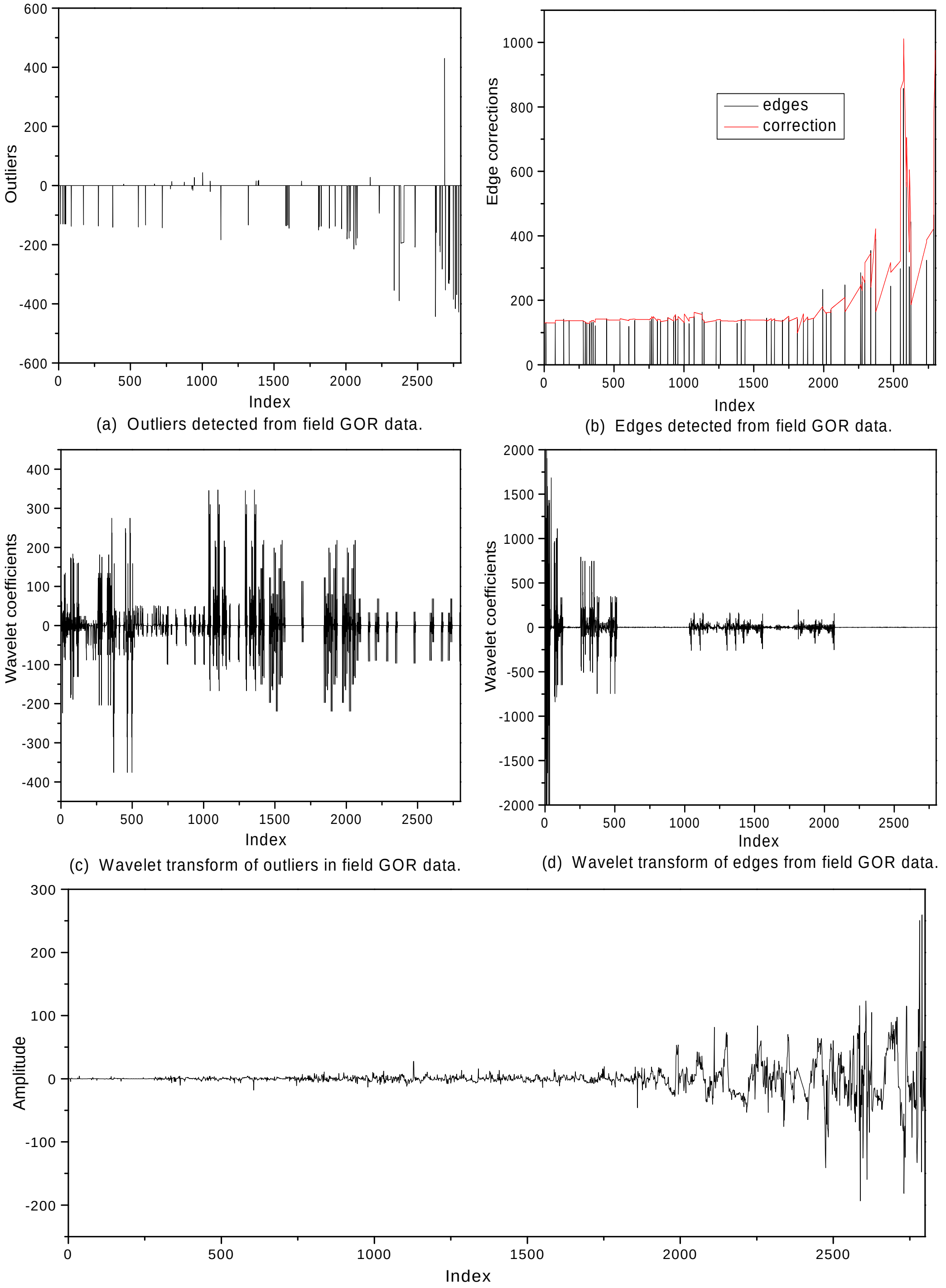
We estimated the covariance as a function of time using Eqs. 7.13 and 7.14. Fig. 7.18(b) shows estimated error, and Fig. 7.18(c) shows the covariances of the estimated error in Fig. 7.18(b). By estimating covariance using moving windows (centers move 100 points at a time), we obtain the approximate change of variance and range as the analyzing window moves from left to right. It is noted that the variances later than the 17th increase dramatically.

7.5.3 Wavelet Denoising

In wavelet denoising, we keep the 4 smoothest scales and apply soft thresholding on the other scales. Fig. 7.19 shows the results of wavelet denoising and the estimated covariances as a function of time. The following comments are made.

1. The results again indicate that the error is non-stationary. The energy in the late time is very strong in all scales. This might be a problem in the following aspects:

   (a) It tends to increase the estimation of error level in Eq. 7.4 because the medians of the components in all scales will be overestimated.



(a) Outliers detected from field GOR data.

(b) Edges detected from field GOR data.

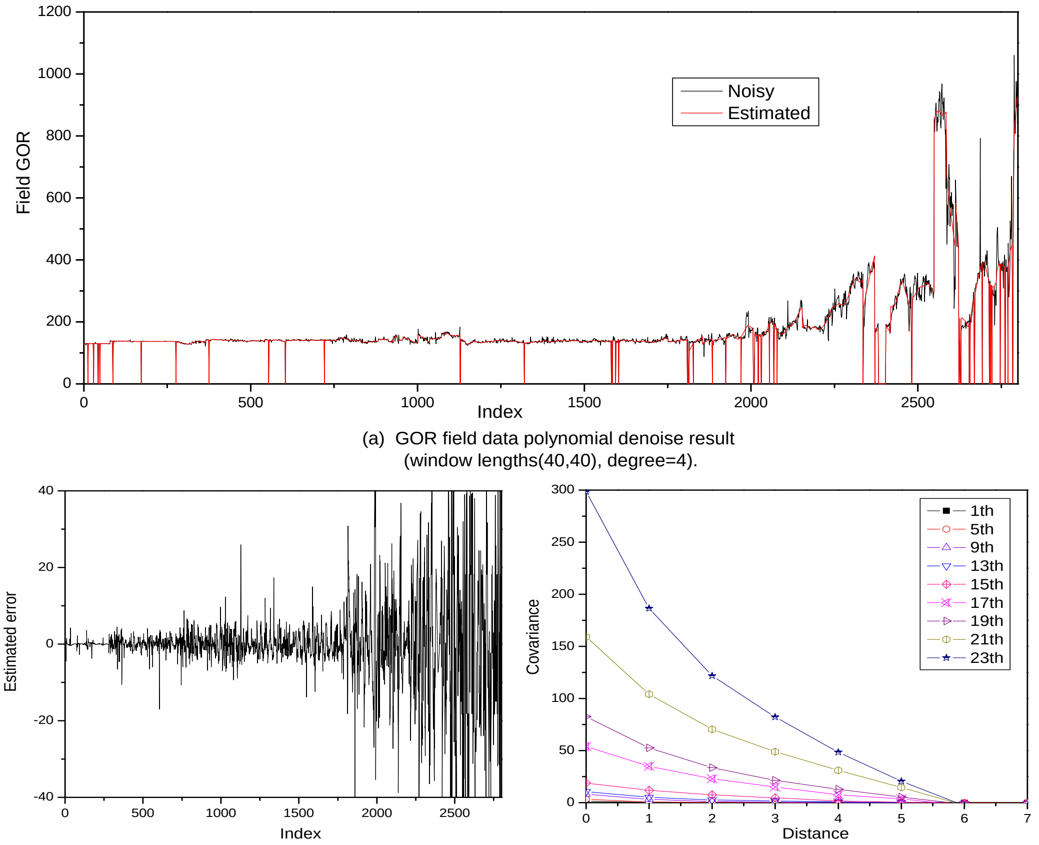
(c) Wavelet transform of outliers in field GOR data.

(d) Wavelet transform of edges from field GOR data.

(e) Field GOR data after eliminating effects of ourlier and edges.

Figure 7.17: Field GOR data preparation results

(b) In the late time, a large number of components exceeds thresholds. Ac-



(a) GOR field data polynomial denoise result (window lengths(40,40), degree=4).

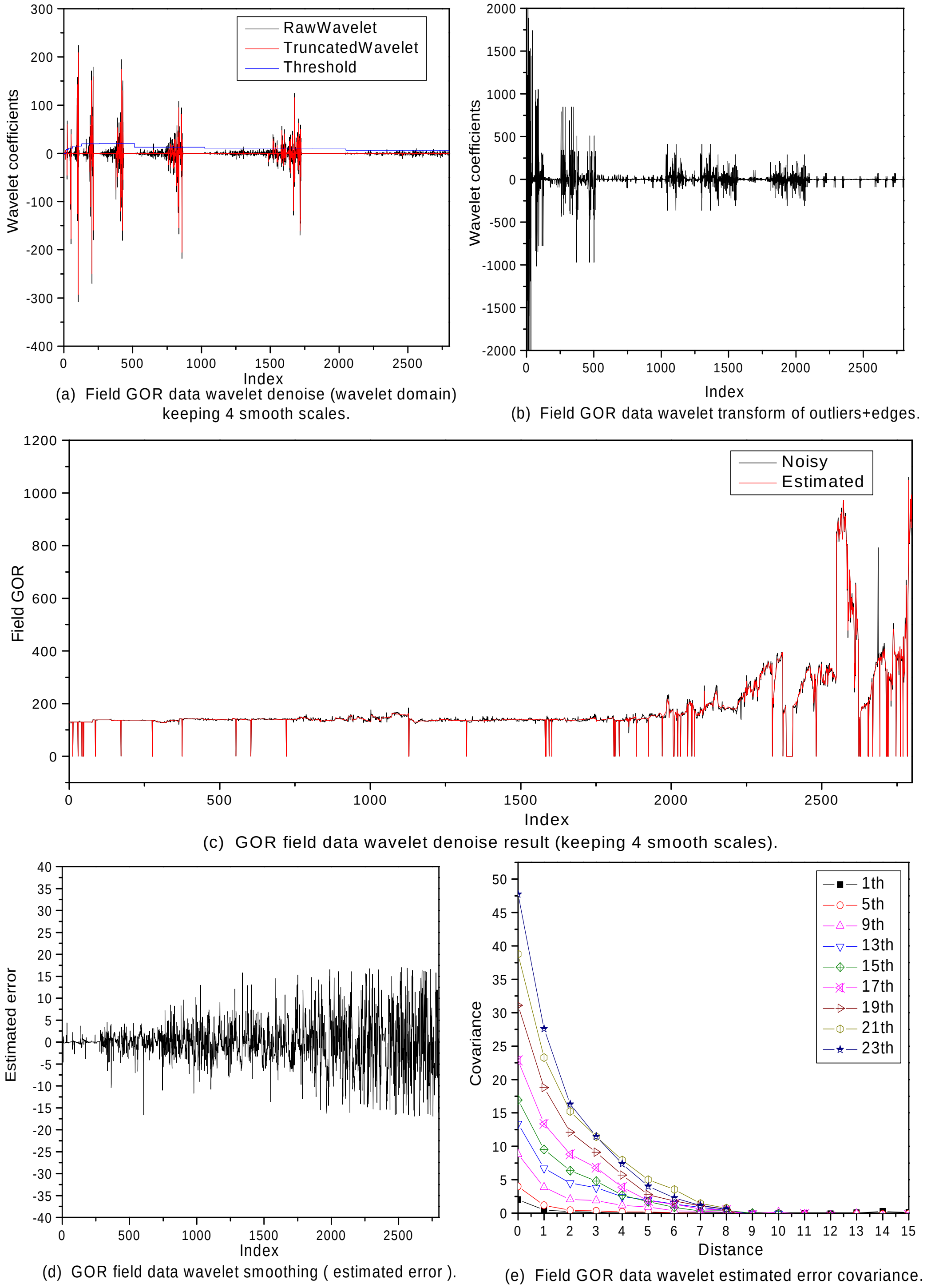
(b) GOR field data polynomial estimated error.

(c) Field GOR data polynomial estimated error covariance.

Figure 7.18: Field GOR Data polynomial smoothing result

cording to soft thresholding, and the error components on these locations are assigned the value with the same amplitude as the threshold. Because threshold is close to the upper limit of the error, this operation tends to overestimate the error on those locations.

2. The estimated covariances on different time intervals are smoother than those estimated by polynomial method, but unfortunately, this does not mean they are more accurate. Wavelet soft thresholding tends to generate a more stable estimation of error than polynomial denoising, since polynomials have much more difficulties in smoothing an irregularly shaped signal.



(a) Field GOR data wavelet denoise (wavelet domain) keeping 4 smooth scales.

(b) Field GOR data wavelet transform of outliers+edges.

(c) GOR field data wavelet denoise result (keeping 4 smooth scales).

(d) GOR field data wavelet smoothing (estimated error).

(e) Field GOR data wavelet estimated error covariance.

Figure 7.19: Field GOR Data wavelet smoothing result

7.6 Summary

Our initial research for separating production data into true signal and error was based on two smoothing techniques: Savitzky-Golay filters and soft threshold denoising of the wavelet transform. Applying the Savitzky-Golay filter is equivalent to performing local least squares fits of data within a moving window with a polynomial. The main difficulty with this procedure is choosing appropriate window lengths. The appropriate choice depends strongly on the correlation length of the error. If the error represents white noise, then relatively short window lengths are appropriate. If the error is correlated and thus has some smooth components, longer window lengths should be used. However, if long window lengths are used, the derivative of the true underlying smooth signal should not change sharply within a window because smoothing across such sharp changes introduces error in the estimated signal.

The main difficulty in decomposing noisy field data into error and true signal is that the procedures depend to some extent on the smoothness of the true signal and its derivatives as well as to whether the noise is stationary and/or correlated. It is particularly important that rapid changes in the observed signal due to events are accounted for so that events are not interpreted as noise when estimating the error.

In order to handle the problems associated with non-stationary and correlated errors, we have made some efforts in the algorithms to reduce their effects on the estimated error:

1. A procedure to eliminate outliers is applied because outliers lead to overestimation of the variance of the measurement error. The median filtering is used to recognize these outliers. The reason is that although it is sensitive to the outliers, it can preserve the structure of the signal especially even close to edges. This is also why we first apply outlier removal before detecting the edges. The outliers detected are replaced by the medians obtained from their neighborhood in order to remove their effects, and we will add them back after denoising when necessary.

2. A procedure to eliminate edge effects is applied so that sharp changes in the underlying signal do not cause over estimates of the noise. The two first order derivatives (left and right) of the signal across the edges have significant differences than those from its neighbors. We use it combined with wavelet transform and median filter to detect these edges. Then the signal is partitioned into several continuous regions by these edges, and is corrected by the linearly interpolated values within each of the regions. The signal after correction should be free of discontinuities. Again, the inverse of these corrections needs to be performed after denoising to recover the structure of the back ground signal.

3. The estimation of the error covariance is based on a moving window allowing for the non-stationarity. We have guaranteed that the moving window smoothing based on edges and outliers removal is quite robust and reliable. In order to account for the non-stationarity of the measurement error, a moving window is used to localize the information in the data. In this way, we have non-stationary covariances depending on the location of the data being studied.

CHAPTER 8
ESTIMATION OF MEASUREMENT ERROR IN 4-D SEISMIC DATA USING AN EM ALGORITHM

In this study, 4-D seismic data are firstly grouped according to their measured values and spatial coordinates using an spatial EM algorithm. By smoothing the data group by group, we can avoid smoothing across discontinuities due to a fluid contact or a fluid front, and obtain improved estimates of measurement errors. We assume the data can be grouped so that data in a group can be approximated with a Gaussian distribution. The data in the whole reservoir is therefore represented by a Gaussian mixture model (GMM). The mean of each Gaussian model reflects the average value of the signal in that group and the variance reflects the variation of the underlying signal. As measurements in a specific group of data will be similar, a group should never contain points on the opposite side of a "discontinuity" such as a flood front or fluid contact. The task for grouping is now simplified to the problem of finding a Gaussian mixture model. The expectation maximization (EM) algorithm [37, 13, 67, 66, 57, 58, 47] has long been used for estimating the Gaussian mixture model parameters based on observed data. The EM algorithm is an iterative procedure and provides an approximation of the maximum likelihood estimate of the Gaussian model parameters using the observed data as well as soft membership information that gives the probability that a certain datum belongs to a certain group (Gaussian). As is shown by Redner et al. [67, 66], although EM algorithm has the problem that the likelihood function is unbounded above for some cases, and that convergence can be very slow, it is still widely used and has an important role in the mixture density estimation problem.

Since we need to implement smoothing within each group, it is desirable that data in a group are spatially continuous. Moreover, except for sharp changes in the data, for example, due to movements of fronts or abrupt changes in a well's operating conditions, we expect that two data measured at nearby locations or at slightly different times will be similar in value. However, the traditional EM algorithm considers only the measured values of the data while ignoring the spatial continuity. This can result in groups such that a group contains data of similar value where the spatial coordinates of the data in the group represent many small disconnected regions in space. Although this is not always a significant problem, it can make smoothing more difficult. Besag [9] forced spatial continuity within each group by conditionally approximating the local Markov random field. Allard and Guillot [3] applied an approximation to the classification EM algorithm (CEM, [11]) to group irregularly spaced data and recover the spatial correlation of measurements in each groups. This algorithm assumes that the measurements in each group are spatially correlated. However, both of these two methods require a priori knowledge of the correct number of groups, and for their methods, it has not been shown that the log-likelihood function increases from iteration to iteration, which is the basic characteristic of the EM algorithm. In this study, we propose an EM algorithm with spatial constraints to enhance the spatial continuity of each group. In our implementation of the spatial EM algorithm, grouping quality coefficients are applied to automatically delete some spatially scattered groups in order to find a parsimonious grouping of the measurements. After grouping, smoothing is done with a moving window quadratic fit within each individual group. The smoothed data represents the estimate of the underlying true signal and the difference of the smoothed data and original observed data is the estimate of measurement error. For our implementation, it can not be proved that the log-likelihood increases from iteration to iteration although examples indicate this is usually the case.

8.1 The Problem of Spatial Measurement Grouping

Although the measurements studied in most EM applications are non-sequential, the measurements discussed here (i.e. 4-D seismic data and production data) are spatially or temporally measured. For the convenience of discussion, we will refer to both spatial or temporal measurements as spatial measurements. The spatially measured data are assumed to be measured in a 2-D lattice composed of N grids: $X = \{x_i, i = 1, 2, \cdots, N\}$, with x_i being the coordinate of the ith grid. The measurements are located at these grids, and are defined as

$$\hat{D} = \{\hat{d}_i, i = 1, 2, \cdots, N\} \qquad \hat{d}_i \in R^n \tag{8.1}$$

Here, $\hat{d}_i$ is the measured value and x_i is the spatial coordinates (or time coordinate) of the ith measurement respectively. An individual $\hat{d}_i$ may be a scalar or a vector containing multiple data types at each location, i.e. in the case of 4-D seismic data, it can be composed of acoustic impedance change (AIC) and Poisson's ratio change (PRC).

The measured value is assumed to be the summation of the underlying true signal and measurement error. In 4-D seismic data, the underlying true signal is the AIC or PRC caused by the production of the reservoir, and the measurement error is the spatially distributed acquisition error and processing error introduced by the two seismic surveys.

In order to characterize the measurement error, we need to separate the two components mixed in the measurements. All we know is the measurements $\hat{D}$ and their spatial coordinates, X. A simple solution to this problem is to smooth the data using a moving window, use the smoothed signal as the estimated true signal, and then the difference between the smoothed data and the observed data represents the estimation of the measurement error. This approach actually assumes the underlying signal is smooth everywhere, which is not true in many cases. When there exist some spatially distinct changes in the measurement, the smoother will move a considerable proportion of the true signal into the measurement error. The estimation can be improved if we assume the true signal is regionally smooth, i.e., if we divide the measurement into multiple spatially distributed regions, and the true signal is smooth except at the sharp changes between regions. In order to do so, it is necessary to find a proper partition (or grouping) of the measurements. If we assume there are M groups, a membership indicator can be defined for each data to indicate the group it belongs. The set of membership indicators are given by

$$Z = \{z_i, i = 1, N\}, \tag{8.2}$$

where each z_i can take any one of the M discrete values in the set $\{1, 2, \cdots, M\}$. Thus, if $z_i = j$, the datum $\hat{d}_i$, measured at x_i, belongs to the jth group.

In this research, the EM algorithm [47], which will be discussed from the next section on, is used to estimate these partitions. In general, the EM algorithm optimizes a log-likelihood function, which represents the likelihood of model parameters when measurements are given. The traditional EM algorithm assumes the measurements are non-sequential, i.e., have no spatial or temporal relation. In the next two sections, the non-sequential EM algorithm (no incorporation of X into the estimation process) and a spatial EM algorithm, which includes information X by applying a spatial constraint, will be discussed.

After the group indicators Z are estimated, either from a MAP (maximum a posteriori) estimate or stochastically, we can use Z to estimate the measurement errors. Specifically a moving window quadratic fitting (Appendix D) is applied to the measurements group by group to avoid smoothing across the boundaries between groups. The smoothed signal is the estimated true signal, which is subtracted from the measurements to get the estimation of the measurement errors. A measurement error covariance matrix can be estimated from the estimated measurement errors, and used to construct the objective functions for automatic history matching.

8.2 Grouping Non-sequential Measurements Using EM Algorithm

The group indicator vector Z is going to be estimated knowing measured values $\hat{D}$ and the spatial coordinates of these measurements X. A Gaussian mixture model can be used to construct a log-likelihood function that the EM algorithm optimizes to solve this problem. Firstly, we discuss the traditional EM algorithm [47], which is used to group non-sequential measurements (without spatial or temporal coordinates). In this case, X is not used in the algorithm. EM stands for the two iterative steps of the algorithm, expectation and maximization.

8.2.1 Gaussian Mixture Model

In the Gaussian mixture model, two random vectors need to be constructed, i.e.

1. $\Delta = \{\delta_1, \delta_2, \cdots, \delta_N\}$ is used to model the group indicators of the N measurements, with the sample space for each δ_i given by $\{1, 2, \cdots, M\}$, where M is the number of groups. If measurements are non-sequential, it is assumed that the group indicators are sampled independently. Hence, the δ_i's are independent and follow the same probability distribution. The discrete PDF of each δ_i is given by

$$P(\delta_i = j|\Pi) = \pi_j, \; j = 1, 2, \cdots, M. \tag{8.3}$$

where P denotes a probability. The model parameter of this distribution is $\Pi = \{\pi_j, j = 1, 2, \cdots, M\}$, where π_j is normally referred to as the mixing proportion of the jth group, and represents the probability that the ith measurement is in the jth group regardless of measured values of data.

