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

THE GRADUATE SCHOOL

PARAMETERIZATION AND DATA ASSIMILATION FOR COMPLEX GEOLOGICAL RESERVOIRS

by
Yu Zhao

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

The Graduate School
The University of Tulsa

2019

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

THE GRADUATE SCHOOL

PARAMETERIZATION AND DATA ASSIMILATION FOR COMPLEX GEOLOGICAL RESERVOIRS

by
Yu Zhao

A THESIS

APPROVED FOR THE DISCIPLINE OF

PETROLEUM ENGINEERING

By Thesis Committee

Albert C. Reynolds, Chair
Mustafa Onur
Ovadia Shoham
William Coberly

COPYRIGHT STATEMENT

ABSTRACT

Yu Zhao (Doctor of Philosophy in Petroleum Engineering)

Parameterization and Data Assimilation for Complex Geological Reservoirs

Directed by Albert C. Reynolds

230 pp., Chapter 6: Conclusions

(661 words)

Over the past decade, it is popular to view the assisted history matching (AHM) problem in a traditional Bayesian framework where we commonly assume a multi-Gaussian description of the prior probability density function (PDF) of model parameters. However, this assumption is often insufficient to capture the complexity of realistic geological structures (e.g., fluvial channels and complex facies distributions) that crucially impact subsurface flow responses. For such complex geological systems, the prior and posterior distributions are poorly described by explicit PDFs which makes it challenging to solve the inverse problem in history matching and obtain posterior models that preserve the geological realism. In this work, we investigate the use of parameterization methods with an ensemble-based algorithm for assisted history matching of complex geological reservoirs (specifically, the channelized reservoirs with multiple facies) where the prior PDF violates the multi-Gaussian assumption.

We develop an integrated history matching workflow by combining parameterization methods and the ensemble smoother with multiple data assimilation (ES-MDA) algorithm. After every iteration step, an improved post-processing technique is used to map updated continuous variables to discrete facies indicators by considering the prior facies proportions.

On the parameterization side, we propose a new implementation method of cosine discrete transform (DCT), i.e., common basis DCT, for parameterization of discrete facies type indicators and model reduction. With the common basis DCT, we construct a set of

common basis functions that describe the geological similarities among different models and show that common basis DCT outperforms traditional implementations of DCT in terms of the quality of data match and the preservation of geological continuity. Different from the global features obtained by common basis DCT, we introduce the non-negative matrix factorization (NMF) to extract local features of model parameters by imposing the non-negativity constraints on the components in the basis and coefficient matrices. With NMF, a posterior reservoir model can be generated as a positive linear combination of updated basis vectors. Based on the results from synthetic examples, NMF is also capable of resolving geologic details and preserving the spatial continuity of channelized facies, and is slightly superior to common basis DCT. Motivated by recent progress in tensor decomposition (TD) and non-negative tensor decomposition (NTD), we also investigate the parameterization of facies fields using TD and NTD. Based on results from various cases, TD and NTD perform similarly and the data match results obtained with these methods are better than those obtained by NMF. Although the uncertainty characterization of facies distributions is slightly better with NTD, the computing time of NTD can be as 10 times longer as that of TD. From probability maps of channel facies, it is shown that more channel gridblocks in the true model are correctly resolved by NTD than TD, but the proportion of missed channel gridblocks is also slightly higher when we implement NTD. In order to determine the truncation level for TD and NTD, we develop a scheme to obtain ranks of different modes by balancing the normalized model mismatch and model reduction ratio. The truncated TD and NTD methods are tested on variations of a 3D example and compared comprehensively with TD and NTD without truncation in terms of history data match, uncertainty characterization of channel distributions and computational cost.

To completely preserve the spatial continuity of channel facies distribution represented by object-based modeling, we also propose a workflow to adjust by history matching the parameters used to generate a realization of channelized system by object-based modeling in Petrel. Although this seems like an obvious approach, the results of a sensitivity analysis show that extremely small changes in some of the parameters used in object-based modeling

with Petrel lead to an extremely large change in the facies distribution, i.e., the object-based realization is not a continuous or stable function of the parameters. Due to the internal randomness associated with Petrel modeling process, it is difficult to obtain desirable approximation of gradient by the data assimilation algorithm. Thus, the proposed workflow for history matching of object-based models was not successful.

ACKNOWLEDGEMENTS

I would like to express my most sincere gratitude to my advisor, Prof. Albert C. Reynolds. It has been an honor to be his Ph.D. student. I appreciate his consistent guidance, encouragement, attention to detail, and endless support during my Ph.D. study at the University of Tulsa. Besides the technical skills, through my many interactions with Prof. Reynolds, I learned great work ethic and professionalism.

I specially thank Prof. Mustafa Onur, Prof. Ovadia Shoham and Prof. William Coberly for serving as members of my defense committee. I would like to extend my thanks to all the other faculty members of the McDougall School of Petroleum Engineering for their guidance through my courses of study as a graduate student.

Thanks also to all my friends at TU and I appreciate their help, support and consideration for the last five years. The friendships with them are very important wealth of my life.

I gratefully acknowledge financial support from the member companies of the University of Tulsa Petroleum Reservoir Exploitation Projects (TUPREP). I also would like to thank the graduate school for awarding me the honorable Bellwether fellowship and thank the McDougall School of Petroleum Engineering for providing me the teaching assistantship during the first four years of my Ph.D. program.

Finally, I would like to express my greatest gratitude to my parents and my girlfriend for their continuous and unconditional love and support. This work is dedicated to them and they deserve the appreciation and love from the deepest part of my heart.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Research Background

In the reservoir engineering community, inverse problems arise when the observed data, $\boldsymbol{d}_{\text{obs}}$, are used to calibrate the model parameters, $\boldsymbol{m}$, for the purpose of accurate prediction of reservoir performance (Oliver et al., 2008). This process is usually referred to as assisted history matching (AHM) which is one crucial stage in the closed-loop reservoir management (Jansen et al., 2005, 2009) (see Fig. 1.1). Over the past decade, the main focus of this area has shifted from finding an individual model that best matches the data to a set of models that are consistent with the observed data. This allows the quantification of uncertainty in the posterior models and production prediction in order to handle the risks in decision making and reservoir management.

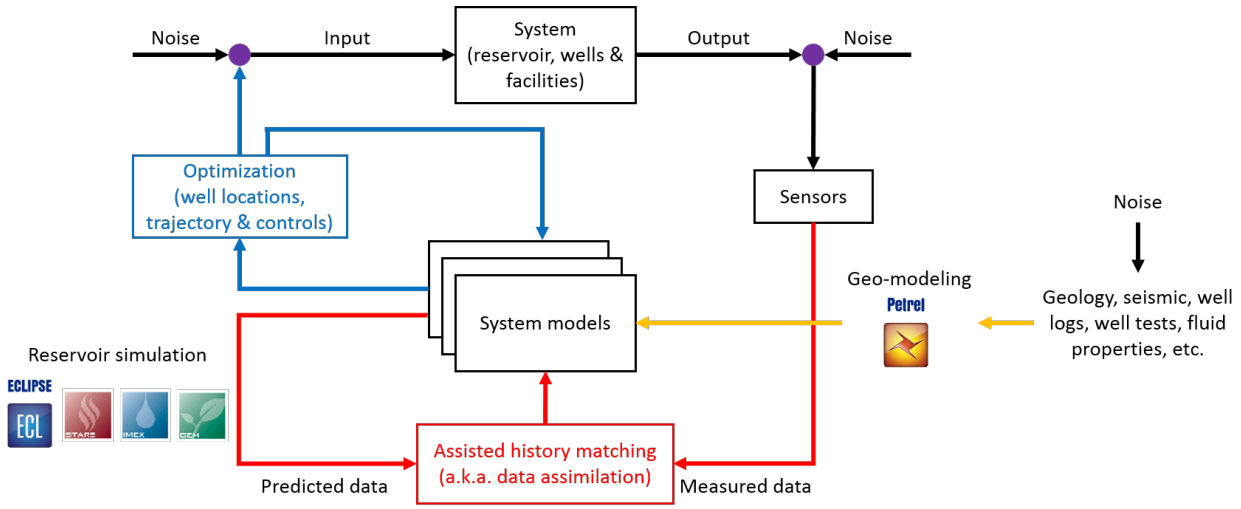


Figure 1.1: Closed-loop reservoir management

For uncertainty quantification, it is convenient to view the inverse problems in a traditional Bayesian framework (Tarantola, 2005; Oliver et al., 2008) where the posterior

probability density function (PDF) of the model parameters, $f(\boldsymbol{m}|\boldsymbol{d}_{\rm obs})$, can be written as

$$f(\boldsymbol{m}|\boldsymbol{d}_{\rm obs}) = \frac{f(\boldsymbol{d}_{\rm obs}|\boldsymbol{m})f(\boldsymbol{m})}{f(\boldsymbol{d}_{\rm obs})} = aL(\boldsymbol{m}|\boldsymbol{d}_{\rm obs})f(\boldsymbol{m}), \tag{1.1}$$

where $\boldsymbol{m}$ is the N_m-dimensional vector of model parameters, $\boldsymbol{d}_{\rm obs}$ is the N_d-dimensional vector of observed data, $f(\boldsymbol{m})$ is the prior PDF of model parameters and $f(\boldsymbol{d}_{\rm obs})$ is the PDF of observed data. Most importantly, $f(\boldsymbol{d}_{\rm obs}|\boldsymbol{m})$ is the conditional PDF of $\boldsymbol{d}_{\rm obs}$ given $\boldsymbol{m}$, which corresponds to the likelihood function of model parameters, $L(\boldsymbol{m}|\boldsymbol{d}_{\rm obs})$, once $\boldsymbol{d}_{\rm obs}$ is given. a is the normalizing constant.

If we assume a multi-Gaussian description of the prior PDF of parameters, then $f(\boldsymbol{m})$ is given by

$$f(\boldsymbol{m}) = \frac{1}{(2\pi)^{N_m/2}\sqrt{\det \boldsymbol{C}_{\rm M}}} \exp\left[-\frac{1}{2}(\boldsymbol{m}-\boldsymbol{m}_{\rm prior})^{\rm T}\boldsymbol{C}_{\rm M}^{-1}(\boldsymbol{m}-\boldsymbol{m}_{\rm prior})\right], \tag{1.2}$$

where $\boldsymbol{C}_{\rm M}$ is the $N_m \times N_m$ prior covariance matrix of model parameters and $\boldsymbol{m}_{\rm prior}$ is the prior mean of model parameters. Similarly, with the assumption of Gaussian-distributed measurement errors, we can obtain a Gaussian likelihood function as

$$L(\boldsymbol{m}|\boldsymbol{d}_{\rm obs}) = \frac{1}{(2\pi)^{N_d/2}\sqrt{\det \boldsymbol{C}_{\rm D}}} \exp\left[-\frac{1}{2}(\boldsymbol{d}_{\rm obs}-\boldsymbol{g}(\boldsymbol{m}))^{\rm T}\boldsymbol{C}_{\rm D}^{-1}(\boldsymbol{d}_{\rm obs}-\boldsymbol{g}(\boldsymbol{m}))\right], \tag{1.3}$$

where $\boldsymbol{C}_{\rm D}$ is the $N_d \times N_d$ covariance matrix of measurement errors and $\boldsymbol{g}(\boldsymbol{m})$ denotes the vector of predicted data obtained by running reservoir simulation given the model parameters, $\boldsymbol{m}$.

By substituting Eqs. 1.2 and 1.3 to Eq. 1.1, we can rewrite the Bayesian posterior PDF, $f(\boldsymbol{m}|\boldsymbol{d}_{\rm obs})$, as

$$f(\boldsymbol{m}|\boldsymbol{d}_{\rm obs}) = a\, \exp\left\{-O(\boldsymbol{m})\right\}, \tag{1.4}$$

where a is the normalizing constant and $O(\boldsymbol{m})$ is called the objective function defined by

$$O(\boldsymbol{m}) = O_m(\boldsymbol{m}) + O_d(\boldsymbol{m}), \tag{1.5}$$

with the model mismatch term given by

$$O_m(\boldsymbol{m}) = \frac{1}{2}(\boldsymbol{m} - \boldsymbol{m}_{\text{prior}})^{\text{T}} \boldsymbol{C}_{\text{M}}^{-1} (\boldsymbol{m} - \boldsymbol{m}_{\text{prior}}), \tag{1.6}$$

and the data mismatch term given by

$$O_d(\boldsymbol{m}) = \frac{1}{2}(\boldsymbol{d}_{\text{obs}} - \boldsymbol{g}(\boldsymbol{m}))^{\text{T}} \boldsymbol{C}_{\text{D}}^{-1} (\boldsymbol{d}_{\text{obs}} - \boldsymbol{g}(\boldsymbol{m})). \tag{1.7}$$

In the linear case where the predicted data, $\boldsymbol{d} = \boldsymbol{g}(\boldsymbol{m})$, is a linear function of model parameters, it is straightforward to obtain an explicit posterior PDF, $f(\boldsymbol{m}|\boldsymbol{d}_{\text{obs}})$, which is also Gaussian; see Oliver et al. (2008). However, the linear assumption is not valid for practical history matching problems.

With the above equations, the history matching problem can be transformed from maximizing the posterior PDF, $f(\boldsymbol{m}|\boldsymbol{d}_{\text{obs}})$, into minimizing the objective function, $O(\boldsymbol{m})$. In other words, the aim of history matching is to find models which honor the heterogeneity of the prior models (low value of the model mismatch term, $O_m(\boldsymbol{m})$) and match the observed data (low value of the model mismatch term, $O_d(\boldsymbol{m})$). Since $\boldsymbol{m}_{\text{prior}}$ usually corresponds to a smooth model, a high value of $O_m(\boldsymbol{m})$ often indicates that we obtain very rough posterior models where there are large number of gridblocks containing unrealistically low (undershooting) or high (overshooting) values of petrophysical properties. On the other hand, if an estimate $\boldsymbol{m}$ gives a high value of $O_d(\boldsymbol{m})$, it means that we obtain a poor data match. Therefore, we focus on minimizing the value of $O(\boldsymbol{m})$ through history matching in order to yield models that give high posterior probability, $f(\boldsymbol{m}|\boldsymbol{d}_{\text{obs}})$.

It is also noteworthy that the number of model parameters is much larger than the number of independent observed data which makes the inverse problem ill-conditioned. So the model mismatch term can be regarded as a regularization term which not only reduces the condition number but constrains the posterior models to honor the prior geological knowledge of the reservoir.

Assuming a multi-Gaussian prior PDF is often insufficient to capture the complex-

ity of realistic geological structures that crucially impact subsurface flow responses (Journel and Zhang, 2006; Kerrou et al., 2008). In reality, most spatial phenomena present strong non-rectilinear shapes and connectivity, e.g., fluvial channels and complex litho-facies distributions. For such complex geological systems, the prior and posterior distributions are poorly described by explicit PDFs which makes it very challenging to solve the inverse problem of characterizing the posterior PDF of Eq. 1.1 and obtain posterior models that preserve the geological realism. This research deals with several technical issues related to the inverse modeling and assisted history matching of complex geological reservoirs (specifically, the multi-facies channelized reservoirs) where $f(\boldsymbol{m})$ violates the multi-Gaussian assumption.

1.2 Literature Review on Geological Modeling Techniques

Modeling subsurface heterogeneity, i.e., constructing $f(\boldsymbol{m})$, is one crucial step for the exploitation of subsurface resources. For more than half a century, there has been significant development progress of subsurface modeling based on geostatistics, and a large suite of tools is available for generating realistic geological models. Traditionally, the subsurface heterogeneity has been mostly modeled using the variogram-based approaches (Goovaerts, 1997; Deutsch and Journel, 1998) which follow the multi-Gaussian spatial framework. Although they are still very popular techniques to model geological heterogeneity (Falivene et al., 2006; Lee et al., 2007; Refsgaard et al., 2014), they cannot describe curvilinear features, such as fluvial channels (Strebelle, 2002) and more complex structures. The truncated pluri-Gaussian simulation (TPS) method (Emery, 2007; Armstrong et al., 2011) addresses this issue by modeling two (or more) random Gaussian fields with underlying variograms. The categorical field is obtained by applying the predefined truncation rules to the multi-Gaussian fields. This method is capable of generating complex facies distributions but one problem is that it is not easy to determine the spatial relations between various facies and the truncation rules a priori. Moreover, the approach has not proved successful for channelized systems (Zhao et al., 2008). On the basis of TPS, Sebacher et al. (2017) developed the adaptive pluri-Gaussian simulation (APS) method by incorporating facies probability

fields and assigning a particular truncation map to each grid. Their results show that the APS method is able to retain more variability in the initial ensemble compared to the traditional TPS method but the critical facies probability fields must be provided by professional experts beforehand.

In the last 20 years, a new class of approaches based on multi-point statistics (MPS) has emerged and gained popularity in the community (Guardiano and Srivastava, 1993; Strebelle, 2002; Caers and Zhang, 2004; Mariethoz and Caers, 2014). It extracts multiple-point patterns from a training image (TI) which contains conceptual geological knowledge (Journel and Zhang, 2006) and store the number of occurrences of each pattern in a search tree. During the simulation, the search tree is used to estimate conditional probabilities and generate spatial continuity model with the patterns that are compatible with available data. The first successful simulation algorithm based on this idea is SNESIM (Strebelle, 2002) and several improved methods have been developed later on to be more efficient and better at reproducing patterns in the training image (Hu and Chugunova, 2008; Tahmasebi et al., 2012; Straubhaar et al., 2013; Strebelle and Cavelius, 2014). Training image based methods can be applied to all kinds of geological environments and it is straightforward to account for hard and soft data (Journel, 2002; Liu et al., 2004; Remy et al., 2009). Although obtaining a suitable training image still remains a challenging task, this type of approach is suggested to be viable for modeling complex geological structures (Linde et al., 2015).

The object-based methods (Deutsch and Wang, 1996; Nordahl and Ringrose, 2008; Ramanathan et al., 2010) provide another way to describe the complex geological features. This type of method consists of pre-defining a set of geologic bodies or objects (e.g., fluvial channels) with orientation, amplitude, width, thickness, presence of levee and so on. The nature of the relationship between objects can also be specified to assemble individual parts together and place them in a domain according to prescribed constraints (e.g., facies proportions) in order to create complete geological structures. One advantage of this approach is that it can handle different geological environments and generate highly realistic representations of subsurface heterogeneity in an efficient way. Among several of the most-used

geological modeling techniques (i.e., two-point statistics, multi-point statistics, object-based, surface-based and process-based) illustrated in Fig. 1.2, the object-based method is thought to achieve a good balance between numerical efficiency, geological realism and conditioning capabilities (Bertoncello, 2011). However, there are usually a large number of parameters that need to be determined and the methods sometimes have difficulty in conditioning to hard data (e.g., well measurements). Currently, a common application for object-based methods is to generate training images for multiple-point statistics.

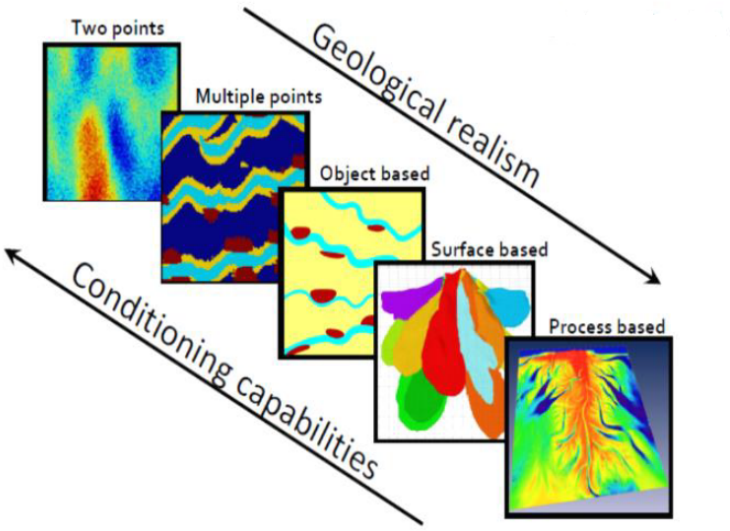


Figure 1.2: Geological modeling techniques (Bertoncello, 2011)

The current trend to generate complex geological models, i.e., multi-facies channelized reservoirs in this work, is to employ MPS algorithm associated with reliable training images or to use object-based methods directly. With these more advanced techniques, large-scale geologic structures, such as a braided fluvial system, can be created efficiently and then the petrophysical properties (e.g., porosity, permeability, net-to-gross ratio, etc.) within distinct facies are simulated by two-point spatial statistics given corresponding variograms or covariance functions. Although there are other geological modeling methods reported by papers in hydrogeology, such as process-based modeling (Koltermann and Gorelick, 1992; Paola, 2000; Nicholas et al., 2013) which solves partial differential equations that describe geologic processes, these techniques are not considered here but a discussion can be found in a recent review paper of Linde et al. (2015).

1.3 Literature Review on Model Parameterization

After producing geologically realistic subsurface models using the approaches described in the previous section, we usually need to parameterize the model in order to solve the inverse problems. Given the original model $\boldsymbol{m}$ (vector of model parameters) of dimension $N_{\boldsymbol{m}}$, the goal of parameterization is to find an appropriate model vector $\hat{\boldsymbol{m}}$ of dimension $N_{\hat{\boldsymbol{m}}}$ with $N_{\hat{\boldsymbol{m}}} < N_{\boldsymbol{m}}$ that describes the most salient features of the original model. Within the multi-Gaussian framework, parameterization can be achieved using schemes such as pilot points (RamaRao et al., 1995), self-sequential calibration (Gómez-Hernández et al., 1997) and gradual deformation (Hu, 2000). With the TPS or APS method (Liu and Oliver, 2004; Sebacher et al., 2017), the geologic facies distribution can be mapped to Gaussian random variables given threshold maps. This type of parameterization is straightforward but the truncation rules or facies probability maps need to be provided by professional experts a priori. A more useful class of approaches relies on feature extraction and model reduction techniques. As a linear version of Karhunen-Loeve transform (KLT), the principal component analysis (PCA) has been used broadly to achieve model reduction by applying the singular value decomposition (SVD) or truncated SVD (TSVD) algorithm to the covariance matrix of the Gaussian random fields (Gavalas et al., 1976). However, when it comes to the complex geological structures characterized by advanced spatial statistics, PCA tends to distort the PDF of prior models so that the critical geological features cannot be preserved in the posterior realizations. To overcome this issue, Sarma et al. (2007, 2008) proposed a procedure to reproduce complex geology by preserving the high-order statistics of random fields using kernel PCA (KPCA). With the "kernel trick", the method is more flexible to do the transform in a feature space but the corresponding back-transformation to the original physical space (termed "pre-image" problem) is not guaranteed to have a unique solution. Recently, Vo and Durlofsky (2014, 2015) developed an optimization-based PCA (OPCA) parameterization for complex geological models. To be specific, an optimization problem involving regularization terms and bound constraints is solved with the standard PCA algorithm to yield a low-dimensional representation of the non-Gaussian features. They also

applied a similar idea to the kernel based PCA and proposed the regularized KPCA (R-KPCA) (Vo and Durlofsky, 2016). By combining the PCA and truncated-pluri-Gaussian, Chen et al. (2016) came up with the Pluri-PCA method for multi-facies systems and the results showed potential for large-scale history matching applications.

Inspired by the advances in sparse coding and compressed sensing theory, Khaninezhad et al. (2012a,b) used the K-SVD algorithm (Aharon et al., 2006) to construct sparse geologic dictionaries from the prior ensemble, and they found that the basis elements obtained with the K-SVD always capture the dominant large-scale features. On the other hand, when using the SVD-based (PCA-based) algorithms mentioned above, only the leading basis elements retain the principal geological features while the non-leading bases only contain trivial details which can be regarded as high frequency noise. With a field case Khaninezhad and Jafarpour (2015) further demonstrated the superior capability of K-SVD over SVD-based algorithms in parameterizing the complex geological structures. However, both traditional SVD-based methods and K-SVD algorithm require vectorization of multi-dimensional models.