2. $D = \{d_1, d_2, \cdots, d_N\}$ is used to model the random data vector, i.e., the actual measured values $\hat{D}$ (Eq. 8.1) is a realization of D. Here, each d_i is a vector with n real components, where in the scalar case $n = 1$. As it is assumed, measured values are sampled independently, the d_i's are independent of each other, the distribution of each d_i depends only on δ_i. The conditional distribution that each d_i follows can be written as a Gaussian function:

$$P(d_i|\delta_i = j, \Theta) = \frac{1}{(2\pi)^{\frac{n}{2}}|C_j|^{\frac{1}{2}}} \exp\big[-\frac{(d_i - \mu_j)^T C_j^{-1}(d_i - \mu_j)}{2}\big]. \tag{8.4}$$

Here, μ_j and C_j are the mean vector and the covariance matrix of the jth Gaussian. For $i = 1, 2, \cdots, N$, δ_i is associated with d_i, i.e., $\delta_i = j$ if and only if the ith data belongs to the jth group, i.e., is a sample from the jth Gaussian. The measurements in the jth group (or partition) follows the jth Gaussian. The set of model parameters for these Gaussian PDFs is given by $\Theta = \{\mu_j, C_j, j = 1, 2, \cdots, M\}$.

For convenience, we use the notation for the jth Gaussian as

$$G(d_i|\Theta_j) = P(d_i|\delta_i = j, \Theta). \tag{8.5}$$

In this model, we assume the measurements are independently sampled. Thus, the PDF for each d_i can be written as a Gaussian mixture:

$$\begin{aligned} P(d_i|\Pi, \Theta) &= \sum_{j=1}^{M} P(d_i, \delta_i = j|\Theta) \\ &= \sum_{j=1}^{M} P(d_i|\delta_i = j, \Theta)P(\delta_i = j|\Pi) \\ &= \sum_{j=1}^{M} \pi_j G(d_i|\Theta_j) \end{aligned} \tag{8.6}$$

In order to group the measurements properly, we need to estimate the model parameters Π and Θ using only the knowledge of D. This is actually a "missing data problem" or incomplete data problem, because the estimation of Π and Θ requires information on both D and Z.

We estimate these model parameters by maximizing a log-likelihood function defined as:

$$\begin{aligned}
L(\Pi, \Theta|\hat{D}) &= \ln\left[P(D = \hat{D}|\Pi, \Theta)\right] \\
&= \ln\left[\prod_{i=1}^{N} P(d_i = \hat{d}_i|\Pi, \Theta)\right] \\
&= \ln\left[\prod_{i=1}^{N}\sum_{j=1}^{M} P(d_i = \hat{d}_i, \delta_i = j|\Pi, \Theta)\right] \\
&= \ln\left[\prod_{i=1}^{N}\sum_{j=1}^{M} P(d_i = \hat{d}_i|\delta_i = j, \Pi, \Theta)P(\delta_i = j|\Pi, \Theta)\right] \\
&= \sum_{i=1}^{N}\ln\left[\sum_{j=1}^{M}[\pi_j G(\hat{d}_i|\Theta_j)]\right], \qquad (8.7)
\end{aligned}$$

where the first equality comes from the assumption that d_i's are independent and the other equalities follows from Eq. 8.6. Because this function is difficult to differentiate, and can not be easily maximized, the EM algorithm was introduced. Next, we show the EM algorithm can be used to estimate a maximum of $L(\Pi, \Theta|\hat{D})$.

8.2.2 Non-sequential EM Algorithm

Although each measurement must be a sample from only one of these Gaussians, the set of Gaussian parameters (Π and Θ) for all the groups are to be estimated by trying to maximize an appropriate Q function discussed in Appendix A. This maximization step is only feasible if we have approximate information about the membership indicators. More specifically, we define an $N \times M$ membership matrix

H with the entry in the ith row and jth column defined by

$$h_i^j(\Pi, \Theta) = P(\delta_i = j | d_i = \hat{d}_i, \Pi, \Theta). \tag{8.8}$$

Note that $h_i^j(\Pi, \Theta)$ represents the probability that the ith measurement belongs to the jth group when the Gaussian model is given as Π and Θ. Here, we have only considered the case that we group by the measured value of the data, the $\hat{d}_i$'s; spatial information will be considered later in the next section. If we let δ_i^j be a random indicator variable which is equal to 1 when the ith data belongs to the jth group, i.e.

$$\delta_i^j = \begin{cases} 1 & \text{if } i\text{th data is from } j\text{th group} \\ 0 & \text{if } i\text{th data is not from } j\text{th group}, \end{cases} \tag{8.9}$$

then it can be shown [38] that Eq. 8.8 can be written as:

$$h_i^j(\Pi, \Theta) = P(\delta_i = j | d_i = \hat{d}_i, \Pi, \Theta) = P(\delta_i^j = 1 | d_i = \hat{d}_i, \Pi, \Theta) = E[\delta_i^j | d_i = \hat{d}_i, \Pi, \Theta], \tag{8.10}$$

where E denotes expectation. The last equation provides the reason why the estimation of h_i^j is called the expectation step in the EM algorithm. For each j, $\delta_i^j = 1$ is equivalent to $\delta_i = j$.

The non-sequential EM algorithm is given by Kung et al. [47], and uses only the measured values to approximate the parameters. The EM algorithm starts from an initial guess of the model Π_0, Θ_0, which can be used to evaluate the initial membership matrix $h_i^j(\Pi_0, \Theta_0)$ and log-likelihood $L(\Pi_0, \Theta_0 | \hat{D})$, and gradually increases the log likelihood function by updating the model through an expectation step and a maximization step.

Suppose the model at the nth iteration is Π_n, Θ_n. In the expectation step, the membership matrix is evaluated as the expectation of the missing data, the δ_i^j's. In the nth iteration, the expression for each entry of the membership matrix $h_i^j(\Pi_n, \Theta_n)$ defined by Eq. 8.10 can be derived by using Bayes theorem and Eq. 8.6 as follows:

$$\begin{aligned}
(h_i^j)_n \equiv h_i^j(\Pi_n, \Theta_n) &= P(\delta_i = j | d_i = \hat{d}_i, \Pi_n, \Theta_n) \\
&= \frac{P(\delta_i = j, d_i = \hat{d}_i | \Pi_n, \Theta_n)}{P(d_i = \hat{d}_i | \Pi_n, \Theta_n)} \\
&= \frac{P(d_i = \hat{d}_i | \delta_i = j, \Pi_n, \Theta_n) P(\delta_i = j | \Pi_n, \Theta_n)}{\sum_{j'=1}^{M} \left\{ P(d_i = \hat{d}_i | \delta_i = j', \Pi_n, \Theta_n) P(\delta_i = j' | \Pi_n, \Theta_n) \right\}} \\
&= \frac{(\pi_j)_n G(\hat{d}_i | (\Theta_j)_n)}{\sum_{j'=1}^{M} \left\{ (\pi_{j'})_n G(\hat{d}_i | (\Theta_{j'})_n) \right\}}.
\end{aligned} \tag{8.11}$$

Here, $(\pi_j)_n$ and $(\Theta_j)_n$ denote the estimated models parameter of the jth group at the nth iteration.

In the maximization step, according to Appendix C, the $(n+1)$th model Π_{n+1} and Θ_{n+1} are calculated using the following equations:

$$\begin{aligned}
(\mu_j)_{n+1} =& \frac{\sum_{i=1}^{N} (h_i^j)_n \hat{d}_i}{\sum_{i=1}^{N} (h_i^j)_n}, \\
(C_j)_{n+1} =& \frac{\sum_{i=1}^{N} (h_i^j)_n (\hat{d}_i - (\mu_j)_{n+1})(\hat{d}_i - (\mu_j)_{n+1})^T}{\sum_{i=1}^{N} (h_i^j)_n}, \\
(\pi_j)_{n+1} =& \frac{1}{N} \sum_{i=1}^{N} (h_i^j)_n,
\end{aligned} \tag{8.12}$$

which are evaluated based on the membership matrix H_n calculated from Π_n and Θ_n as the output of the expectation step. In order to group the data using the spatial information, the spatial EM is considered next, after which the step by step procedure will be described.

8.3 Spatial EM Algorithm

The traditional EM algorithm groups data according only to their values, while ignoring the spatial relationship between them. When this algorithm is applied to spatially correlated data, it gathers all the data with similar values together regardless of their locations. As noted earlier, this can create groups which are highly discontinuous spatially, which can make smoothing within groups tenuous. Moreover, in most cases we expect data measured at close locations to have similar values. In sequential measurement cases, we can still use the two random vectors Δ and D to model the group indicators and measured values. We can still assume d_i depends only on δ_i, but the δ_i's are not independent any more. Reviewing the Eqs. 8.6 and 8.7 we used in non-sequential cases, the term $P(\delta_i = j|\Pi, \Theta)$ become intractable because the δ_is are dependent to each other.

In the literature, two strategies have been proposed to construct a spatially-constrained EM. One strategy is to smooth the membership matrix h_i^j [16] group by group (for different j's). The basic idea of smoothing h_i^j is that when two data points are spatially close to each other, they tend to fall into the same group, which means they should have a similar probability (h_i^j) for being in any particular group (or are likely to be samples from the same Gaussian model). In the spatially-constrained EM algorithm used by Diplaros et al. [16], a smoothing step is added between the E-step and the M-step. Another strategy consists of adding a spatial penalty term to the log-likelihood function (neighborhood EM algorithm [4]). Since the penalty term of the spatial neighborhood EM algorithm does not contain the Gaussian mixture model parameters, Θ, defined in the previous section, the maximization step of this algorithm is equivalent to the traditional EM algorithm, but the membership matrix is modified to include the spatial information. This is equivalent to smoothing the membership matrix using the spatial information implicitly. Besag [9] uses a more natural way to incorporate spatial information into the log-likelihood function: the term $P(\delta_i = j|\Pi, \Theta)$ is modified to a conditional probability (conditional to a temporary evaluation of the membership indicator Z), which is especially suitable for the EM algorithm because the membership indicator is always re-evaluated from iteration to iteration. Our method for imposing a spatial constraint uses an idea similar to Besag [9], but the way we incorporate the spatial information is different. More importantly, our final algorithm incorporates a method to delete groups so we do not need to know the number of groups a priori.

As is shown later, the δ_i's are dependent on each other, and only a conditional discrete PDF can be defined based on an evaluation of the group indicator vector $\hat{Z} = \{\hat{z}_i, i = 1, 2 \cdots, N\}$, where we recall that $\hat{z}_i = j$ if $\hat{d}_i$ is from the jth group (Gaussian). In this study, the conditional PDF is defined as:

$$P(\delta_i = j | \Pi, \Theta, \hat{Z}) = \frac{\sum_{\hat{z}_k = j} \frac{1}{2\pi r_0^2} \exp\left[- \frac{\|x_i - x_k\|^2}{2r_0^2} \right]}{\sum_{k=1}^{N} \frac{1}{2\pi r_0^2} \exp\left[- \frac{\|x_i - x_k\|^2}{2r_0^2} \right]}, \tag{8.13}$$

for $j = 1, 2, \cdots, M$, and $i = 1, 2, \cdots, N$. In the numerator, the notation indicates the sum is over all k such that $\hat{z}_k = j$; in the denominator, the sum is over all data. Hence we have

$$\sum_{j=1}^{M} P(\delta_i = j | \Pi, \Theta, \hat{Z}) = 1.$$

Here, r_0 is a distance weighting factor used to construct the spatial constraints. Originally, we tried to estimate r_0 as part of the process but were unable to do so. Thus we fix it to be 2 or 3 (2 or 3 times width of a grid-block), and found that these values can work for the cases we have tried.

Assuming d_i is dependent only on δ_i, and the δ_i's are conditionally independent (conditional to $\hat{Z}$ as shown in Eq. 8.13). The conditional PDF $P(\Delta, D | \Theta, \Pi, \hat{Z})$ can be written as

$$\begin{aligned} P(\Delta, D | \Theta, \Pi, \hat{Z}) &= P(D | \Delta, \Theta, \Pi, \hat{Z}) P(\Delta | \Theta, \Pi, \hat{Z}) \\ &= \prod_{i=1}^{N} P(d_i | \delta_i, \Theta, \Pi, \hat{Z}) \prod_{i=1}^{N} P(\delta_i | \Theta, \Pi, \hat{Z}) \\ &= \prod_{i=1}^{N} P(\delta_i, d_i | \Theta, \Pi, \hat{Z}). \end{aligned} \tag{8.14}$$

Consequently, with spatially measured data, a modified log-likelihood function based on Eq. 8.13 is constructed for the estimation of the group indicator Z:

$$\begin{aligned}
L_c(\Pi, \Theta, \hat{Z}|\hat{D}) &= \ln\left[P(D = \hat{D}|\Pi, \Theta, \hat{Z})\right] \\
&= \ln\left[\prod_{i=1}^{N}\sum_{j=1}^{M} P(d_i = \hat{d}_i, \delta_i = j|\Pi, \Theta, \hat{Z})\right] \\
&= \ln\left[\prod_{i=1}^{N}\sum_{j=1}^{M} P(d_i = \hat{d}_i|\delta_i = j, \Pi, \Theta, \hat{Z})P(\delta_i = j|\Pi, \Theta, \hat{Z})\right] \\
&= \sum_{i=1}^{N} \ln\left[\sum_{j=1}^{M}[P(\delta_i = j|\Pi, \Theta, \hat{Z})G(\hat{d}_i|\Theta_j)]\right]. \qquad (8.15)
\end{aligned}$$

where $P(\delta_i = j|\Pi, \Theta, \hat{Z})$ is defined in Eq. 8.13. In our implementation of the EM algorithm, we always set $\hat{Z}$ as the current guess of membership Z_n at the nth iteration. The denominator of Eq. 8.13 is a constant value for all i and j, and the log-likelihood function of the nth iteration (according to Eq. 8.15) can be written as

$$L_c(\Pi_n, \Theta_n, Z_n|\hat{D}) = \sum_{i=1}^{N} \ln\left\{\sum_{j=1}^{M}\Big[\sum_{(z_k)_n=j} \frac{1}{2\pi r_0^2}\exp\left[-\frac{\|x_i - x_k\|^2}{2r_0^2}\right]G(\hat{d}_i|(\Theta_j)_n)\Big]\right\} + \text{Constant} \qquad (8.16)$$

An iterative two step strategy used in Besag [9] is similar to the EM algorithm we use here to estimate Π, Θ and Z:

1. E-Step: Membership matrix (H_n will be shown in Eq. 8.18) can be calculated based on the current guess of $\Pi = \Pi_n$, $\Theta = \Theta_n$ and $Z = Z_n$. The corresponding log-likelihood can be calculated as $L_c(\Pi_n, \Theta_n, Z_n|\hat{D})$ from Eq. 8.16.

2. M-Step: Update estimation of parameters $\Pi = \Pi_{n+1}$ and $\Theta = \Theta_{n+1}$ and $Z = Z_{n+1}$ using (H_n). The details will be shown later.

Using Eq. 8.13, and referring to Eq. 8.11, the membership matrix at the nth iteration can be written as:

$$h_i^j(\Pi_n, \Theta_n, Z_n) = \frac{P(\delta_i = j|\Pi_n, \Theta_n, Z_n)G(\hat{d}_i|(\Theta_j)_n)}{\sum_{j'=1}^{M}\left\{P(\delta_i = j'|\Pi_n, \Theta_n, Z_n)G(\hat{d}_i|(\Theta_{j'})_n)\right\}}$$
$$= \frac{\sum_{(z_k)_n=j}\frac{1}{2\pi r_0^2}\exp\left[-\frac{\|x_i-x_k\|^2}{2r_0^2}\right]G(\hat{d}_i|(\Theta_j)_n)}{\sum_{j'=1}^{M}\left\{\sum_{(z_k)_n=j'}\frac{1}{2\pi r_0^2}\exp\left[-\frac{\|x_i-x_k\|^2}{2r_0^2}\right]G(\hat{d}_i|(\Theta_{j'})_n)\right\}} \tag{8.17}$$

Although the global proportion of each group Π_n is still a parameter of interest, Eqs. 8.16 and 8.17 do not have this parameter in the right hand sides, because we are using a local proportion defined in Eq. 8.13. Suppose $(N_j)_n$ is the number of measurements that are from the jth group according to the current estimation of membership indicator Z_n. We have the global proportion of the jth group as $(\pi_j)_n \approx \frac{(N_j)_n}{N}$. By inserting the terms as $(\pi_j)_n\frac{N}{(N_j)_n} \approx 1$ into the numerator and denominator of Eq. 8.17, we obtain the modified membership matrix:

$$(h_i^j)_n = h_i^j(\Pi_n, \Theta_n, Z_n) = \frac{(\pi_j)_n G(\hat{d}_i|(\Theta_j)_n)S_j(x_i|r_0, Z_n)}{\sum_{j'=1}^{M}\left\{(\pi_{j'})_n G(\hat{d}_i|(\Theta_{j'})_n)S_{j'}(x_i|r_0, Z_n)\right\}}. \tag{8.18}$$

where S_j's are spatial weight functions for the jth group defined as

$$S_j(x_i|r_0, Z_n) = \frac{1}{(N_j)_n}\sum_{(z_k)_n=j}\frac{1}{2\pi r_0^2}\exp\left[-\frac{\|x_i-x_k\|^2}{2r_0^2}\right],\ i = 1, 2, \cdots, N, \tag{8.19}$$

for $j = 1, 2, \cdots, M$. Hence, in the nth iteration ($\hat{Z} = Z_n$), Eq. 8.13 can be written as

$$P(\delta_i = j|\Pi_n, \Theta_n, Z_n) = C(\pi_j)_n S_j(x_i|r_0, Z_n), \tag{8.20}$$

where C is a constant value. Note that for $i = 1, 2, \cdots, N$,

$$\sum_{j=1}^{M} h_i^j(\Pi_n, \Theta_n, Z_n) = 1. \tag{8.21}$$

For same reason, the log-likelihood function Eq. 8.16 can be written as

$$L_c(\Pi_n, \Theta_n, Z_n|\hat{D}) = \sum_{i=1}^{N} \ln \left\{ \sum_{j=1}^{M} \left[(\pi_j)_n S_j(x_i|r_0, Z_n) G(\hat{d}_i|(\Theta_j)_n)\right] \right\} + \text{Constant.} \tag{8.22}$$

In the implementation, an initial guess of grouping and model parameters are provided as Z_1, Π_1 and $\Theta_{s,1}$ as will be shown in the Section 5. They can be used to construct the first guess of the modified membership matrix H_1 (Eq. 8.18). In each iteration (i.e. the nth iteration), the parameter for the next iteration Θ_{n+1} and Π_{n+1} as well as the grouping Z_{n+1} can be estimated from H_n. The updated indicator variable Z_{n+1} can be generated as maximum a posteriori estimate (MAP estimate) or stochastically, i.e.,

1. MAP grouping: $(z_i)_{n+1}$ is calculated as

   $$(z_i)_{n+1} = \ell$$

   where

   $$\ell = \text{Argmax}_j \left[(h_i^j)_n\right].$$

2. Stochastic grouping: $(z_i)_{n+1}$ can be evaluated stochastically by sampling the cumulative distribution function for the ith row of the current membership matrix ($\{(h_i^j)_n, j = 1, 2, \cdots, M\}$).

At the end of the next section, we will see when to use MAP grouping and when to use stochastic grouping in the implementations.

As is shown in Appendix C, we still use Eq. 8.12 to update the model parameters in the maximization step except that $(h_i^j)_n$ terms in this equation are given by Eq. 8.18. Similar to comments made in Besag [9], we failed to find a way to prove that the log-likelihood function is non-decreasing from iteration to iteration (see Appendices A, B and C), but we observed it is increasing for all our cases when the number of groups is fixed as known. The next section will be used to show how we solve for the number of groups when it is uncertain.

8.4 Spatial EM Algorithm with Group Quality Coefficient

The above spatial EM algorithm does not account for the uncertainty in the number of groups. With this algorithm, we find that the number of groups tends to be conserved, i.e., we end up with the same number of groups as the number of groups in the initial grouping. From iteration to iteration in most cases, we need to estimate the number of groups (Gaussians). Richardson and Green [70] use a reversible jump Markov chain Monte Carlo (McMC) method to find the optimal number of groups. Because the McMC method is very computationally demanding, we use another approach to find the number of groups.

We start from a fairly large number of groups, and apply additional multipliers in the membership matrix. The membership matrix is modified to

$$h_i^j(\Pi_n, \Theta_n, Z_n) = \frac{(\pi_j)_n (F_j)_n G(\hat{d}_i|(\Theta_j)_n) S_j(x_i|r_0, Z_n)}{\sum_{j'=1}^{M} (\pi_{j'})_n (F_{j'})_n G(\hat{d}_i|(\Theta_{j'})_n) S_{j'}(x_i|r_0, Z_n)}. \tag{8.23}$$

According to Eqs. 8.4 and 8.19,

$$\begin{aligned} G(\hat{d}_i|(\Theta_j)_n) &= P(d_i = \hat{d}_i|\delta_i = j, \Theta_n) \qquad (8.24) \\ &= \frac{1}{(2\pi)^{\frac{n}{2}} |(C_j)_n|^{\frac{1}{2}}} \exp\big[-\frac{(\hat{d}_i - (\mu_j)_n)^T (C_j)_n^{-1} (\hat{d}_i - (\mu_j)_n)}{2} \big] \\ S_j(x_i|r_0, Z_n) &= \frac{1}{(N_j)_n} \sum_{(z_k)_n = j} \Big\{ \frac{1}{2\pi r_0^2} \exp\big[-\frac{\|x_i - x_k\|^2}{2r_0^2} \big] \Big\}. \qquad (8.25) \end{aligned}$$

Here, $(N_j)_n$ is the number of measurements in the jth group according to Z_n, and $(z_k)_n$ is the group indicator of the kth measurement in the nth iteration. $(F_j)_n$ is a variable used to indicate the quality of the jth group at the nth iteration. It is used to cause groups of low quality to be absorbed by groups of higher quality so that the number of groups will be gradually decreased until convergence. The quality of each group is judged by its continuity, and the groups that are sparsely distributed spatially tend to have lower quality than those continuously distributed.

Before defining $(F_j)_n$, the continuity of groups based on each single measurement has to be quantified. We use a grouping score matrix $(E_n)_{M \times N}$ at the nth iteration. $(E_i^j)_n$, the element in the ith row and jth column of this matrix, is used to indicate the continuity of the jth group at the ith measurement. $(E_i^j)_n$ is a number between 0 and 1. The continuity is good if the corresponding value is close to 1, and poor if it is close to 0.

Suppose that according to the current membership information, the ith measurement $\hat{d}_i$ is assigned to the jth group, then we wish $(E_i^j)_n$ to have the following properties:

1. If most points with a small distance of x_i (say within a distance $3r_0$ of x_i) are within group j, then we wish $(E_i^j)_n$ to be greater than $\frac{1}{2}$, and close to 1, if all points within a distance of $3r_0$ are assigned to group j based on the current membership information. Note in this situation, the jth group is expected to correspond to a spatially continuous region around x_i.

2. If most points within a distance of $3r_0$ of x_i do not fall within group j, we wish $(E_i^j)_n$ to be smaller than $\frac{1}{2}$, and in the extreme case, where all the points except x_i itself that are within $3r_0$ of x_i are not in group j, we wish $(E_i^j)_n$ to be close to zero.

We view groups which have a large percentage of their data satisfying the first situation to be of higher quality than groups which have a large percentage of their data satisfying the second situation. Based on this, the group quality coefficient for the jth group is defined as the average of the grouping scores of the measurements in the jth group. The grouping score matrix E_n can be defined in different ways. In this research, we use the spatial constraint defined in Eq. 8.25 to evaluate it with the current guess of indicator variables. i.e. for the measurement at x_i:

$$(E_i^j)_n = E_i^j(Z_n) = \frac{(N_j)_n S_j(x_i|r_0, Z_n)}{\sum_{j'=1}^{M}(N_{j'})_n S_{j'}(x_i|r_0, Z_n)}. \tag{8.26}$$

From Eq. 8.25, Eq. 8.26 can be rewritten as

$$(E_i^j)_n = E_i^j(Z_n) = \frac{\sum_{(z_k)_n=j}\left\{\exp\left[-\frac{\|x_i-x_k\|^2}{2r_0^2}\right]\right\}}{\sum_{j'=1}^{M}\sum_{(z_{k'})_n=j'}\left\{\exp\left[-\frac{\|x_i-x_{k'}\|^2}{2r_0^2}\right]\right\}}. \tag{8.27}$$

Because $\exp(-\frac{(3r_0)^2}{2r_0^2}) \approx 0.01$, we say the effective radius of the spatial constraint is $3r_0$. We define the group quality coefficients by

$$(F_j)_n = F_j(Z_n) = \frac{a}{(N_j)_n}\sum_{(z_k)_n=j}(E_k^j)_n, \tag{8.28}$$

for $j = 1, 2, \cdots M$ where

$$a = \left(\sum_{j=1}^{M}\left[(\pi_j)_n \frac{1}{(N_j)_n}\sum_{(z_k)_n=j}(E_k^j)_n\right]\right)^{-1} \tag{8.29}$$

is a normalizing constant which ensures that

$$\sum_{j=1}^{M}(\pi_j)_n(F_j)_n = 1, \tag{8.30}$$

because the term $(\pi_j)_n(F_j)_n$ is effectively the modified mixing proportion of the jth group in the nth iteration, and we want them to sum to 1. Here, the value of a does not have any effect on the results because it can be canceled from the nominator and denominator of the membership matrix shown in Eq. 8.23. Note that in the definition of $(F_j)_n$ only one entry in each row of E_n is used, i.e. for the ith row, only $(E_i^j)_n$ is used if the ith measurement falls into the jth group at the nth iteration. When the grouping quality coefficients are not equal to 1.0, we compute log-likelihood by considering $(\pi_j)_n(F_j)_n$ as modified mixing proportion, i.e.,

$$L_{c,f}(\Pi_n, \Theta_n, Z_n|\hat{D}) = \sum_{i=1}^{N} \ln \left\{ \sum_{j=1}^{M} \left[(\pi_j)_n (F_j)_n S_j(x_i|r_0, Z_n) G(\hat{d}_i|(\Theta_j)_n) \right] \right\} + \text{Constant}. \tag{8.31}$$

Because the number of groups is not conserved as constant during the process of estimating the number of groups, the log-likelihood calculated from Eq. 8.31 may decrease as the number of groups decrease. Here, F_j actually serves as a factor that can cause biases in π_j's. Therefore, a second run without F_j, using the obtained final groups as the initial, should be followed to remove the biases caused by F_j. In the second run, the log-likelihood function is expected to increase.

In this study, stochastic grouping will be used in the case when the number of group is exactly known as a priori. If the number of groups is not known, stochastic grouping is used in the first EM process when the number of groups is being determined. This is because computational experiments indicate that it is more robust than grouping based on the MAP (maximum a posteriori) estimate in two ways:

1. More robust in finding the number of groups: From the experiments we made in Zhao et al. [85], we found that the stochastic grouping has some superior properties over MAP grouping, which leads to better robustness in the initial grouping.

2. Help the EM processes to escape from some local maxima and reach higher maxima: Celeux and Govaert [11] explained that stochastic grouping can relax the problem of strong dependence on the initial guess of parameters for EM algorithm.

However, in the case with the number of groups uncertain, in the second run when the number of groups has been solved in the first run, we normally use the MAP estimate to determine Z_{n+1}. This is because we might not solve the number of groups precisely, although the result is closer to the correct values than the initial estimate of the number of groups. Some small redundant groups might exist at the end of the first run, and using the MAP estimate in this case appear to prevent some of these groups from growing into unrealistic groups.

8.5 Application Procedures

The spatial EM algorithm starts from an initial grouping and iteratively adjusts the model parameters in each iteration until the grouping stops changing and the proportions stop changing. Here, we discuss the steps of the spatial EM algorithm when the group quality coefficients are used.

If the number of groups is known, we fix the group quality coefficients equal to 1, and use stochastic grouping discussed later in the maximization step. On the other hand, if the number of groups is not known, we use the basic algorithm twice: once with F_j to decrease the number of groups, and then with a fixed number of groups and all $F_j = 1$. The first run with F_j generate groups stochastically. The second run uses the final grouping and model parameters obtained at the end of the first run, and generate groups with maximum a posteriori (MAP grouping).

8.5.1 Initialization

The initialization step is used to construct the first membership matrix H_1. To do so, we simply divide the data into a fairly large number of groups (say 50 to 100). The initial grouping can be done either by value, by spatial location or randomly. This grouping results in an evaluation of the initial group indicator vector denoted by Z_1.

All the other parameters are also constructed from the initial guess

$$Z_1 = \{(z_1)_1, (z_2)_1, \cdots, (z_N)_1\},$$

where if $(z_i)_1 = j$, then $\hat{d}_i$ is assigned to the group j. For $j = 1, 2, \cdots M$, we let $(N_j)_1$ denote the number of measurements in the jth group according to Z_1. In particular, $(\Pi_j)_1$ and $(\Theta_j)_1$ are estimated as

$$\begin{aligned}
(\pi_j)_1 &= \frac{(N_j)_1}{N}, \\
(\mu_j)_1 &= \frac{1}{(N_j)_1} \sum_{(z_k)_1 = j} \hat{d}_k, \\
(C_j)_1 &= \frac{1}{(N_j)_1} \sum_{(z_k)_1 = j} (\hat{d}_k - (\mu_j)_1)(\hat{d}_k - (\mu_j)_1)^T.
\end{aligned} \tag{8.32}$$

With these initial guesses of model parameters Π_1, Θ_1 and Z_1, and setting $(F_j)_1 = 1$ for $j = 1, 2, \cdots, M$ in the first iteration, we evaluate the membership matrix H_1 using Eq. 8.23.

8.5.2 Membership Matrix Update (E-step)

With the above definitions, we can start the expectation step for the first iteration by calculating the initial membership matrix H_1.

In the expectation step, the probabilities of each datum belonging to each group are evaluated based on the current model, which is the membership matrix calculated from Eq. 8.23. Here, $H_n = \{(h_i^j)_n\}$ is evaluated at:

$$\{(\pi_j)_n, (\mu_j)_n, (C_j)_n, Z_n, j = 1, 2, \cdots, M\},$$

which are updated in the maximization step of the previous iteration. The terms in Eq. 8.23 can be computed as following

1. $G(\hat{d}_i|(\Theta_j)_n)$ is evaluated at $\{(\mu_j)_n, (C_j)_n, j = 1, 2, \cdots, M\}$ using Eq. 8.24, i.e.,

$$G(\hat{d}_i|(\Theta_j)_n) = \frac{1}{(2\pi)^{\frac{n}{2}}|(C_j)_n|^{\frac{1}{2}}} \exp\big[-\frac{(\hat{d}_i - (\mu_j)_n)^T (C_j)_n^{-1} (\hat{d}_i - (\mu_j)_n)}{2}\big]. \quad (8.33)$$

2. $S_j(x_i|r_0, Z_n)$ is evaluated at Z_n using Eq. 8.25, i.e.,

$$S_j(x_i|r_0, Z_n) = \frac{1}{(N_j)_n} \sum_{(z_k)_n = j} \left\{ \frac{1}{2\pi r_0^2} \exp\big[-\frac{\|x_i - x_k\|^2}{2r_0^2}\big] \right\}. \quad (8.34)$$

3. In the first iteration, we always set $(F_j)_1 = 1$. From the second iteration on, if the number of groups is unknown, $(F_j)_n$ is evaluated as Eq. 8.28 when we are trying to reduce the number of groups in the first run, i.e.,

$$\begin{aligned} (F_j)_n = F_j(Z_n) &= \frac{a}{(N_j)_n} \sum_{(z_k)_n = j} (E_k^j)_n \\ (E_i^j)_n = E_i^j(Z_n) &= \frac{(N_j)_n S_j(x_i|r_0, Z_n)}{\sum_{j'=1}^{M} (N_{j'})_n S_{j'}(x_i|r_0, Z_n)} \\ a &= \left(\sum_{j=1}^{M} \big[(\pi_j)_n \frac{1}{(N_j)_n} \sum_{(z_k)_n = j} (E_k^j)_n\big] \right)^{-1}. \end{aligned} \quad (8.35)$$

   If the number of groups is fixed, we set all the F_j's equal to 1.

Using these terms, the membership matrix of the current iteration can be computed as

$$h_i^j(\Pi_n, \Theta_n, Z_n) = \frac{(\pi_j)_n (F_j)_n G(\hat{d}_i|(\Theta_j)_n) S_j(x_i|r_0, Z_n)}{\sum_{j'=1}^{M} (\pi_{j'})_n (F_{j'})_n G(\hat{d}_i|(\Theta_{j'})_n) S_{j'}(x_i|r_0, Z_n)}. \quad (8.36)$$

The log-likelihood functions for the current iteration is evaluated as

$$L_{c,f}(\Pi_n, \Theta_n, Z_n|\hat{D}) = \sum_{i=1}^{N} \ln \left\{ \sum_{j=1}^{M} \big[(\pi_j)_n (F_j)_n S_j(x_i|r_0, Z_n) G(\hat{d}_i|(\Theta_j)_n)\big] \right\} + \text{Constant}, \quad (8.37)$$

in which the constant value has no effect on maximizing the log-likelihood. For H_1, set $n = 1$ in the Eqs. 8.33 to 8.37.

8.5.3 Model Parameter Update (M-Step)

In the maximization step, the model parameters are updated using the membership matrix evaluated in the E-step. Here we calculate the updated parameter Θ_{n+1} and Π_{n+1} using Eq. 8.12, i.e.,

$$\begin{aligned}(\mu_j)_{n+1} =& \frac{\sum_{i=1}^N (h_i^j)_n \hat{d}_i}{\sum_{i=1}^N (h_i^j)_n}, \\ (C_j)_{n+1} =& \frac{\sum_{i=1}^N (h_i^j)_n (\hat{d}_i - (\mu_j)_{n+1})(\hat{d}_i - (\mu_j)_{n+1})^T}{\sum_{i=1}^N (h_i^j)_n}, \\ (\pi_j)_{n+1} =& \frac{1}{N}\sum_{i=1}^N (h_i^j)_n.\end{aligned} \tag{8.38}$$

The membership matrix Z_{n+1} may be generated from H_n by stochastic grouping or MAP grouping (used only in the second EM process).

At this point, all the parameters that are necessary for calculating the new membership matrix H_{n+1} are evaluated. Then we go to the E-step of iteration $n+1$.

In the case where we are trying to determine the number of groups, after the model parameter update, we delete some groups with very small group probabilities, if no data tends to fall into them. To do this, the groups are sorted by their proportions, so that $\pi_1 \geq \pi_2 \geq \cdots \geq \pi_M$. We delete groups $j' = j_0, j_0+1, \cdots, M$, if group j_0 satisfies the following two criteria:

1. For $j' = j_0, j_0+1, \cdots, M$, $(z_i)_{n+1} \neq j'$ for any i;

2. $(\pi_{j_0})_{n+1} \leq \epsilon_g$, where ϵ_g is a small value set by the user (i.e. 0.0001).

Under these conditions, groups j_0 to M have a negligible effect on the Q function and are not expected to increase in size. Thus, we delete these groups and set $M = j_0 - 1$. For the first criteria, it means no data belongs to the group $j_0, j_0+1, \cdots$. We also need the second criteria because even if no data is evaluated (stochastically or MAP) to fall in a specific group there will still be chances that some data fall in this group in followed iterations if the proportion of this group is not too small.

8.5.4 Stopping Criteria

We use the following two conditions for the criteria to terminate the iterative process:

1. No group is deleted in the current iteration;

2. $\frac{\|(\pi_j)_{n+1}-(\pi_j)_n\|}{\|(\pi_j)_n\|} < \epsilon_1$ for $j = 1, 2, \cdots, M$; normally, we use $\epsilon_1 = 10^{-4}$.

8.6 Case Study

Synthetic 4-D seismic and field 4-D seismic data are presented to check the ability of the spatial EM algorithm to solve the measurement error estimation problem. For both cases, the measurement errors are spatially correlated. So we need to estimate the covariance of the measurement error. Since the measurements are located in uniform 2-D grids, the covariance matrix will be estimated in the form of matrix:

$$C(dx) = \frac{1}{N(dx)} \sum_{(x_i - x_j)=dx} e_i e_j. \tag{8.39}$$

where $dx = \{dx_1, dx_2\}$ is spatial displacement vector between measurements in the X and Y directions, which can take discrete values as $\{dx_1 = i\delta x_1, dx_2 = j\delta x_2,\ i = 0, \pm 1, \pm 2, \cdots, \pm N_{cov}\ j = 0, \pm 1, \pm 2, \cdots \pm N_{cov}\}$ with δx_1 and δx_2 as the size of grid-blocks in the two directions. x_i and x_j are the spatial coordinates of the ith and jth grid-block respectively. e_i and e_j are the corresponding measurement errors. The summation is made on the pairs of measurements with distance of dx in the two directions. $N(dx)$ is the number of such pairs. Hence, the covariance matrix will have $2 \times N_{cov} + 1$ rows and columns, with the entry of $dx = \{0, 0\}$ as the variance. In this study we always set $N_{cov} = 10$.

In the synthetic case, we know that an approximate number of groups is 4. However, as we wish to test our algorithm, we assume the number of groups is unknown. In addition the sensitivity of the algorithm to different initializations will also be studied. We use two types of initializations, random initialization and value initialization.

1. In random initialization, we assign a group indicator to each measurement randomly by sampling from a uniform distribution with the requirement that each group should be expected to have the same number of data.
2. In value initialization, we sort the measurements based on the measured value of data, and then the measurements are evenly divided into a specified number of groups.

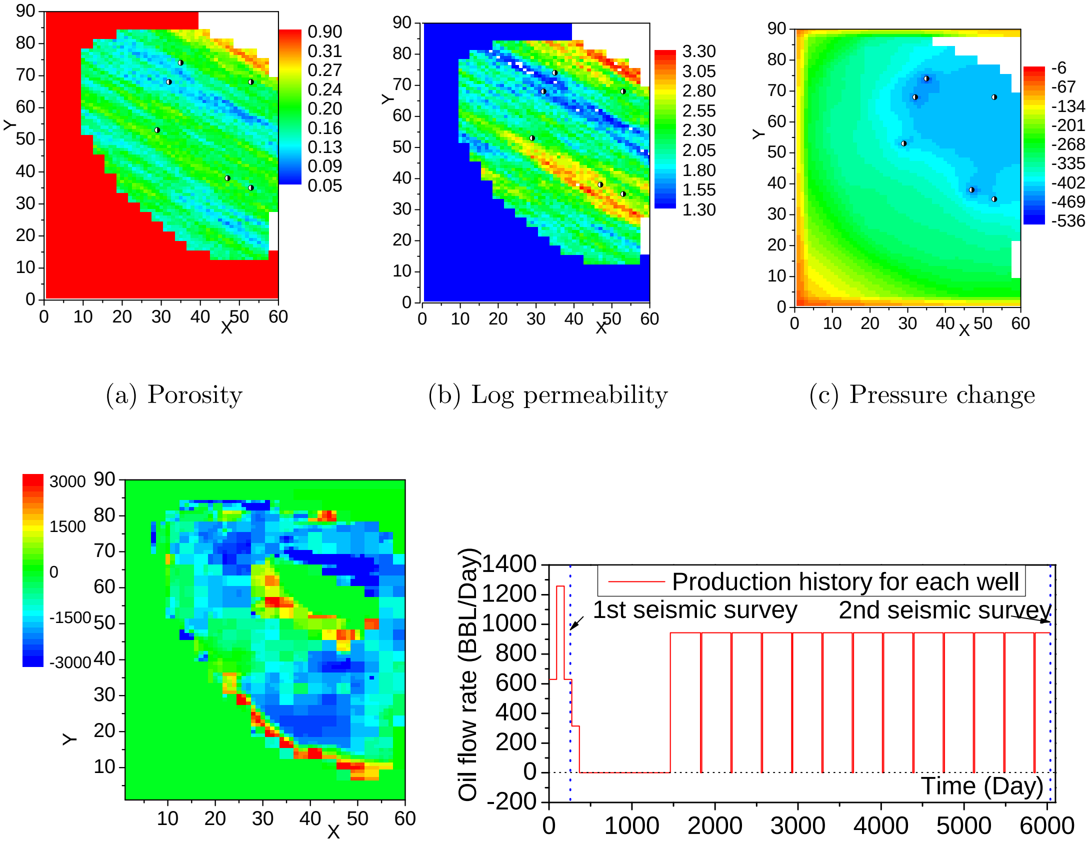
Therefore, we will have four cases for the synthetic data set, known or unknown number of groups, random or value initialization.

In the field data case, the number of groups is not known. Then we will have two variations: random or value initializations.

The estimated measurement error will also compared with those from a moving average as used by Aanonsen et al. [1]. In the implementation, we use a constant weight for all the data inside the window. Because our algorithm uses a moving quadratic fit in each individual group, a direct moving quadratic fit is also compared.

8.6.1 Synthetic Case

The synthetic acoustic impedance change data are generated based on the PUNQS3 reservoir model used in the previous chapters except that the reservoir is expanded to a $20 \times 30 \times 5$ with a numerical aquifer attached to the reservoir; the aquifer is simulated by inputting high porosities (0.95) and high water saturation (1.0) in the aquifer grid blocks. In this chapter we call this model PUNQ. In this case, because seismic data always has a much finer grid than that for simulation, the model is down-scaled to $60 \times 90 \times 5$ grid by assigning the same values of depth in



(a) Porosity

(b) Log permeability

(c) Pressure change

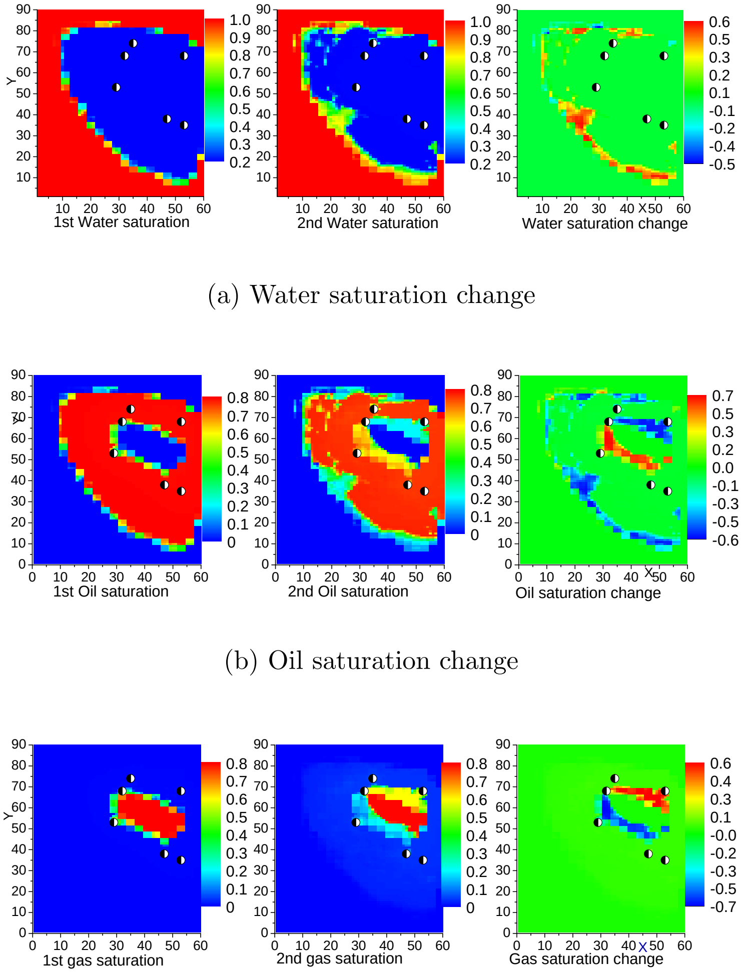
(d) Acoustic impedance change

(e) Production history

Figure 8.1: Synthetic PUNQ 3-D, 3-phase case simulations.

the 9 fine grids that fall into the same coarse grid. Figs. 8.1(a) and 8.1(b) shows the porosity and permeability assigned to each grid block of the second layer. We simulated this reservoir with an in-house reservoir simulator (CLASS) developed by Chevron. There are 6 production wells completed in the oil column around the gas cap. The locations are shown as black circles in Figs. 8.1(a) and 8.1(b). The production constraint is the same for all wells. For each well, the flow rates are specified as in Fig. 8.1(e), and the minimum bottom hole pressure is 1764 psi.

Since the acoustic impedance is implicitly a function of pressure and phase saturations, the production activities in the reservoir which results in saturation changes or pressure changes will cause a change in the acoustic impedance. The



(a) Water saturation change

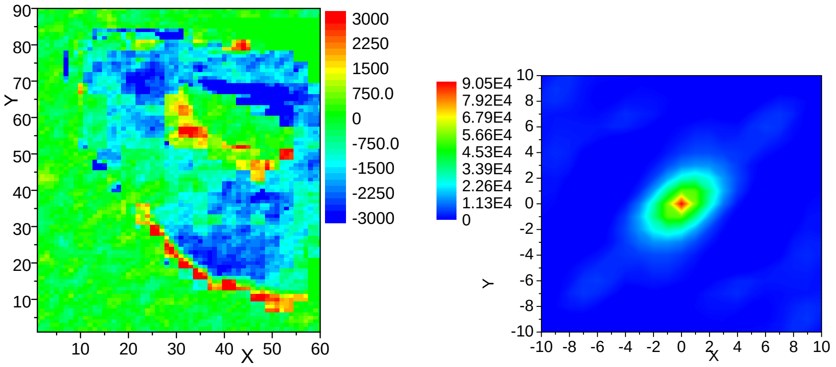
(b) Oil saturation change

(c) Gas saturation change

Figure 8.2: Synthetic PUNQ 3-D, saturation changes.

acoustic impedance change is usually derived from a 4-D seismic survey, in which two seismic surveys of the same field are done at two different times. In general, geophysicists wish to infer the change in fluid distribution from this survey. For the synthetic example, instead of doing two seismic surveys, we use a reservoir simulator (CLASS) as the forward model to calculate the fluid and pressure distributions as a function of time. With the calculated pressures and saturations, we use the rock physics models discussed in Chapter 2 to determine the acoustic impedance change.