Tensor (i.e., multidimensional array) factorization methods (De Lathauwer et al., 2000a,b; Kolda and Bader, 2009) are the natural extension of the matrix factorization for decomposition of high-dimensional data sets, that extract various low-dimensional features hidden in different dimensions of the data tensor. Afra and Gildin (2013, 2016) introduced a tensor-based technique with higher order singular value decomposition (HOSVD) for parameterizing the permeability fields of channelized reservoirs. On the basis of their work, Sebacher and Hanea (2018) proposed implementation of the normal score transform for parameterization of facies fields and then applied HOSVD to the tensor defined by the parameter fields.

Jafarpour and McLaughlin (2007, 2009b) first introduced the discrete cosine transform (DCT) to parameterize non-Gaussian variables. This type of method has been widely used for many applications like pattern recognition and image compression (Rao and Yip, 1990; Gonzalez and Woods, 2002). Compared to the SVD-based methods, the DCT is more

computationally efficient and has comparable accuracy in the aspects of feature extraction and energy compression. By retaining a small number of low-frequency basis vectors which correspond to large DCT coefficients, key characteristics of the original model are preserved and the dimensionality of the model vector is significantly reduced as well. However, one major drawback of the traditional DCT algorithm is that it is only implemented for single model without considering the mutual geological patterns throughout a set of realizations. Similar to the predefined cosine bases used in the DCT, the parameterization approaches that involve Fourier bases or wavelets, e.g., discrete wavelet transform (DWT) (Sahni and Horne, 2005; Jafarpour, 2011; Gentilhomme et al., 2015), are also developed to extract localized features of the geological model. One important property of wavelets is its space-frequency localization since the spatial resolution of wavelet bases are different at different frequencies. Thus, the geological features at different scales can be effectively represented using wavelets.

One unique set of parameterization approaches is based on the level set functions. Moreno and Aanonsen (2007) first introduced the level set method originally proposed by Osher and Sethian (1988) to model the boundaries of categorical facies. By using the level set function, the facies distribution can be parameterized as the signed distance to a given surface (Chang et al., 2010; Chang and Zhang, 2014; Ping and Zhang, 2014).

In very recent years, the great success of deep learning (DL) in different areas (e.g., computer vision, natural language processing, etc.) has inspired applications to the solution of inverse problems in the geosciences. Several researchers have attempted to parameterize the complex geological models using advanced deep learning techniques, such as the autoencoder and its variants (Canchumuni et al., 2017, 2018) and the generative adversarial network (GAN) (Chan and Elsheikh, 2017, 2018). Because the DL techniques usually involve a large number of hyperparameters, the number of training samples is correspondingly very high (i.e., $> 10,000$) which makes the training process computationally demanding.

1.4 Literature Review on History Matching Complex Geological Models

Over the past decades, great progress has also been made on the data assimilation

and assisted history matching methods for solving the inverse problem (Eq. 1.1) in order to obtain the posterior subsurface models that not only honor the observed data but preserve the complex geological realism at the same time.

In the Bayesian inverse framework, the sampling methods aim at approximating $f(\boldsymbol{m}|\boldsymbol{d}_{\text{obs}})$ by drawing random samples from this distribution. The Markov chain Monte Carlo (MCMC) methods, such as Metropolis sampler (Metropolis et al., 1953; Hastings, 1970) or Gibbs sampler (Geman and Geman, 1984), can be used to sample from the posterior distribution rigorously. However, a very long Markov chain is usually required for convergence so the direct application of the MCMC to practical problems has not been feasible until very recently due to the work of Sousa and Reynolds (2019) who replaced the reservoir simulator by a support vector regression proxy model when evaluating the Metropolis-Hastings acceptance probability. Alternatively, Kitanidis (1995) and Oliver et al. (1996) introduced the randomized maximize likelihood (RML) method to generate approximate conditional realizations. RML has been successfully applied to reservoir history matching problems (Gao et al., 2006; Tavakoli and Reynolds, 2011). A limitation of the RML method is that the efficient calculation of gradients requires the adjoint method, which is not commonly available in most commercial reservoir simulators. Moreover, although RML does generate samples around modes of the posterior PDF, it only provides an approximate characterization of this PDF.

Due to the lack of an adjoint module in commercial simulators, the ensemble-based stochastic methods, such as ensemble Kalman filter (EnKF) (Evensen, 1994) and ensemble smoother (ES) (van Leeuwen and Evensen, 1996) have become very attractive and popular in the field of reservoir history matching in the last decade. The EnKF updates the model parameters and state variables (e.g., pressure, saturations) by assimilating the observed data sequentially in time. It does not require an adjoint gradient and often performs well in practical applications (Bianco et al., 2007; Evensen et al., 2007; Aanonsen et al., 2009) provided the initial ensemble is chosen with some care. However, the simultaneous updating of reservoir state variables and parameters with EnKF is theoretically valid only if the

updated states are statistically consistent with updated model parameters. There exists cases where the strong nonlinearity introduces inconsistencies between updated states and parameters that causes EnKF to perform poorly (Thulin et al., 2007; Wang et al., 2010). Unlike the EnKF, the ES method assimilates all data simultaneously and updates only the reservoir model parameters to avoid the inconsistency problem. However, it is reported that the ES often results in a worse data match than is obtained with EnKF (Emerick and Reynolds, 2013b). In order to further improve the performance of EnKF and ES, several iterative ensemble-based methods have been proposed (Gu and Oliver, 2007; Li and Reynolds, 2009; Chen and Oliver, 2013; Emerick and Reynolds, 2013a; Luo et al., 2015). Chen and Oliver (2013) developed the LM-ensemble randomized maximum likelihood (LM-EnRML) based on the Levenberg-Marquardt (LM) algorithm. They avoided the explict computation of the sensitivity matrix and used the LM damping parameter, λ, to improve the numerical stability. But according to Le et al. (2015a), the performance of LM-EnRML can be sensitive to the choice of λ and a poor initial choice of λ may give rise to a poor history match in some cases. Another popular iterative algorithm is the ensemble smoother with multiple data assimilation (ES-MDA) developed by Emerick and Reynolds (2012, 2013a). Motivated by the analogy between ES and one Gauss-Newton iteration (Reynolds et al., 2006), the ES-MDA assimilates the same set of observed data multiple times with an inflated covariance matrix of the measurement errors so that it avoids the statistical inconsistency between parameters and states, and appears to outperform both the ES and the EnKF (Emerick and Reynolds, 2013b).

Due to complex geological deposition process, there are always distinct rock bodies or facies distributed in the fluvial or channelized reservoirs while the rock properties usually change abruptly from one facies to another. For such a highly non-Gaussian multi-facies problem, it is well recognized that the direct applications of the ensemble-based methods often result in a degraded performance where the posterior models are not able to preserve the plausible geological and geostatistical features of the categorical facies if one uses gridblock porosities and permeabilities as the model parameters. This is because the ensemble-based

methods only work exactly correctly for linear Gaussian problems as the ensemble size goes to infinity (Thulin and Nævdal, 2006), and they are suitable for the estimation of continuous (in particular, multi-Gaussian) properties rather than the discrete variables (e.g., facies type indicator). A common problem-solving strategy is: (1) convert the discrete variables to continuous parameters with previously illustrated parameterization approaches, (2) apply an ensemble-based method to update the parameters by assimilating the observed data, (3) perform back-transformation to the original physical space. There is a large amount of published papers regarding the combination of various parameterization approaches and the ensemble-based methods, such as TPS with EnKF (Liu and Oliver, 2005a,b; Agbalaka and Oliver, 2008), PCA-based methods with EnKF or variants of ES (Sarma and Chen, 2009, 2013; Chen et al., 2014; Emerick, 2017), DCT or DWT with EnKF (Jafarpour and McLaughlin, 2007, 2009b), level set method with EnKF (Chang et al., 2010; Chang and Zhang, 2014) and DL methods with ES-MDA (Canchumuni et al., 2017, 2018).

In most cases, the back-transformation usually gives a continuous field rather than a discrete distribution of facies, and the spatial continuity of geological structures (e.g., sand channels) may not be maintained. Thus, either the truncation operation (Vo and Durlofsky, 2014, 2015) or the spatial clustering techniques (e.g., neighborhood expectation maximization (Han et al., 2011; Le et al., 2015b)) have been employed to rebuild the facies boundaries and enhance the connectivity of geological structures. But these post-processing techniques cannot guarantee the posterior models preserve the geological realism shown in the prior ensemble. Some researchers have developed methods to integrate the geological simulation techniques (e.g., MPS or object-based method) into the history matching process. Caers and Hoffman (2006) proposed the probability perturbation method (PPM) which takes a linear combination of two probability fields to obtain a single probability field that is then used as soft data to guide MPS simulations. In the process of perturbing the pre-posterior probability and generating new realizations, the lack of feedback information from the flow data leads to very low computational efficiency of the PPM approach. Based on this observation, Jafapour and Khodabakhsi (2011) developed the probability conditioning method (PCM)

which calculates the log-permeability ensemble mean after each update step in order to generate a facies probability map. The probability map is used as soft data in the MPS algorithm for the subsequent conditional resampling. The similar idea has been adapted in other researches to ensure the posterior models are consistent with the prior geologic information presented in the training image (Tavakoli et al., 2014; Le et al., 2015b; Sebacher et al., 2016). Elsheikh et al. (2013, 2015) investigated an efficient sampling algorithm known as nested sampling (NS) for channelized reservoirs. They showed that the estimated Bayesian evidence by the NS algorithm can be used for prior model selection. Tavakoli et al. (2014, 2015) combined the model classification with multidimensional scaling (MDS) and the EnKF to rapidly update models of a channelized reservoir. For this class of methods, the variability of ensemble members can be maintained when assimilating a large number of observed data without suffering the ensemble collapse problem. However, the objective function usually oscillates after the resampling procedure and the computational cost would be an issue for application to large-scale real field cases.

1.5 Research Objectives and Dissertation Outline

1.5.1 Research Objectives

The main objectives of this research are as follows:

1. Develop parameterization approaches that can be reliably used for categorical variables (i.e., facies type), so that the number of model parameters is reduced but the non-Gaussian geological and geostatistical features are preserved.

2. Develop efficient and robust ensemble-based history matching workflow for complex geological systems (i.e., multi-facies channelized reservoirs) for simultaneous calibration of the non-Gaussian facies and the petrophysical properties by assimilating various types of data.

3. Combine ensemble-based method with object-based geological modeling technique for assisted history matching of data from complex geological systems.

1.5.2 Dissertation Outline

This dissertation includes five chapters. In Chapter 2, we present an improved discrete cosine transform method, named common basis DCT, and compare it with traditional DCT parameterizations. In Chapter 3, we introduce a new parameterization approach based on non-negative matrix factorization (NMF) and develop an ensemble-based history matching workflow for simultaneous calibration of the facies distributions and the petrophysical properties within each facies. The performance of NMF is compared with other popular methods when applied to synthetic 2D and 3D reservoir problems. In Chapter 4, we thoroughly investigate the properties of tensor and non-negative tensor decomposition for facies parameterization, and extend their applications to the history matching of complex 3D cases. In Chapter 5, we combine the ensemble-based data assimilation algorithm with object-based modeling technique for history matching of complex geological reservoirs. Chapter 6 presents our conclusions and summarizes the research outcome of this work. Appendices regarding data assimilation algorithm and tensor decomposition are given at the end of this dissertation.

CHAPTER 2

ASSISTED HISTORY MATCHING OF CHANNELIZED RESERVOIRS USING ES-MDA WITH COMMON BASIS DCT

Given the complexity of channelized reservoir system, it is desirable to compactly represent key geological structures from a prior ensemble of representative and plausible reservoir models. In this chapter, we first introduce a parameterization algorithm referred to as common basis DCT. By selecting a set of common basis functions which are the same for all ensemble members, it is shown that the common basis DCT is able to capture the geological similarities among different ensemble models and describe the complex geological connectivity of a channelized system. Through assimilating the observed data and updating the DCT coefficients using the ES-MDA algorithm, the facies field of each realization is calibrated with the updated coefficients and the common basis set. Instead of using a resampling procedure to preserve the spatial continuity of geological structures, an improved post-processing technique based on a regularization framework is then employed at the end of each iteration in order to retain the discrete nature of the facies field and obtain the corresponding reservoir properties fields.

This chapter is organized as follows: we first briefly review the DCT technique and show the implementation details of common basis DCT in next section. Then the outline of the ES-MDA algorithm with common basis DCT is presented followed by the description of the post-processing procedure. In the subsequent sections, the computational results of two synthetic examples are shown and discussed.

2.1 Discrete Cosine Transform (DCT)

The discrete cosine transform (DCT) is a Fourier-based transform which has been

widely used for many applications like pattern recognition and image compression (Rao and Yip, 1990; Gonzalez and Woods, 2002), or dimensionality reduction in optimization and history matching (Jafarpour and McLaughlin, 2009a,b; Bhark et al., 2010). With DCT, the number of model parameters of a history matching problem can be considerably reduced by only retaining the low frequency components of reservoir properties which are sensitive to production data. The most common forward DCT of a 2D input field $\boldsymbol{u}(m,n)$ can be written as (Rao and Yip, 1990):

$$\boldsymbol{v}(\mu,\omega) = \frac{\alpha(\mu)\,\alpha(\omega)}{\sqrt{N_x N_y}} \sum_{m=0}^{N_x-1} \sum_{n=0}^{N_y-1} \boldsymbol{u}(m,n) \cos\left[\frac{\pi(2m+1)\mu}{2N_x}\right] \cos\left[\frac{\pi(2n+1)\omega}{2N_y}\right], \qquad (2.1)$$

$$\alpha(j) = \begin{cases} \frac{1}{\sqrt{2}} & j = 0 \\ 1 & \text{otherwise} \end{cases}$$

where $\mu = 0, \ldots, N_x - 1$ and $\omega = 0, \ldots, N_y - 1$.

Similarly, the inverse 2D DCT is given by (Rao and Yip, 1990):

$$\boldsymbol{u}(m,n) = \frac{2}{\sqrt{N_x N_y}} \sum_{\mu=0}^{N_x-1} \sum_{\omega=0}^{N_y-1} \alpha(\mu)\,\alpha(\omega)\,\boldsymbol{v}(\mu,\omega) \cos\left[\frac{\pi(2m+1)\mu}{2N_x}\right] \cos\left[\frac{\pi(2n+1)\omega}{2N_y}\right],$$

$$\alpha(j) = \begin{cases} \frac{1}{\sqrt{2}} & j = 0 \\ 1 & \text{otherwise} \end{cases} \qquad (2.2)$$

where $m = 0, \ldots, N_x - 1$ and $n = 0, \ldots, N_y - 1$.

Actually, the implementation of DCT can be represented in a vector form. After computing the coefficient matrix $\boldsymbol{v}(\mu,\omega)$ using Eq. 2.1, we can rewrite it as a $N_x N_y$-dimensional vector $\boldsymbol{v} = [v_1, v_2, \cdots, v_{N_x N y}]^{\mathrm{T}}$. In order to obtain the corresponding basis function matrix $\boldsymbol{\Phi} \in \mathbb{R}^{N_x N_y \times N_x N_y}$, Eq. 2.2 can be expressed in another way as follows:

$$\boldsymbol{u} = \boldsymbol{\Phi}\boldsymbol{v} = \left[\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \cdots, \boldsymbol{\phi}_N, \cdots, \boldsymbol{\phi}_{N_x N_y}\right] \left[v_1, v_2, \cdots, v_N, \cdots, v_{N_x N_y}\right]^{\mathrm{T}}, \qquad (2.3)$$

where $\boldsymbol{u}$ is a N_xN_y-dimensional vector and $\boldsymbol{\phi}_j$ is the jth column of the $N_xN_y \times N_xN_y$ basis matrix $\boldsymbol{\Phi}$. In Eq. 2.3, the subscript N denotes the coefficient index from which it is straightforward to compute the corresponding row/column index, μ and ω, respectively. Therefore, the basis function (vector) $\boldsymbol{\phi}_N$ which is a N_xN_y-dimensional column vector, corresponding to the coefficient v_N is given by

$$\boldsymbol{\phi}_N = \frac{2\alpha\left(\mu\right)\alpha\left(\omega\right)}{\sqrt{N_xN_y}} \begin{bmatrix} \cos\left[\frac{\pi(2\times 0+1)\mu}{2N_x}\right]\cos\left[\frac{\pi(2\times 0+1)\omega}{2N_y}\right] \\ \vdots \\ \cos\left[\frac{\pi(2\times(N_x-1)+1)\mu}{2N_x}\right]\cos\left[\frac{\pi(2\times(N_y-1)+1)\omega}{2N_y}\right] \end{bmatrix}_{N_xN_y\times 1}, \tag{2.4}$$

$$\text{for } \mu = 0,\ldots,N_x-1 \text{ and } \omega = 0,\ldots,N_y-1,$$

where $N = \omega N_x + \mu + 1$.

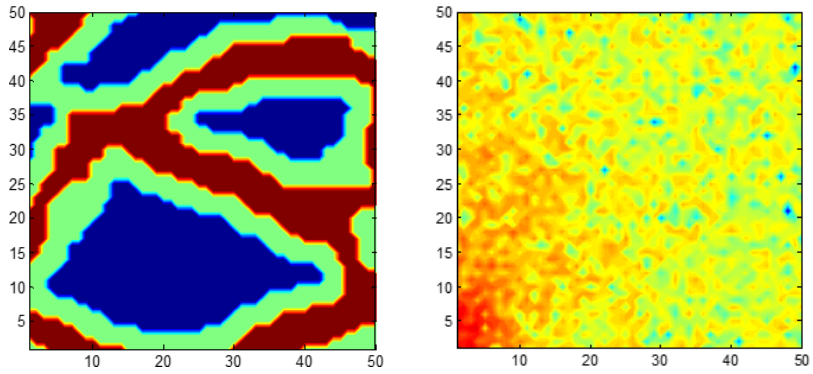
2.1.1 Complete DCT

For the jth realization of a N_m-dimensional property vector $\boldsymbol{m}_j$ (e.g., facies field), we can perfectly reconstruct it by multiplying the $N_m \times N_m$ basis matrix, $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \cdots, \boldsymbol{\phi}_{N_m}]$, with the corresponding N_m-dimensional coefficient vector, $\boldsymbol{v} = [v_1, v_2, \cdots, v_{N_m}]^{\mathrm{T}}$ based on Eqs. 2.1 to 2.4, as follows

$$\boldsymbol{m}_j = \sum_{i=1}^{N_m} v_{j,i}\boldsymbol{\phi}_i = \boldsymbol{\Phi}\boldsymbol{v}_j, \quad \text{for } j = 1,\cdots,N_e, \tag{2.5}$$

where j denotes the index of the realization, $v_{j,i}$ denotes the ith component of $\boldsymbol{v}_j$, and N_e is the size of the ensemble. Because the DCT basis functions, $\boldsymbol{\Phi}$, are prespecified and data-independent, they only need to be computed and saved once.

Given a reservoir property field shown in Fig. 2.1(a), we can obtain the DCT coefficient distribution (see Fig. 2.1(b)) by implementing the forward DCT to the original model. Through retaining all the DCT coefficients and corresponding basis functions, we are able to completely reconstruct the original reservoir property field with Eq. 2.5. Thus, this implementation of DCT is referred to as the "complete DCT" in this dissertation.



(a) Original field (b) DCT coefficients

Figure 2.1: Complete DCT

2.1.2 Truncated Basis DCT

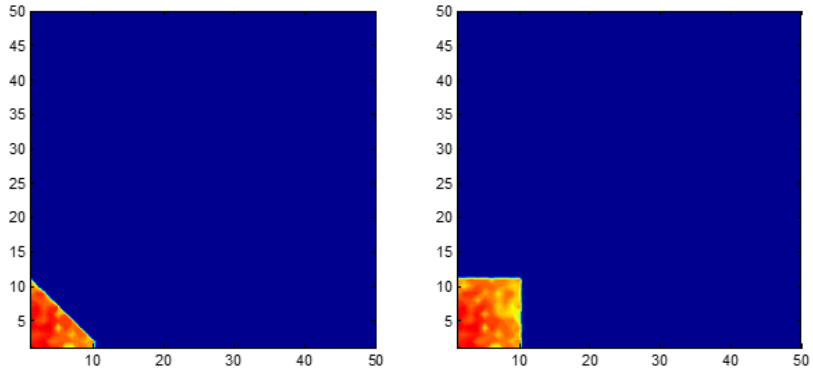
It can be seen from Fig. 2.1(b) that most of the large DCT coefficients (red pixels) are concentrated around the left corner of the coefficient field. These large coefficients correspond to low frequency basis functions which describe the main features of the original field. Based on this property, it is possible to truncate the DCT coefficient field and only retain a small number of large DCT coefficients in order to reduce the number of parameters without losing the most critical features of the original field. Following the common practice in the literature, we can retain the DCT coefficients distributed in a triangular or rectangular area of the field (see Fig. 2.2) in order to "compactly" describe the most important characteristics of $\boldsymbol{m}_j$. Note that the DCT coefficients in the deep blue area of Fig. 2.2 and the corresponding high frequency basis functions are simply abandoned during the truncation process.

As a result of model reduction, it is inevitable that some detailed information about the original field is lost where the detail lost depends on the number of truncated basis functions. For the jth ensemble member, we can have the approximate field, $\hat{\boldsymbol{m}}_j^{\text{truncated}}$, given by

$$\hat{\boldsymbol{m}}_j^{\text{truncated}} = \sum_{i=1}^{N_c} v_{j,i}\boldsymbol{\phi}_{c,i} = \boldsymbol{\Phi}_c \boldsymbol{v}_j, \quad \text{for } j = 1, \cdots, N_e, \tag{2.6}$$

where $\boldsymbol{\Phi}_c$ denotes the $N_m \times N_c$ matrix with columns composed of the reduced (truncated) set of basis functions which is identical for every ensemble member since the truncated area is

predefined and remains the same for each realization, and N_c denotes the number of retained DCT coefficients. In Eq. 2.6, $\boldsymbol{\phi}_{c,i}$ denotes the ith basis vector of the truncated basis set and $v_{j,i}$ is the ith component of the coefficient vector, $\boldsymbol{v}_j$. We refer to this implementation as the "truncated basis DCT" in this dissertation.



(a) Triangular area (b) Rectangular area

Figure 2.2: Truncated basis DCT

2.1.3 Particular Basis DCT

The limitation of the truncated basis DCT is that the linear combination of retained DCT coefficients and corresponding basis functions in the predefined area may not be optimal for every realization. To better reflect the key features of every original model, we can construct a particular set of basis functions for each realization by retaining the N_c largest (in magnitude) DCT coefficients and their corresponding basis fuctions. This gives the approximate field, $\hat{\boldsymbol{m}}_j^{\text{particular}}$, as

$$\hat{\boldsymbol{m}}_j^{\text{particular}} = \sum_{i=1}^{N_c} v_{j,i}\boldsymbol{\phi}_{j,i} = \boldsymbol{\Phi}_j \boldsymbol{v}_j, \quad \text{for } j = 1, \cdots, N_e, \tag{2.7}$$

where the particular set of basis function, $\boldsymbol{\Phi}_j$, is different for every prior realization and needs to be saved separately. Therefore, this implementation is referred to as the "particular basis DCT" in this dissertation.