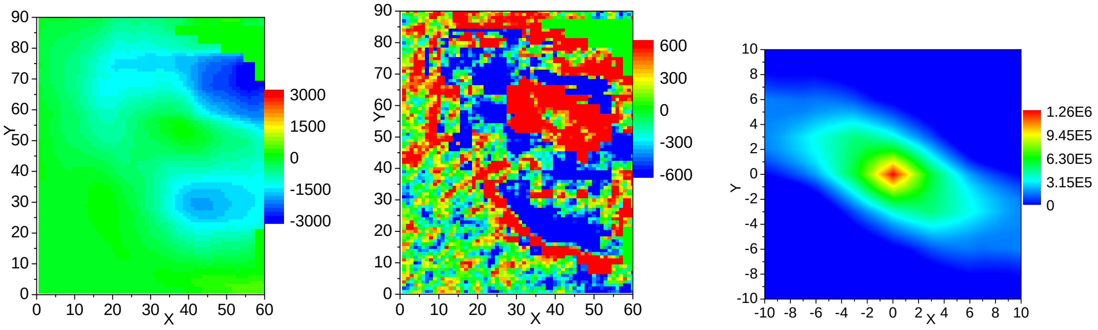
In this synthetic example, the initial seismic survey is taken 273 days after the



(a) Data with correlated measurement error

(b) True covariance matrix

Figure 8.3: Synthetic PUNQ 3-D, noisy data and covariance of measurement error.



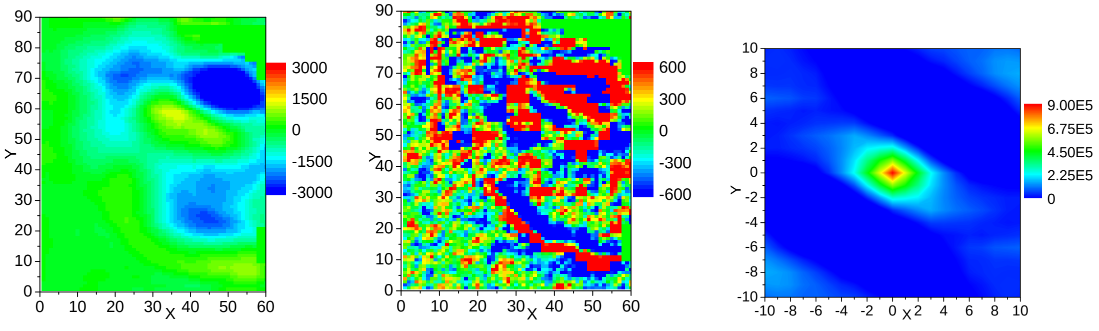
(a) Estimated mean

(b) Estimated measurement error

(c) Estimated covariance

Figure 8.4: PUNQ with correlated measurement error; Moving average with window length of 21 × 21, without EM algorithm.

reservoir was put in production and the 2nd seismic survey is taken at the 6032nd day. Although we simulate the PUNQ reservoir with 5 layers, for the purpose of error estimation here, we only present results for the 2nd layer. Figs. 8.2(a) through 8.2(c), respectively, show the water, oil and gas saturation distributions at the first survey time (the 273rd day), at the second survey time (the 6032nd day) as well as the change of the saturations during the time interval. In the left and middle panels in Fig. 8.2(c), the blue means zero gas saturation and the red gives the highest gas saturation of 0.8 with an irreducible water saturation of 0.2 in the reservoir. The

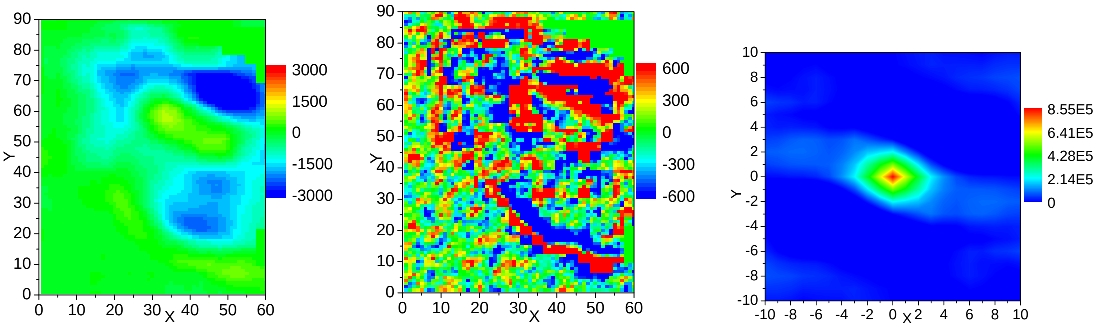


(a) Estimated mean

(b) Estimated measurement error

(c) Estimated covariance

Figure 8.5: PUNQ with correlated measurement error; Moving fitting with window length of 21 × 21.



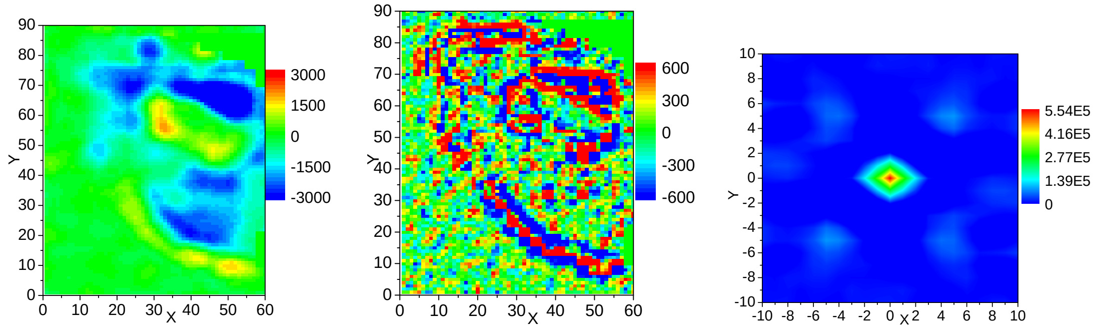
(a) Estimated mean

(b) Estimated measurement error

(c) Estimated covariance

Figure 8.6: PUNQ with correlated measurement error; Moving average with window length of 11 × 11, without EM algorithm.

right panel of Fig. 8.2(c) shows the change in the gas saturation. It is noted that the gas cap invaded the oil column as evidenced by the increasing gas saturation (red color) near the top right corner of Fig. 8.2(c). Somewhat surprisingly, the oil invaded the gas cap as indicated by the decreasing gas saturation (blue color) in Fig. 8.2(c). While the physics of why the oil moved up to the gas cap is not completely clear, it is apparently due to the strength of the aquifer. Simulation runs with a different reservoir simulator (Eclipse) shows the same phenomenon. The right plot of Fig. 8.2(a) shows the water saturation change between the two surveys. From



(a) Estimated mean

(b) Estimated measurement error

(c) Estimated covariance

Figure 8.7: PUNQ with correlated measurement error; Moving fitting with window length of 11 × 11.

the graph, the major water saturation change occurred around the original water-oil contact (WOC) due to the water invasion into the oil zone. Fig. 8.2(b) shows the oil saturation change, which is similar to the combination of Figs. 8.2(a) and 8.2(c), due to the saturation constraint, $S_w + S_g + S_o = 1$. The change of the reservoir pressure is displayed in Fig. 8.1(c). Compared to the change in saturation, the pressure change is not localized to certain regions, instead it occurred over the whole range of the reservoir around the edge of original gas cap. The maximum change occurred in the gridblocks containing the production wells shown as the dark blue spots in the oil zone. Note that the pure white regions along the east side and at the top right corner represent inactive reservoir cells.

Fig. 8.1(d) shows the acoustic impedance change obtained based on the change in the saturation and pressure distributions. The procedure for the acoustic impedance change calculation uses the following steps: 1) Run a reservoir simulation; 2) Calculate the acoustic impedance (Z_1) at time t_1 by applying a rock physics model with the saturation and pressure distribution as well as the rock and fluid properties of the reservoir at that specific time. 3) Calculate the acoustic impedance (Z_2) at time t_2. 4) Subtract (Z_1) from (Z_2) to obtain the acoustic impedance change over the time interval ($\Delta t = t_2 - t_1$). From Fig. 8.1(d), the acoustic impedance change resembles the combination of the change in pressure and saturations. The increasing water saturation along the original WOC shows high positive signal in red (The acoustic impedance change is about 3000 lb/ft^2/s). Along the original GOC, the region where the oil invaded the gas cap also shows a high positive impedance change, whereas the region where the gas cap expanded into the oil zone shows high negative impedance change with dark blue color in Fig. 8.1(d). In the oil zone, the change is due to the pressure decrease, which is represented by another blue color region right above the water influx region (red) in Fig. 8.1(d). It can be seen that the saturation change has a larger influence than the pressure change on the acoustic impedance change. Note that the acoustic impedance does not change much around (X=25, Y=40). This is because a gas saturation increase is combined with a water saturation increase, and these two types of changes have the opposite effect on the acoustic impedance change, and tend to cancel each other out. The original aquifer and the part of the gas cap where there is no saturation change and with fairly small pressure changes shows little change in acoustic impedance.

After the true synthetic acoustic impedance change signal is generated, correlated measurement errors are added to the true signal to test the applicability of our method for estimating measurement error parameters. In the following discussion, we use the terms noise and measurement error interchangeably. Fig. 8.3(a) gives the seismic data with correlated noise added to the “true” signal. The standard deviation of the noise added is 300 lb/ft^2/s. Fig. 8.3(b) shows the covariance matrix estimated from the correlated measurement error. The coordinate in the X and Y axes represents the correlation distances in the two directions. The correlation length for this correlated noise is about 9 gridblocks in the major principle direction (about 40 degrees), and 5 gridblocks in the minor principle direction. Comparing before and after adding noise (Figs. 8.1(d) and 8.3(a)), the image of the noisy seismic data gets rougher.

The purpose of this case is to check the ability of the EM algorithm to recognize the boundaries of the regions of different saturation changes, and estimate the measurement error by smoothing the data within these regions.

For each of the cases, there will be seven (or eight) plots, the initial and final grouping, the log-likelihood function, the estimated Gaussian of each group, the estimated mean, the estimated measurement error and the estimated covariance of measurement error. For the cases where number of groups is unknown, there will be two final groupings, (1) the final grouping after the first run (stochastic grouping) with F_j (2) the final grouping after the second run (MAP grouping) without F_j. Consequently, there will also be two log-likelihood function curves: the black one is for the first run (decreasing number of groups using F_j), and the red one is for the second run (with F_j fixed equal to 1).

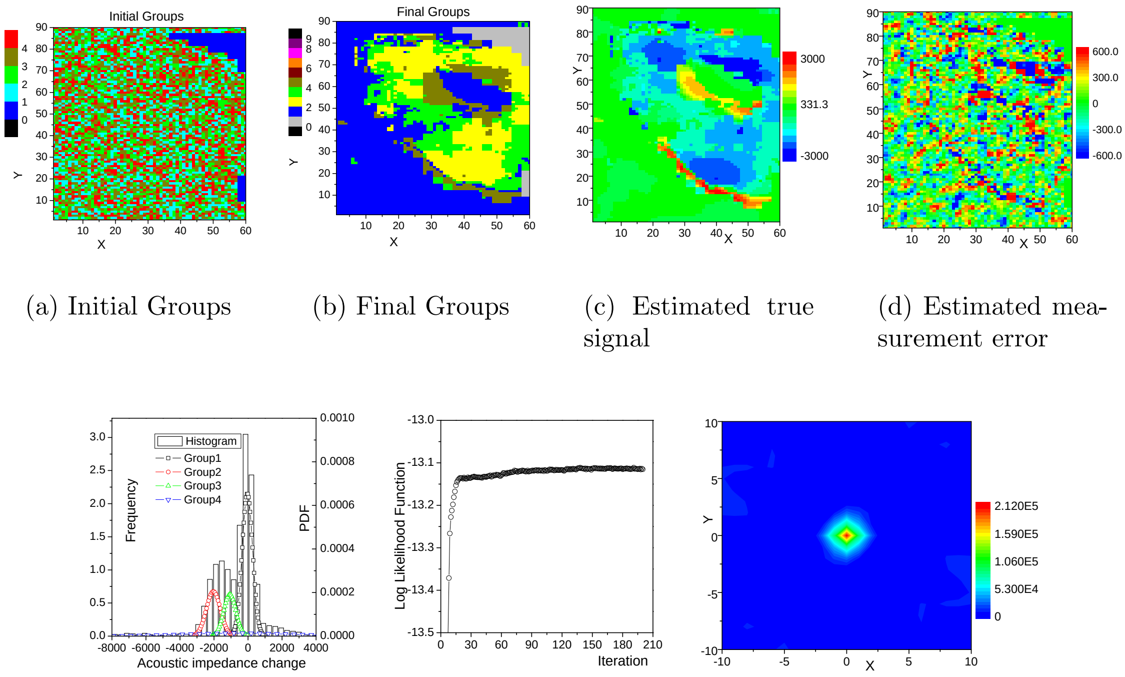
The estimated measurement error from the spatial EM algorithm is compared with those from the direct moving average and direct moving fitting without using EM groups. Fig. 8.4 shows the result with a constant window size of 21×21. Although the estimated true signal (Fig. 8.4(a)) shows some trends of the true signal, it is far smoother. The variance of the estimated measurement error (1.25×10^6 lb^2/ft^4/s^2) is much larger than the true (0.91×10^5 lb^2/ft^4/s^2). The covariance map also indicates a long correlation length in the direction with an angle of about 170 degrees, which is totally wrong compared with the true covariance shown in Fig. 8.3(b). Note that the map of the estimated measurement error (Fig. 8.4(b)) shows a structure which has features similar to the true signal. The red band indicates the water invasion into the oil zone. The oil invasion into the gas cap and the gas cap expansion into the oil can be seen around the original gas cap in continuous red or blue colored regions. The correlation of the estimated error reflects the correlation of the true signals. The constant window average over-smoothed the data. The main reason for the over-smoothing is largely due to changes in saturation with respect to the movement of fluid contacts. At the boundaries, there is a sharp change in the signal, the moving-window average tends to smooth out this sharp edge, which in turn incorporates some of the true signal into the estimated measurement error.

Fig. 8.5 shows the result of a moving quadratic fit with a constant window size of 21×21. Even though the result is closer to the true than obtained with the moving average, the estimated correlation of measurement error is also wrong as shown in Fig. 8.5(c), and the estimated variance is 9.0×10^5 lb^2/ft^4/s^2 versus the true value of 0.91×10^5 lb^2/ft^4/s^2. We also tried to repeat moving average and moving fitting using a smaller window (11×11, close to the correlation length of measurement error), and the results are shown in Figs. 8.6 and 8.7. The small window still resulted in too high variance and incorrect correlation directions. This cross boundary fitting problem can be effectively solved using the EM groups.

Cases with Fixed Number of Groups.

For the synthetic data, 4 seems to be a reasonable value as the number of groups based on the change in acoustic impedance in Fig. 8.3(a). We started from two types of initial groups (value and random initialization). Figs. 8.8 and 8.9 shows the results from these two cases. Value initialization case successfully recognized the water region, oil region and influx regions. However, random initialization (Fig. 8.8) grouped the water, oil and gas influx regions into one single group to give a higher estimated value of the variance than was obtained from value initialization.

Figs. 8.8 and 8.9 both show estimated true signal, measurement error and covariance matrix. Recall the the variation of the true measurement error is 9.1×10^4 lb^2/ft^4/s^2, and the range of the correlation is nine gridblocks in the major direction, and five gridblocks in the minor direction. For both of these cases, the variances are slightly overestimated (2.12×10^5 lb^2/ft^4/s^2 for random initialization and 1.91×10^5 lb^2/ft^4/s^2 for value initialization versus the true value of 0.91×10^5 lb^2/ft^4/s^2) and the



(a) Initial Groups (b) Final Groups (c) Estimated true signal (d) Estimated measurement error

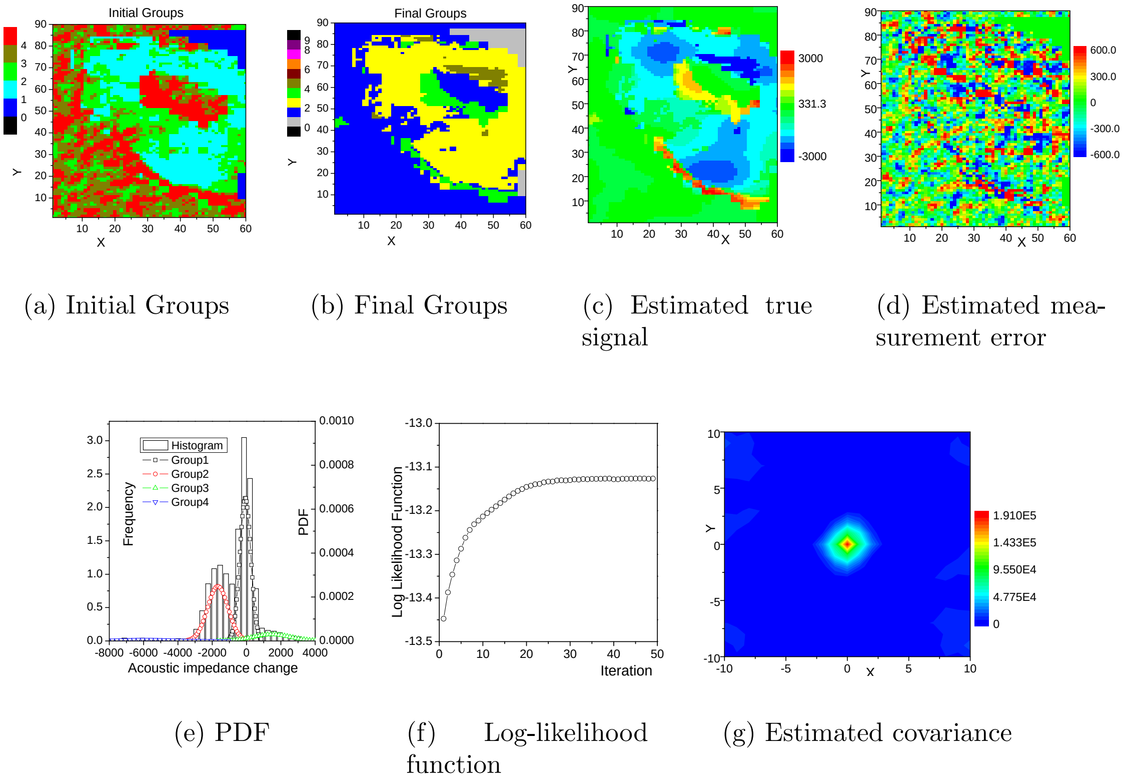
(e) PDF (f) Log-likelihood function (g) Estimated covariance

Figure 8.8: PUNQ with correlated measurement error; no F_j, $r_0 = 2.0$, random initialization (4 initial groups), stochastic grouping.

correlation are underestimated. In the value initialization case, the overestimation of variance is because the Gaussian mixture model can not accurately represent the true signal near the boundary , as is illustrated by by a small belt shaped structure near the boundary between the water region and the water influx region (Figs. 8.9(d)). However, the estimated covariance gives a reasonable characterization of the measurement error.

Cases with Uncertain Number of Groups.

Because F_j serves as the weight for each group to indicate their spatial continuity, the groups with weaker spatial continuity relative to other groups will be penalized with a smaller weight. When the continuity of the groups become better, the penalty will gradually be removed. This mechanism helps the spatial EM to automatically remove those groups with poor continuity, and automatically find a



(a) Initial Groups

(b) Final Groups

(c) Estimated true signal

(d) Estimated measurement error

(e) PDF

(f) Log-likelihood function

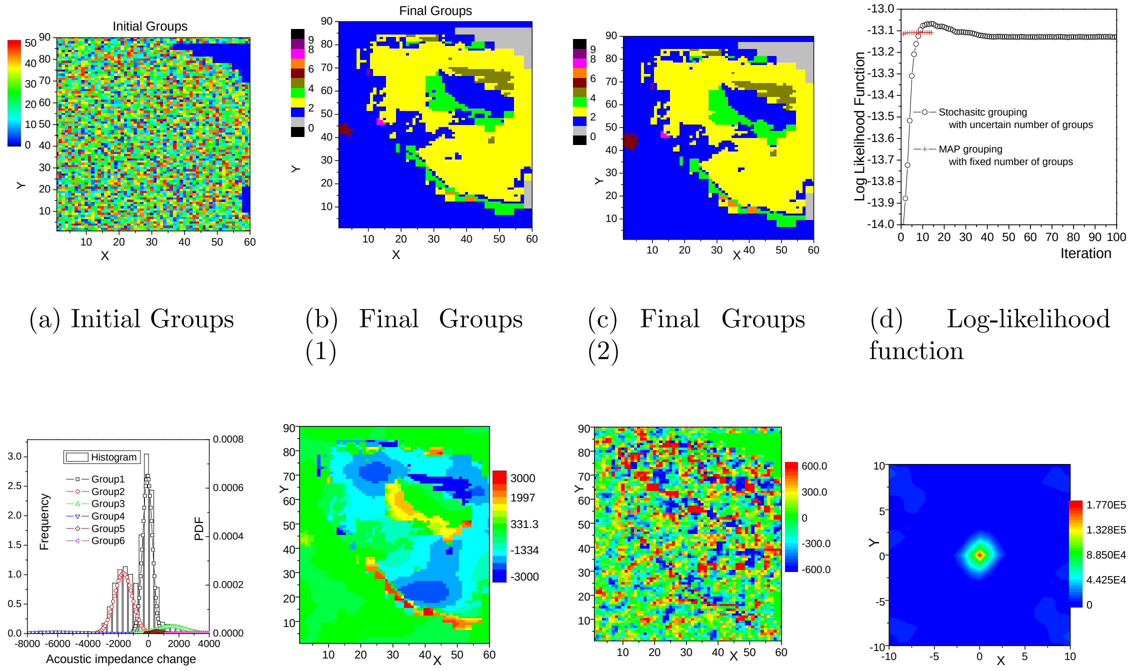
(g) Estimated covariance

Figure 8.9: PUNQ with correlated measurement error; no F_j, $r_0 = 2.0$, value initialization (4 initial groups), stochastic grouping.

reasonable number of groups.

The spatial EM algorithm is applied to the synthetic cases with two types of initialization, initialize by sorting the measured values, and initialize randomly. Here we show that the two different initial groupings give very close results. This illustrates the robustness of this algorithm. There are 50 initial groups and $r_0 = 2.0$ for each case.