2.2 Common Basis DCT

In this work, from the N_e different combinations (each combination includes one basis matrix $\mathbf{\Phi}_j \in \mathbb{R}^{N_m \times N_c}$ and one coefficient vector $\boldsymbol{v}_j \in \mathbb{R}^{N_c \times 1}$), we select N_c specific basis functions to construct a $N_m \times N_c$ common basis set, $\bar{\mathbf{\Phi}}$, which is identical and fixed for the whole ensemble during the history matching process. The corresponding implementation is referred to as the "common basis DCT", and this common basis set eliminates the need to save the particular set of basis functions for each realization specifically. To obtain the set of common basis functions $\bar{\mathbf{\Phi}}$, we first retain the N_c largest coefficients for each ensemble member using Eq. 2.7. Consequently, each model can be estimated with N_c DCT coefficients and their corresponding basis functions. It needs to be noted that the N_c retained coefficients vary between different realizations which means the whole ensemble has N_e different combinations of retained coefficients and corresponding basis functions. Then, we loop through the total N_e sets of coefficients (each set has N_c coefficients sorted in descending order) to select the N_c largest coefficients. To be specific, we go through the first largest coefficient of each realization and retain the current coefficient as long as it is different from any previously retained ones. The second largest coefficient of every realization will be checked subsequently and we repeat the same process sequentially until the retained number N_c is reached, and the $\bar{\mathbf{\Phi}}$ consists of N_c basis functions corresponding to the retained coefficients. The flowchart for the construction of common basis set $\bar{\mathbf{\Phi}}$ is illustrated in Fig. 2.3.

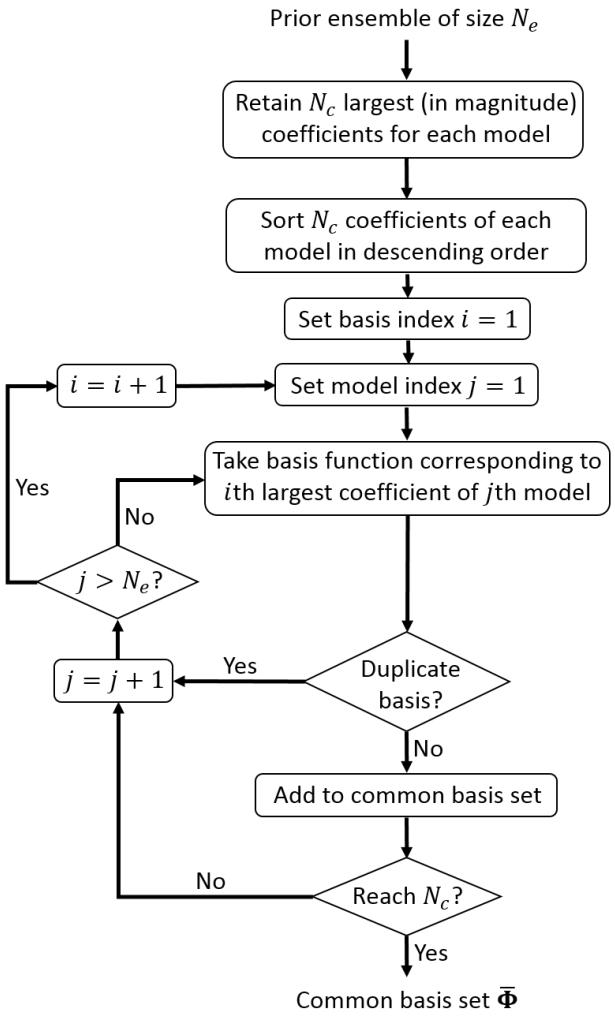


Figure 2.3: Flowchart for the construction of common basis set

Since the common basis set usually differs from the particular basis set for each ensemble member, it is necessary to recalculate the expansion coefficients of each model with respect to the common set of basis functions. This corresponds to finding $\hat{\boldsymbol{v}}_j$ such that

$$\hat{\boldsymbol{m}}_j^{\text{common}} = \bar{\boldsymbol{\Phi}}\hat{\boldsymbol{v}}_j, \quad \text{for } j = 1, \cdots, N_e, \tag{2.8}$$

where the coefficient vector, $\hat{\boldsymbol{v}}_j$, is obtained by solving the least squares problem defined below

$$\hat{\boldsymbol{v}}_j = \underset{\boldsymbol{v}}{\operatorname{argmin}} \left\| \boldsymbol{m}_j - \bar{\boldsymbol{\Phi}}\boldsymbol{v} \right\|_2^2, \quad \text{for } j = 1, \cdots, N_e, \tag{2.9}$$

where $\boldsymbol{m}_j$ is the original N_m-dimensional model vector. These $\hat{\boldsymbol{v}}_j$'s are then the model

parameters which will be updated by assimilating observed data during history matching.

2.2.1 Reconstruction Experiment

To compare the feature extraction capacity of different implementations of DCT, we design an experiment to reconstruct true facies fields, $\boldsymbol{m}_{\text{true}} \in \mathbb{R}^{N_m \times 1}$, from different initial models using truncated, particular and common basis DCT, respectively, as follows:

$$\hat{\boldsymbol{v}} = \underset{\boldsymbol{v}}{\operatorname{argmin}} \|\boldsymbol{m}_{\text{true}} - \boldsymbol{\Phi}\boldsymbol{v}\|_2^2 \tag{2.10}$$

and

$$\hat{\boldsymbol{m}}_{\text{true}} = \boldsymbol{\Phi}\hat{\boldsymbol{v}}, \tag{2.11}$$

where $\hat{\boldsymbol{m}}_{\text{true}}$ is the N_m-dimensional approximation of the true model; $\hat{\boldsymbol{v}}$ is the N_c-dimensional coefficient vector; $\boldsymbol{\Phi} \in \mathbb{R}^{N_m \times N_c}$ denotes the set of DCT basis functions so it can be $\boldsymbol{\Phi}_c$ (see Eq. 2.6) for truncated basis DCT, $\boldsymbol{\Phi}_j$ (see Eq. 2.7) for particular basis DCT and $\bar{\boldsymbol{\Phi}}$ (see Eq. 2.8) for common basis DCT, respectively. Each initial model is also a facies field and the retained coefficients, $\hat{\boldsymbol{v}}$, are calculated by solving a least squares problem defined in Eq. 2.10 after the basis matrix, $\boldsymbol{\Phi}$, is obtained by different implementations of DCT.

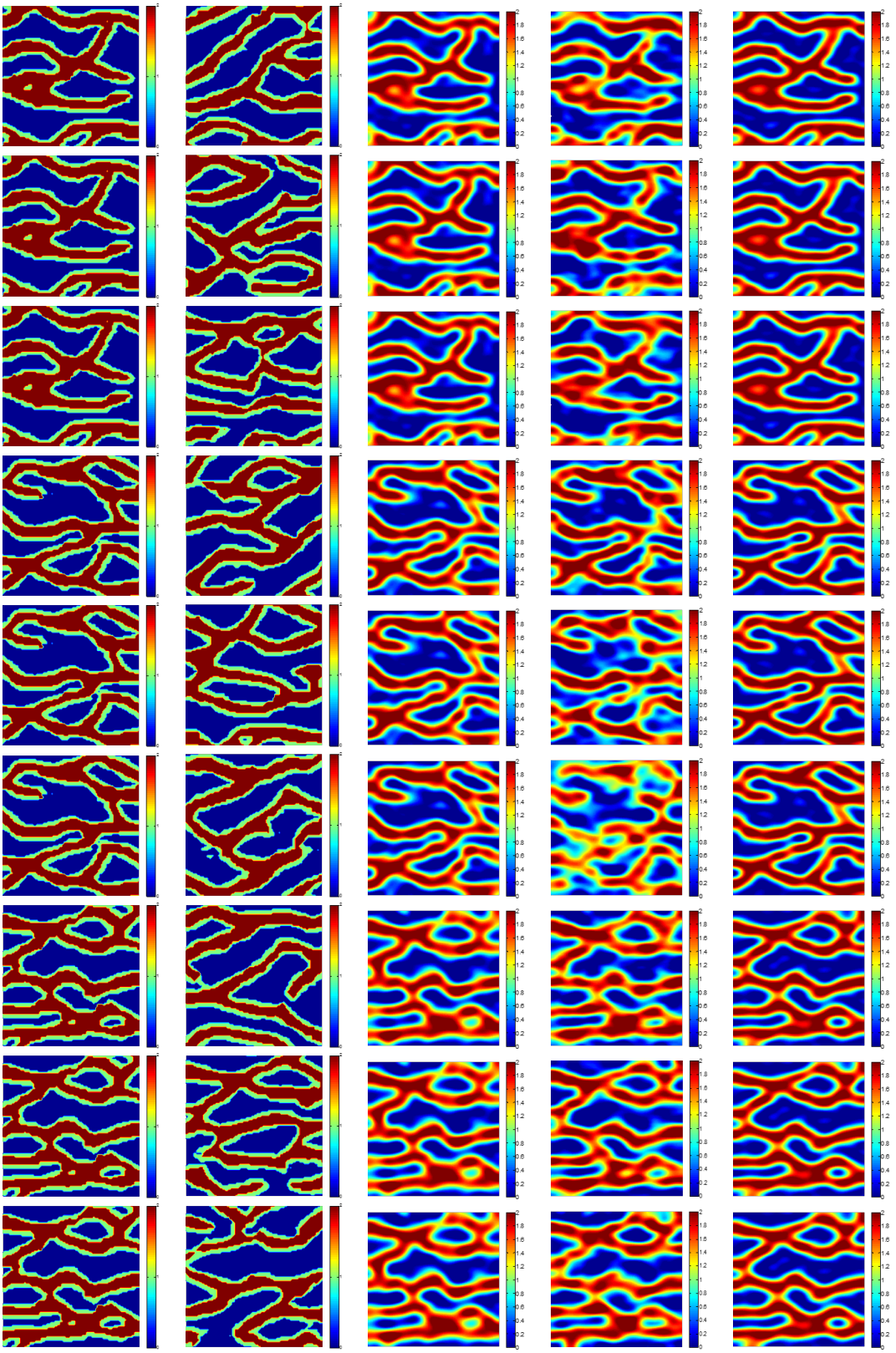
To be specific, $\boldsymbol{\Phi}_c$ is selected in a predefined rectangular or triangular area given the retained number N_c, and the choice of the initial models does not affect the truncated set of basis functions. For the particular basis DCT, the $\boldsymbol{\Phi}_j$ is model-dependent so we retain different particular basis functions with different initial models. For the common basis DCT, the common basis set, $\bar{\boldsymbol{\Phi}}$, are constructed from a training ensemble consisting of 100 facies models (see Fig. 2.3) and it is identical for different initial models. Note that the true models which are being reconstructed are not included in the ensemble.

Figures 2.4 and 2.5 present the results of this reconstruction experiment, where the first column shows the true facies fields (three models in total) which are being reconstructed, the second column shows the initial facies models and the columns (c) to (e) are the reconstructed facies fields using truncated, particular and common basis DCT given the retained

number of basis functions, N_c. We set the retained number, N_c, to be 200 (see results in Fig. 2.4) and 400 (see results in Fig. 2.5), respectively, in order to investigate the influence of N_c on the model approximation. Through visual comparison, the true facies fields are shown to be better reconstructed with the common set of basis functions, $\bar{\mathbf{\Phi}}$. The probable reason is that the common basis set can describe the geological similarity throughout the whole ensemble by taking the features of different models into account. It is shown that the complex channel patterns in the three true facies fields can be clearly captured by using the common basis DCT even with $N_c = 200$. However, we cannot reproduce all the continuous geological structures using the particular basis DCT with the same retained number because the basis functions selected based on one particular model may have poor generalization capability to reconstruct other models. Although the reconstructed fields obtained by the truncated basis DCT appear to be better than those obtained by the particular basis DCT, some of them still display apparent discontinuity around the sinuous parts of the channels (see column (c) in Fig. 2.4). The advantage of the common basis DCT can be further demonstrated by the model mismatch ($\|\boldsymbol{m}_{\text{true}} - \mathbf{\Phi}\hat{\boldsymbol{v}}\|_2^2$) shown in Table 2.1, but it should be noted that the metric is only applicable for the synthetic models used in this work. For the same true facies field, various initial models lead to different reconstructed fields when we implement the particular basis DCT. For instance, the model mismatch obtained by using initial model 3 to reconstruct true field 2 is much larger than those obtained with the other two initial models. On the other hand, since the choice of the initial models does not affect the retained basis functions in the truncated and common basis DCT, we can observe identical model mismatch for different initial models.

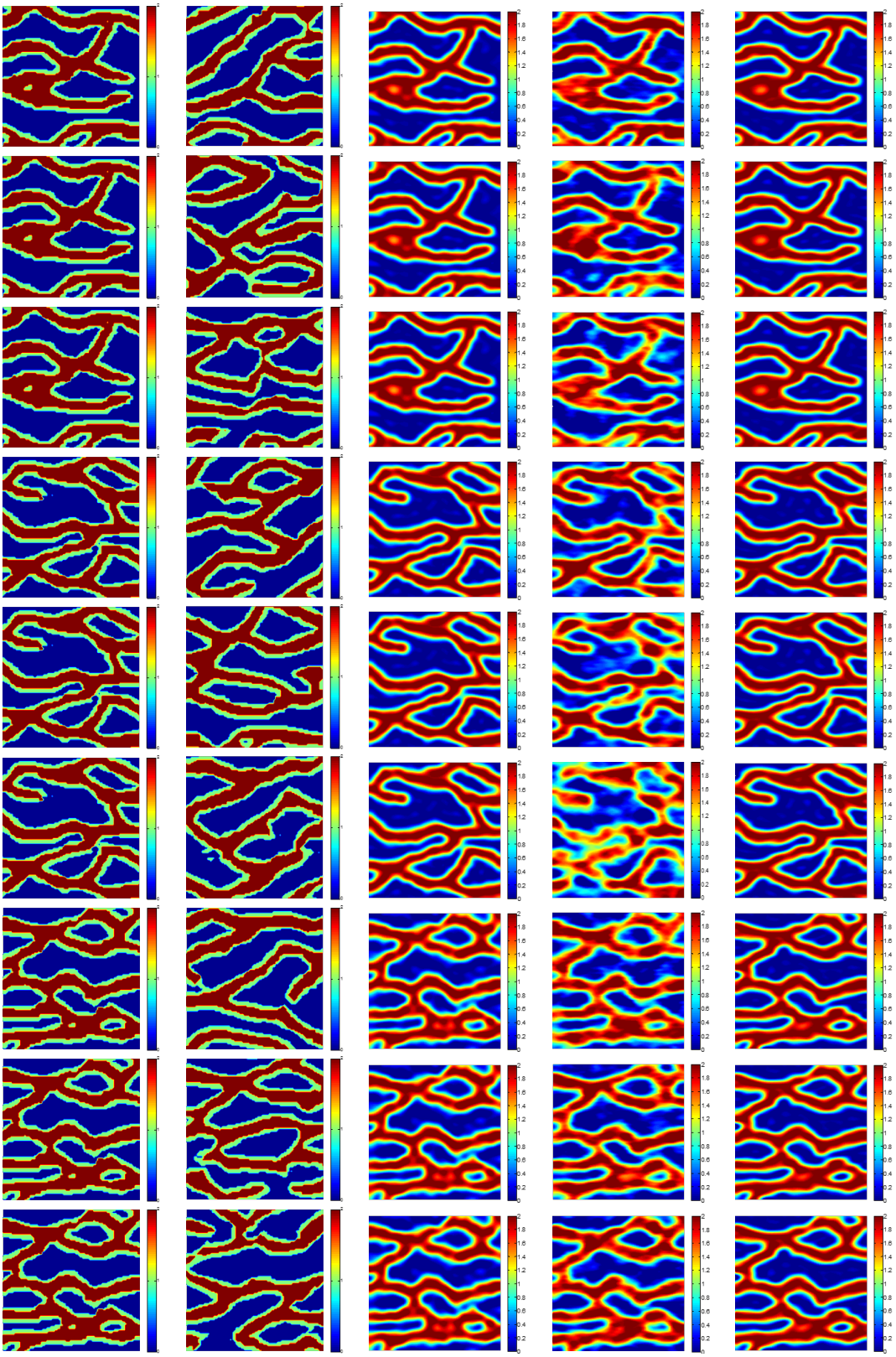
Table 2.1: Model mismatch

		Truncated basis DCT		Particular basis DCT		Common basis DCT	
		$N_c = 200$	$N_c = 400$	$N_c = 200$	$N_c = 400$	$N_c = 200$	$N_c = 400$
True 1	Initial 1	1105.3	601.8	1536.6	998.6	767.3	547.6
	Initial 2	1105.3	601.8	1489.9	1176.5	767.3	547.6
	Initial 3	1105.3	601.8	1444.0	1190.2	767.3	547.6
True 2	Initial 1	1206.3	631.9	1451.6	1176.5	823.7	590.5
	Initial 2	1206.3	631.9	1892.2	7744.0	823.7	590.5
	Initial 3	1206.3	631.9	3329.3	2683.2	823.7	590.5
True 3	Initial 1	1348.8	784.9	1436.4	1218.1	846.8	610.1
	Initial 2	1348.8	784.9	1232.0	948.6	846.8	610.1
	Initial 3	1348.8	784.9	1332.2	894.0	846.8	610.1



(a) True (b) Initial (c) Truncated basis (d) Particular basis (e) Common basis

Figure 2.4: Comparison of reconstruction performance between three DCT implementations when $N_c = 200$



(a) True (b) Initial (c) Truncated basis (d) Particular basis (e) Common basis

Figure 2.5: Comparison of reconstruction performance between three DCT implementations when $N_c = 400$

2.3 History Matching Workflow

2.3.1 Ensemble Smoother with Multiple Data Assimilation (ES-MDA)

Motivated by the analogy between the ES method (van Leeuwen and Evensen, 1996) and one single iteration of Gauss-Newton correction (Reynolds et al., 2006), the ensemble smoother with multiple data assimilation (ES-MDA) was first proposed by Emerick and Reynolds (2013a), and it was used for assimilating seismic and production data to estimate the permeability field of a reservoir. Later Emerick and Reynolds (2013b) showed that the ES-MDA gives better performance for matching observed data and reducing uncertainty in model description compared to the ES and EnKF methods. The history matching workflow by using the original ES-MDA algorithm to update the DCT coefficients is as follows

(1) Choose the number of data assimilation steps, N_a, and the corresponding inflation coefficients, α_ℓ, for $\ell = 1, \cdots, N_a$, such that $\sum_{\ell=1}^{N_a} \frac{1}{\alpha_\ell} = 1$.

(2) Initialize reservoir model $\boldsymbol{m}_j^1$ and coefficient vector $\boldsymbol{v}_j^1$, for $j = 1, \cdots, N_e$.

(3) For $\ell = 1$ to N_a,

- Run all the ensemble members from time zero until the end of the history matching period in order to obtain the predicted data as

$$\boldsymbol{d}_j^\ell = \boldsymbol{g}\left(\boldsymbol{m}_j^\ell\right), \quad \text{for } j = 1, \cdots, N_e, \tag{2.12}$$

  where $\boldsymbol{g}(\cdot)$ denotes the nonlinear forward model, i.e., $\boldsymbol{d}_j^\ell$ is the N_d-dimensional vector of predicted data obtained by running the Eclipse reservoir simulator with the model parameters given by the vector $\boldsymbol{m}_j^\ell$ from time zero.

- Perturb the vector of the observed data by

$$\boldsymbol{d}_{\mathrm{uc},j}^\ell = \boldsymbol{d}_{\mathrm{obs}} + \sqrt{\alpha_\ell}\boldsymbol{C}_{\mathrm{D}}^{1/2}\boldsymbol{z}_{d,j}, \quad \text{for } j = 1, \cdots, N_e, \tag{2.13}$$

where $\boldsymbol{C}_{\mathrm{D}}$ is a $N_d \times N_d$ covariance matrix containing measurement errors of the observed data and $\boldsymbol{z}_{d,j} \sim \mathcal{N}(0, \boldsymbol{I}_{N_d})$.

- Update the DCT coefficients using

$$\hat{\boldsymbol{v}}_j^{\ell+1} = \boldsymbol{v}_j^{\ell} + \boldsymbol{C}_{\mathrm{VD}}^{\ell}(\boldsymbol{C}_{\mathrm{DD}}^{\ell} + \alpha_\ell \boldsymbol{C}_{\mathrm{D}})^{-1}(\boldsymbol{d}_{\mathrm{uc},j}^{\ell} - \boldsymbol{d}_j^{\ell}), \quad \text{for } j = 1, \cdots, N_e, \tag{2.14}$$

where we denote $\boldsymbol{v}_j^{\ell}$ as the jth vector of DCT coefficients corresponding to the truncated, particular or common basis functions; N_c as the number of retained coefficients; $\boldsymbol{C}_{\mathrm{VD}}^{\ell}$ is the $N_c \times N_d$ cross-covariance matrix between the DCT coefficients and the predicted data and $\boldsymbol{C}_{\mathrm{DD}}^{\ell}$ is the $N_d \times N_d$ auto-covariance matrix of the predicted data. Similar to the EnKF and ES methods, the covariance matrices $\boldsymbol{C}_{\mathrm{VD}}^{\ell}$ and $\boldsymbol{C}_{\mathrm{DD}}^{\ell}$ are estimated using the ensemble of reservoir models and their corresponding predicted data by

$$\boldsymbol{C}_{\mathrm{VD}}^{\ell} = \frac{1}{N_e - 1} \sum_{j=1}^{N_e} \left(\boldsymbol{v}_j^{\ell} - \bar{\boldsymbol{v}}^{\ell}\right) \left(\boldsymbol{d}_j^{\ell} - \bar{\boldsymbol{d}}^{\ell}\right)^{\mathrm{T}}, \tag{2.15}$$

$$\boldsymbol{C}_{\mathrm{DD}}^{\ell} = \frac{1}{N_e - 1} \sum_{j=1}^{N_e} \left(\boldsymbol{d}_j^{\ell} - \bar{\boldsymbol{d}}^{\ell}\right) \left(\boldsymbol{d}_j^{\ell} - \bar{\boldsymbol{d}}^{\ell}\right)^{\mathrm{T}}, \tag{2.16}$$

where $\bar{\boldsymbol{v}}^{\ell}$ and $\bar{\boldsymbol{d}}^{\ell}$ denote the ensemble mean of the DCT coefficients and the predicted data, respectively.

- Update the facies field by

$$\hat{\boldsymbol{m}}_j^{\ell+1} = \boldsymbol{\Phi} \hat{\boldsymbol{v}}_j^{\ell+1}, \quad \text{for } j = 1, \cdots, N_e, \tag{2.17}$$

where $\boldsymbol{\Phi}$ denotes the set of basis functions so it can be $\boldsymbol{\Phi}_c$ (see Eq. 2.6) for truncated basis DCT, $\boldsymbol{\Phi}_j$ (see Eq. 2.7) for particular basis DCT and $\bar{\boldsymbol{\Phi}}$ (see Eq. 2.8) for common basis DCT, respectively. It is noteworthy that after the update step, $\hat{\boldsymbol{m}}_j^{\ell+1}$ only contains continuous values rather than discrete facies type indicators,

thus a post-processing procedure is added to do the discrete mapping.

- Obtain the discrete facies field by

$$\boldsymbol{m}_j^{\ell+1} = \text{post-processing}\left(\hat{\boldsymbol{m}}_j^{\ell+1}\right), \quad \text{for } j = 1, \cdots, N_e, \tag{2.18}$$

where post-processing $(\cdot)$ denotes a procedure based on a regularization framework which is discussed in next section to map most of the continuous values in $\hat{\boldsymbol{m}}_j^{\ell+1}$ to discrete facies indicators in $\boldsymbol{m}_j^{\ell+1}$ based on which we can obtain the reservoir properties fields for simulation.

- Recalculate the DCT coefficients by

$$\boldsymbol{v}_j^{\ell+1} = \underset{\boldsymbol{v}}{\text{argmin}} \left\| \boldsymbol{m}_j^{\ell+1} - \boldsymbol{\Phi}\boldsymbol{v} \right\|_2^2, \quad \text{for } j = 1, \cdots, N_e, \tag{2.19}$$

where $\boldsymbol{\Phi}$ denotes the set of basis functions so it can be $\boldsymbol{\Phi}_c$ (see Eq. 2.6) for truncated basis DCT, $\boldsymbol{\Phi}_j$ (see Eq. 2.7) for particular basis DCT and $\bar{\boldsymbol{\Phi}}$ (see Eq. 2.8) for common basis DCT, respectively. Since the DCT coefficients updated by history matching (i.e., $\hat{\boldsymbol{v}}_j^{\ell+1}$ in Eq. 2.17) are not consistent with the facies field after the post-processing procedure (i.e., $\boldsymbol{m}_j^{\ell+1}$ in Eq. 2.18), thus we need to solve the least squares problem of Eq. 2.19 in order to recalculate the coefficients for next data assimilation step.