Figs. 8.10 and 8.11 show the results for these two cases. Both random and value initializations resulted in 4 major groups and a few very small groups. Similar to the results with a fixed number of groups, the variances of measurement error are overestimated, and the correlation lengths are underestimated. Random initialization gave smaller estimated variances than the other cases because it has more small groups. We note also that the log-likelihood function for the first run (sto-



(a) Initial Groups

(b) Final Groups (1)

(c) Final Groups (2)

(d) Log-likelihood function

(e) PDF

(f) Estimated true signal

(g) Estimated measurement error

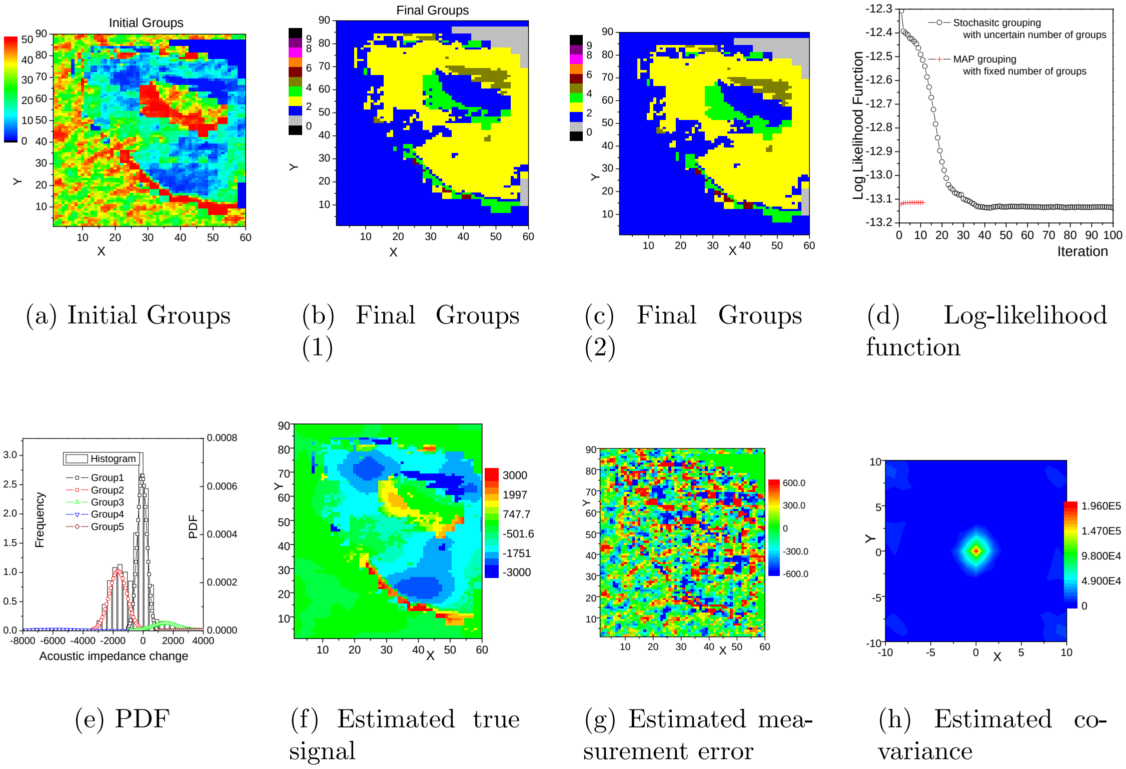
(h) Estimated covariance

Figure 8.10: PUNQ with correlated measurement error; with uncertain number of groups, $r_0 = 2.0$, random initialization (50 initial groups).

chastic grouping with F_j) is no longer non-decreasing, because the log-likelihood is constructed as if the number of groups are correctly known. When an additional term of F_j is applied, and some low quality groups are gradually killed, the value of log-likelihood may decrease because the number of groups is no longer conserved. In the second run, the log-likelihood function experienced only a small increase, and only the size of each group is changed, but only slightly. According to the estimated covariance of measurement error, Figs. 8.10(h) and 8.11(h), the variances are overestimated (1.77×10^5 lb^2/ft^4/s^2 for random initialization and 1.96×10^5 lb^2/ft^4/s^2 for value initialization) and correlation length are underestimated.

8.6.2 Field Case

As is shown in Fig. 8.12, a layer of 4-D acoustic impedance change (AIC) and Poisson's ratio change (PRC) is chosen as the data to be analyzed using the EM



(a) Initial Groups

(b) Final Groups (1)

(c) Final Groups (2)

(d) Log-likelihood function

(e) PDF

(f) Estimated true signal

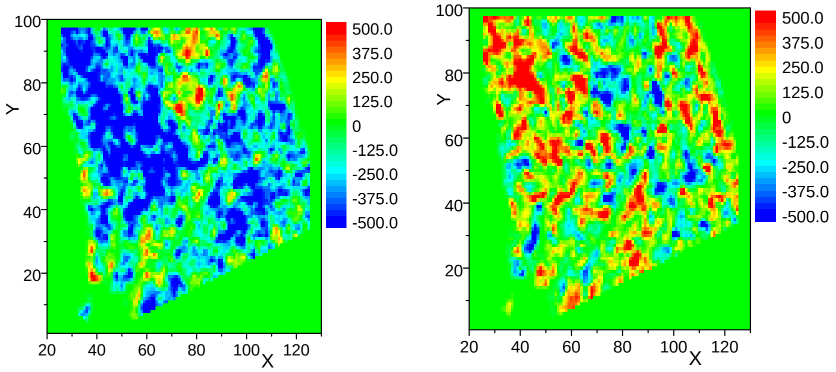
(g) Estimated measurement error

(h) Estimated covariance

Figure 8.11: PUNQ with correlated measurement error; with uncertain number of groups, $r_0 = 2.0$, value initialization (50 initial groups).

algorithms. The data set is defined on 7049 active gridblocks. Note that the data have some trends, for example, in the top-left part, the AIC is comparatively low, and PRC is comparatively high. However, the number of groups is uncertain.

The EM algorithm allows us to expand from grouping a single data type to grouping multiple data types, so that we can group the data using both AIC and PRC. The grouping results will be the same for both data. The relationship of the two data provides more valuable information for grouping, and the grouping is expected to be more accurate and with higher resolution. As is shown by Dvorkin [21], the cross plot of acoustic impedance (P-impedance) and Poisson's ratio can be an effective pore pressure and pore fluid diagnostic chart. As the pore pressure increases, the Poisson's ratio will increase for a saturated rock, and will decrease for a dry rock. The P wave velocity, and here acoustic impedance, will decrease as the



(a) Acoustic impedance change (AIC)

(b) Poisson's ratio change (PRC)

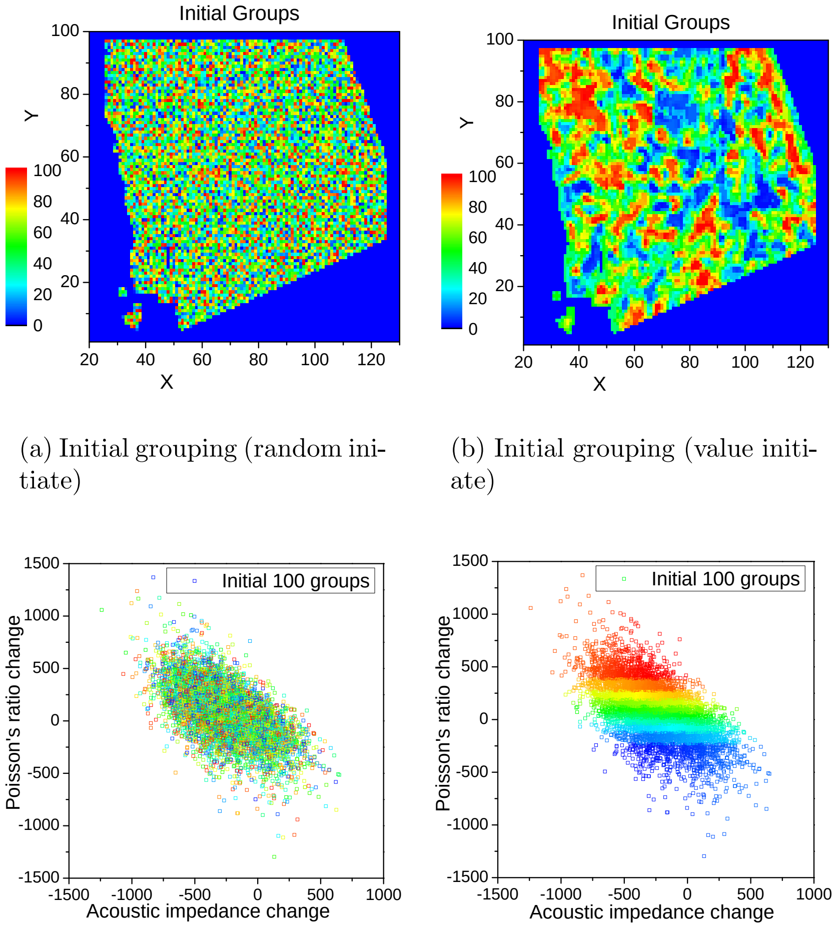
Figure 8.12: Field 4-D seismic data.

pore pressure increases.

In the case of two data types, value initialization of grouping is more complicated. In this study, we set PRC as the primary data, and AIC as the secondary data. If we want n_1 slots for the primary data and n_2 slots for the secondary data, the entire number of groups will be $n_1 \times n_2$. we can first evenly divide the data into n_1 groups according to the measured value of the primary data, and then evenly divide each group we have obtained into n_2 groups according to the measured value of the secondary data. In this case the observed data $\hat{d_i}$ is a vector with first entry equal to the PRC and second entry equal to AIC. The value initialization result will also be compared with that of random initialization. In the results, we compute the correlation coefficient for both types of data in each groups although they are not model parameters in the spatial EM algorithm.

Figs. 8.13 to 8.17 are the estimated grouping (maps and cross-plots), Gaussians, and log-likelihood from these two types of initializations. There are 8 plots for each case. The colors of the second through seventh plots are consistent.

1. Initial and final grouping (maps and cross-plots) of the first run with F_j, and final grouping (maps and cross-plots) of the second run without F_j. In Figs. 8.14,



(a) Initial grouping (random initiate)

(b) Initial grouping (value initiate)

(c) Initial cross-plot (random initiate)

(d) Initial cross-plot (value initiate)

Figure 8.13: Initial Grouping (map and cross-plot) for field 4-D seismic data; $r_0 = 2.0$, random and value initialization (100 initial groups).

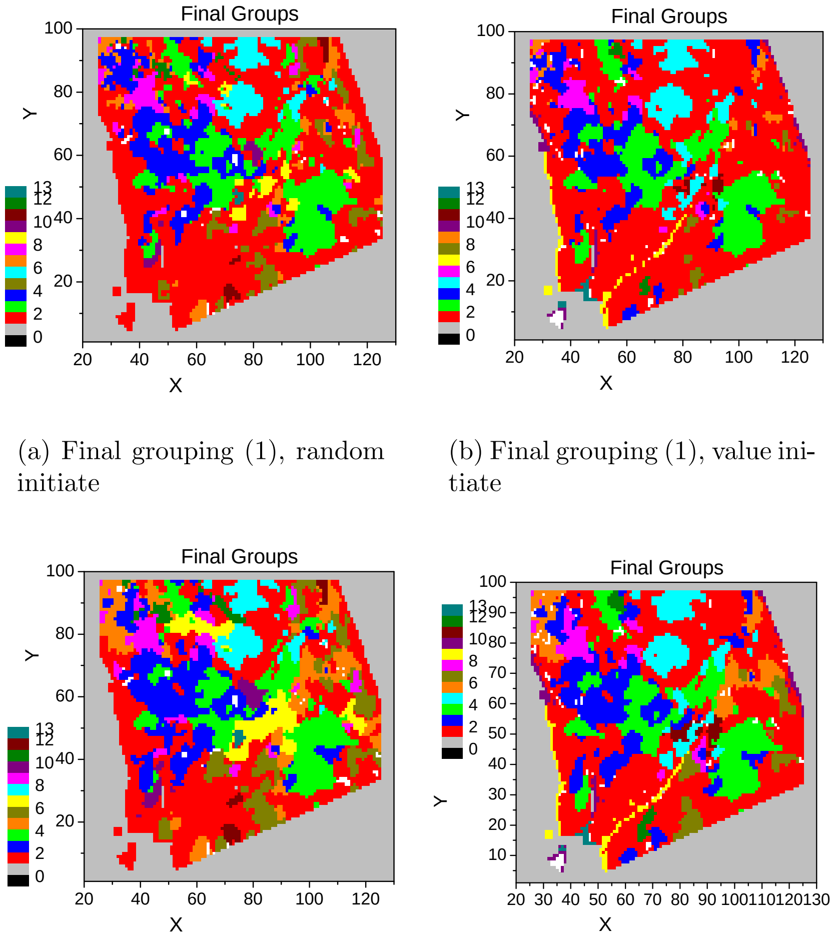
and 8.15 we use label "(1)" as the first run, and "(2)" as the second run.

2. Estimated Gaussian functions of both AIC and PRC.

3. Log-likelihood function of the first run (black) and second run (red).

The cross-plots in Fig. 8.13(c) and Fig. 8.13(d) show the initial grouping for random initialization and value initialization respectively, a different color is used for each group. Fig. 8.13(d) shows the PRC is primary data in value initialization case.

For both cases, the estimated grouping of the first 7 groups are similar, and



(a) Final grouping (1), random initiate

(b) Final grouping (1), value initiate

(c) Final grouping (2), random initiate

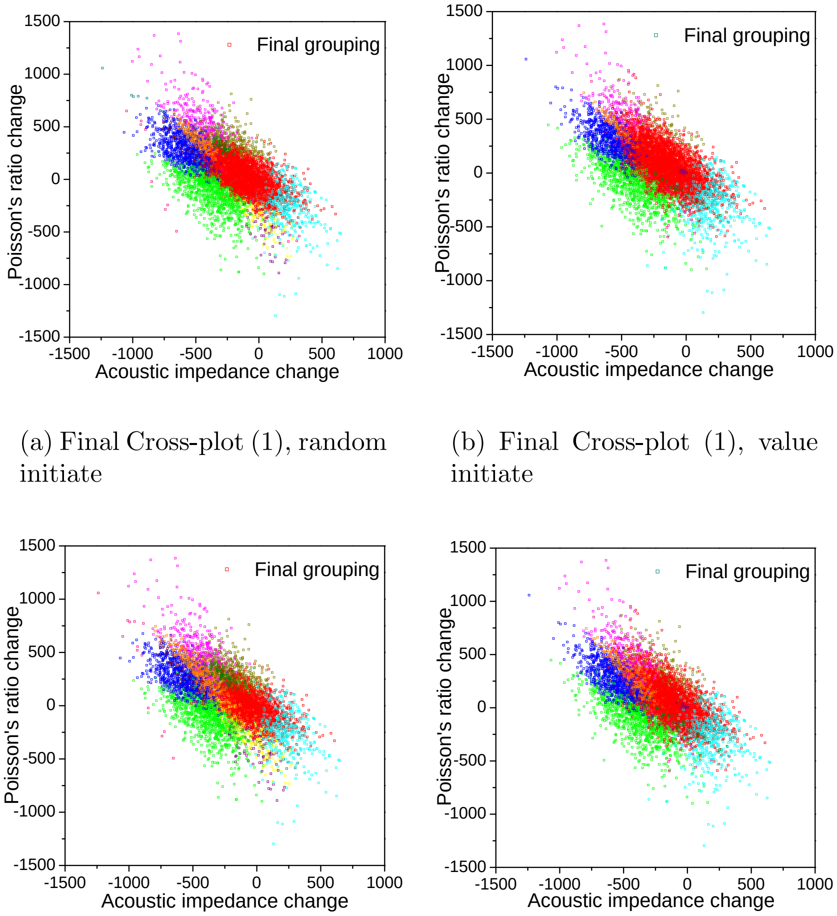
(d) Final grouping (2), value initiate

Figure 8.14: Final grouping map for field 4-D seismic data; $r_0 = 2.0$, random and value initialization (100 initial groups).

the final log-likelihood of these two cases are very close. However, there are some differences in small groups, and the sizes of each groups. This reflects the fact that there are some uncertainty in the grouping because the noise level is quite high. The following are some detailed comparison of these results.

For the two cases with different initializations, the estimated groups are very close to each other:

1. As is shown in the final groups (Figs. 8.14(c) and 8.14(d)), and the summary statistics of final groups in Tables 8.2 and 8.3, the first 5 random initiated final



(a) Final Cross-plot (1), random initiate

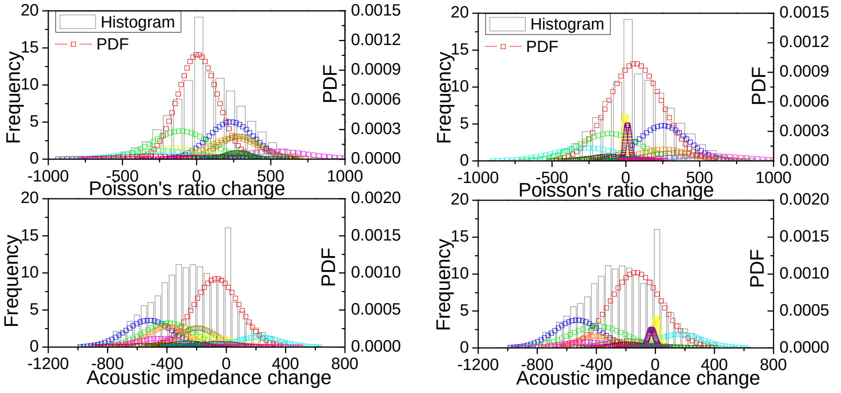
(b) Final Cross-plot (1), value initiate

(c) Final Cross-plot (2), random initiate

(d) Final Cross-plot (2), value initiate

Figure 8.15: Final grouping cross-plot for field 4-D seismic data; $r_0 = 2.0$, random and value initialization (100 initial groups).

groups are close to the first 5 value initiated final groups. The correlation coefficients computed from each group are all negative. Except for the 6th group in the value initialization case, the negative correlations are strong (around -0.7). From the final grouping maps, we can see the final resulted groups are quite continuous. The grouping quality coefficients F_j of the largest 7 final groups (at the end of the first run with F_j) of these two cases are quite high. Most of them are above 0.8, and the rest of them are around 0.5. Therefore, the estimated groups are acceptable.



(a) Random initiate

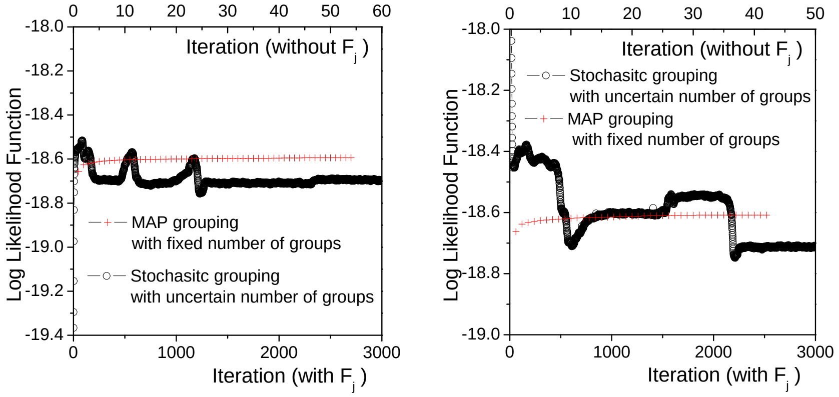
(b) Value initiate

Figure 8.16: Estimated Gaussian of field 4-D seismic data; $r_0 = 2.0$, random initialization (100 initial groups).

2. The information from AIC and PRC compliment with each other and result in a fairly fine description of the final grouping. i.e. from Fig. 8.16(a), the blue and dark yellow groups occupy almost same range of PRC values, but are separated in AIC values. Similarly, the green and orange groups occupy the same range of PRC values, but are separated in AIC values.

3. The cross plot of AIC and PRC are shown in Fig. 8.15(c) and 8.15(d). Different colors are used to indicate the different initial and final groups, and the histograms are computed from the values of the measurements in each groups. These final groups are concentrated and overlap with each other in measured values, and are spatially continuous. This is exactly what we want to do, gather the data with close spatial coordinates and measured values into the same groups.

Figs. 8.18, 8.19, 8.20 and 8.21 show the estimated true values, measurement errors and covariance matrix for both cases. The estimated true signal from these two cases has higher resolution, because ideally the boundaries between different regions are preserved by smoothing within each group. This is also the reason why



(a) Random initiate

(b) Value initiate

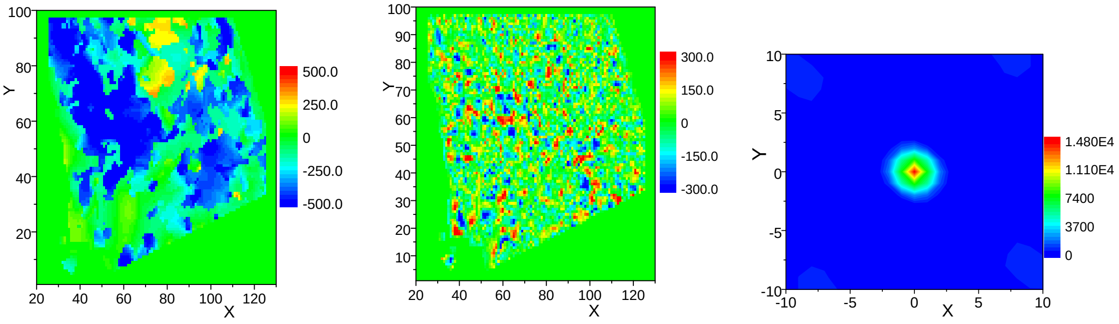
Figure 8.17: Log-likelihood of field 4-D seismic data; $r_0 = 2.0$, value initialization (100 initial groups).

the estimated measurement error should be much more dependable than those from the direct moving average and moving fitting. A summary of estimated covariance for AIC (acoustic impedance change) and PRC (Poisson's ratio change) are listed in Table 8.1, in which the first two rows are from the final grouping of spatial EM algorithm, and the last two rows are for the results from direct moving average and moving fitting. Quantitatively, the results from the spatial EM algorithm gives much smaller variance and smaller correlation range. This is consistent with the synthetic case.

Initiate	Var. (AIC)	Range (AIC)	Var. (PRC)	Range (PRC)
Value	1.73×10^4	3×3	2.22×10^4	3×3
Random	1.48×10^4	3×3	1.79×10^4	3×3
Direct Average	4.08×10^4	5×5	5.04×10^4	5×5
Direct Fitting	3.18×10^4	4×4	4.14×10^4	4×4

Table 8.1: Field 4-D seismic case summary

Compared to the results from the moving average and moving fitting without EM groups as shown in Fig. 8.22 to Fig. 8.25 and Table 8.1, our new results seem superior: 1) The moving average and moving fitting smoothes out the boundaries

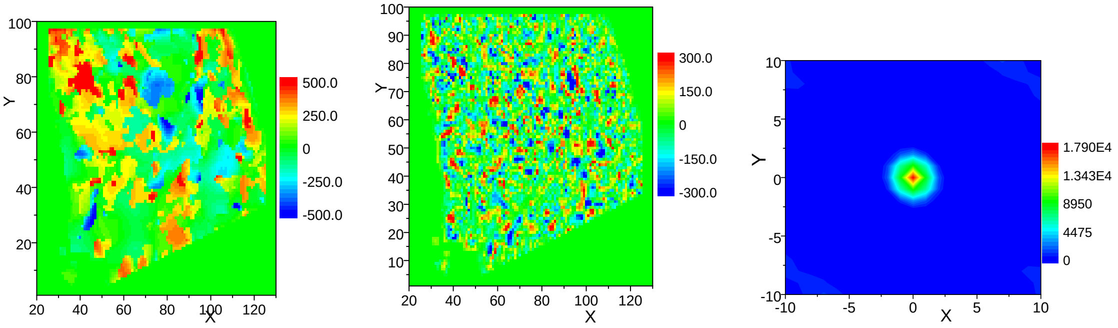


(a) Estimated true signal

(b) Estimated measurement error

(c) Estimated covariance

Figure 8.18: Field acoustic impedance change: moving quadratic fitting (21 × 21) from random initiated EM groups.