End (For)

2.3.2 Post-Processing Based on Regularization Framework

The adopted post-processing procedure in this work is based on the regularization framework proposed by Vo and Durlofsky (2014, 2015) which is designed to enhance the connectivity of key geological features by introducing regularization. For a three-facies reservoir system, the facies are designated by 0, 1 and 2 (shale, levee and sand, respectively). Vo and Durlofsky (2015) derived a series of analytical equations to obtain the facies type indicator

of a gridblock, x_i, from its continuous value a_i. It needs to be noted that this algorithm is implemented in a gridblock-by-gridblock manner, so a_i represents a component of the continuous facies field, $\hat{\boldsymbol{m}}_j^{\ell+1}$, obtained by Eq. 2.17. To obtain the value of facies indicator, x_i^*, in each gridblock, the following function is minimized:

$$f(x_i) = (a_i - x_i)^2 + \gamma_1 R_1^i(x_i) + \gamma_2 R_2^i(x_i), \quad x_i \in \left[x^l, x^u\right], \tag{2.20}$$

where x_i^* denotes the value of x_i which minimizes $f(x_i)$, $R_1^i(x_i)$ and $R_2^i(x_i)$ denote the regularization terms, γ_1 and γ_2 are the regularization weights, and $\left[x^l, x^u\right]$ represent the lower and upper bounds of x_i which actually depend on the value of a_i.

According to the solutions presented in the original paper (Vo and Durlofsky, 2015), two cases need to be considered depending on the value of a_i:

Case 1. If $a_i \leq 1$, the bound constraints are $x^l = 0$ and $x^u = 1$, and the regularization terms are defined by $R_1^i = x_i(1 - x_i/2)$ and $R_2^i = -x_i^2/2$, with weights γ_{11} and γ_{12}. The objective function in this case is

$$\begin{aligned} f(x_i) &= (a_i - x_i)^2 + \gamma_{11} x_i \left(1 - \frac{1}{2}x_i\right) + \gamma_{12}\left(-\frac{1}{2}x_i^2\right) \\ &= \left(1 - \frac{\gamma_{11} + \gamma_{12}}{2}\right) x_i^2 - 2\left(a_i - \frac{\gamma_{11}}{2}\right) x_i + a_i^2, \quad x_i \in [0,1], \end{aligned} \tag{2.21}$$

where $\gamma_{11}, \gamma_{12} \geq 0$, and $\frac{\gamma_{11}+\gamma_{12}}{2} < 1$. The solution for this case follows

(1) If $a_i \leq \gamma_{11}/2$, then $x_i^* = 0$ (facies0, shale);

(2) If $\gamma_{11}/2 < a_i < 1 - \gamma_{12}/2$, then $x_i^* = (a_i - \gamma_{11}/2)/(1 - (\gamma_{11} + \gamma_{12})/2)$;

(3) If $a_i \geq 1 - \gamma_{12}/2$, then $x_i^* = 1$ (facies1, levee).

Case 2. If $a_i > 1$, the bound constraints are $x^l = 1$ and $x^u = 2$, and the regularization terms are defined by $R_1^i = (x_i - 1)(1 - (x_i - 1)/2)$ and $R_2^i = -(x_i - 1)^2/2$, with weights γ_{21}

and γ_{22}. The objective function in this case is

$$\begin{aligned} f(x_i) &= (a_i - x_i)^2 + \gamma_{21}(x_i - 1)\left[1 - \frac{1}{2}(x_i - 1)\right] + \gamma_{22}\left[-\frac{1}{2}(x_i - 1)^2\right] \\ &= \left(1 - \frac{\gamma_{21} + \gamma_{22}}{2}\right) x_i^2 - 2\left(a_i - \frac{2\gamma_{21} + \gamma_{22}}{2}\right) x_i + a_i^2 - \frac{3\gamma_{21} + \gamma_{22}}{2}, \quad x_i \in [1, 2], \end{aligned} \tag{2.22}$$

where $\gamma_{21}, \gamma_{22} \geq 0$, and $\frac{\gamma_{21}+\gamma_{22}}{2} < 1$. The solution for this case follows

(1) If $1 < a_i \leq 1 + \gamma_{21}/2$, then $x_i^* = 1$ (facies1, levee);

(2) If $1 + \gamma_{21}/2 < a_i < 2 - \gamma_{22}/2$, then $x_i^* = (a_i - (2\gamma_{21} + \gamma_{22})/2)/(1 - (\gamma_{21} + \gamma_{22})/2)$;

(3) If $a_i \geq 2 - \gamma_{22}/2$, then $x_i^* = 2$ (facies2, sand).

For situations (1) and (3) in the two cases, it's straightforward to obtain the reservoir property fields by assigning the permeability or porosity value to every gridblock according to its facies type indicator, x_i^*. However, for situation (2), x_i^* is still a continuous value. In order to assign the permeability or porosity to the gridblock, a simple weighted interpolation is performed between the property values of different facies. For instance, if $\gamma_{11}/2 < a_i < 1 - \gamma_{12}/2$ in Case 1, then $x_i^* = (a_i - \gamma_{11}/2)/(1 - (\gamma_{11} + \gamma_{12})/2)$ and $\ln(k)_i = (1 - x_i^*) \times \ln(k)_{\text{facies0}} + (x_i^* - 0) \times \ln(k)_{\text{facies1}}$.

In the original papers by Vo and Durlofsky (2014, 2015), the authors proved that the objective function of the optimization problem is quadratic and convex when the four regularization weights satisfy the following constraints:

$$\begin{aligned} &\gamma_{11}, \gamma_{12} \geq 0 \text{ and } \gamma_{11} + \gamma_{12} < 2, \\ &\gamma_{21}, \gamma_{22} \geq 0 \text{ and } \gamma_{21} + \gamma_{22} < 2. \end{aligned} \tag{2.23}$$

However, it's not trivial to set proper values to the four regularization weights (γ_{11}, γ_{12}, γ_{21} and γ_{22}). Here we propose a scheme to calculate the four regularization weights based on the average facies fractions of the prior ensemble. Basically, the four regularization

parameters in Eq. 2.23 correspond to different facies and constrain their fractions. In order to keep the estimated facies proportions (FP) as close as possible to those of the prior ensemble, the following equations are solved to obtain the values of four regularization parameters.

$$\frac{\gamma_{11}}{\gamma_{12}} = \frac{\text{FP(shale)}}{\text{FP(levee)}},\ \gamma_{11} + \gamma_{12} = \beta, \tag{2.24}$$

$$\frac{\gamma_{21}}{\gamma_{22}} = \frac{\text{FP(levee)}}{\text{FP(sand)}},\ \gamma_{21} + \gamma_{22} = \beta, \tag{2.25}$$

where FP(shale), FP(levee) and FP(sand) denote the average facies proportions of the prior ensemble, and the value β of the summations of two parameters ($\gamma_{11} + \gamma_{12}$, $\gamma_{21} + \gamma_{22}$) aims to balance the impacts of the regularization terms and the quadratic mismatch part in the optimization function, $f(x_i)$. Based on our experimental results, the summation of $\beta = 1.5$ is a reasonable value to retain the average facies proportions shown in the prior ensemble but it may vary from application to application. With this post-processing technique, we are able to enhance the spatial connectivity of channel facies and keep the facies proportions within an acceptable range.

2.4 Computational Results and Discussion

In this section, we test the proposed assisted history matching workflow on two synthetic reservoir examples. The first example is a 2D three-facies channelized reservoir with a relatively high number of channels. In this example, we first show the history matching results with the implementation of common basis DCT then compare its parameterization performance with truncated basis DCT and complete DCT. The second example is a 3D three-facies five-layer channelized reservoir where sinuous channels have various distributions in both horizontal and vertical directions, and the five layers are divided into two geological zones with different channel patterns.

2.4.1 Example 1: 2D Three-Facies Reservoir Model

The first example considers a 2D three-facies (shale, levee and sand) fluvial channel-

ized reservoir in a horizontal domain. The model contains 100×100 gridblocks and the size of each gridblock is 100ft $\times$ 100ft $\times$ 50ft. The facies distribution of the true model and prior ensemble members are generated by using the MPS simulation algorithm. All the models are subject to the geostatistics inferred from a training image shown in Fig. 2.6 and honor the hard data of observed facies at the well locations. Note that the petrophysical properties (i.e., permeability and porosity) within each facies are assumed to be homogeneous and known a priori as our objective is to investigate whether we can maintain the geological structures of the facies field when history data matched. The basic properties of facies are summarized in Table 2.2.

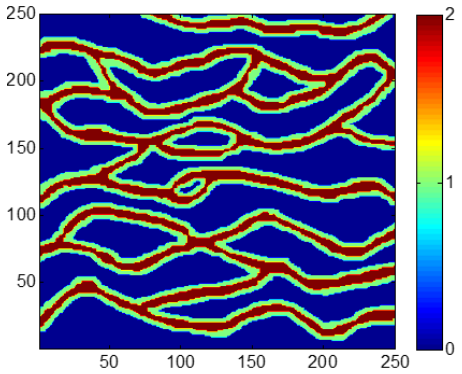


Figure 2.6: Training image (Example 1)

Table 2.2: Properties of facies

Facies type	Indicator	Color	ϕ	k, mD	$\ln(k)$
Shale	0	Blue	0.1	5	1.609
Levee	1	Green	0.2	500	6.215
Sand	2	Red	0.3	1500	7.313

There are four injection wells and nine production wells which are deployed in a five-spot pattern in the reservoir. All producers and injectors, respectively, operate under constant bottom hole pressure (BHP) of 3000 psi and 5500 psi. The initial reservoir pressure and water saturation are uniform and set equal to 5000 psi and 0.2, respectively. The history matching period is 10 years (3600 days) and all wells start operating from time zero which

is the beginning of the simulation. In this work, the subsurface flow responses are obtained by running the Eclipse reservoir simulator. The dynamic data to be assimilated in this example are collected every 30 days during the history matching period, so we have a total data collection times of $N_t = 3600/30 = 120$. The production data include water injection rate (q_{inj}) of the injectors and oil production rate (q_{o}) and water production rate (q_{w}) of the producers. Thus, the vector of observed data is constructed as

$$\boldsymbol{d}_{\text{obs}} = \left[\boldsymbol{q}_{\text{inj}}^T, \boldsymbol{q}_{\text{o}}^T, \boldsymbol{q}_{\text{w}}^T, \boldsymbol{d}_{\text{hard}}^T\right]_{N_d \times 1}^T, \tag{2.26}$$

where $\boldsymbol{q}_{\text{inj}}$ denotes the water injection rate of all injection wells collected in the history matching period, and similarly for $\boldsymbol{q}_{\text{o}}$ and $\boldsymbol{q}_{\text{w}}$. The vector of hard data, $\boldsymbol{d}_{\text{hard}}$, is used to constrain the facies type at well locations during the history matching process. This mitigates the need for using geological simulation techniques to honor the hard data in a resampling procedure. The observed data are obtained by adding random Gaussian noise to the simulation results of the true model. The covariance matrix of the observed data, $\boldsymbol{C}_{\text{D}}$ in Eq. 2.14, is given by

$$\boldsymbol{C}_{\text{D}} = \begin{bmatrix} \sigma_{d_1}^2 & & \\ & \ddots & \\ & & \sigma_{d_{N_d}}^2 \end{bmatrix}_{N_d \times N_d}, \tag{2.27}$$

where the standard deviation of measurement error, σ, is 3% of the true data for the injection and production rates and is set to 0.01 for the facies observation at well locations. The facies field of the true model as well as the well locations are shown in Fig. 2.7. After the history matching period, the updated posterior models are used to predict the future reservoir performance for another 5 years (1800 days).

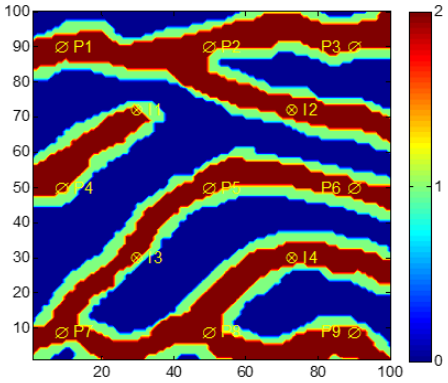
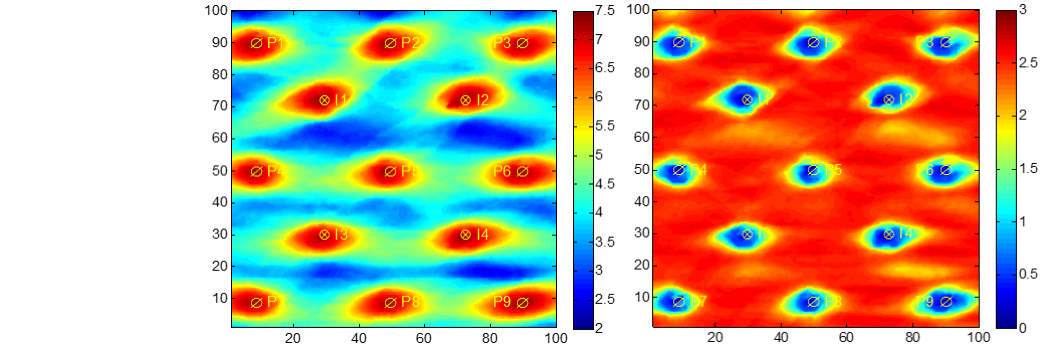
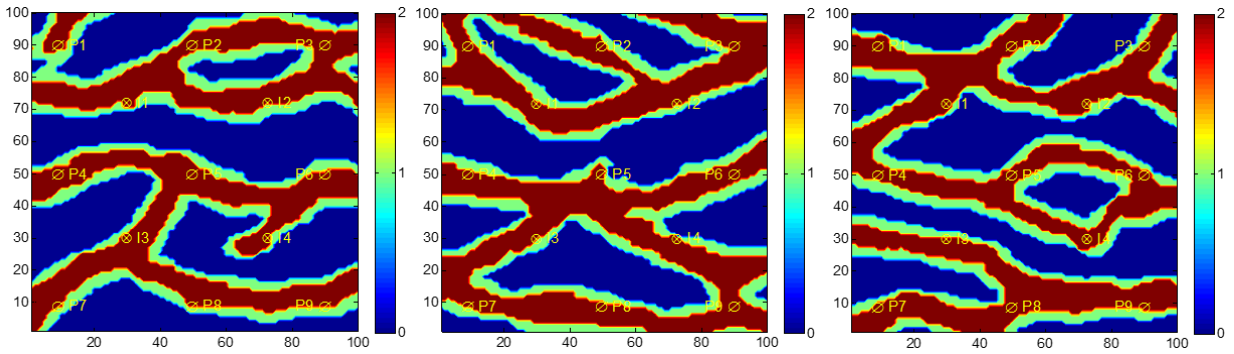


Figure 2.7: True model (Example 1)

In this example, we generated 400 prior realizations (ensemble size $N_e = 400$) by using the MPS algorithm. Since we deal with complex channelized reservoirs, a larger ensemble is necessary to obtain desired data match results. In Fig. 2.8, we show the ensemble mean and standard deviation (STD) of the ln-permeability ($\ln(k)$) field based on 400 ensemble members, and also show three unconditional realizations of facies field. As can be inferred from Figs. 2.8(a) and 2.8(b), although all ensemble members honor the sand facies as hard data at the well locations, there is still a large variability among realizations when it comes to the orientation and location of high-permeability channels. In addition, since the ln-permeability values of the sand and levee facies are close, the problems involving more than two facies are much more challenging as the history matching algorithm is not only required to preserve the connectivity of the channel facies but also to distinguish facies from each other.



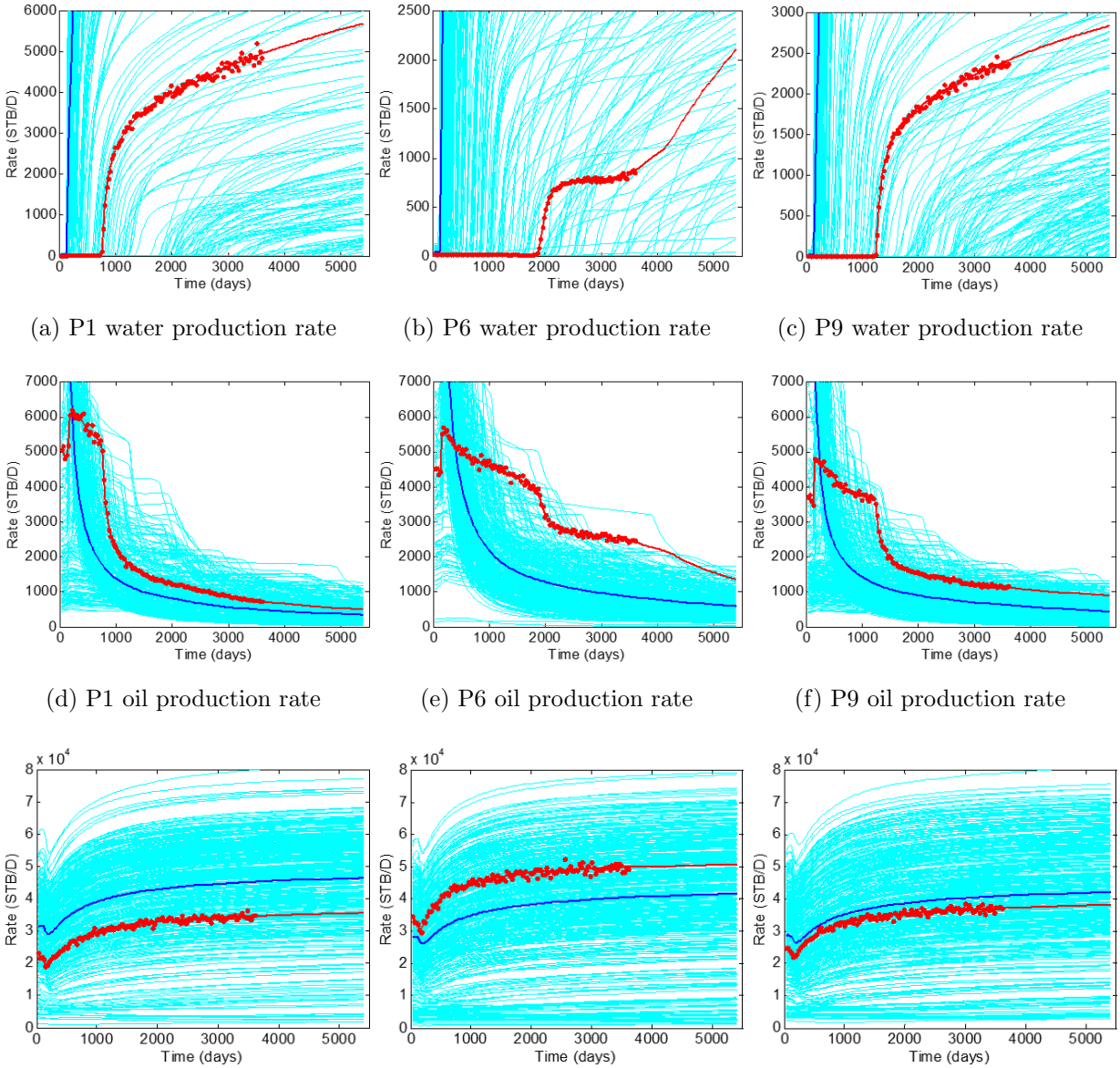
(a) Prior mean ($\ln(k)$) (b) Prior STD ($\ln(k)$)



(c) Prior model #1 (facies) (d) Prior model #2 (facies) (e) Prior model #3 (facies)

Figure 2.8: Mean and STD of the prior ensemble and three prior realizations (Example 1)

Figure 2.9 presents the production data obtained by running the prior ensemble from time zero to the end of the forecast period. The red dots denote the observed data and the red curve is obtained from the true model. The light blue curves represent the production data of every prior model and the dark blue curve corresponds to the prior ensemble mean. In this reservoir example, we observe water breakthrough at all production wells during the history matching period. Since the initial realizations differ greatly from each other in terms of channel locations, the prior uncertainty of production data is very high. For the water production rate (q_{w}) of P1, P6 and P9, most of the prior members have much earlier water breakthrough than that of the true model and the final water flow rates deviate significantly from the observed data.



(a) P1 water production rate (b) P6 water production rate (c) P9 water production rate

(d) P1 oil production rate (e) P6 oil production rate (f) P9 oil production rate

(g) I1 water injection rate (h) I3 water injection rate (i) I4 water injection rate

Figure 2.9: Prior production data (Example 1). True (red curves), observed data (red dots), simulated data of realizations (light blue curves), and ensemble mean (dark blue curves). History: $0 < t \leq 3600$ days, forecast: $t > 3600$ days.

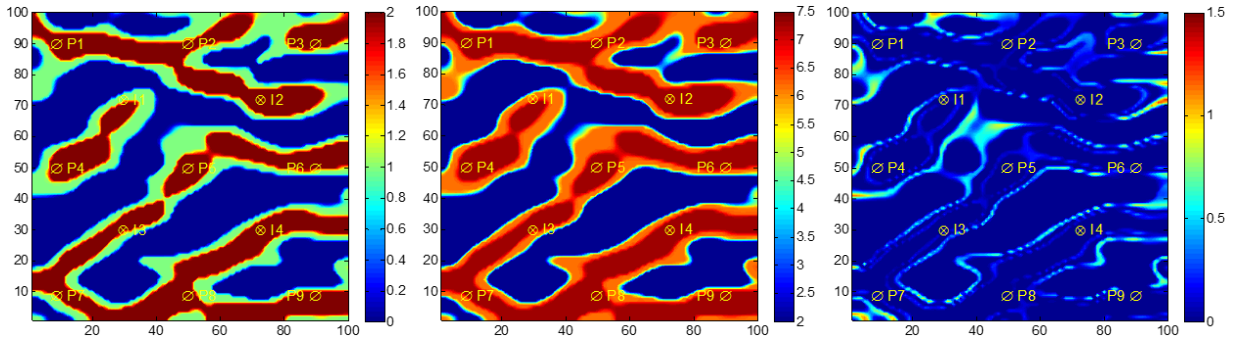
As stated before, for this example we parameterize the realizations by using common basis DCT and implement the ES-MDA algorithm with an ensemble of 400 realizations due to the large amount of production data and relatively high number of complex channels. We first apply the DCT to each of the prior realizations in order to obtain the particular DCT coefficients corresponding to the basis functions. The number of DCT coefficients and

basis functions for each realization is equal to the number of uncertain model parameters of the reservoir, which is $100 \times 100 = 10000$ in this example. For this example, the retained number of DCT coefficients and their corresponding basis functions is set to be $N_c = 200$. In fact, there are no rigorous rules to determine the value of N_c, but 2% to 5% of the total gridblock number is an empirically reasonable value of N_c and large enough to handle complex channelized reservoir system according to our experience. Therefore, the parameters to be calibrated through the history matching process become the DCT coefficients corresponding to the particular set of basis function for each model (see Eq. 2.7). This is the implementation of particular basis DCT.

As discussed previously, with the concept of common basis DCT, we select a set of common basis functions which correspond to the largest coefficients throughout the whole ensemble in order to be able to reflect the similarity of geological features between different realizations. After a set of common basis functions has been picked from all basis functions of the prior ensemble, the DCT coefficients for each realization with respect to the set of common basis functions are recomputed in order to satisfy the requirement shown in Eq. 2.9. That is to say, we have same common basis functions for all realizations and the DCT coefficients or weights ($\hat{\boldsymbol{v}}_j$ in Eq. 2.8) are the parameters for every realization to be updated by assimilating the production data.

The inflation factors are determined a priori by setting the values in a descending order and at early iterations, much larger damping factors are used to ensure that the first several update steps will not be too large. We should probably note that in order to improve the robustness of the original ES-MDA, Le et al. (2015a) proposed an adaptive ES-MDA algorithm in which the inflation factor are chosen for the next data assimilation step automatically and adaptively as the history match proceeds. In this example, we performed the ES-MDA algorithm to calibrate the DCT coefficients by assimilating the observed data for 8 times ($N_a = 8$) with the inflation coefficients α_i specified as $\{400.0,\ 200.0,\ 100.0,\ 30.0,\ 15.0,\ 8.0,\ 4.0,\ 2.0\}$. After updating the DCT coefficients, we can reconstruct the facies field by using Eq. 2.17. Note that the reconstructed facies field only contains continuous values instead of discrete

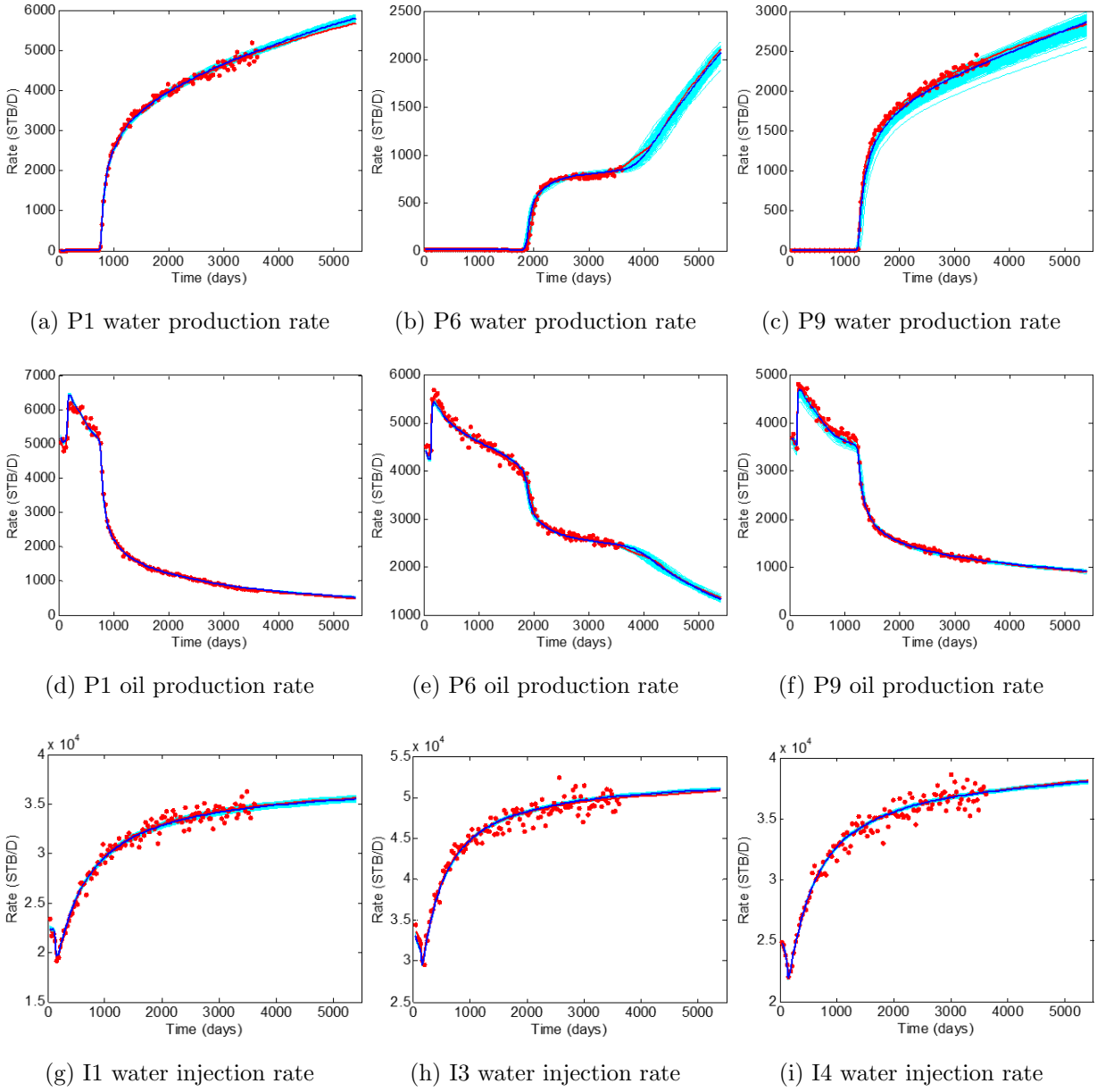
facies indicators, so the post-processing procedure is employed subsequently to remap the continuous values in order to obtain the updated facies field which better reflects the geological realism of the reservoir. Since the permeability and porosity within each facies are assumed to be homogeneous and hold fixed, therefore it's easy to obtain the updated permeability and porosity field for the next simulation. Figure 2.10 shows that the mean of ensemble members has captured the main geological features of the true model (Fig. 2.7) and preserved the continuity of most channels. Besides that, the isolation of I1 and P4 from other wells is also clearly described in the posterior ensemble. The STD of ln-permeability field in Fig. 2.10 indicates that the proposed workflow has significantly reduced the geological uncertainty associated with the prior models, and most of the remaining variability is around the boundaries of channels far from the wells.



(a) Posterior mean (facies) (b) Posterior mean ($\ln(k)$) (c) Posterior STD ($\ln(k)$)

Figure 2.10: Mean and STD of the posterior ensemble (Example 1)

Figure 2.11 shows the production data obtained with the posterior ensemble. Compared to the prior results (see Fig. 2.9), we obtain very good history data matches and a low level of uncertainty with the posterior ensemble. The forecast production data of the true model (red curve) also lie within the band formed by the results of the posterior realizations, and only the water production rate of P1 deviates slightly from the ensemble average (dark blue curve) by the end of the predication period.



(a) P1 water production rate (b) P6 water production rate (c) P9 water production rate

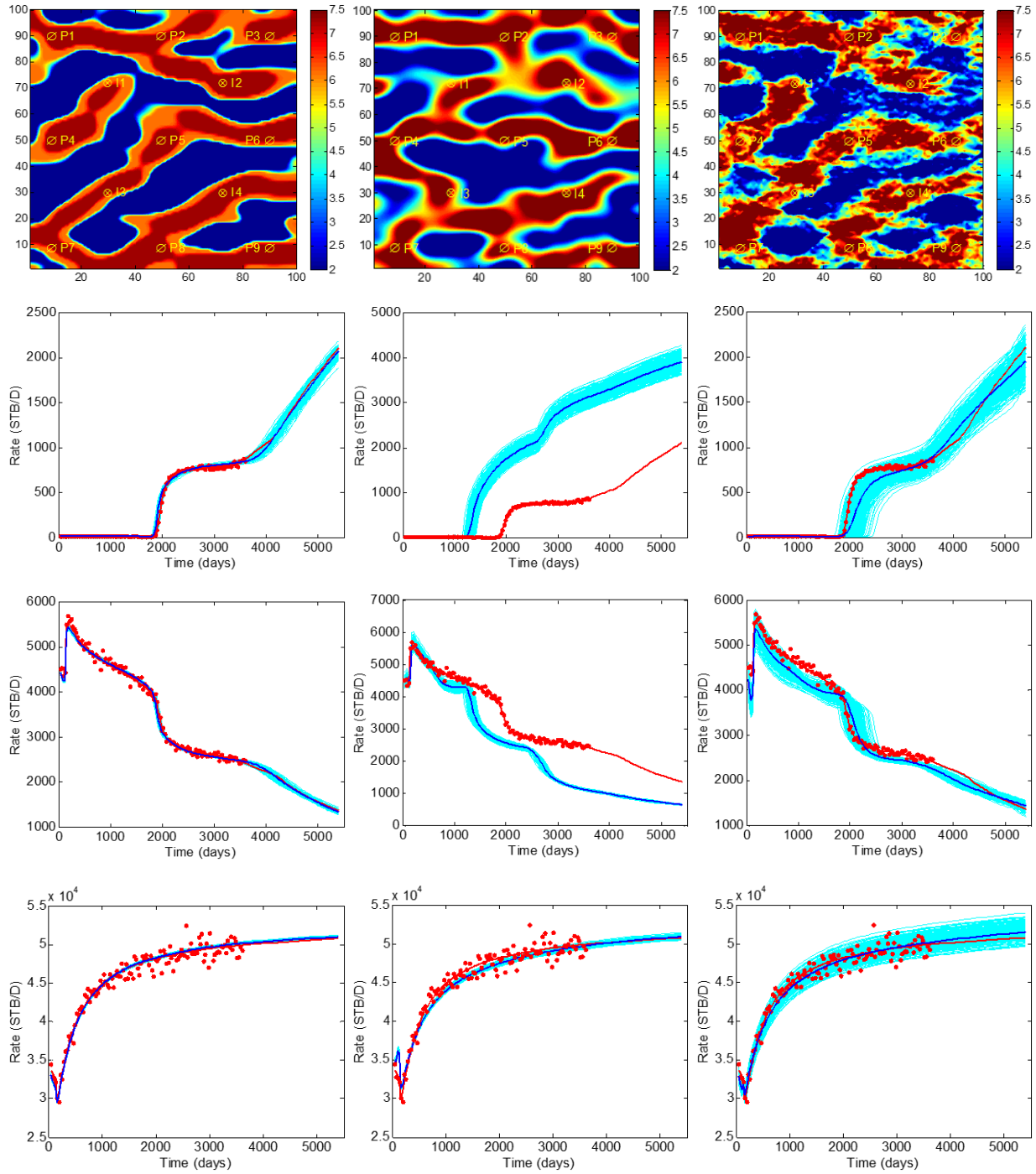
(d) P1 oil production rate (e) P6 oil production rate (f) P9 oil production rate

(g) I1 water injection rate (h) I3 water injection rate (i) I4 water injection rate

Figure 2.11: Posterior production data (Example 1). True (red curves), observed data (red dots), simulated data of realizations (light blue curves), and ensemble mean (dark blue curves). History: $0 < t \leq 3600$ days, forecast: $t > 3600$ days.

To evaluate the parameterization performance of common basis DCT, we compare it with the truncated basis and complete DCT for the same history matching problem and the comparison of results is shown in Fig. 2.12. Different from the implementation of common basis DCT, we retrain N_c DCT coefficients in the triangular area for the truncated basis DCT implementation and keep all the DCT coefficients for the complete DCT implementation,

then employ ES-MDA algorithm to calibrate the specific coefficients for every model by assimilating the production data. As a common practice in the literature, the truncated basis DCT and complete DCT are used to parameterize the reservoir property fields (i.e., permeability and porosity fields) instead of the facies field as is done with the common basis DCT. In the history matching process, the DCT coefficients of permeability and porosity fields are updated through ES-MDA iterations using Eq. 2.14 separately. After the update, estimated coefficients of each model are multiplied with the corresponding set of basis functions in order to obtain the renewed reservoir property fields for next iteration. In Fig. 2.12, the column (a) is the results obtained by implementing the common basis DCT with the retained number of 200 ($N_c = 200$), the column (b) denotes the results of truncated basis DCT with the same retained number and the column (c) represent the results of complete DCT which retain all the basis functions ($N_c = N_m$) in the history matching process. From the first row, we can clearly see that the estimated ln-permeability field obtained by the truncated basis DCT cannot well preserve the geological features, such as the continuous channels, of the reservoir, and the boundaries between different rock facies are also smoothed out to some extent. Although the updated models obtained by the complete DCT can yield the correct locations of high-permeability channels, the distinctions between levee and sand facies have been seriously distorted. The characteristics of geological structures can be confirmed by the data match results shown in Fig. 2.12. With the truncated basis DCT, the updated models give a bad data match for water and oil production rates at P6 (see the second column), and the result of the true model (red curve) is not even within the range formed by the posterior ensemble (light blue curves). Compared to the results obtained by the truncated basis DCT, it is shown that the data matches appear to be much better when we use the complete DCT to preserve more detailed information (see the third column). In addition, this implementation can provide more variability in the posterior ensemble, but the storage of all DCT coefficients and computational cost needs to be considered in the practical applications, and increasing the number of parameters may make the inverse problem more ill-conditioned.



(a) Common basis DCT ($N_c = 200$) (b) Truncated basis DCT ($N_c = 200$) (c) Complete DCT ($N_c = N_m$)

Figure 2.12: Comparison of results obtained by different parameterization schemes (Example 1): ensemble mean of $\ln(k)$ field (1st row), P6 water production rate (2nd row), P6 oil production rate (3rd row) and I3 water injection rate (4th row).

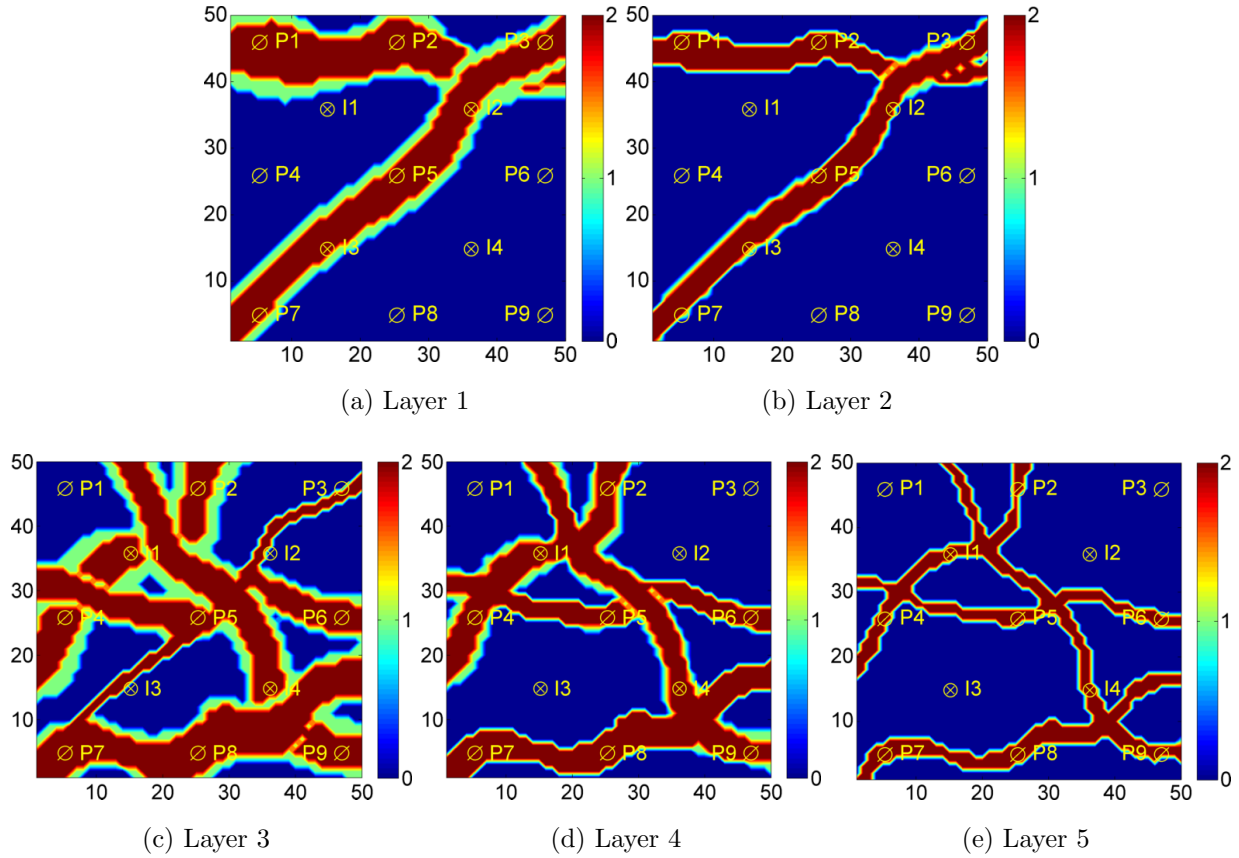
2.4.2 Example 2: 3D Three-Facies Reservoir Model

In this example, we consider a five-layer three-facies reservoir model defined on a 3D $50 \times 50 \times 5$ grid system and the gridblock size is 100ft $\times$ 100ft $\times$ 10ft. The five layers are divided into two geological zones with different channel patterns. To be specific, layers 1 and 2 belong to Zone 1, and the bottom three layers belong to Zone 2. The facies distributions of the prior ensemble members and the true model are generated by using the object-based modeling with the facies at each well location observed and used as hard data. The ln-permeability and porosity fields are assumed to be homogeneous within each facies. The properties of each facies are the same as shown in Table 2.2.

There are four injectors and nine producers arranged in a five-spot pattern in the reservoir. All wells perforate five reservoir layers and the facies observations at the well locations are given in Table 2.3. The injection wells are constrained to a constant bottom hole pressure (BHP) of 5500 psi, and the production wells also operate under constant BHP control of 3000 psi. The reservoir has uniform initial pressure and water saturation equal to 5000 psi and 0.2, respectively. The history matching period is 1500 days and all wells start operating at the beginning of the simulation. The production data are collected every 30 days during this period and they correspond to water injection rate (q_{inj}) of the injectors and oil flow rate (q_{o}) and water flow rate (q_{w}) of the producers. In this synthetic example, the observed data are obtained by adding uncorrelated Gaussian random noise to the simulation results of the true model with a noise level equal to 3%. Similar to the 2D problem, the form of the observed data $\boldsymbol{d}_{\mathrm{obs}}$ and the covariance matrix of observation error $\boldsymbol{C}_{\mathrm{D}}$ are written as Eqs. 2.26 and 2.27. After the history matching, the updated models are used to predict the reservoir performance for additional forecast period of 1500 days. The model parameters which are estimated through the data assimilation process are still the DCT coefficients corresponding to the set of common basis functions and the facies distribution of the true model as well as the well locations are shown in Fig. 2.13. It should be noted that the common basis DCT is applied in a layer-by-layer manner to this example.

Table 2.3: Hard data at the well locations (Example 2)

	Facies	Wells
Zone 1	Sand	I2, I3, P1, P2, P3, P5, P7
Zone 2	Sand	I1, I4, P2, P4, P5, P6, P7, P8, P9



(a) Layer 1 (b) Layer 2

(c) Layer 3 (d) Layer 4 (e) Layer 5

Figure 2.13: Facies fields of the true model (Example 2)

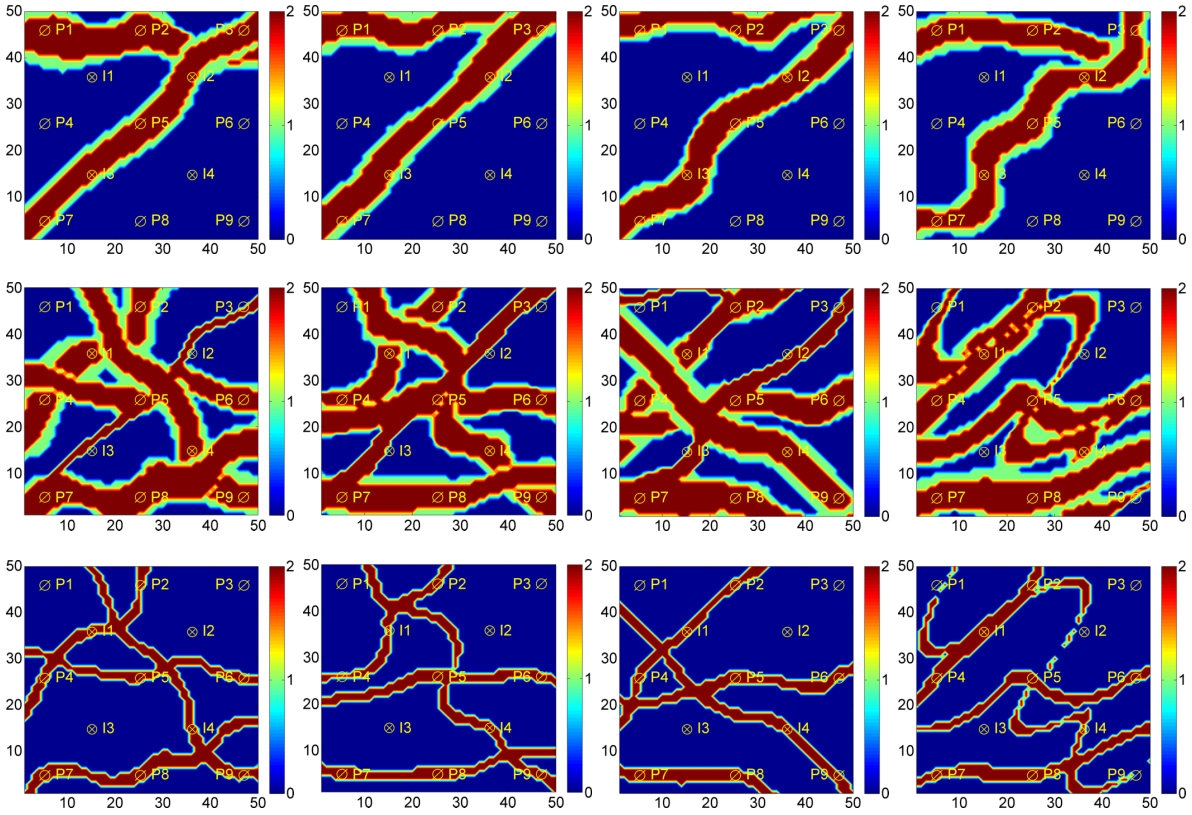
From Fig. 2.13, we can see that in the same geological zone, the facies distributions and orientation of channels appear to be similar throughout different layers. For instance, the two main channels in layer 2 are actually the bottom parts of the channels in layer 1, and it is the same case for layers 3 to 5. In Zone 1 (layers 1 and 2), five production wells (P1, P2, P3, P5 and P7) and two injection wells (I2 and I3) perforate the sand facies of the two channels. And in Zone 2 (layers 3 to 5), seven production wells (P2, P4 to P9) and two injection wells (I1 and I4) perforate through several overlapping channels. By stochastic

object-based modeling, we are able to control the width, thickness and fraction of each facies to construct the interrelated geological zones.

In this example, we implemented the common basis DCT to do parameterization and conducted history matching with the ES-MDA algorithm. To investigate the influence of ensemble size and retained number of basis functions, we designed three cases as described in Table 2.4. Note that the value of N_c in Table 2.4 denotes the retained number of basis functions for each layer. Figure 2.14 shows facies fields of three layers of the true model and three unconditional realizations in the prior ensemble. Although the prior realizations honor the hard data at the well locations, the sinuosity and distributed patterns of complex channels in the prior ensemble still differ greatly from those in the true model. Therefore, it is challenging to obtain an updated model that can accurately represent the actual distribution of rock facies and achieve a good history data match. Correspondingly, the prior predicted data are shown in Fig. 2.15 from which we can see that the production data of prior realizations (light blue curves) cover a wide range, indicating the prior uncertainty is very large.