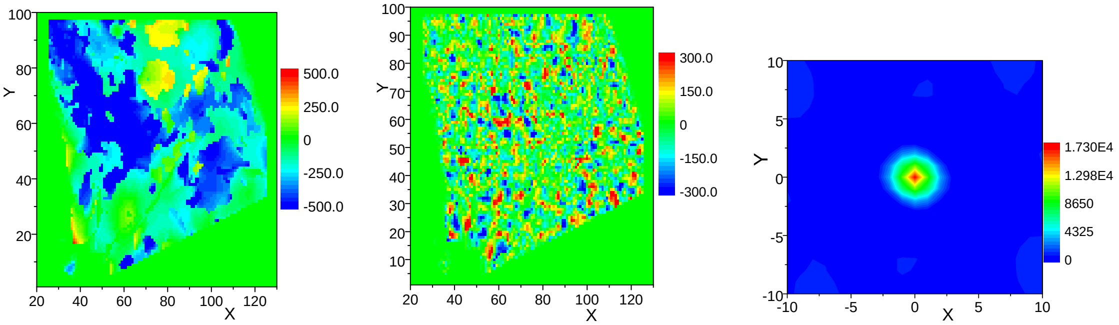
(a) Estimated true signal

(b) Estimated measurement error

(c) Estimated covariance

Figure 8.19: Field Poisson's ratio change: moving quadratic fitting (21 × 21) from random initiated EM groups.

and gives a poor resolution of the true signal; 2) The estimated variance of the measurement error in AIC and PRC are 1.48×10^4 and 1.79×10^4 respectively for the random initialization case, and 1.73×10^4 and 2.22×10^4 respectively for the value initialization case. The moving average and moving fitting gives a higher estimation of measurement error because near boundaries, the smoothing window include values from both side of the boundary even though there may be sharp changes across boundaries that represent flooding fronts. To sum up, we believe that the grouping obtained from the combination of AIC and PRC is reliable and gives a more reliable

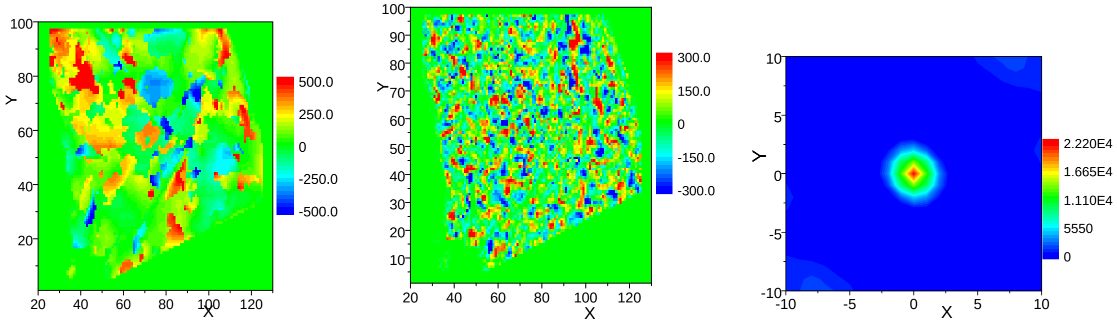


(a) Estimated true signal

(b) Estimated measurement error

(c) Estimated covariance

Figure 8.20: Field acoustic impedance change: moving quadratic fitting (21 × 21) from value initiated EM groups.



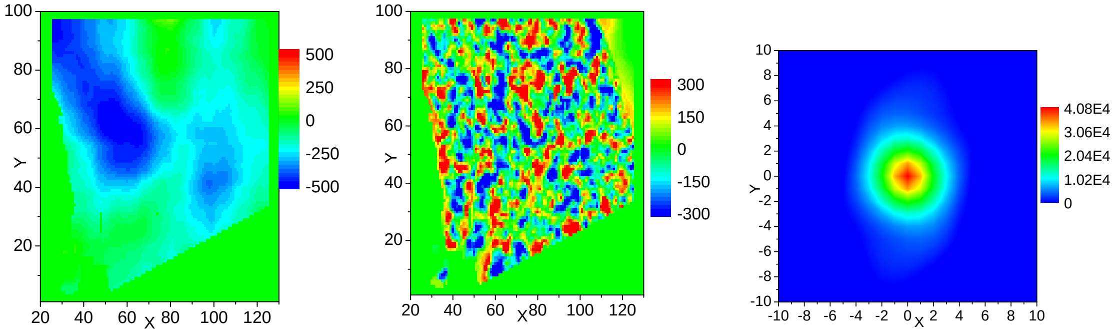
(a) Estimated true signal

(b) Estimated measurement error

(c) Estimated covariance

Figure 8.21: Field Poisson's ratio change: moving quadratic fitting (21 × 21) from value initiated EM groups.

characterization of the data and measurement error.

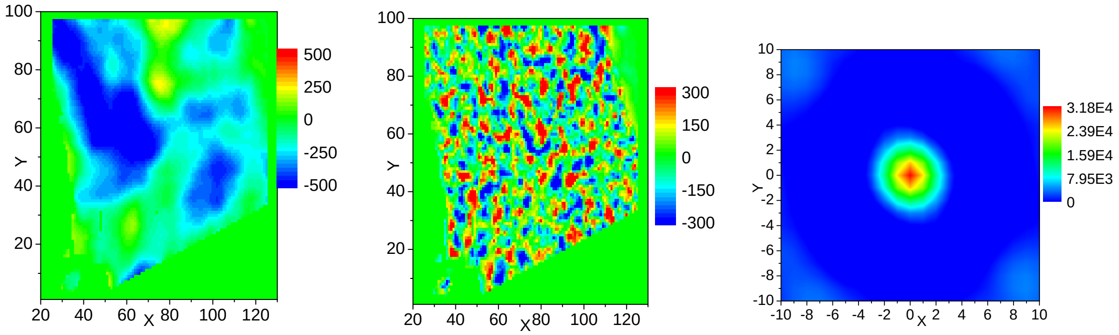


(a) Estimated true signal

(b) Estimated measurement error

(c) Estimated covariance

Figure 8.22: Field acoustic impedance change: direct moving average (21 × 21).

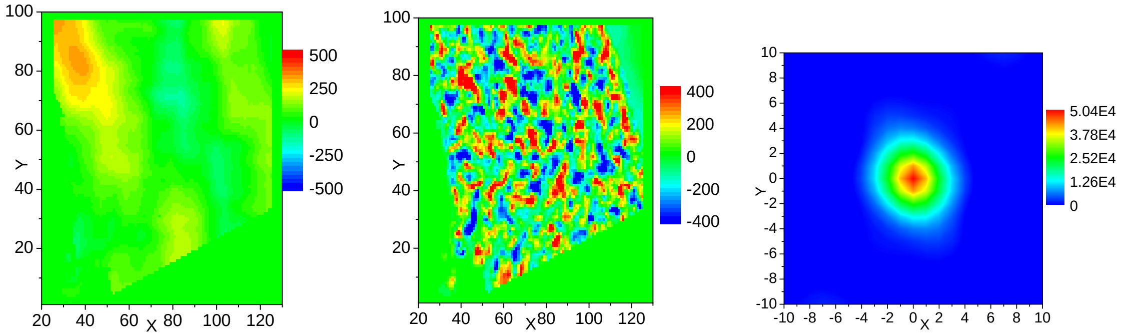


(a) Estimated true signal

(b) Estimated measurement error

(c) Estimated covariance

Figure 8.23: Field acoustic impedance change: direct moving quadratic fitting (21 × 21).

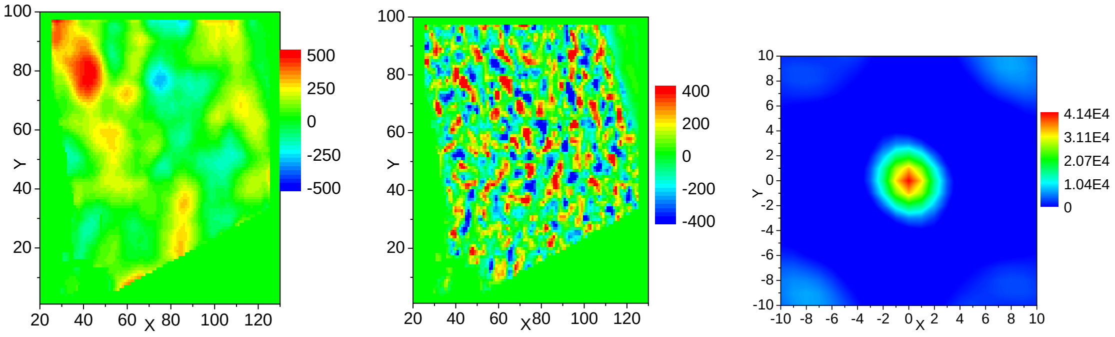


(a) Estimated true signal

(b) Estimated measurement error

(c) Estimated covariance

Figure 8.24: Field Poisson's ratio change: direct moving average (21 × 21).



(a) Estimated true signal

(b) Estimated measurement error

(c) Estimated covariance

Figure 8.25: Field Poisson's ratio change: direct moving quadratic fitting (21 × 21).

Group	π_j	μ_j(AIC)	σ_j(AIC)	μ_j(PRC)	σ_j(PRC)	ρ_j	F_j
1	0.36	-66.3	154.4	6.4	134.7	-0.59	1.18
2	0.14	-520.6	158.5	232.8	151.6	-0.65	0.97
3	0.14	-389.2	167.5	-107.1	190.0	-0.62	0.86
4	0.09	-397.3	129.2	251.3	164.2	-0.87	0.67
5	0.08	-193.3	125.3	287.7	138.8	-0.53	0.97
6	0.05	-121.7	157.6	-152.2	192.5	-0.83	0.57
7	0.05	217.4	140.8	-244.3	223.2	-0.36	0.62

Table 8.2: Field 4-D seismic case (2 data type, random initialization)

Group	π_j	μ_j(AIC)	σ_j(AIC)	μ_j(PRC)	σ_j(PRC)	ρ_j	F_j
1	0.45	-133.2	176.1	62.6	182.0	-0.63	1.17
2	0.15	-528.2	155.7	251.8	164.3	-0.70	0.97
3	0.14	-390.7	190.4	-110.5	201.8	-0.70	0.74
4	0.05	-399.2	128.8	255.3	174.3	-0.89	0.65
5	0.04	-184.3	152.8	333.8	176.6	-0.65	0.51
6	0.07	162.4	153.8	-266.8	217.3	-0.21	0.84
7	0.03	-475.0	161.7	536.5	248.7	-0.72	0.48

Table 8.3: Field 4-D seismic case (2 data type, value initialization)

8.7 Summary

We have provided a modified expectation maximization (EM) algorithm as a tool to separate a set of measured data into groups in a way so that groups tend to be spatially continuous where the measurements in each group follow a Gaussian distribution. By grouping time-lapse seismic data into subsets based on regions prior to smoothing, we avoid applying a smoothing window to a data subset where the true underlying signal changes sharply. The smoothed signal provides an estimate of the "true" data. Subtraction of the smooth data from the corresponding observed data gives an estimate of the measurement error. In all synthetic cases tried here and in other examples, reasonable estimates of the variance(s) of the measurement error are obtained and the estimate of the associated covariance is somewhat reasonable but the correlation length is always underestimated. The estimate of covariance of the measurement error based on quadratic fitting within each group is superior to that obtained with the same moving window applied to the entire data set.

The improved EM algorithm can be used to determine an appropriate number of groups. We have introduced a grouping quality coefficient to enhance spatial continuity within each group and eliminate low continuity groups. After convergence of this process, a second run of the EM iterations is made with a fixed number of groups (without F_j) and MAP grouping in the maximization step. The second run normally changes only the size of each group accompanied with a slight increase of log-likelihood function. Results from the synthetic data presented here indicate that the final grouping can be used to obtain a reasonably reliable characterization of measurement error. Interestingly, our procedure tends to yield groups which correspond to physical changes in the reservoir; for example, one group may correspond to water influx into an original oil column and a second to a region where gas has displaced oil between the two seismic surveys.

We have studied the final groups obtained for different initial groups. In most cases, the differences between the final groups obtained are small, and more important to the study, the characterization of measurement error does not depend significantly on the initial grouping. Different initial guesses were also used for field 4-D seismic data. Although the main structure of the results are similar as are the estimated covariance of the measurement error, the differences in the grouping results indicate that the initial grouping can somewhat affect the final set of groups obtained. Based on all examples considered, we conclude that the estimate of the measurement error does not depend strongly on the initial grouping.

CHAPTER 9

CONCLUSIONS AND DISCUSSIONS

9.1 Conclusions on the Pluri-Gaussian Geological Model

The truncation scheme using two intersecting ellipses is able to model the connectivity of a channelized reservoir. The location of the of channel facies can be moved by changing the Gaussian random fields values in each gridblock using EnKF to match production data and seismic data. The same algorithm is also applied to a three-layer, three facies model, in which three lines are used to define truncation. Facies porosity and permeability can be included as model parameter in the state vectors. EnRML gives a more dependable estimate of these porosity and permeability values than does the standard EnKF algorithm.

We presented a procedure to generate an initial ensemble that honors the hard data (observed facies at well gridblocks) as well as a procedure to ensure that hard data are honored at each data assimilation step. The results presented indicate that these methods are robust.

9.2 Conclusions on EnKF and EnRML

Applying global and local normal score transforms to the water saturations to account for non-Gaussian saturation distributions at a gridblock did not yield any significant improvement over standard EnKF. This may be because the normal score transform is designed to handle non-Gaussian distributions, but may not alleviate the negative effects of strong nonlinearities. Local normal score transform gives less oscillation in the analyzed saturations during EnKF assimilation, but we could not distinguish between the performance of the two transform schemes in the terms of the reliability of reservoir performance prediction.

Assimilating water breakthrough time did not give significant improvements in the EnKF results, but gave some improvement in predictions and data matches when EnRML was used.

The modified Gu-Oliver scheme (EnRML) with individual line search was compared to the standard EnKF. EnRML gives a somewhat better data match and more reliable predictions, but as each iteration requires rerunning the reservoir simulator from time zero to the current assimilation step, the method requires significantly more computational time than EnKF. Moreover, the method failed completely when trying to assimilate seismic data with a global analysis algorithm.

9.3 Conclusions on Assimilating Seismic Data

Global analysis is less efficient than local analysis in assimilating seismic data and can result in filter divergence even for a relatively large ensemble size. Local analysis eliminates filter divergence but results in unrealistically non-smooth facies distribution. In some cases, these distributions may be so unrealistic, that the reservoir simulator can not run with these properties. A projection method was introduced to alleviate this problem. When combined with local analysis, it was shown that this method gives an ensemble of geologically reasonable facies distributions that still yield good estimates of performance predictions for both the PUNQS3 and three facies models. However, this scheme is unable to effectively assimilate seismic data in the channel model even though the projection method preserves the continuity of the channel facies distributions.

9.4 Conclusions on Measurement Error Estimation

Outliers and discontinuities in 1-D production data can be effectively detected using the algorithms developed in this study. Dependable estimates of measurement errors are obtained by applying Savitzky-Golay smoothing or wavelet smoothing to the corrected the signal, which is based on the detected outliers and discontinuities.

The spatial EM algorithm is successful in grouping the 4-D seismic data with an unknown number of groups according to the measured values and their spatial coordinates. Smoothing across the boundaries can be avoided by smoothing data group by group, and the estimated measurement error is superior to those obtained by smoothing without grouping.

BIBLIOGRAPHY

[1] S. I. Aanonsen, I. Aavatsmark, T. Barkve, A. Cominelli, R. Gonard, O. Gosselin, M. Kolasinski, and H. Reme. Effect of scale dependent data correlations in an integrated history matching loop combining production data and 4D seismic data (SPE 79665). In *Proceedings of the 2003 SPE Reservoir Simulation Symposium*, 2003.

[2] F. Abramovich, T. C. Bailey, and T. Sapatinas. Wavelet analysis and its statistical application. *J. Royal Statist. Soc. Ser. D*, 49:1–29, 2000.

[3] D. Allard and G. Guillot. Clustering geostatistical data. *Geostatistics 2000 Cape Town*, 1:49–63, 1999.

[4] C. Ambroise and G. Govaert. Spatial clustering and the em algorithm. *http://www.isip.msstate.edu/projects/speech/support/help/bibliography/index.html*, 1995.

[5] Margaret Armstrong, Alain G. Galli, Gaëlle Le Loc'h, François Geffroy, and Rémi Eschard. *Plurigaussian Simulations in Geosciences.* Springer, New York, 2003.

[6] Suwaat Athichanagorn, Roland N. Horne, and Jitendra Kikani. Processing and interpretation of long-term data from permanent downhole pressure gauges, SPE 56419. In *Proc. of SPE Annual Tech Conf.*, 1999.

[7] John W. Barker, Maarten Cuypers, and Lars Holden. Quantifying uncertainty in production forecasts: Another look at the PUNQ-S3 problem. *SPE Journal*, 6(4):433–441, 2001.

[8] L. Bertino, Geir Evensen, and Hans Wackernagel. Sequential data assimilation techniques in oceanography. *International Statistical Review*, 71(2):223–241, 2003.

[9] Julian E. Besag. On the statistical analysis of dirty pictures. *J. R. Statist. Soc. Series B*, 48(3):295–302, 1986.

[10] A. Bianco, A. Cominelli, L. Dovera, G. Naevdal, and B. Valles. History matching and production forecast uncertainty by means of the ensemble Kalman filter: A real field application (SPE-107161). In *EUROPEC/EAGE Conference and Exhibition*, 2007.

[11] Gilles Celeux and Gerard Govaert. A classification EM algorithm for clustering and two stochastic versions. *Rapports de Recherche.*, January 1991.

[12] Ingrid Daubechies. *Ten Lectures on Wavelets*. SIAM, Philadelphia, 1992.

[13] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *J. Roy. Statist. Soc. B*, 39(1):1–38, 1977.

[14] Clayton V. Deutsch and André G. Journel. *GSLIB: Geostatistical Software Library and User's Guide*. Oxford University Press, New York, 1992.

[15] Deepak Devegowda, Elkin Arroyo-Negrete, Akhil Datta-Gupta, and S. G. Douma. Efficient and robust reservoir model updating using ensemble Kalman filter with sensitivity-based covariance localization (SPE-106144). In *SPE Reservoir Simulation Symposium*, 2007.

[16] A. Diplaros, T. Gevers, and N. Vlassis. Skin detection using the em algorithm with spatial constraints. 4, 2004.

[17] Yannong Dong and Dean Oliver. Automatic history matching of production data and seismic impedance change data. In *TUPREP Research Report 20.* The University of Tulsa, 2003.

[18] Yannong Dong and Dean S. Oliver. Quantitative use of 4d seismic data for reservoir description. *SPE Journal*, 10(1):51–65, 2005.

[19] D. L. Donoho and I. M. Johnstone. Adapting to unknown smoothness via wavelet shrinkage. *J. Amer. Statist. Assn.*, 90:1200–1224, 1995.

[20] D. L. Donoho and I. M. Johnstone. Mimimax estimation via wavelet shrinkage. *Ann. Statist.*, 26:879–921, 1998.

[21] Jack Dvorkin. Detecting overpressure from seismic velocity calibrated to log and core measurements. *32nd Annual Offshore Technology Conference*, May 2000.

[22] Geir Evensen. Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *Journal of Geophysical Research*, 99:10143–10162, 1994.

[23] Geir Evensen. Advanced data assimilation for strongly nonlinear dynamics. *Monthly Weather Review*, 125(6):1342–1354, 1997.

[24] Geir Evensen. The ensemble Kalman filter: theoretical formulation and practical implementation. *Ocean Dynamics*, 53:343–367, 2003.

[25] Geir Evensen. Sampling strategies and square root analysis schemes for EnKF. Technical report, Hydro Research Center and Nansen Environmental and Remote Sensing Center, 2004.

[26] Geir Evensen. *Data Assimilation: The Ensemble Kalman Filter.* Springer, Berlin, 2006.

[27] Geir Evensen. *Data Assimilation: The Ensemble Kalman Filter*. Springer, 2007.

[28] Geir Evensen, J. Hove, H. C. Meisingset, E. Reiso, K. S. Seim, and (SPE-90896) O. Espelid. Using the EnKF for assisted history matching of a north sea reservoir. In *Proceeding of the SPE Reservoir Simulation Symposium*, 2007.

[29] Frans J. T. Floris, 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.

[30] Guohua Gao and A. C. Reynolds. An improved implementation of the LBFGS algorithm for automatic history matching (SPE-90058). In *2004 SPE Annual Technical Conference and Exhibition*, 2004.

[31] Guohua Gao and A. C. Reynolds. An improved implementation of the LBFGS algorithm for automatic history matching. *SPE Journal*, 11(1):5–17, 2006.

[32] F. Gassmann. Elastic waves through a packing of spheres. *Geophysics*, 16:673–685, 1951.

[33] Yaqing Gu and D. S. Oliver. History matching of the PUNQ-S3 reservoir model using the ensemble Kalman filter. *SPE Journal*, 10(2):51–65, 2005.

[34] Yaqing Gu and D. S. Oliver. An iterative ensemble Kalman filter for multiphase fluid flow data assimilation (SPE-108438). *SPE Journal*, 12(4):438–446, 2007.

[35] Yaqing Gu and Dean S. Oliver. The ensemble Kalman filter for continuous updating of reservoir simulation models. *J. of Energy Resources Technology*, 128(1):79–87, 2006.

[36] D. Han. *Effects of porosity and clay content on acoustic properties of sandstones and unconsolidated sediments*. PhD thesis, Stanford University, 1986.

[37] H. Hartley. Maximum likelihood from incomplete data. *Biometrics*, 14(1):174–194, 1958.

[38] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer-Verlag, New York, 2001.

[39] V. Haugen, L.-J. Natvik, G. Evensen, A. Berg, K. Flornes, and G. Nævdal. History matching using the ensemble Kalman filter on a North Sea field case (SPE-102430). In *SPE Annual Technical Conference and Exhibition*, 2006.

[40] P. L. Houtekamer and Herschel L. Mitchell. A sequential ensemble Kalman filter for atmospheric data assimilation. *Monthly Weather Review*, 129(1):123–137, 2001.

[41] P. L. Houtekamer, Herschel L. Mitchell, Gérard Pellerin, Mark Buehner, Martin Charron, Lubos Spacek, and Bjarne Hansen. Atmospheric data assimilation with an ensemble Kalman filter: Results with real observations. *Monthly Weather Review*, 133(3):604–620, 2005.

[42] I. M. Johnstone and B. W. Silverman. Wavelet threshold estimators for data with correlated noise. *J. Royal Statist. Soc. Ser. B*, 59:319–351, 1997.

[43] Jeffrey D. Kepert. On ensemble representation of the observation-error covariance in the Ensemble Kalman Filter. *Ocean Dynamics*, 54(6):561–569, 2004.

[44] Jitendra Kikani and Meiquing He. Multi-resolution analyis of long-term pressure transient data using wavelet methods, SPE 48966. In *Proc. of SPE Annual Tech Conf.*, pages 117–126, 1998.

[45] Peter K. Kitanidis. Quasi-linear geostatistical theory for inversing. *Water Resour. Res.*, 31(10):2411–2419, 1995.

[46] Steven G. Krantz. *Handbook of Complex Variables*. MA: Birkhauser, 1999.