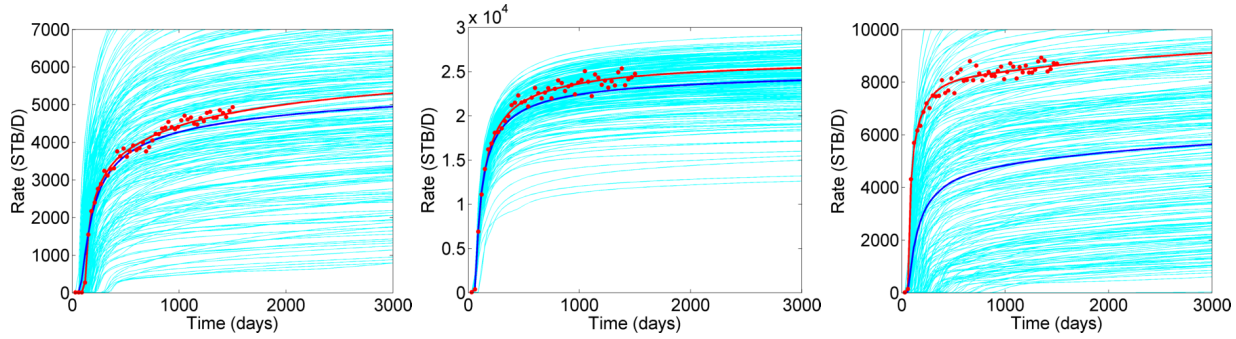
Table 2.4: Descriptions of three cases (Example 2)

Case	N_e	N_c
2.1	200	100
2.2	500	100
2.3	500	200

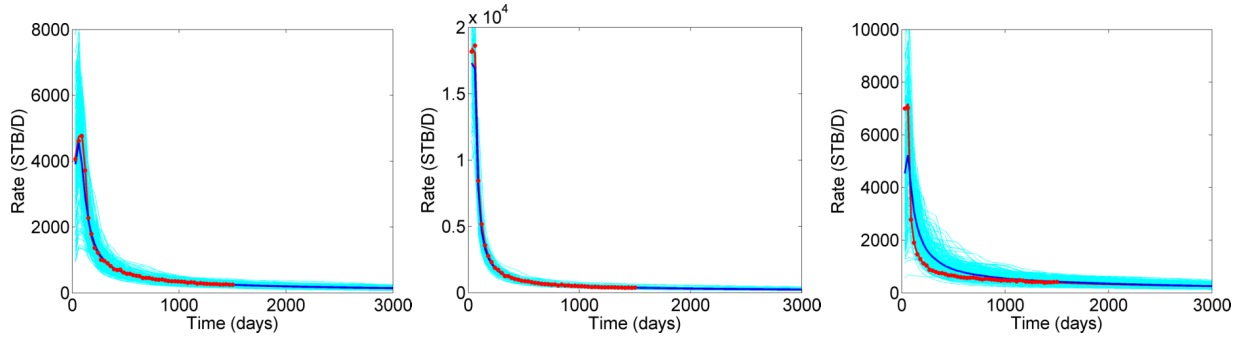


(a) True (b) Prior model #1 (c) Prior model #2 (d) Prior model #3

Figure 2.14: Facies fields of the true model and three prior realizations (Example 2): layer 1 (1st row), layer 3 (2nd row) and layer 5 (3rd row).



(a) P1 water production rate (b) P5 water production rate (c) P8 water production rate

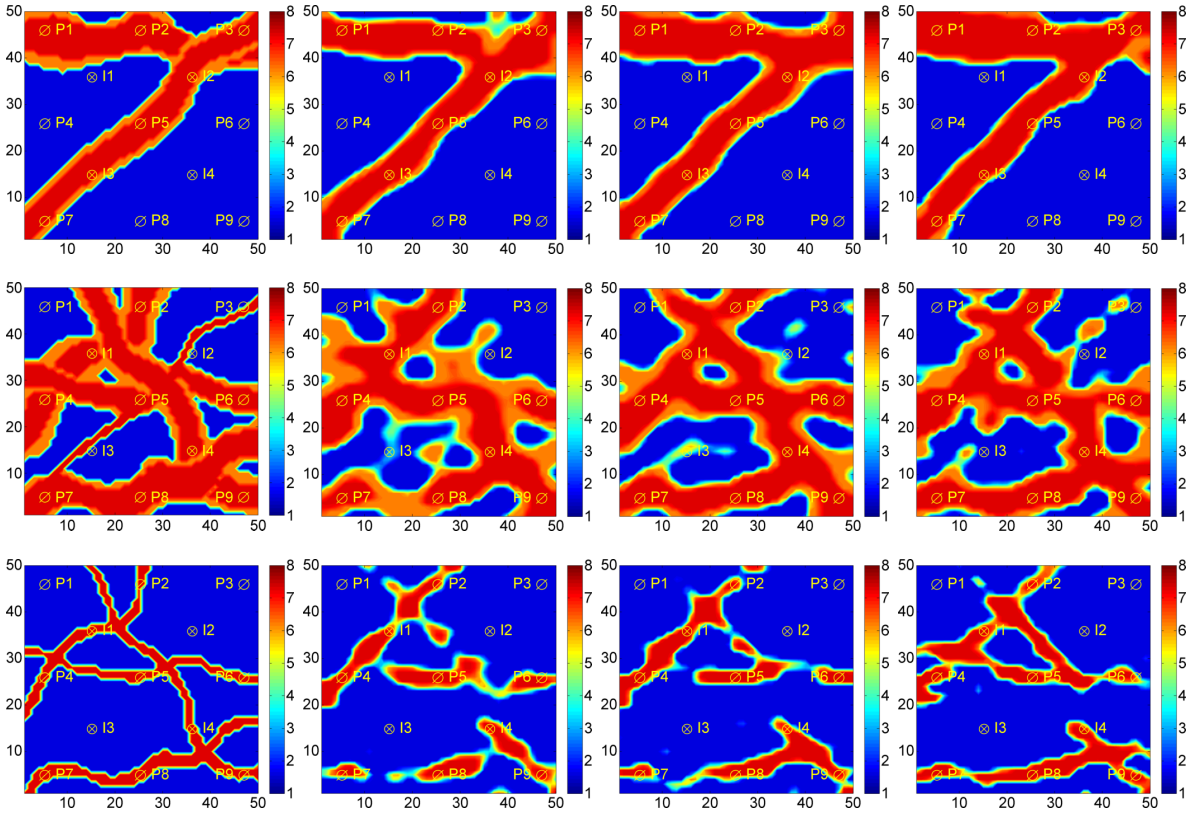


(d) P1 oil production rate (e) P5 oil production rate (f) P8 oil production rate

Figure 2.15: Prior production data (Example 2). True (red curves), observed data (red dots), simulated data of realizations (light blue curves), and ensemble mean (dark blue curves). History: $0 < t \leq 1500$ days, forecast: $t > 1500$ days.

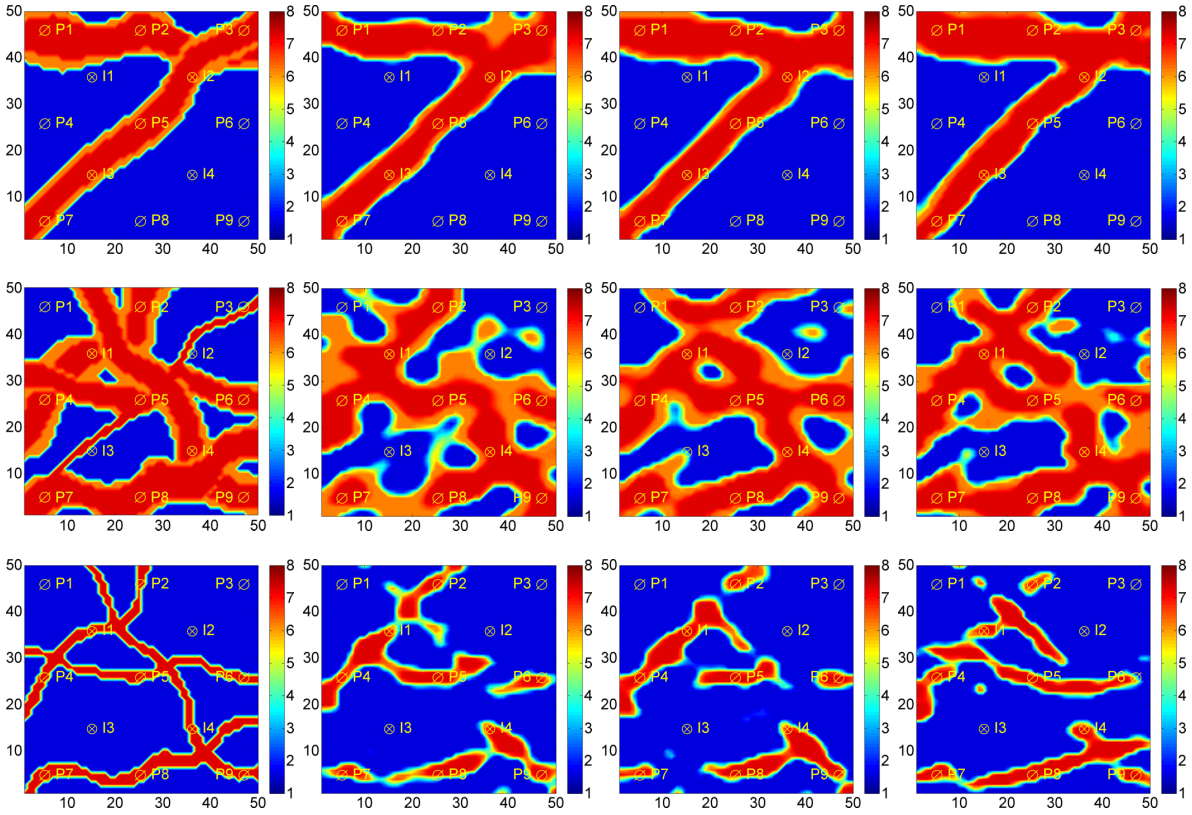
The number of data assimilation step in the ES-MDA algorithm is set to $N_a = 8$ and the corresponding inflation coefficients, α_i, are $\{400.0,\ 200.0,\ 100.0,\ 30.0,\ 15.0,\ 8.0,\ 4.0,\ 2.0\}$. Figures 2.16 and 2.17 present the $\ln(k)$ fields of the true model and two posterior realizations obtained with different ensemble sizes (N_e) and retained number of DCT basis functions (N_c). It is shown that the posterior realizations obtained for different values of N_e and N_c are able to capture the main geological structures of channelized facies in the true model. The continuity of channels (especially in layer 3) obtained with $N_e = 200$ is not as good as those obtained with a larger ensemble ($N_e = 500$). But, for different values of N_e and N_c considered, we are unable to resolve the two very thin diagonal channels in layer 3 of the true reservoir. When the ensemble size is fixed at 500 and we increase the retained number of basis functions from 100 to 200, more geological details are resolved through history matching (see the 3rd rows in Figs. 2.16 and 2.17) since more high frequency basis functions are re-

tained after parameterization. But for layer 5, the narrow channels are somewhat marred in all cases, and the reason may come from the fact that even though the wells perforate all five layers, the narrow channels in layer 5 contribute very little to the production data and cannot be resolved appropriately through the history matching process. Table 2.5 gives the facies proportions of the true, prior and posterior models. Note that even with the post-processing procedure after every iteration of ES-MDA, there still remains a small number of gridblocks with continuous facies type indicators and the fraction of these gridblocks is referred to as "Unresolved Facies" in the table. Therefore, the facies proportions in the bottom three rows are actually normalized proportions with respect to the number of gridblocks with discrete facies type indicators. Compared to the prior values, the average facies proportions of sand and shale for the posterior ensemble obtained in Case 2.2 ($N_e = 500$, $N_c = 100$) and Case 2.3 ($N_e = 500$, $N_c = 200$) are shown to be in reasonable range after history matching whereas less sand facies are resolved and the proportion of shale facies is overestimated in Case 2.1 ($N_e = 200$, $N_c = 100$). On the other hand, the levee facies proportion is closest to the true proportion for the $N_e = 200$, $N_c = 100$ case.



(a) True (b) $N_e = 200$, $N_c = 100$ (c) $N_e = 500$, $N_c = 100$ (d) $N_e = 500$, $N_c = 200$

Figure 2.16: $\ln(k)$ fields of the true model and posterior realization #1 (Example 2): layer 1 (1st row), layer 3 (2nd row) and layer 5 (3rd row).



(a) True (b) $N_e = 200$, $N_c = 100$ (c) $N_e = 500$, $N_c = 100$ (d) $N_e = 500$, $N_c = 200$

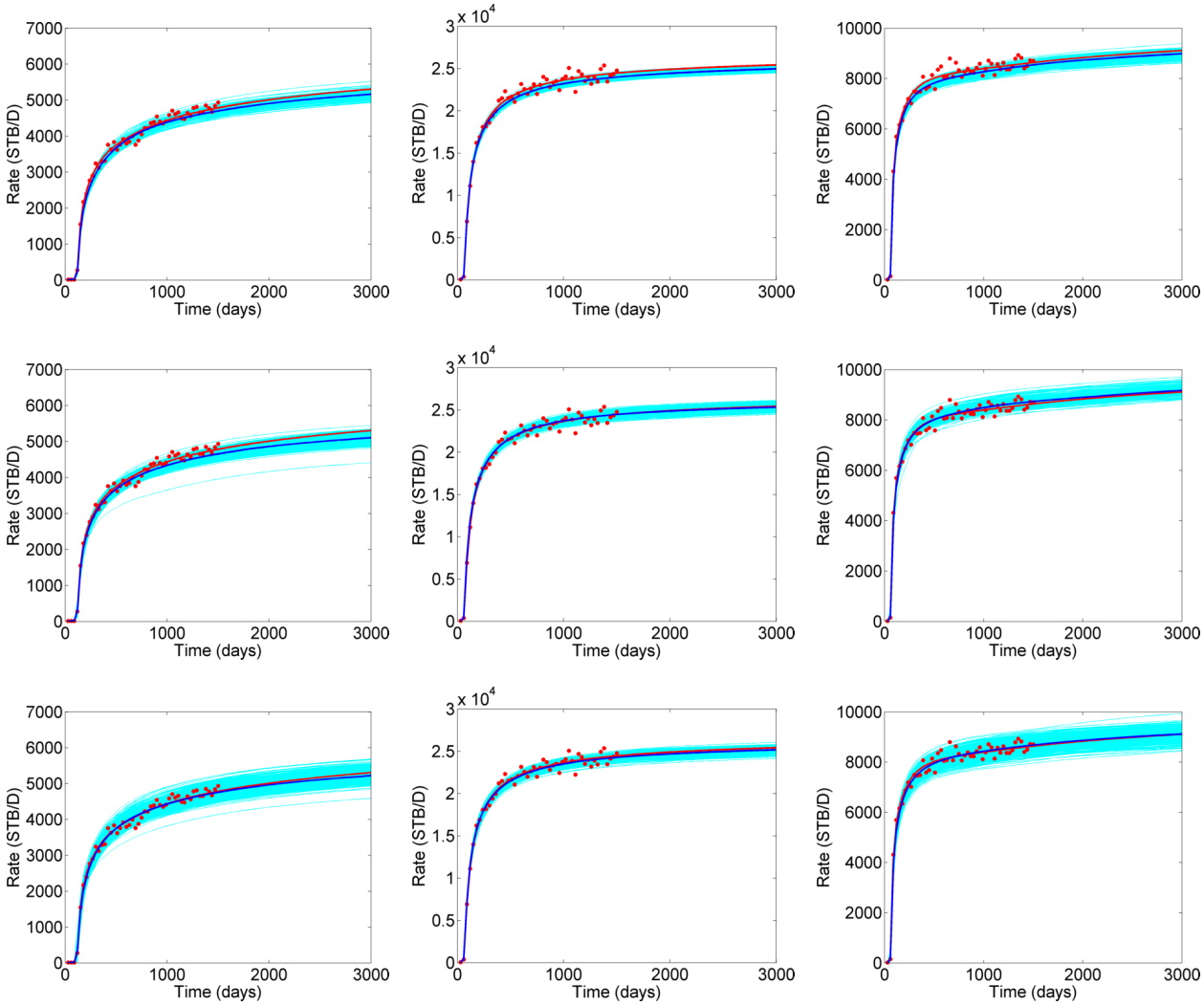
Figure 2.17: $\ln(k)$ fields of the true model and posterior realization #2 (Example 2): layer 1 (1st row), layer 3 (2nd row) and layer 5 (3rd row).

Table 2.5: Facies proportions (Example 2)

	Sand	Levee	Shale	Unresolved Facies
True	0.263	0.084	0.653	/
Prior	0.260	0.076	0.664	/
Posterior ($N_e = 200$, $N_c = 100$)	0.201	0.081	0.718	0.163
Posterior ($N_e = 500$, $N_c = 100$)	0.250	0.061	0.689	0.124
Posterior ($N_e = 500$, $N_c = 200$)	0.243	0.062	0.695	0.127

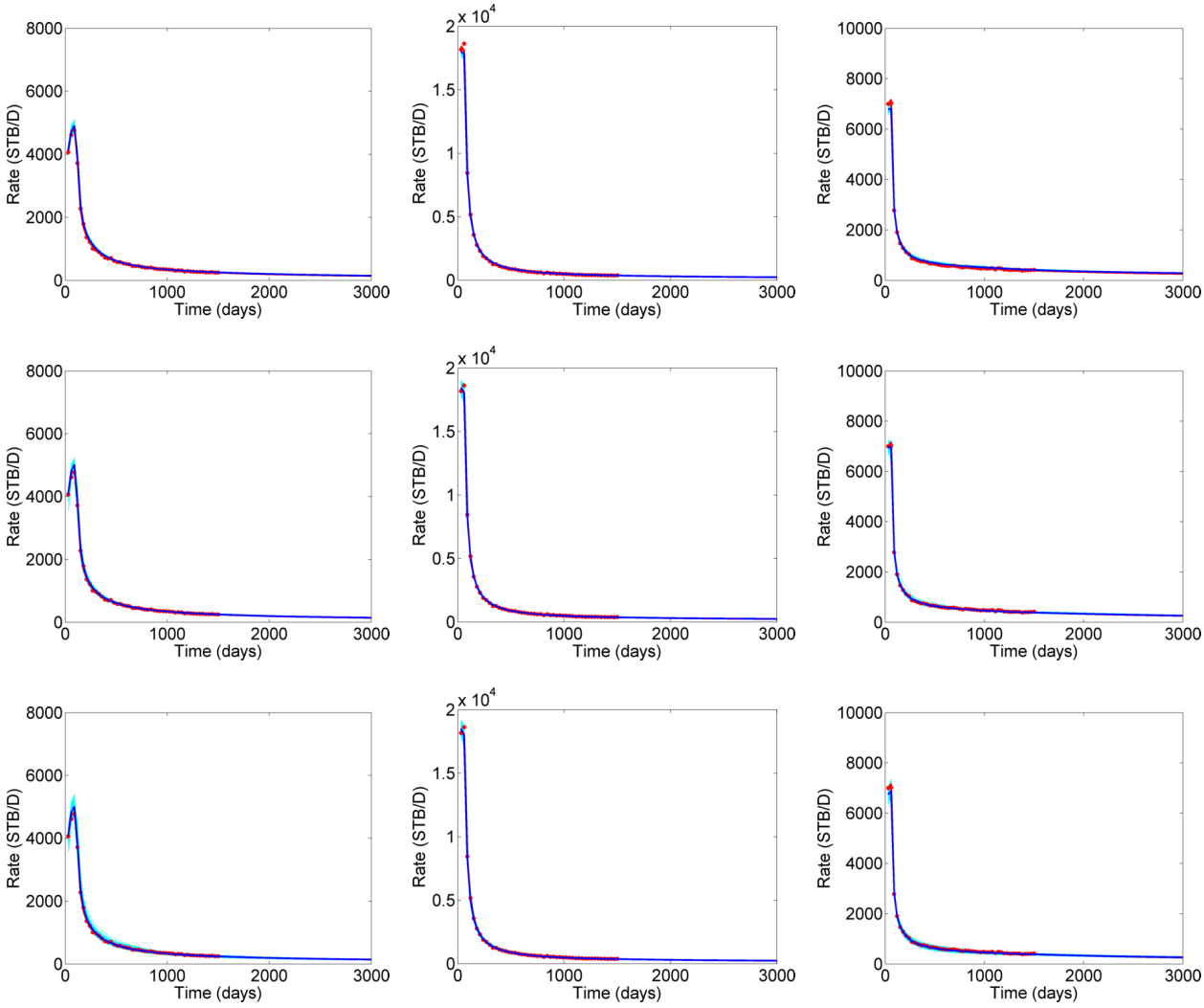
In Figs. 2.18 and 2.19, we can see the posterior water and oil rates of P1, P5 and P8 in three cases. The uncertainty associated with the prior ensemble is shown to be reduced significantly through history matching. It is clear that more geological variability of

facies distributions are preserved with a large ensemble of 500 realizations and the posterior ensemble in Case 2.1 almost collapses to one model when $N_e = 200$.



(a) P1 (b) P5 (c) P8

Figure 2.18: Posterior water rates (Example 2): Case 2.1, $N_e = 200$, $N_c = 100$ (1st row); Case 2.2, $N_e = 500$, $N_c = 100$ (2nd row); Case 2.3, $N_e = 500$, $N_c = 200$ (3rd row). True (red curves), observed data (red dots), simulated data of realizations (light blue curves), and ensemble mean (dark blue curves). History: $0 < t \leq 1500$ days, forecast: $t > 1500$ days.



(a) P1 (b) P5 (c) P8

Figure 2.19: Posterior oil rates (Example 2): Case 2.1, $N_e = 200$, $N_c = 100$ (1st row); Case 2.2, $N_e = 500$, $N_c = 100$ (2nd row); Case 2.3, $N_e = 500$, $N_c = 200$ (3rd row). True (red curves), observed data (red dots), simulated data of realizations (light blue curves), and ensemble mean (dark blue curves). History: $0 < t \leq 1500$ days, forecast: $t > 1500$ days.

Figure 2.20 shows the normalized data mismatch of the posterior ensemble in three cases calculated by

$$O(\boldsymbol{d}) = \frac{1}{N_d} \left(\boldsymbol{d} - \boldsymbol{d}_{\text{obs}}\right)^{\text{T}} \boldsymbol{C}_{\text{D}}^{-1} \left(\boldsymbol{d} - \boldsymbol{d}_{\text{obs}}\right), \tag{2.28}$$

where $\boldsymbol{d}$ and $\boldsymbol{d}_{\text{obs}}$ only include production data during the historical period. It can be seen that the data matches obtained with $N_e = 200$ and $N_c = 100$ is the best but the variance

in the posterior realizations is very small. When we increase the ensemble size to 500, there is more variability in the posterior ensemble and the data mismatch is slightly larger than that when $N_e = 200$. Although more geological details are resolved when $N_c = 200$, the continuity of key structures is also marred since more high frequency basis functions are retained which may lead to a larger data mismatch result in Case 2.3.

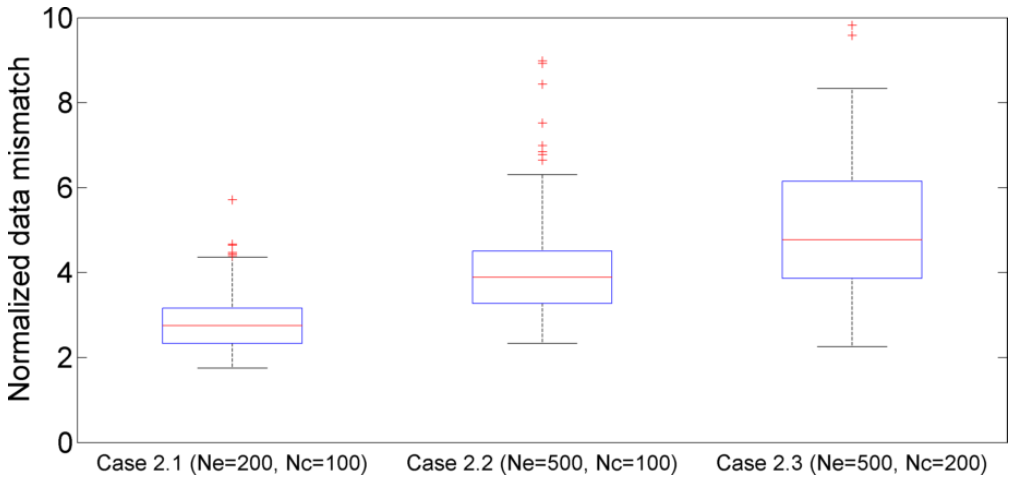


Figure 2.20: Normalized data mismatch (Example 2). Median (central red mark), 25th and 75th percentiles (bottom and top edges of the box), extreme data points (whiskers), and outliers ("+" symbol).

CHAPTER 3

SIMULTANEOUS CALIBRATION OF THE DISTRIBUTION OF FACIES AND PETROPHYSICAL PROPERTIES WITH NON-NEGATIVE MATRIX FACTORIZATION

As mentioned in the Introduction chapter, the methods based on the principal component analysis (PCA) and discrete cosine transform (DCT) methods are very effective in reducing the number of parameters by retaining the coefficients and corresponding basis functions which contain most critical information in the models. These methods have two characteristics: (1) They tend to obtain large-scale not parts-based representations of the original model and allow the components in the retained bases to be of arbitrary sign; (2) The bases arise directly from the prior models. On the other hand, the methods that use Fourier bases or wavelets (e.g., discrete wavelet transform (DWT)) extract localized features of the reservoir model but do not incorporate the prior ensemble. Lee and Seung (1999, 2001) proposed the non-negative matrix factorization (NMF), which includes non-negativity constraints for the purpose of obtaining localized representation of the original model by interpretable basis functions (Hoyer, 2004). More importantly, the information of prior ensemble can be learned and integrated through the NMF. Inspired by their work, here we investigate the properties of the NMF method and compare its performance with other parameterization approaches in a Bayesian inversion framework.

In this chapter, we first define the inversion problem based on an ensemble-based data assimilation algorithm. The NMF parameterization approach is presented and discussed in the subsequent section. Then, a brief review of the post-processing technique is provided after which we show the whole history matching workflow. The methodology is tested on two synthetic reservoirs with complex geometry and we demonstrate the superior features of

the NMF compared to PCA-based and DCT-based approaches by considering computational examples.

3.1 Methodology

3.1.1 Data Assimilation Algorithm

To adjust the model parameters to honor historical production data, the ensemble smoother with multiple data assimilation (ES-MDA) algorithm (Emerick and Reynolds, 2012, 2013a) is employed. In the original ES-MDA, same observed data are assimilated N_a times but with the measurement error covariance matrix $\boldsymbol{C}_{\mathrm{D}}$ replaced by $\alpha_\ell \boldsymbol{C}_{\mathrm{D}}$ at the ℓth assimilation step where the inflation factor $\alpha_\ell \geq 1$ and

$$\sum_{\ell=1}^{N_a} \frac{1}{\alpha_\ell} = 1. \tag{3.1}$$

For the linear Gaussian case, condition in Eq. 3.1 ensures that ES-MDA does the sampling correctly (Emerick and Reynolds, 2012, 2013a). Since ES-MDA was proposed, much work on the algorithm has focused on how to choose the inflation factors. Here, we use the ES-MDA-GEO (Rafiee and Reynolds, 2017) algorithm where the "GEO" refers to the fact that after the largest inflation factor α_1 is chosen, all other factors are calculated by

$$\alpha_{\ell+1} = \beta \alpha_\ell, \quad \text{for } \ell = 1, \cdots, N_a - 1, \tag{3.2}$$

where the factor β is determined so that Eq. 3.1 is satisfied. As in Rafiee and Reynolds (2017), the inflation factors meet the discrepancy principle which was first advocated by Iglesias (2015) to choose inflation factors in an iterative ensemble smoother algorithm. More details about the ES-MDA-GEO can be found in Appendix A.1. Although the discrepancy method has a theoretical basis, ES-MDA-GEO only enforces the discrepancy principle at the first data assimilation step of ES-MDA and hence is partially heuristic. In fact, how best to choose inflation factors in ES-MDA is an unsolved problem and is not a focus of this work.

We do, however, ensure we use a consistent procedure for choosing the inflation factors when we compare history matching results for different parameterizations of the reservoir models.

Here we calibrate the facies and the permeability within each facies simultaneously, i.e., the vector of model parameters, $\boldsymbol{y}$, is given by

$$\boldsymbol{y} = [\boldsymbol{m}_{\text{facies}}^{\text{T}}, \boldsymbol{m}_{\text{perm}}^{\text{T}}]^{\text{T}}, \tag{3.3}$$

where $\boldsymbol{m}_{\text{facies}}$ and $\boldsymbol{m}_{\text{perm}}$ represent the model parameters associated with the facies field and the permeability within each facies, respectively. We denote the number of gridblocks in the reservoir model by N_g and the number of specific facies by N_f. Since the parameterization is applied to the facies indicator field in this work, $\boldsymbol{m}_{\text{facies}}$ can be replaced by a reduced N_c-dimensional vector, $\boldsymbol{v}$, which is obtained with a parameterization approach where $N_c \ll N_g$. Besides that, $\boldsymbol{m}_{\text{perm}}$ is a $N_f N_g$-dimensional vector containing the ln-permeability ($\ln(k)$) values within facies. Thus, the dimension of $\boldsymbol{y}$ is $N_y = N_c + N_f N_g$.

Here we consider three-facies (i.e., $N_f = 3$) systems where shale, levee and sand are represented by 0, 1 and 2, respectively. Then $\boldsymbol{m}_{\text{perm}}$ is given by

$$\boldsymbol{m}_{\text{perm}} = [\boldsymbol{m}_{\text{sand}}^{\text{T}}, \boldsymbol{m}_{\text{levee}}^{\text{T}}, \boldsymbol{m}_{\text{shale}}^{\text{T}}]^{\text{T}}, \tag{3.4}$$

where $\boldsymbol{m}_{\text{sand}}$, $\boldsymbol{m}_{\text{levee}}$ and $\boldsymbol{m}_{\text{shale}}$ denote N_g-dimensional vector containing $\ln(k)$ values within sand, levee and shale facies, respectively. Therefore, the vector of model parameters given in Eq. 3.3 can be rewritten as

$$\boldsymbol{y} = [\boldsymbol{v}^{\text{T}}, \boldsymbol{m}_{\text{sand}}^{\text{T}}, \boldsymbol{m}_{\text{levee}}^{\text{T}}, \boldsymbol{m}_{\text{shale}}^{\text{T}}]^{\text{T}}. \tag{3.5}$$

3.1.2 Non-Negative Matrix Factorization (NMF)

Formulation:

Before presenting the detailed formulation of parameterization approach, we first

define the $N_g \times N_{te}$ training ensemble matrix $\boldsymbol{M}$ as

$$\boldsymbol{M} = [\boldsymbol{m}_{\text{facies},1}, \boldsymbol{m}_{\text{facies},2}, \cdots, \boldsymbol{m}_{\text{facies},j}, \cdots, \boldsymbol{m}_{\text{facies},N_{te}}], \tag{3.6}$$

where N_{te} denotes the training ensemble size and $\boldsymbol{m}_{\text{facies},j}$ is a N_g-dimensional vector containing the facies type indicator in each gridblock of the jth ensemble member.

Through parameterization, the discrete facies indicators are converted to continuous variables based on the following matrix factorization framework:

$$\boldsymbol{M} \approx \hat{\boldsymbol{M}} = \boldsymbol{\Phi}\boldsymbol{V}, \tag{3.7}$$

where $\hat{\boldsymbol{M}} \in \mathbb{R}^{N_g \times N_{te}}$ is the approximate estimation of the ensemble matrix $\boldsymbol{M} \in \mathbb{R}^{N_g \times N_{te}}$; $\boldsymbol{\Phi} \in \mathbb{R}^{N_g \times N_c}$ and $\boldsymbol{V} \in \mathbb{R}^{N_c \times N_{te}}$, respectively, represent the basis matrix and coefficient matrix given respectively by

$$\boldsymbol{\Phi} = [\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \cdots, \boldsymbol{\phi}_k, \cdots, \boldsymbol{\phi}_{N_c}], \tag{3.8}$$

$$\boldsymbol{V} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_j, \cdots, \boldsymbol{v}_{N_{te}}], \tag{3.9}$$

where $\boldsymbol{\phi}_k$ is a N_g-dimensional basis vector (i.e., $\boldsymbol{\phi}_k = [\phi_{k,1}, \phi_{k,2}, \cdots, \phi_{k,N_g}]^{\mathrm{T}}$) in the kth column of the basis matrix $\boldsymbol{\Phi}$ and $\boldsymbol{v}_j$ is a N_c-dimensional coefficient vector of the jth ensemble member (i.e., $\boldsymbol{v}_j = [v_{j,1}, v_{j,2}, \cdots, v_{j,N_c}]^{\mathrm{T}}$). As a result, each column of the training ensemble matrix of Eq. 3.6 can be approximated by

$$\boldsymbol{m}_{\text{facies},j} \approx \hat{\boldsymbol{m}}_{\text{facies},j} = \sum_{k=1}^{N_c} v_{j,k}\boldsymbol{\phi}_k = \boldsymbol{\Phi}\boldsymbol{v}_j, \quad \text{for } j = 1, \cdots, N_{te}. \tag{3.10}$$

Because the entries in the matrix $\boldsymbol{M}$ are facies type indicators which are all positive, the distinct requirement of NMF that the components in the resulting $\boldsymbol{\Phi}$ and $\boldsymbol{V}$ must be all positive as well is appropriate. To obtain a factorization which satisfies this constraint, we

minimize the cost function given by

$$O(\boldsymbol{\Phi}, \boldsymbol{V}) = \frac{1}{2} \left\| \boldsymbol{M} - \boldsymbol{\Phi}\boldsymbol{V} \right\|_{\mathrm{F}}^2 , \tag{3.11}$$

subject to the constraints that

$$\boldsymbol{\Phi} \geq 0 \quad \text{and} \quad \boldsymbol{V} \geq 0. \tag{3.12}$$

In Eq. 3.11, the subscript "F" denotes the Frobenius norm, so we have

$$O(\boldsymbol{\Phi}, \boldsymbol{V}) = \frac{1}{2} \left\| \boldsymbol{M} - \boldsymbol{\Phi}\boldsymbol{V} \right\|_{\mathrm{F}}^2 = \frac{1}{2} \sum_{i=1}^{N_g} \sum_{j=1}^{N_{te}} \left(\boldsymbol{M}_{ij} - (\boldsymbol{\Phi}\boldsymbol{V})_{ij} \right)^2 , \tag{3.13}$$

where $\boldsymbol{M}_{ij}$ and $(\boldsymbol{\Phi}\boldsymbol{V})_{ij}$ denote the components in the ith row and jth column of the matrices $\boldsymbol{M}$ and $\boldsymbol{\Phi}\boldsymbol{V}$, respectively. In fact, $(\boldsymbol{\Phi}\boldsymbol{V})_{ij}$ can be rewritten as $\sum_{k=1}^{N_c} \boldsymbol{\Phi}_{ik}\boldsymbol{V}_{kj}$ then the derivative of $(\boldsymbol{\Phi}\boldsymbol{V})_{ij}$ with respect to $\boldsymbol{\Phi}_{ik}$ is

$$\frac{\partial (\boldsymbol{\Phi}\boldsymbol{V})_{ij}}{\partial \boldsymbol{\Phi}_{ik}} = \boldsymbol{V}_{kj}. \tag{3.14}$$

Therefore, the partial derivative of the cost function shown in Eq. 3.13 with respect to $\boldsymbol{\Phi}_{ik}$ is given by

$$\begin{aligned} \frac{\partial O(\boldsymbol{\Phi}, \boldsymbol{V})}{\partial \boldsymbol{\Phi}_{ik}} &= \sum_{j=1}^{N_{te}} \left[\boldsymbol{V}_{kj} \left((\boldsymbol{\Phi}\boldsymbol{V})_{ij} - \boldsymbol{M}_{ij} \right) \right] \\ &= \sum_{j=1}^{N_{te}} (\boldsymbol{\Phi}\boldsymbol{V})_{ij} \boldsymbol{V}_{kj} - \sum_{j=1}^{N_{te}} \boldsymbol{M}_{ij} \boldsymbol{V}_{kj} \\ &= \left(\boldsymbol{\Phi}\boldsymbol{V}\boldsymbol{V}^{\mathrm{T}} \right)_{ik} - \left(\boldsymbol{M}\boldsymbol{V}^{\mathrm{T}} \right)_{ik} . \end{aligned} \tag{3.15}$$

Similar to Eq. 3.14, the derivative of $(\boldsymbol{\Phi}\boldsymbol{V})_{ij}$ with respect to $\boldsymbol{V}_{kj}$ is

$$\frac{\partial (\boldsymbol{\Phi}\boldsymbol{V})_{ij}}{\partial \boldsymbol{V}_{kj}} = \boldsymbol{\Phi}_{ik}. \tag{3.16}$$

Then, the partial derivative of $O(\boldsymbol{\Phi}, \boldsymbol{V})$ with respect to $\boldsymbol{V}_{kj}$ is

$$\begin{aligned}\frac{\partial O(\boldsymbol{\Phi}, \boldsymbol{V})}{\partial \boldsymbol{V}_{kj}} &= \sum_{i=1}^{N_g} \left[\boldsymbol{\Phi}_{ik} \left((\boldsymbol{\Phi}\boldsymbol{V})_{ij} - \boldsymbol{M}_{ij}\right)\right] \\ &= \sum_{i=1}^{N_g} \boldsymbol{\Phi}_{ik}(\boldsymbol{\Phi}\boldsymbol{V})_{ij} - \sum_{i=1}^{N_g} \boldsymbol{\Phi}_{ik}\boldsymbol{M}_{ij} \\ &= \left(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\boldsymbol{V}\right)_{kj} - \left(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{M}\right)_{kj}.\end{aligned} \tag{3.17}$$

With the partial derivatives in Eqs. 3.15 and 3.17, a steepest gradient descent method as in Lee and Seung (2001) is implemented to minimize the cost function as follows:

$$\boldsymbol{\Phi}_{ik}^{\ell+1} = \boldsymbol{\Phi}_{ik}^{\ell} - \alpha_{ik}^{\ell} \frac{\partial O(\boldsymbol{\Phi}^{\ell}, \boldsymbol{V}^{\ell})}{\partial \boldsymbol{\Phi}_{ik}^{\ell}} = \boldsymbol{\Phi}_{ik}^{\ell} - \alpha_{ik}^{\ell} \left[\left(\boldsymbol{\Phi}^{\ell}\boldsymbol{V}^{\ell}(\boldsymbol{V}^{\ell})^{\mathrm{T}}\right)_{ik} - \left(\boldsymbol{M}^{\ell}(\boldsymbol{V}^{\ell})^{\mathrm{T}}\right)_{ik}\right], \tag{3.18}$$

$$\boldsymbol{V}_{kj}^{\ell+1} = \boldsymbol{V}_{kj}^{\ell} - \beta_{kj}^{\ell} \frac{\partial O(\boldsymbol{\Phi}^{\ell}, \boldsymbol{V}^{\ell})}{\partial \boldsymbol{V}_{kj}^{\ell}} = \boldsymbol{V}_{kj}^{\ell} - \beta_{kj}^{\ell} \left[\left((\boldsymbol{\Phi}^{\ell})^{\mathrm{T}}\boldsymbol{\Phi}^{\ell}\boldsymbol{V}^{\ell}\right)_{kj} - \left((\boldsymbol{\Phi}^{\ell})^{\mathrm{T}}\boldsymbol{M}^{\ell}\right)_{kj}\right], \tag{3.19}$$

where ℓ denotes the iteration index; α_{ik}^{ℓ} and β_{kj}^{ℓ} are the step size for updating $\boldsymbol{\Phi}_{ik}^{\ell}$ and $\boldsymbol{V}_{kj}^{\ell}$, respectively. However, Eqs. 3.18 and 3.19 cannot guarantee the non-negativity of $\boldsymbol{\Phi}$ and $\boldsymbol{V}$ after every iteration. To address this issue, we first rewrite the right-hand side of Eqs. 3.18 and 3.19 as

$$\boldsymbol{\Phi}_{ik}^{\ell+1} = \boldsymbol{\Phi}_{ik}^{\ell} - \alpha_{ik}^{\ell} \left(\boldsymbol{\Phi}^{\ell}\boldsymbol{V}^{\ell}(\boldsymbol{V}^{\ell})^{\mathrm{T}}\right)_{ik} + \alpha_{ik}^{\ell} \left(\boldsymbol{M}^{\ell}(\boldsymbol{V}^{\ell})^{\mathrm{T}}\right)_{ik}, \tag{3.20}$$

$$\boldsymbol{V}_{kj}^{\ell+1} = \boldsymbol{V}_{kj}^{\ell} - \beta_{kj}^{\ell} \left((\boldsymbol{\Phi}^{\ell})^{\mathrm{T}}\boldsymbol{\Phi}^{\ell}\boldsymbol{V}^{\ell}\right)_{kj} + \beta_{kj}^{\ell} \left((\boldsymbol{\Phi}^{\ell})^{\mathrm{T}}\boldsymbol{M}^{\ell}\right)_{kj}. \tag{3.21}$$

Lee and Seung (2001) proposed the multiplicative update rules to calculate the values of α_{ik}^{ℓ} and β_{kj}^{ℓ} by letting the first two items on the right-hand side of Eqs. 3.20 and 3.21 be zero as

$$\boldsymbol{\Phi}_{ik}^{\ell} - \alpha_{ik}^{\ell} \left(\boldsymbol{\Phi}^{\ell}\boldsymbol{V}^{\ell}(\boldsymbol{V}^{\ell})^{\mathrm{T}}\right)_{ik} = 0, \tag{3.22}$$

$$\boldsymbol{V}_{kj}^{\ell} - \beta_{kj}^{\ell} \left((\boldsymbol{\Phi}^{\ell})^{\mathrm{T}}\boldsymbol{\Phi}^{\ell}\boldsymbol{V}^{\ell}\right)_{kj} = 0. \tag{3.23}$$

With the above equations, the values of α_{ik}^{ℓ} and β_{kj}^{ℓ} can be obtained by

$$\alpha_{ik}^{\ell} = \frac{\boldsymbol{\Phi}_{ik}^{\ell}}{\left(\boldsymbol{\Phi}^{\ell}\boldsymbol{V}^{\ell}(\boldsymbol{V}^{\ell})^{\mathrm{T}}\right)_{ik}}, \tag{3.24}$$

$$\beta_{kj}^{\ell} = \frac{\boldsymbol{V}_{kj}^{\ell}}{\left((\boldsymbol{\Phi}^{\ell})^{\mathrm{T}}\boldsymbol{\Phi}^{\ell}\boldsymbol{V}^{\ell}\right)_{kj}}. \tag{3.25}$$

Substituting Eqs. 3.24 and 3.25, respectively, into Eqs. 3.18 and 3.19, respectively, yield the modified updating equations as

$$\boldsymbol{\Phi}_{ik}^{\ell+1} = \boldsymbol{\Phi}_{ik}^{\ell}\frac{\left(\boldsymbol{M}^{\ell}(\boldsymbol{V}^{\ell})^{\mathrm{T}}\right)_{ik}}{\left(\boldsymbol{\Phi}^{\ell}\boldsymbol{V}^{\ell}(\boldsymbol{V}^{\ell})^{\mathrm{T}}\right)_{ik}}, \tag{3.26}$$

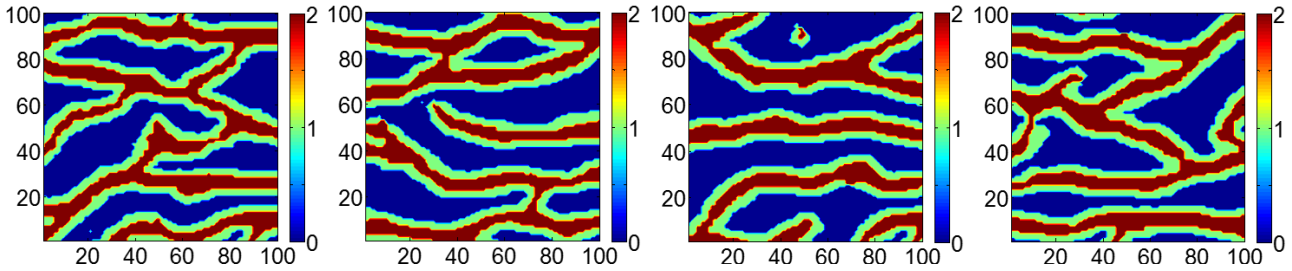
$$\boldsymbol{V}_{kj}^{\ell+1} = \boldsymbol{V}_{kj}^{\ell}\frac{\left((\boldsymbol{\Phi}^{\ell})^{\mathrm{T}}\boldsymbol{M}^{\ell}\right)_{kj}}{\left((\boldsymbol{\Phi}^{\ell})^{\mathrm{T}}\boldsymbol{\Phi}^{\ell}\boldsymbol{V}^{\ell}\right)_{kj}}. \tag{3.27}$$

It is obvious that every update can be guaranteed to be positive using Eqs. 3.26 and 3.27 which are iteratively applied until the difference between $\boldsymbol{\Phi V}$ and $\boldsymbol{M}$ reaches the stopping criteria. In the original paper of Lee and Seung (2001), they proved that the application of the update rules in Eqs. 3.26 and 3.27 is guaranteed to find at least locally optimal solution of the problem defined in Eqs. 3.11 and 3.12. The convergence proofs rely upon defining an appropriate auxiliary function similar to that used in the expectation-maximization algorithm (Dempster et al., 1977).

Properties:

A reconstruction experiment was conducted with a training ensemble of size $N_{te} = 1000$ to demonstrate the properties of NMF. Each ensemble member corresponds to the facies field of a channelized reservoir model containing three distinct facies, i.e., shale (0), levee (1) and sand (2). The reservoir model is based on a 100×100 grid system so that $N_g = 10,000$. It is straightforward to construct a $N_g \times N_{te}$ ensemble matrix $\boldsymbol{M}$ as defined in Eq. 3.6. Figure 3.1 presents four original facies models where we can observe a high number

of complex channels distributed in the reservoir. The retained number of NMF basis vectors is arbitrarily set to be $N_c = 200$ for model reduction but we provide a heuristic scheme for estimating N_c later. Through NMF, the original ensemble matrix is approximated by the product of the basis matrix, $\boldsymbol{\Phi} \in \mathbb{R}^{N_g \times N_c}$, and the coefficient matrix, $\boldsymbol{V} \in \mathbb{R}^{N_c \times N_{te}}$, as shown in Eq. 3.7. The approximate ensemble matrix, $\hat{\boldsymbol{M}} = \boldsymbol{\Phi}\boldsymbol{V} \in \mathbb{R}^{N_g \times N_{te}}$, contains reconstructed facies fields in its columns. Figure 3.2 shows the first 16 bases plotted on the same reservoir simulation grid used in the plots of Fig. 3.1. On the same scale (i.e., $[0, 2]$) as the reservoir models in Fig. 3.1, the dark blue pixels in the basis images are zero or very close to zero; thus, each basis can be regarded as a local patch of the large-scale geological structures (i.e., sand channels surrounded by levee facies embedded in the background shale matrix). In addition, we observe that some basis vectors contain relatively large features which can be treated as common characteristics presented in various models. Due to the localized properties, the separate patches are additively assembled together to generate the approximate model. From Fig. 3.3, it is shown that the salient geological structures (e.g., sinuous sand channels) are well reproduced in the reconstructed models obtained with the NMF approach. Note that there are continuous values not discrete facies indicators in the reconstructed models. Besides, the retained NMF coefficients are all guaranteed to be non-negative according to the distributions of coefficients in the bottom row of Fig. 3.3.



(a) Model #1 (b) Model #2 (c) Model #3 (d) Model #4

Figure 3.1: Three-facies channelized reservoir models: sand facies (red), levee facies (green) and shale facies (blue).