[47] S.Y. Kung, M.W. Mak, and S.H. Lin. *Biometric Authentication: A Machine Learning Approach.* Prentice Hall, 2004.

[48] G. Le Loc'h and A. Galli. Truncated plurigaussian method: Theoretical and practical points of view. In E. Y. Baafi and N. A. Schofield, editors, *Geostatistics Wollongong '96*, volume 1, pages 211–222. Kluwer Academic, 1997.

[49] Gaoming Li and A. C. Reynolds. An iterative ensemble kalman filter for data assimilation, (SPE-109808). In *Proceedings of the 2007 SPE Annual Technical Conference and Exhibition*, 2007.

[50] Ruijian Li. *Conditioning Geostatistical Models to Three-Dimensional Three-Phase Flow Production Data by Automatic History Matching.* Ph.D. thesis, University of Tulsa, Tulsa, Oklahoma, 2001.

[51] Ruijian Li, A. C. Reynolds, and D. S. Oliver. History matching of three-phase flow production data. *SPE J.*, 8(4):328–340, 2003.

[52] Ruijian Li, Albert C. Reynolds, and Dean S. Oliver. Sensitivity coefficients for three-phase flow history matching. *J. Canadian Pet. Tech.*, 42(4):70–77, 2003.

[53] Ning Liu and Dean S. Oliver. Automatic history matching of geologic facies. *Soc. Petrol. Eng. J.*, 8(2):188–195, 2004.

[54] Ning Liu and Dean S. Oliver. Critical evaluation of the ensemble Kalman filter on history matching of geologic facies. *SPE Reservoir Evaluation and Engineering*, 8(4):470–477, 2005.

[55] Gaelle Le Loc'h and Alain Galli. Truncated plurigaussian method: Theoretical and practical point of view. *Geostatistics Wollongong'96*, 1:211–222, 1997.

[56] Rolf J. Lorentzen, Geir Nævdal, Brice Vàlles, Aina M. Berg, and alv Arne Grimstad. Analysis of the ensemble Kalman filter for estimation of permeability and

porosity in reservoir models. In *SPE Annual Technical Conference and Exhibition*, 2005.

[57] Xiao-Li Meng. Maximum likelihood estimation via the ECM algorithm: A general framework. *Biometrika*, 80(2):267–278, Jun. 1993.

[58] Xiao-Li Meng. On the rate of convergence of the ECM algorithm. *Annals of Statistics*, 22(1):326–339, Mar. 1994.

[59] Herschel L. Mitchell, P. L. Houtekamer, and Gérard Pellerin. Ensemble size, balance, and model-error representation in an ensemble Kalman filter. *Monthly Weather Review*, 130(11):2791–2808–433, 2002.

[60] G. Naevdal, L. M. Johnsen, S. I. Aanonsen, and E. H. Vefring. Reservoir monitoring and continuous model updating using ensemble Kalman filter (SPE-84372). In *2003 SPE Annual Technical Conference and Exhibition*, 2003.

[61] G. Naevdal, 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.

[62] G. Naevdal, T. Mannseth, and E. H. Vefring. Near-well reservoir monitoring through ensemble Kalman filter (SPE-75235). In *Proceeding of SPE/DOE Improved Oil Recovery Symposium*, 2002.

[63] Dean S. Oliver, Nanqun He, and Albert C. Reynolds. Conditioning permeability fields to pressure data. In *European Conference for the Mathematics of Oil Recovery, V*, pages 1–11, 1996.

[64] Edward Ott, Brian R. Hunt, Istvan Szunyogh, Aleksey V. Zimin, Eric J. Kostelich, Matteo Corazza, Eugenia Kalnay, D. J. Patil, and James A. Yorke.

A local ensemble Kalman filter for atmospheric data assimilation. *Tellus A*, 56(5):415–428, 2004.

[65] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in FORTRAN: The Art of Scientific Computing*. Cambridge University Press, Cambridge, England, 1992.

[66] Richard A. Redner, Richard J. Hathaway, and James C. Bezdek. Estimating the parameters of mixture models with modal estimators. *Communications in Statistics - Theory and Methods*, 16(9):2639–2660, 1987.

[67] Richard A. Redner and Homer F. Walker. Mixture densities, maximum likelihood and the em algorithm. *SIAM Review*, 26(2):195–239, 1984.

[68] A. C. Reynolds, Mohammad Zafari, and Gaoming Li. Iterative forms of the ensemble kalman filter. *Proceedings of 10th European Conference on the Mathematics of Oil Recovery*, Amsterdam, 4–7 September 2006.

[69] Albert C. Reynolds, Nanqun He, and Dean S. Oliver. Reducing uncertainty in geostatistical description with well testing pressure data. In Richard A. Schatzinger and John F. Jordan, editors, *Reservoir Characterization—Recent Advances*, pages 149–162. American Association of Petroleum Geologists, 1999.

[70] Sylvia Richardson and Peter J. Green. On Bayesian analysis of mixtrues with an unknown number of components. *J. R. Statist. Soc.*, 59(4):731–792, June 1997.

[71] T. Schaaf, M. Mezghani, and G. Chavent. Direct conditioning of fine-scale facies models to dynamic data by combining gradual deformation and numerical upscaling techniques. In *8th European Conference on the Mathematics of Oil Recovery*, 2002.

[72] James R. Schott. *Matrix Analysis for Statistics.* Willey Series in Probability and Statistics, 1997.

[73] J. A. Skjervheim, 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–95789. In *SPE Annual Technical Conference and Exhibition, 9-12 October, Dallas, Texas*, 2005.

[74] J-A. Skjervheim, 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.

[75] J. A. Skjervheim, B. O. Ruud, S. I. Aanonsen, G. Evensen, S. I. Aanonsen, and T. A. Johansen. Using the ensemble Kalman filter with 4D data to estimate properties and lithology of reservoir rocks. In *Proceedings of 10th European Conference on the Mathematics of Oil Recovery*, Amsterdam, 4–7 September 2006.

[76] Albert Tarantola. *Inverse Problem Theory: Methods for Data Fitting and Model Parameter Estimation.* Elsevier, Amsterdam, The Netherlands, 1987.

[77] Kristian Thulin, Gaoming Li, Sigurd Ivar Aanonsen, and Albert C. Reynolds. Estimation of initial fluid contacts by assimilation of production data with EnKF, (SPE-109975). In *Proceedings of the 2007 SPE Annual Technical Conference and Exhibition*, 2007.

[78] Xian-Huan Wen and Wen H. Chen. Real-time reservoir model updating using ensemble Kalman filter with confirming option. *SPE Journal*, 11(4):431–442, 2006.

[79] Xian-Huan Wen and Wen H. Chen. Some practical issues on real-time reservoir model updating using ensemble Kalman filter. *SPE Journal*, 12(2):156–166, 2007.

[80] Mohammad Zafari and Albert C. Reynolds. Assessing the uncertainty in reservoir description and performance predictions with the ensemble Kalman filter (SPE 95750). In *Proceedings of the SPE Annual Technical Conference and Exhibition*, 2005.

[81] Mohammad Zafari and Albert C. Reynolds. Assessing the uncertainty in reservoir description and performance predictions with the ensemble Kalman filter. *SPE Journal*, 12(3):382–391, 2007.

[82] F. Zhang and A. C. Reynolds. Optimization algorithms for automatic history matching of production data. *Proceedings of 8th European Conference on the Mathematics of Oil Recovery*, pages 1–10, 2002.

[83] F. Zhang, 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.

[84] F. Zhang, A. C. Reynolds, and D. S. Oliver. Conditioning a stochastic channel to pressure data: Generation of an initial guess for fast convergence of the Levenberg-Marquardt algorithm. *Mathematical Geology*, 35(1):67–88, 2003.

[85] Yong Zhao, Gaoming Li, and A. C. Reynolds. Characterizing measurement errors with the EM algorithm. In *TUPREP Research Report 23*, pages 14–105. The University of Tulsa, 2006.

[86] M. Zupanski. Maximum likelihood ensemble filter: Theoretical aspects. *Monthly Weather Review*, 133(6):1710–1726, 2005.

APPENDIX A

CONVERGENCE OF EM ALGORITHM

The convergence of the EM algorithm are discussed in the case of non-sequential measurements (no spatial constraint) and the case where we use a spatial constraint. The so-called Q function (see [67, 66]) and Q_c function derived in these two cases are discussed separately. The optimization of these two functions will be discussed in Appendix C. Recall that $\Delta = \{\delta_1, \delta_2, \cdots, \delta_N\}$ and $D = \{d_1, d_2, \cdots, d_N\}$ are the random vectors used to model the group indicators and measured values of the N grid-blocks.

A.1 Convergence of EM (Non-Sequential)

To estimate the Gaussian mixture model parameters $\bar{\Theta} = \{\Theta, \Pi\}$ based on the measured value D, we maximize the following log-likelihood function:

$$L(\bar{\Theta}|D) = \ln[P(D|\bar{\Theta})]. \tag{A.1}$$

We let $\bar{\Theta}_n$ denote the estimates of the parameters of the Gaussian mixture model at the nth iteration of the EM algorithm. We must always have

$$\sum_{\Delta} P(\Delta|D, \bar{\Theta}_n) = 1, \tag{A.2}$$

which means discrete probability of all possible values of Δ sum to 1.

Multiplying Eq. A.1 by Eq. A.2 yields

$$\begin{aligned}
L(\bar{\Theta}|D) &= \ln[P(D|\bar{\Theta})] \sum_{\Delta} P(\Delta|D, \bar{\Theta}_n) \\
&= \sum_{\Delta} \Big(P(\Delta|D, \bar{\Theta}_n) \ln[P(D|\bar{\Theta})] \Big) \\
&= \sum_{\Delta} \Big(P(\Delta|D, \bar{\Theta}_n) \ln \Big[\frac{P(D, \Delta|\bar{\Theta})}{P(\Delta|D, \bar{\Theta})} \Big] \Big) \\
&= \sum_{\Delta} \Big(P(\Delta|D, \bar{\Theta}_n) \ln[P(D, \Delta|\bar{\Theta})] \Big) - \sum_{\Delta} \Big(P(\Delta|D, \bar{\Theta}_n) \ln[P(\Delta|D, \bar{\Theta})] \Big) \\
&= Q(\bar{\Theta}|\bar{\Theta}_n) + R(\bar{\Theta}|\bar{\Theta}_n),
\end{aligned} \tag{A.3}$$

where the Q function and the R function are defined, respectively, as

$$\begin{aligned}
Q(\bar{\Theta}|\bar{\Theta}_n) &= \sum_{\Delta} \Big(P(\Delta|D, \bar{\Theta}_n) \ln[P(D, \Delta|\bar{\Theta})] \Big) \text{ and} \\
R(\bar{\Theta}|\bar{\Theta}_n) &= -\sum_{\Delta} \Big(P(\Delta|D, \bar{\Theta}_n) \ln[P(\Delta|D, \bar{\Theta})] \Big).
\end{aligned}$$

In the following, we will show the log-likelihood function (Eq. A.3) can be increased by maximizing the Q function. To do this, we need to show that $L(\bar{\Theta}_{n+1}|D) \geq L(\bar{\Theta}_n|D)$ if the Q function increases from the nth iteration to the $(n+1)$th iteration.

In Appendix B, the following inequality is proved for any $\bar{\Theta}_{n+1}$ using Jensen's inequality:

$$\begin{aligned}
L(\bar{\Theta}_{n+1}|D) &= Q(\bar{\Theta}_{n+1}|\bar{\Theta}_n) + R(\bar{\Theta}_{n+1}|\bar{\Theta}_n) \\
&\geq Q(\bar{\Theta}_{n+1}|\bar{\Theta}_n) + R(\bar{\Theta}_n|\bar{\Theta}_n).
\end{aligned} \tag{A.4}$$

The preceding equation indicates that for any $\bar{\Theta}_{n+1}$,

$$R(\bar{\Theta}_{n+1}|\bar{\Theta}_n) \geq R(\bar{\Theta}_n|\bar{\Theta}_n).$$

Thus, if we can ensure that the Q function increases from the nth iteration to the $n+1$th iteration so that

$$Q(\bar{\Theta}_{n+1}|\bar{\Theta}_n) \geq Q(\bar{\Theta}_n|\bar{\Theta}_n), \tag{A.5}$$

then it follows from Eq. A.4 that

$$\begin{aligned} L(\bar{\Theta}_{n+1}|D) &= Q(\bar{\Theta}_{n+1}|\bar{\Theta}_n) + R(\bar{\Theta}_{n+1}|\bar{\Theta}_n) \\ &\geq Q(\bar{\Theta}_{n+1}|\bar{\Theta}_n) + R(\bar{\Theta}_n|\bar{\Theta}_n) \\ &\geq Q(\bar{\Theta}_n|\bar{\Theta}_n) + R(\bar{\Theta}_n|\bar{\Theta}_n) \\ &= L(\bar{\Theta}_n|D). \end{aligned} \tag{A.6}$$

To improve the Q function, the following optimization problem is normally solved:

$$\bar{\Theta}_{n+1} = \mathrm{argmax}_{\bar{\Theta}} Q(\bar{\Theta}|\bar{\Theta}_n). \tag{A.7}$$

We attempt to maximize the Q function at each iteration by setting the derivative of Q with respect to each parameter in Q to zero.

A.2 Convergence of EM (Spatial)

In the case of spatial measurement, we need to maximize the modified log-likelihood function shown in Eq. 8.22, which can be written as:

$$L_c(\bar{\Theta}, \hat{Z}|D) = \ln[P(D|\bar{\Theta}, \hat{Z})] \tag{A.8}$$

We let $\bar{\Theta}_n$ and Z_n denote the estimates of the parameters and group indicator of the Gaussian mixture model at the nth iteration of the EM algorithm. Using

$\hat{Z} = Z_n$, we also always have:

$$\sum_{\Delta} P(\Delta|D, \bar{\Theta}_n, Z_n) = 1. \tag{A.9}$$

which means discrete probability of all possible values of Δ sum to 1.

Multiplying Eq. A.8 by Eq. A.9 yields

$$\begin{aligned}
L_c(\bar{\Theta}, Z_n|D) =& \ln[P(D|\bar{\Theta}, Z_n)] \sum_{\Delta} P(\Delta|D, \bar{\Theta}_n, Z_n) \\
=& \sum_{\Delta} \Big(P(\Delta|D, \bar{\Theta}_n, Z_n) \ln[P(D|\bar{\Theta}, Z_n)] \Big) \\
=& \sum_{\Delta} \Big(P(\Delta|D, \bar{\Theta}_n, Z_n) \ln \Big[\frac{P(D, \Delta|\bar{\Theta}, Z_n)}{P(\Delta|D, \bar{\Theta}, Z_n)} \Big] \Big) \\
=& \sum_{\Delta} \Big(P(\Delta|D, \bar{\Theta}_n, Z_n) \ln[P(D, \Delta|\bar{\Theta}, Z_n)] \Big) \\
& - \sum_{\Delta} \Big(P(\Delta|D, \bar{\Theta}_n, Z_n) \ln[P(\Delta|D, \bar{\Theta}, Z_n)] \Big) \\
=& Q_c(\bar{\Theta}|\bar{\Theta}_n, Z_n) + R_c(\bar{\Theta}|\bar{\Theta}_n, Z_n),
\end{aligned} \tag{A.10}$$

where the Q_c function and the R_c function are defined, respectively, as

$$\begin{aligned}
Q_c(\bar{\Theta}|\bar{\Theta}_n, Z_n) &= \sum_{\Delta} \Big(P(\Delta|D, \bar{\Theta}_n, Z_n) \ln[P(D, \Delta|\bar{\Theta}, Z_n)] \Big) \\
\text{and } R_c(\bar{\Theta}|\bar{\Theta}_n, Z_n) &= -\sum_{\Delta} \Big(P(\Delta|D, \bar{\Theta}_n, Z_n) \ln[P(\Delta|D, \bar{\Theta}, Z_n)] \Big).
\end{aligned}$$

The log-likelihood function (Eq. A.10) can be shown to increased from iteration to iteration, i.e. $L_c(\bar{\Theta}_{n+1}, Z_{n+1}|D) \geq L_c(\bar{\Theta}_{n+1}, Z_n|D) \geq L_c(\bar{\Theta}_n, Z_n|D)$, if we can prove the following two points:

1. $L_c(\bar{\Theta}_{n+1}, Z_{n+1}|D) \geq L_c(\bar{\Theta}_{n+1}, Z_n|D)$ as the first inequity;

2. $L_c(\bar{\Theta}_{n+1}, Z_n|D) \geq L_c(\bar{\Theta}_n, Z_n|D)$ (second inequity) if the Q_c function increases

from the nth iteration to the $(n+1)$th iteration, i.e.

$$Q_c(\bar{\Theta}_{n+1}|\bar{\Theta}_n, Z_n) \geq Q_c(\bar{\Theta}_n|\bar{\Theta}_n, Z_n).$$

In the following, we will show that the second point is true. We found that it is intractable to find the updated model model $\bar{\Theta}_{n+1}$ that satisfy the first point strictly, which prevent us from proving that the log-likelihood is non-decrease from iteration to iteration. However, in the Appendix C, we will still show how we optimize the Q_c function.

In Appendix B, the following inequality is proved using Jensen's inequality:

$$\begin{aligned} L_c(\bar{\Theta}_{n+1}, Z_n|D) &= Q_c(\bar{\Theta}_{n+1}|\bar{\Theta}_n, Z_n) + R_c(\bar{\Theta}_{n+1}|\bar{\Theta}_n, Z_n) \\ &\geq Q_c(\bar{\Theta}_{n+1}|\bar{\Theta}_n, Z_n) + R_c(\bar{\Theta}_n|\bar{\Theta}_n, Z_n). \end{aligned} \tag{A.11}$$

The preceding equation indicates that

$$R_c(\bar{\Theta}_{n+1}|\bar{\Theta}_n, Z_n) \geq R_c(\bar{\Theta}_n|\bar{\Theta}_n, Z_n)$$

holds for any two different models for $\bar{\Theta}_n$ and $\bar{\Theta}_{n+1}$. Here, if $L_c(\bar{\Theta}_{n+1}, Z_n|D) \geq L_c(\bar{\Theta}_n, Z_n|D)$, then we have $Q_c(\bar{\Theta}_{n+1}|\bar{\Theta}_n, Z_n) \geq Q_c(\bar{\Theta}_n|\bar{\Theta}_n, Z_n)$. Thus, if we can ensure that the Q_c function increases from the nth iteration to the $n+1$th iteration so that

$$Q_c(\bar{\Theta}_{n+1}|\bar{\Theta}_n, Z_n) \geq Q_c(\bar{\Theta}_n|\bar{\Theta}_n, Z_n), \tag{A.12}$$

then it follows from Eq. A.11 that

$$\begin{aligned} L_c(\bar{\Theta}_{n+1}, Z_n|D) &= Q_c(\bar{\Theta}_{n+1}|\bar{\Theta}_n, Z_n) + R_c(\bar{\Theta}_{n+1}|\bar{\Theta}_n, Z_n) \\ &\geq Q_c(\bar{\Theta}_{n+1}|\bar{\Theta}_n, Z_n) + R_c(\bar{\Theta}_n|\bar{\Theta}_n, Z_n) \\ &\geq Q_c(\bar{\Theta}_n|\bar{\Theta}_n, Z_n) + R_c(\bar{\Theta}_n|\bar{\Theta}_n, Z_n) \\ &= L_c(\bar{\Theta}_n, Z_n|D). \end{aligned} \tag{A.13}$$

To improve the Q_c function, the following optimization problem is normally solved:

$$\bar{\Theta}_{n+1} = \text{argmax}_{\bar{\Theta}} Q_c(\bar{\Theta}|\bar{\Theta}_n, Z_n). \tag{A.14}$$

We attempt to maximize the Q_c function at each iteration by setting the derivative of Q_c with respect to each parameter in Q_c to zero. As is shown earlier in the first point of proof, we need also to make sure that $L_c(\bar{\Theta}_{n+1}, Z_{n+1}|D) \geq L_c(\bar{\Theta}_{n+1}, Z_n|D)$, which is intractable and prevent us from proving the non-decreasing properties of the log-likelihood.

A.3 Discussions on Q function (Non-Sequential)

As noted previously, the Q function is defined as

$$Q(\bar{\Theta}|\bar{\Theta}_n) = \sum_{\Delta} \left(P(\Delta|D, \bar{\Theta}_n) \ln \left[P(\Delta, D|\bar{\Theta}) \right] \right). \tag{A.15}$$

In non-sequential cases, data are independent, $P(\Delta, D|\bar{\Theta})$ can be written as

$$P(\Delta, D|\bar{\Theta}) = \prod_{i=1}^{N} P(\delta_i, d_i|\bar{\Theta}). \tag{A.16}$$

If we define the random indicator variable δ_i^j by

$$\delta_i^j = \begin{cases} 1, \text{ if } \delta_i = j \\ 0, \text{otherwise} \end{cases} \tag{A.17}$$

then $P(\delta_i, d_i|\bar{\Theta})$ can be written as

$$P(\delta_i, d_i|\bar{\Theta}) = \sum_{j=1}^{M} \delta_i^j P(\delta_i = j, d_i|\bar{\Theta}). \tag{A.18}$$

Note that exactly one entry in the summation in the preceding equation can be non-zero.

Substituting Eq. A.18 into Eq. A.16 yields

$$P(\Delta, D|\bar{\Theta}) = \prod_{i=1}^{N} \sum_{j=1}^{M} \delta_i^j P(\delta_i = j, d_i|\bar{\Theta}), \tag{A.19}$$

and

$$\ln[P(\Delta, D|\bar{\Theta})] = \sum_{i=1}^{N} \sum_{j=1}^{M} \delta_i^j \ln[P(\delta_i = j, d_i|\bar{\Theta})]. \tag{A.20}$$

Note that because only one δ_i^j is nonzero in the summation part, we moved the ln operator inside in Eq. A.20.

Using Eq. A.20 in Eq. A.15 yields:

$$\begin{aligned} Q(\bar{\Theta}|\bar{\Theta}_n) &= \sum_{\Delta} \left\{ P(\Delta|D, \bar{\Theta}_n) \sum_{i=1}^{N} \sum_{j=1}^{M} \delta_i^j \ln\left[P(\delta_i = j, d_i|\bar{\Theta})\right] \right\} \\ &= \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{\Delta} \left[P(\Delta|D, \bar{\Theta}_n)\delta_i^j\right] \ln\left[P(\delta_i = j, d_i|\bar{\Theta})\right]. \end{aligned} \tag{A.21}$$

The inner summation of Eq. A.21 is actually the expectation of δ_i^j with respect to the whole sampling space of Δ. Since δ_i^j is only related to δ_i, it can be written as

$$\begin{aligned}\sum_{\Delta}[P(\Delta|D,\bar{\Theta}_n)\delta_i^j] &= \sum_{k=1}^{M}\delta_i^j P(\delta_i = k|D,\bar{\Theta}_n)\\ &= P(\delta_i = j|D,\bar{\Theta}_n)\\ &= P(\delta_i = j|d_i,\bar{\Theta}_n)\\ &= (h_i^j)_n\end{aligned} \tag{A.22}$$

where $(h_i^j)_n$ is defined in Eq. 8.11 as the expectation of δ_i^j with respect to Δ evaluated at $\bar{\Theta}_n$ with given measurement D. Therefore,

$$Q(\bar{\Theta}|\bar{\Theta}_n) = \sum_{i=1}^{N}\sum_{j=1}^{M}(h_i^j)_n \ln\big[P(\delta_i = j, d_i|\bar{\Theta})\big]. \tag{A.23}$$

Here again, h_i^j represents the conditional probability that the ith datum is sampled from the jth Gaussian model or belongs to the jth group given $\bar{\Theta}_n$ and d_i. As in Eq. 8.8, the collection of memberships (the $(h_i^j)_n$'s) is defined for all the data as the membership matrix in the nth iteration. More specifically, the membership matrix is given by the $M \times N$ matrix

$$H_n = \{(h_i^j)_n\}$$

where the subscript n indicates that the membership matrix H_n is evaluated at the model $\bar{\Theta}_n$ obtained at the nth iteration.

A.4 Discussions on Q_c function (Spatial)

As noted previously, the Q_c function is defined as

$$Q_c(\bar{\Theta}|\bar{\Theta}_n, Z_n) = \sum_{\Delta}\big(P(\Delta|D,\bar{\Theta}_n, Z_n)\ln\big[P(\Delta, D|\bar{\Theta}, Z_n)\big]\big). \tag{A.24}$$

For spatially measured data, the group indicators are independent, and the measured value d_i is dependent only on δ_i. According Eq. 8.14, the conditional PDF can be written as:

$$P(\Delta, D|\bar{\Theta}, Z_n) = \prod_{i=1}^{N} P(\delta_i, d_i|\bar{\Theta}, Z_n). \tag{A.25}$$

If we define the random indication variable δ_i^j by

$$\delta_i^j = \begin{cases} 1, \text{ if } \delta_i = j \\ 0, \text{otherwise} \end{cases} \tag{A.26}$$

then $P(\delta_i, d_i|\bar{\Theta}, Z_n)$ can be written as

$$P(\delta_i, d_i|\bar{\Theta}, Z_n) = \sum_{j=1}^{M} \delta_i^j P(\delta_i = j, d_i|\bar{\Theta}, Z_n). \tag{A.27}$$

Note that exactly one entry in the summation in the preceding equation can be non-zero.

Substituting Eq. A.27 into Eq. A.25 yields

$$P(\Delta, D|\bar{\Theta}, Z_n) = \prod_{i=1}^{N} \sum_{j=1}^{M} \delta_i^j P(\delta_i = j, d_i|\bar{\Theta}, Z_n), \tag{A.28}$$

and

$$\ln[P(\Delta, D|\bar{\Theta}, Z_n)] = \sum_{i=1}^{N} \sum_{j=1}^{M} \delta_i^j \ln[P(\delta_i = j, d_i|\bar{\Theta}, Z_n)]. \tag{A.29}$$

Note that because only one δ_i^j is nonzero in the summation part, we moved the ln operator inside in Eq. A.29.

Using Eq. A.29 in Eq. A.24 yields:

$$\begin{aligned} Q_c(\bar{\Theta}|\bar{\Theta}_n, Z_n) &= \sum_{\Delta} \left\{ P(\Delta|D, \bar{\Theta}_n, Z_n) \sum_{i=1}^{N} \sum_{j=1}^{M} \delta_i^j \ln \left[P(\delta_i = j, d_i|\bar{\Theta}, Z_n) \right] \right\} \\ &= \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{\Delta} \left[P(\Delta|D, \bar{\Theta}_n, Z_n) \delta_i^j \right] \ln \left[P(\delta_i = j, d_i|\bar{\Theta}, Z_n) \right]. \end{aligned} \tag{A.30}$$

The inner summation of Eq. A.30 is actually the expectation of δ_i^j with respect to the whole sampling space of Δ. Since δ_i^j is only related to δ_i, it can be written as

$$\begin{aligned} \sum_{\Delta} [P(\Delta|D, \bar{\Theta}_n, Z_n) \delta_i^j] &= \sum_{k=1}^{M} \delta_i^j P(\delta_i = k|D, \bar{\Theta}_n, Z_n) \\ &= P(\delta_i = j|D, \bar{\Theta}_n, Z_n) \\ &= P(\delta_i = j|d_i, \bar{\Theta}_n, Z_n) \\ &= (h_i^j)_n \end{aligned} \tag{A.31}$$

where $(h_i^j)_n$ is defined in Eq. 8.18 as the expectation of δ_i^j with respect to Δ evaluated at $\bar{\Theta}_n$ and Z_n with given measurement D. Therefore,

$$Q(\bar{\Theta}|\bar{\Theta}_n, Z_n) = \sum_{i=1}^{N} \sum_{j=1}^{M} (h_i^j)_n \ln \left[P(\delta_i = j, d_i|\bar{\Theta}, Z_n) \right]. \tag{A.32}$$

Here again, h_i^j represents the conditional probability that the ith datum is sampled from the jth Gaussian model or belongs to the jth group given $\bar{\Theta}_n$, Z_n and d_i. As in Eq. 8.18, the collection of memberships (the $(h_i^j)_n$'s) is defined for all the data as the membership matrix in the nth iteration. More specifically, the membership matrix is given by the $M \times N$ matrix

$$H_n = \{(h_i^j)_n\}$$

where the subscript n indicates that the membership matrix H_n is evaluated at the model $\bar{\Theta}_n$ obtained at the nth iteration.

For non-sequential and spatial measurements, H_n is evaluated using different equations. However, for both cases, the evaluation of H_n is called the expectation step, while the maximization of the Q or Q_c function is called the maximization step. In the maximization step, the $\bar{\Theta}_{n+1}$ is evaluated as the updated model.

APPENDIX B

JENSEN'S INEQUALITY IN THE Q FUNCTION

Jensen's inequality([46]): For any real-valued concave function $f(x)$ defined on $[a, b]$,

$$f\Big(\sum_{k=1}^{n} \lambda_k x_k\Big) \geq \sum_{k=1}^{n} \lambda_k f(x_k), \tag{B.1}$$

for any x_k in $[a, b]$, where $0 \leq \lambda_k \leq 1$, for $1 \leq k \leq n$, and

$$\lambda_1 + \lambda_2 + ... + \lambda_n = 1.$$

If $\bar{\Theta}_n$ is the model at iteration n, and $\bar{\Theta}_{n+1}$ is the model at iteration $n+1$, then

$$\begin{aligned} L(\bar{\Theta}_{n+1}|D) &= \ln\big[P(D|\bar{\Theta}_{n+1})\big] \\ &= \ln\big[\sum_{\Delta} P(D, \Delta|\bar{\Theta}_{n+1})\big] \\ &= \ln\big[\sum_{\Delta} P(\Delta|D, \bar{\Theta}_n)\frac{P(D, \Delta|\bar{\Theta}_{n+1})}{P(\Delta|D, \bar{\Theta}_n)}\big] \end{aligned} \tag{B.2}$$

Since $\ln x$ is a concave function on $(0, \infty)$, it follows from Jensen's inequality that

$$\begin{aligned} L(\bar{\Theta}_{n+1}|D) &\geq \sum_{\Delta} P(\Delta|D, \bar{\Theta}_n) \ln\big[\frac{P(D, \Delta|\bar{\Theta}_{n+1})}{P(\Delta|D, \bar{\Theta}_n)}\big] \\ &= \sum_{\Delta} P(\Delta|D, \bar{\Theta}_n) \ln\big[P(D, \Delta|\bar{\Theta}_{n+1})\big] - \sum_{\Delta} P(\Delta|D, \bar{\Theta}_n) \ln\big[P(\Delta|D, \bar{\Theta}_n)\big] \\ &= Q(\bar{\Theta}_{n+1}|\bar{\Theta}_n) + R(\bar{\Theta}_n|\bar{\Theta}_n), \end{aligned} \tag{B.3}$$

where

$$
\begin{aligned}
Q(\bar{\Theta}_{n+1}|\bar{\Theta}_n) &= \sum_{\Delta} \left(P(\Delta|D, \bar{\Theta}_n) \ln \big[P(D, \Delta|\bar{\Theta}_{n+1}) \big] \right), \\
R(\bar{\Theta}_{n+1}|\bar{\Theta}_n) &= -\sum_{\Delta} \left(P(\Delta|D, \bar{\Theta}_n) \ln \big[P(\Delta|D, \bar{\Theta}_{n+1}) \big] \right).
\end{aligned}
\tag{B.4}
$$

APPENDIX C

OPTIMIZATION OF THE Q-FUNCTION

As is derived in Appendix A, according to Eqs. A.22 and A.23, the Q function for non-sequential measurements in the nth iteration can be written as:

$$\begin{aligned} Q(\Pi, \Theta | \Pi_n, \Theta_n) &= \sum_{i=1}^{N} \sum_{j=1}^{M} P(\delta_i^j = 1 | D = \hat{D}, \Pi_n, \Theta_n) \ln \left[P(\delta_i = j, d_i = \hat{d}_i | \Pi, \Theta) \right] \\ &= \sum_{i=1}^{N} \sum_{j=1}^{M} (h_i^j)_n \ln \left[P(\delta_i = j, \hat{d}_i = d_i | \Pi, \Theta) \right]. \end{aligned} \tag{C.1}$$

where $(h_i^j)_n$ is defined as the expectation of δ_i^j with respect to Δ (see Eq. A.22) evaluated at Π_n, Θ_n with given measurement $\hat{D}$, and is evaluated in Eq. 8.11.

In the case of spatial measurements, according to Eqs. A.31 and A.32, the Q_c function has the additional parameter Z_n and

$$\begin{aligned} Q_c(\Pi, \Theta | \Pi_n, \Theta_n, Z_n) &= \sum_{i=1}^{N} \sum_{j=1}^{M} P(\delta_i^j = 1 | D = \hat{D}, \Pi_n, \Theta_n, Z_n) \ln \left[P(\delta_i = j, d_i = \hat{d}_i | \Pi, \Theta, Z_n) \right] \\ &= \sum_{i=1}^{N} \sum_{j=1}^{M} (h_i^j)_n \ln \left[P(\delta_i = j, \hat{d}_i = d_i | \Pi, \Theta, Z_n) \right]. \end{aligned} \tag{C.2}$$

where $(h_i^j)_n$ is re-defined as the expectation of δ_i^j with respect to Δ (see Eq. A.31) evaluated at Π_n, Θ_n, Z_n with given measurement $\hat{D}$, and is evaluated in Eq. 8.18. For simplicity, in the case of spatial measurement.

C.1 Optimization in Non-Sequential EM Algorithm

As is shown in Appendix A, in the case of non-sequential measurements, the log-likelihood can be guaranteed to be non-decrease if in the nth iteration we have:

$$Q(\Pi_{n+1}, \Theta_{n+1}|\Pi_n, \Theta_n) \geq Q(\Pi_n, \Theta_n|\Pi_n, \Theta_n).$$

To do this, we normally maximize the Q function in the maximization step. The Q function in Eq. C.1 can be further expanded, according to the non-sequential PDF defined in Eqs. 8.6, as

$$\begin{aligned} Q(\Theta, \Pi|\Theta_n, \Pi_n) &= \sum_{i=1}^{N}\sum_{j=1}^{M}(h_i^j)_n \ln\big[P(d_i = \hat{d}_i|\delta_i = j, \Theta, \Pi)P(\delta_i = j|\Theta, \Pi)\big] \\ &= \sum_{i=1}^{N}\sum_{j=1}^{M}(h_i^j)_n\Big\{\ln[G(\hat{d}_i|\Theta_j)] + \ln(\pi_j)\Big\} \\ &= \sum_{i=1}^{N}\sum_{j=1}^{M}(h_i^j)_n\Big\{\ln\big\{\frac{1}{(2\pi)^{\frac{n}{2}}|C_j|^{\frac{1}{2}}}\exp\big[-\frac{(\hat{d}_i - \mu_j)^T C_j^{-1}(\hat{d}_i - \mu_j)}{2}\big]\big\} + \ln(\pi_j)\Big\} \\ &= \sum_{i=1}^{N}\sum_{j=1}^{M}(h_i^j)_n\Big\{\ln(\pi_j) - \ln[(2\pi)^{\frac{n}{2}}|C_j|^{\frac{1}{2}}] - \frac{(\hat{d}_i - \mu_j)^T C_j^{-1}(\hat{d}_i - \mu_j)}{2}\Big\}, \end{aligned} \tag{C.3}$$

where $|C_j|$ is the determinant of C_j. With a known model of (Π_n, Θ_n), the maximizer of the Q function satisfies the following equation:

$$\frac{\partial Q(\Pi, \Theta|\Pi_n, \Theta_n)}{\partial \mu_j} = 0,\ j = 1, M \tag{C.4}$$

$$\frac{\partial Q(\Pi, \Theta|\Pi_n, \Theta_n)}{\partial C_j} = 0,\ j = 1, M \tag{C.5}$$

$$\frac{\partial}{\partial \pi_j}\Big[Q(\Pi, \Theta|\Pi_n, \Theta_n) + \lambda(\sum_{l=1}^{M}\pi_l - 1)\Big] = 0,\ j = 1, M \tag{C.6}$$

Note that in the second equation, the derivative of a scalar function with respect the matrix C_j is a matrix containing the derivative of the function with respect to each entry of C_j ([72]); in the last equation, a Lagrangian multiplier λ is applied.

Denote the model parameter for the $(n+1)$th iteration as:$\{(\pi_j)_{n+1}, (\mu_j)_{n+1}, (C_j)_{n+1}\}$. The optimal choice of $(\mu_j)_{n+1}$ is obtained from Eq. C.4 which is

$$\frac{\partial Q(\Theta|\Theta_n)}{\partial \mu_j} = \sum_{i=1}^{N} (h_i^j)_n C_j^{-1}(\hat{d}_i - \mu_j) = 0,$$

so

$$\sum_{i=1}^{N} (h_i^j)_n (\mu_j)_{n+1} = \sum_{i=1}^{N} (h_i^j)_n \hat{d}_i,$$

and

$$(\mu_j)_{n+1} = \frac{\sum_{i=1}^{N} (h_i^j)_n \hat{d}_i}{\sum_{i=1}^{N} (h_i^j)_n}. \tag{C.7}$$

In order to find $(C_j)_{n+1}$, define

$$(\Upsilon_j)_{n+1} = (C_j)_{n+1}^{-1}.$$

Here, we need to assume that $(C_j)_{n+1}$ is non-singular. To differentiate with respect to Υ_j, we need to use the calculus of matrix ([72]) which gives

$$\frac{\partial \Upsilon_j}{\partial \Upsilon_j} = I, \tag{C.8}$$

and

$$\frac{\partial |\Upsilon_j|}{\partial \Upsilon_j} = |\Upsilon_j| (\Upsilon_j)^{-1}, \tag{C.9}$$

where I is the identity matrix. Then, differentiating Q function with respect to Υ_k gives

$$\begin{aligned} \frac{\partial Q(\Pi, \Theta|\Pi_n \Theta_n)}{\partial \Upsilon_j} &= \sum_{i=1}^{N} (h_i^j)_n \frac{1}{2} \left[\Upsilon_j^{-1} - (\hat{d}_i - \mu_j)(\hat{d}_i - \mu_j)^T \right] = 0, \\ (\Upsilon_j^{-1})_{n+1} = (C_j)_{n+1} &= \frac{\sum_{i=1}^{N} (h_i^j)_n (\hat{d}_i - (\mu_j)_{n+1})(\hat{d}_i - (\mu_j)_{n+1})^T}{\sum_{i=1}^{N} (h_i^j)_n}. \end{aligned} \tag{C.10}$$

The optimal choice of $(\pi_j)_{n+1}$ is obtained from Eq. C.6, i.e.,

$$\frac{\partial}{\partial \pi_j}\big[Q(\Pi,\Theta|\Pi_n,\Theta_n)+\lambda(\sum_{l=1}^{M}\pi_l-1)\big]=\sum_{i=1}^{N}(h_i^j)_n\frac{1}{\pi_j}+\lambda=0. \tag{C.11}$$

By summing both sides of the last equality in Eq. C.11 over the M groups, we obtain

$$\begin{aligned}\sum_{i=1}^{N}\sum_{j'=1}^{M}(h_i^j)_n+\lambda\sum_{j'=1}^{M}\pi_{j'}&=0,\\ N+\lambda\sum_{j'=1}^{M}\pi_{j'}&=0,\\ \lambda&=-N.\end{aligned} \tag{C.12}$$

Using $\lambda=-N$ the last equality in Eq. C.11, we find that

$$(\pi_j)_{n+1}=\frac{1}{N}\sum_{i=1}^{N}(h_i^j)_n. \tag{C.13}$$

To summarize, we have the following solutions for the maximization step:

$$\begin{aligned}(\mu_j)_{n+1}=&\frac{\sum_{i=1}^{N}(h_i^j)_n\hat{d}_i}{\sum_{i=1}^{N}(h_i^j)_n},\\ (C_j)_{n+1}=&\frac{\sum_{i=1}^{N}(h_i^j)_n(\hat{d}_i-(\mu_j)_{n+1})(\hat{d}_i-(\mu_j)_{n+1})^T}{\sum_{i=1}^{N}(h_i^j)_n},\\ (\pi_j)_{n+1}=&\frac{1}{N}\sum_{i=1}^{N}(h_i^j)_n.\end{aligned} \tag{C.14}$$

C.2 Optimization in Spatial EM Algorithm

For the case of spatial measurements, the membership matrix term $h_i^j(\Pi_n,\Theta_n,Z_n)$ in Eq. C.2 is given by Eq. 8.18. According to Eqs. 8.19 and 8.20, and under the assumption we made that d_i is dependent only to δ_i, and δ_i's are conditionally independent (conditional to Z_n), we have:

$$\begin{aligned} P(\delta_i = j, d_i = \hat{d}_i|\Pi, \Theta, Z_n) &= P(\delta_i = j|\Pi, \Theta, Z_n)P(d_i = \hat{d}_i|\delta_i = j, \Pi, \Theta, Z_n) \\ &= C\pi_j G(d_i|\Theta_j)S_j(x_i|r_0, Z_n), \end{aligned} \tag{C.15}$$

where C is a constant value. The Q_c function can be written as:

$$\begin{aligned} Q_c(\Theta, \Pi|\Theta_n, \Pi_n, Z_n) &= \sum_{i=1}^{N}\sum_{j=1}^{M}(h_i^j)_n \ln\left[P(d_i = \hat{d}_i|\delta_i = j, \Theta, \Pi, Z_n)P(\delta_i = j|\Theta, \Pi, Z_n)\right] \\ &= \sum_{i=1}^{N}\sum_{j=1}^{M}(h_i^j)_n\Big\{\ln[G(\hat{d}_i|\Theta_j)] + \ln(\pi_j) + \ln[S_j(x_i|r_0, Z_n)] + \text{Constant}\Big\} \\ &= Q_{c,a}(\Theta, \Pi|\Theta_n, \Pi_n, Z_n) + Q_{c,s}(\Theta, \Pi|\Theta_n, \Pi_n, Z_n), \end{aligned} \tag{C.16}$$

where

$$\begin{aligned} Q_{c,a}(\Theta, \Pi|\Theta_n, \Pi_n, Z_n) &= \sum_{i=1}^{N}\sum_{j=1}^{M}(h_i^j)_n\Big\{\ln[G(\hat{d}_i|\Theta_j)] + \ln(\pi_j)\Big\} \\ Q_{c,s}(\Theta, \Pi|\Theta_n, \Pi_n, Z_n) &= \sum_{i=1}^{N}\sum_{j=1}^{M}(h_i^j)_n\Big\{\ln[S_j(x_i|r_0, Z_n)] + \text{Constant}\Big\} \end{aligned} \tag{C.17}$$

In Eq. C.17, the $Q_{c,a}$ function is same as the Q function Eq. C.3 except that the definition of membership matrix term $(h_i^j)_n$ is spatial (Eq. 8.18). The $Q_{c,s}$ function is not related to the parameter Θ and Π. Hence, maximizing Q_c function shares exactly same procedure of maximizing Q function (Eqs. C.4 to C.14); and the maximizer of the Q_c function should have same form as Eq. C.14, except that the membership matrix term $(h_i^j)_n$ includes a spatial term (Eq. 8.18).

APPENDIX D

MEASUREMENT ERROR ESTIMATION AND THE QUADRATIC FITTING

After the data are divided into groups, smoothing is done within each group to obtain an estimate of the true signal. A moving-window average is the simplest method for smoothing, however, the weight on each data point within the window can be *ad hoc*. As we mentioned earlier, this method is accurate when the data are a linear function of spatial (or temporal) coordinate. If not, a better approximation of the true signal is to fit the data with a curve (1-D problem) or with a surface (2-D problem). For surface-fitting, we apply a generalized least square to fit the data with a quadratic surface.

D.1 Generalized Least Squares Fitting

Suppose there are N_0 observations $(\hat{d}_i, i = 1, N_0)$ in a selected window of a group. The coordinate of each d_i is (x_i, y_i). To fit these N_0 data points with a quadratic surface, we have the following expression

$$\begin{aligned} d_i = \beta_1 x_i^2 + \beta_2 y_i^2 + \beta_3 x_i y_i + \beta_4 x_i + \beta_5 y_i + \beta_6 + e_i \\ i = 1, \ldots, N_0, \end{aligned} \tag{D.1}$$

where e_i is the measurement error.

In matrix form, Eq. D.1 becomes

$$\begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_{N_0} \end{pmatrix}, = \begin{pmatrix} x_1^2 & y_1^2 & x_1 y_1 & x_1 & y_1 & 1 \\ x_2^2 & y_2^2 & x_2 y_2 & x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{N_0}^2 & y_{N_0}^2 & x_{N_0} y_{N_0} & x_{N_0} & y_{N_0} & 1 \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \\ \beta_5 \\ \beta_6 \end{pmatrix} + \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_{N_0} \end{pmatrix} \tag{D.2}$$

or

$$Y = A\beta + E, \tag{D.3}$$

where Y is the column vector with N_0 entries given by

$$Y = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_{N_0} \end{pmatrix}. \tag{D.4}$$

A is an $N_0 \times 6$ matrix and, according to Eq. D.2, is given by

$$A = \begin{pmatrix} x_1^2 & y_1^2 & x_1 y_1 & x_1 & y_1 & 1 \\ x_2^2 & y_2^2 & x_2 y_2 & x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{N_0}^2 & y_{N_0}^2 & x_{N_0} y_{N_0} & x_{N_0} & y_{N_0} & 1 \end{pmatrix}, \tag{D.5}$$

and β is a column vector with 6 entries for a quadratic fit, i.e.

$$\beta = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \\ \beta_5 \\ \beta_6 \end{pmatrix}. \tag{D.6}$$

E is a column vector with N_0 entries.

$$E = \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_{N_0} \end{pmatrix}. \tag{D.7}$$

The known parameters in Eq. D.3 are Y and A, and the unknowns are β and E. The generalized least square solution to β is

$$\beta = (A^T A)^{-1}(A^T Y). \tag{D.8}$$

The estimated error is calculated by

$$E = Y - A\beta. \tag{D.9}$$

To calculate β from Eq. D.8, the matrix $A^T A$ has to be invertible (positive definite). If the above condition is not satisfied, the data are fit with a plane.