In the reconstruction experiment, we retain $N_c = 200$ basis vectors with 1000 training models. Through comparing the basis vectors obtained with different retained numbers

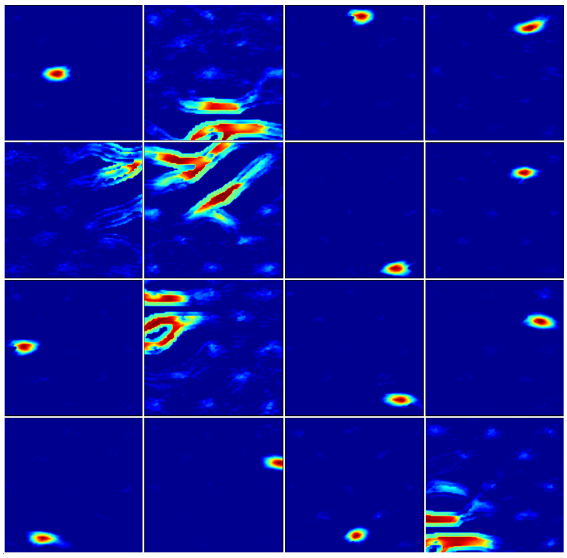


Figure 3.2: First 16 bases obtained by NMF ($N_c = 200,\ N_{te} = 1000$). The scale of basis images is $[0, 2]$ and dark blue pixels denote zero values.

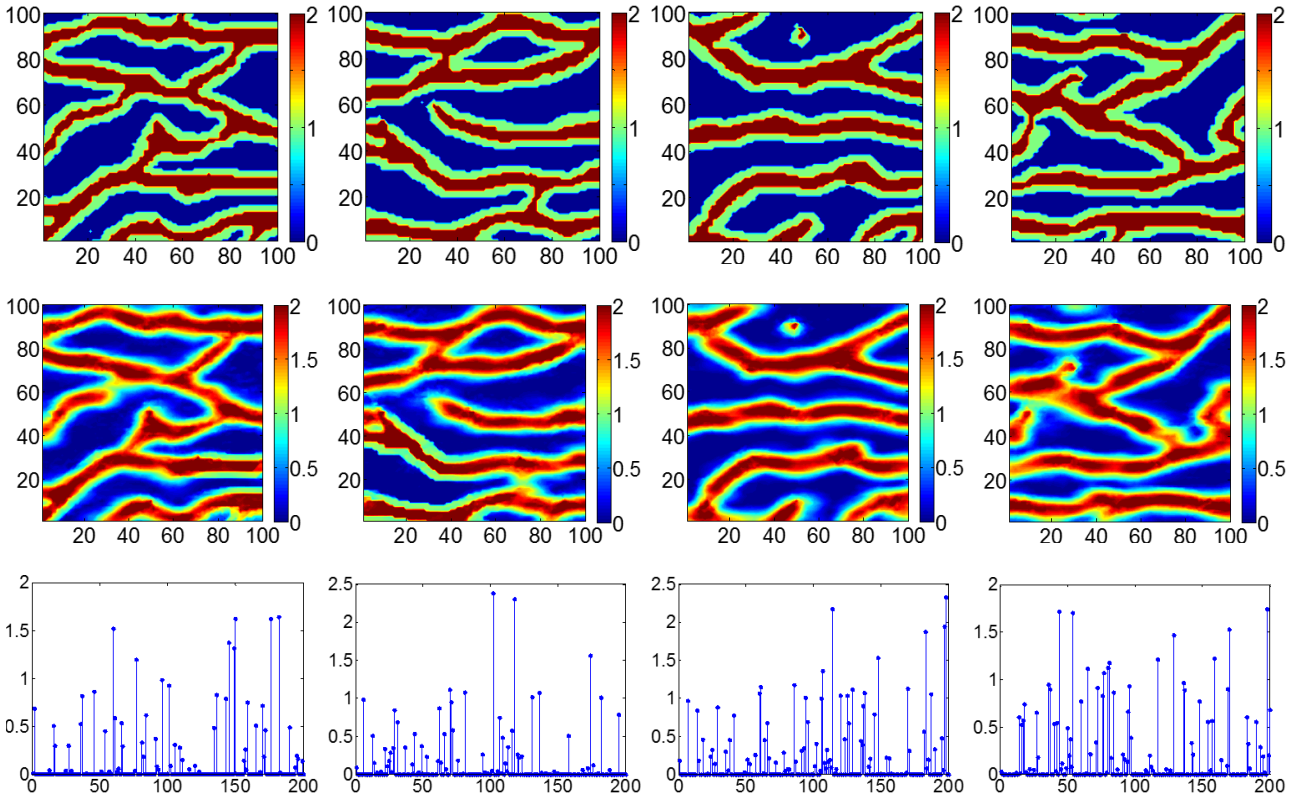
($N_c = \{50,\ 200,\ 600\}$) in Fig. 3.4, we can see that the geological features preserved in the basis vectors tend to be more localized when $N_c = 50$ and more relatively large-scale features appear in the leading basis vectors when $N_c = 600$. That is probably because when the number of retained basis vectors is small, we need more localized features to have flexibility and generalization capability in order to reconstruct different models whereas $N_c = 600$ is large enough so more large-scale and small-scale features are included in the basis vectors. This illustrates that the number of retained basis vectors, N_c, has an impact on the feature patterns of NMF bases when the training ensemble is fixed. However, a "good" choice of the retained number is problem-dependent and no clear criteria for selecting N_c has been presented in the literature. Below, we develop a scheme to determine the value of the retained number of basis vectors based on the reconstruction performance of NMF.

For use in our method to choose N_c, the model mismatch of the approximate model is defined as follows

$$\varepsilon = \left\| \boldsymbol{M} - \hat{\boldsymbol{M}} \right\|_{\mathrm{F}} = \left\| \boldsymbol{M} - \boldsymbol{\Phi}\boldsymbol{V} \right\|_{\mathrm{F}}, \tag{3.28}$$

where $\hat{\boldsymbol{M}}$ denotes the approximate $N_g \times N_{te}$ ensemble matrix.

Through more numerical experiments, we find that if the retained number is larger



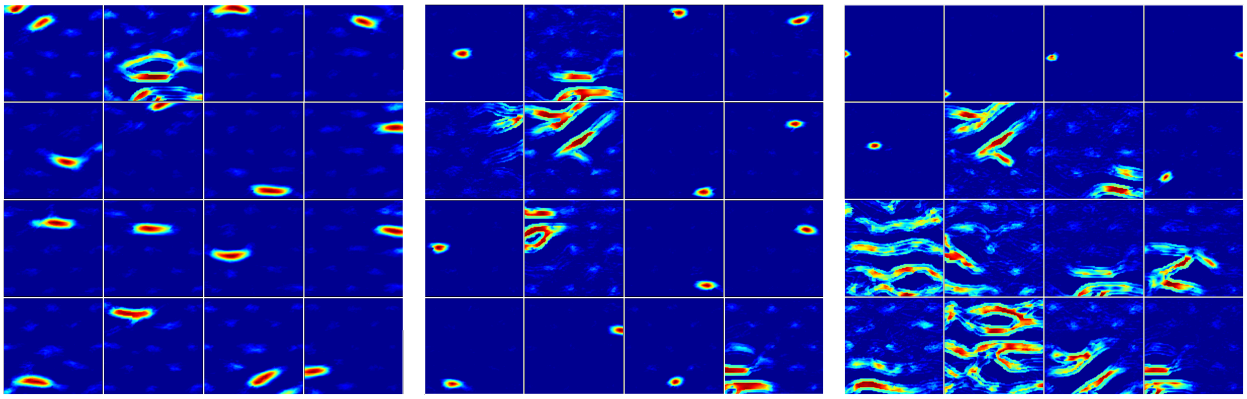
(a) Model #1 (b) Model #2 (c) Model #3 (d) Model #4

Figure 3.3: Reconstruction performance of NMF: original models (1st row), reconstructed models (2nd row) and coefficients distributions (3rd row).

than the training ensemble size, some basis vectors are very similar to specific original models. Such basis vectors have weak generality and cannot be used to represent models not in the training set. Moreover, the retained number should satisfy $N_c < N_{te}$ in order to ensure that the selected basis vectors are able to describe the common features throughout the whole ensemble and for computational efficiency. If a set of N_c basis functions can be used to accurately represent all models in a training set of size N_{te}, where $N_c < N_{te}$, then the bases can potentially represent any possible model. We define a normalized model mismatch by

$$\xi = \frac{\left\| \boldsymbol{M} - \hat{\boldsymbol{M}} \right\|_{\mathrm{F}}}{\left\| \boldsymbol{M} \right\|_{\mathrm{F}}}, \tag{3.29}$$

and conduct a sensitivity analysis of ξ with respect to the retained number, N_c, which is in the range of $[10, 1000]$. The upper bound of 1000 corresponds to the training ensemble size,



(a) $N_c = 50$ (b) $N_c = 200$ (c) $N_c = 600$

Figure 3.4: First 16 bases obtained with different retained number ($N_{te} = 1000$). The scale of basis images is $[0, 2]$ and dark blue pixels denote zero values.

N_{te}, and the step size change in N_c is set equal to 10 for the sake of efficiency in the analysis. In Fig. 3.5, it is shown that the normalized model mismatch declines rapidly as N_c increases. To guarantee the quality of reconstruction performance, a reasonable value of ξ is typically between 0.1 and 0.2, and more NMF basis vectors are retained with a smaller ξ.

For parameterization of the facies field, we apply the NMF approach as we did in the reconstruction experiment and obtain the coefficient vector. Then, the vector of model parameters, $\boldsymbol{y}$, is assembled with the NMF coefficient vector and three ln-permeability fields; see Eq. 3.5. We also compare the performance of two other parameterization approaches, i.e., OPCA (Vo and Durlofsky, 2014, 2015) and common basis DCT discussed in Chapter 2 (also see Zhao et al. (2016, 2017)) with the performance of NMF. Note that the three methods are referred to as "NMF", "PCA" and "DCT" here.

3.1.3 Post-Processing

As discussed in Chapter 2, when using common basis DCT and NMF, we apply slight modification of the OPCA post-processing procedure of Vo and Durlofsky (2014, 2015). Specifically, after the ℓth iteration of ES-MDA-GEO, the updated facies field, $\hat{\boldsymbol{m}}^{\ell+1}_{\text{facies},j}$, is computed with $\hat{\boldsymbol{v}}^{\ell+1}_j$ using Eq. 3.10. However, $\hat{\boldsymbol{m}}^{\ell+1}_{\text{facies},j}$ only contains continuous values; thus a post-processing procedure based on the work of Vo and Durlofsky (2014, 2015) is adapted to map $\hat{\boldsymbol{m}}^{\ell+1}_{\text{facies},j}$ to $\boldsymbol{m}^{\ell+1}_{\text{facies},j}$ where most facies type indicators become discrete. Although this

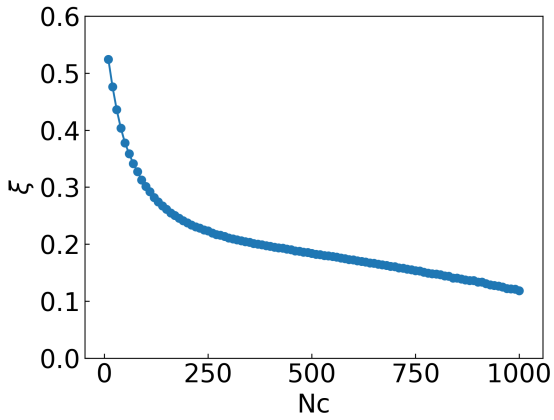


Figure 3.5: Normalized model mismatch as a function of the retained number of basis functions, N_c, which is in the range of $[10, 1000]$.

procedure is discussed in Chapter 2, we provide the basic details again here to make this chapter self-contained. The basic idea of this treatment is to minimize the following function for each gridblock.

$$f(x_i) = (a_i - x_i)^2 + \gamma_1 R_1^i(x_i) + \gamma_2 R_2^i(x_i), \quad x_i \in \left[x^l, x^u\right], \tag{3.30}$$

where a_i is the ith continuous component in $\hat{\boldsymbol{m}}_{\text{facies},j}^{\ell+1}$ and x_i is the facies indicator in $\boldsymbol{m}_{\text{facies},j}^{\ell+1}$. $R_1^i(x_i)$ and $R_2^i(x_i)$ denote the regularization terms, γ_1 and γ_2 represent the regularization weights, and x^l and x^u, respectively denote the lower and upper bound constraints for different a_i; see Vo and Durlofsky (2014, 2015). In a three-facies reservoir, shale, levee and sand are represented by 0, 1 and 2, respectively; thus we have (1) $x^l = 0$ and $x^u = 1$ if $a_i \leqslant 1$; (2) $x^l = 1$ and $x^u = 2$ if $a_i > 1$.

Through solving the minimization problem, we are able to obtain analytical solutions where four regularization weights (γ_{11}, γ_{12}, γ_{21} and γ_{22}) are required to map the continuous value, a_i, to the facies indicator, x_i (see the conceptual precedure in Fig. 3.6). According to the original papers, the four regularization weights are determined by numerical experiments a priori. In the previous work of Zhao et al. (2016, 2017), a different scheme was proposed to easily estimate the four weights by considering the prior facies proportions and the details can be referred to Section 2.3.2 of Chapter 2.

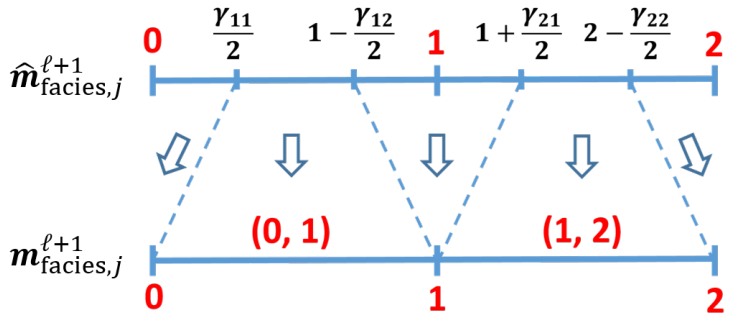


Figure 3.6: Post-processing procedure (Zhao and Forouzanfar, 2017)

After the ℓth iteration, we can also obtain updated ln-permeability fields (i.e., $\boldsymbol{m}^{\ell+1}_{\text{sand},j}$, $\boldsymbol{m}^{\ell+1}_{\text{levee},j}$ and $\boldsymbol{m}^{\ell+1}_{\text{shale},j}$) and note each field is for the entire set of reservoir gridblocks. If the facies type of the kth gridblock in $\boldsymbol{m}^{\ell+1}_{\text{facies},j}$ (i.e., $m^{\ell+1}_{\text{facies},j,k}$) is sand, then the ln-permeability value of the kth gridblock in $\boldsymbol{m}^{\ell+1}_{j}$ (i.e., $m^{\ell+1}_{j,k}$) is set equal to the value in the kth gridblock of $\boldsymbol{m}^{\ell+1}_{\text{sand},j}$ (i.e., $m^{\ell+1}_{\text{sand},j,k}$). Otherwise, if $m^{\ell+1}_{\text{facies},j,k}$ still has a real value, i.e., not an integer, then the permeability of the gridblock is obtained by simple interpolation shown below

$$\begin{aligned} m^{\ell+1}_{j,k} &= (1 - m^{\ell+1}_{\text{facies},j,k}) \times m^{\ell+1}_{\text{shale},j,k} \\ &\quad + m^{\ell+1}_{\text{facies},j,k} \times m^{\ell+1}_{\text{levee},j,k}, \quad \text{if } 0 < m^{\ell+1}_{\text{facies},j,k} < 1, \end{aligned} \tag{3.31}$$

$$\begin{aligned} m^{\ell+1}_{j,k} &= (2 - m^{\ell+1}_{\text{facies},j,k}) \times m^{\ell+1}_{\text{levee},j,k} \\ &\quad + (m^{\ell+1}_{\text{facies},j,k} - 1) \times m^{\ell+1}_{\text{sand},j,k}, \quad \text{if } 1 < m^{\ell+1}_{\text{facies},j,k} < 2. \end{aligned} \tag{3.32}$$

To summarize, the complete history matching workflow is shown in Fig. 3.7. It should be noted that since we use ES-MDA-GEO and the training ensemble is typically very large (e.g., $N_{te} = 1000$), we arbitrarily select N_e realizations to form the prior ensemble, where $N_e < N_{te}$.

3.2 Computational Results and Discussion

In this section, three parameterization approaches, i.e., OPCA, common basis DCT and NMF, are integrated into the history matching workflow and tested by two synthetic examples. The 2D example considers a reservoir with complex patterns of channels and

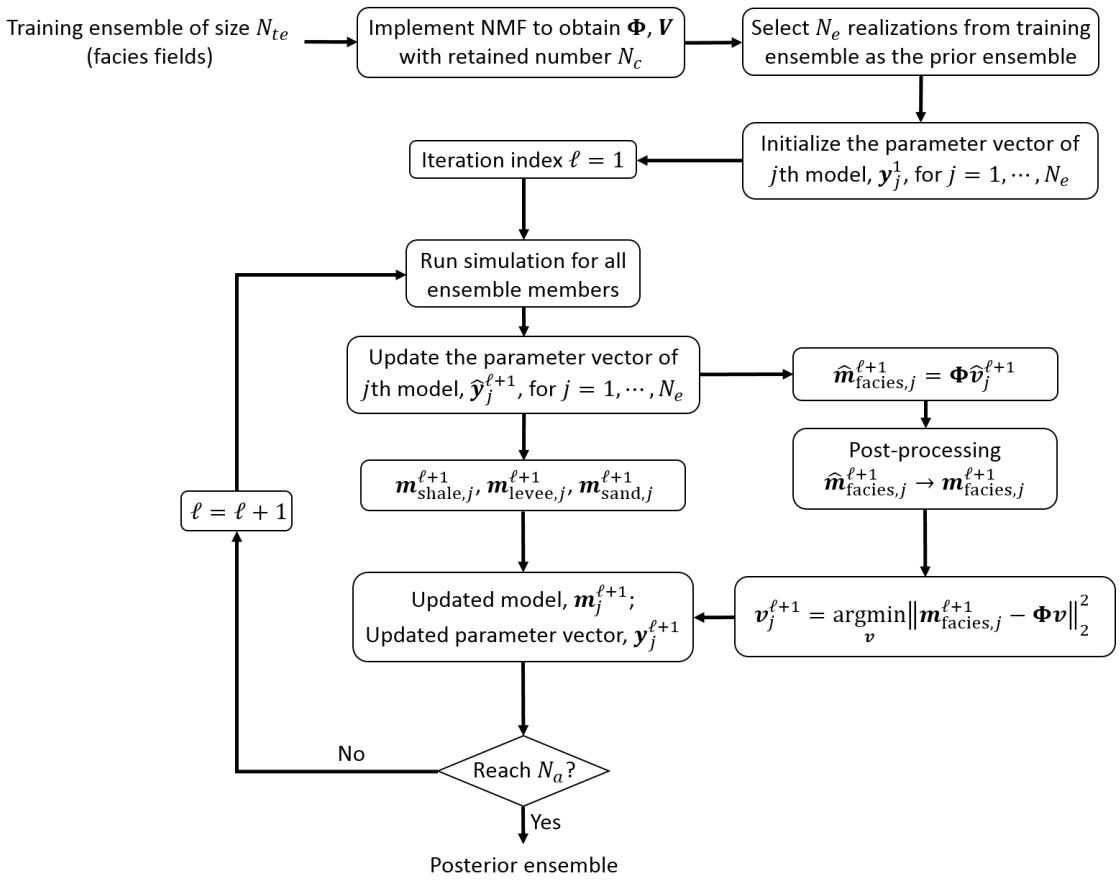


Figure 3.7: History matching workflow with NMF parameterization

not very much hard data information. The 3D example involves two geological zones in a multi-layer reservoir.

3.2.1 Example 1: 2D Three-Facies Channelized Reservoir

In the first example, we consider a three-facies (shale, levee and sand), two-dimensional fluvial reservoir model which has a 100 × 100 grid system. The size of each gridblock is 100ft × 100ft with a constant thickness of 50 ft. Given the facies type at wells as hard data, the MPS algorithm (Strebelle, 2002) is used to generate the ensemble of facies field with the pre-prepared training image (see Fig. 3.8). Due to distinct rock characteristics in the reservoir, the ln-permeability values follow Gaussian distribution with the mean and standard deviation specified differently for each facies. In this work the ln-permeability fields are obtained by the sequential Gaussian simulation technique where an isotropic spherical

Table 3.1: Properties of facies

Facies	Indicator	Color	ϕ	$\overline{k}$, mD	$\overline{\ln(k)}$	$\sigma(\ln(k))$
Sand	2	Red	0.3	2000	7.601	0.3
Levee	1	Green	0.2	200	5.298	0.3
Shale	0	Blue	0.1	20	2.996	0.3

variogram is used with correlation range equal to ten times the width of a gridblock. The detailed properties of individual facies are given in Table 3.1.

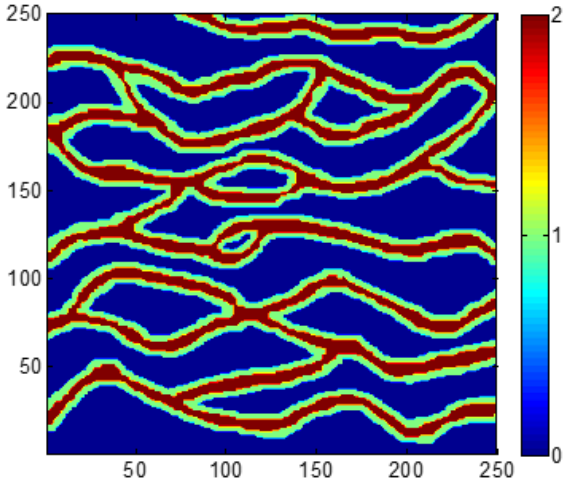
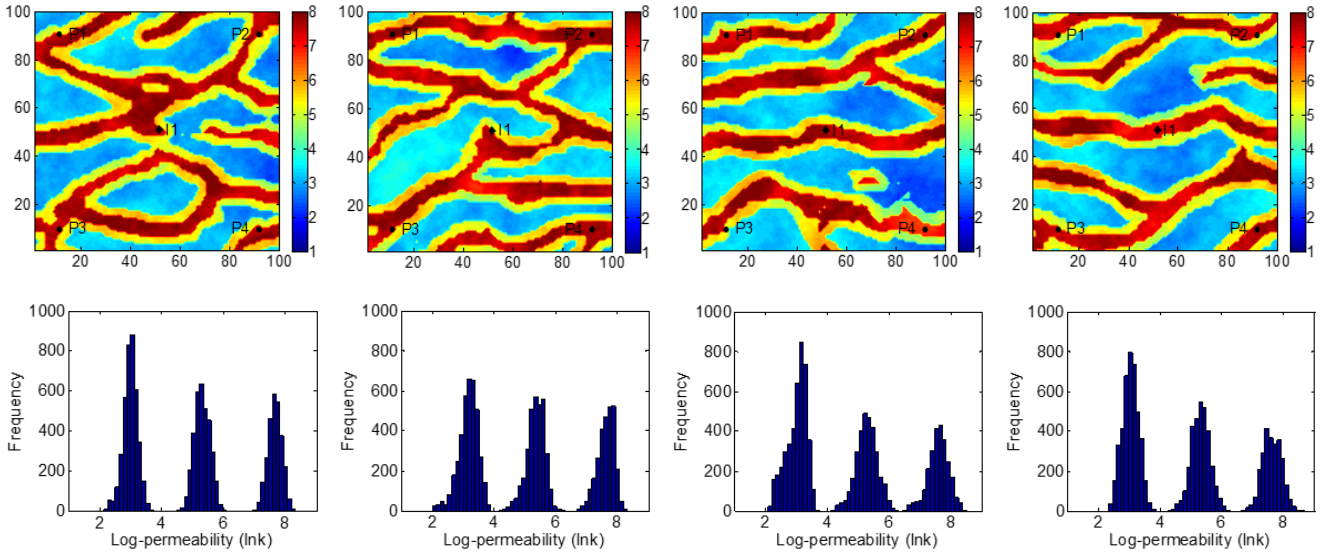


Figure 3.8: Training image (Example 1)

From the training ensemble of 1000 members, we randomly select 200 realizations to form the prior ensemble for history matching, i.e., $N_e = 200$. Figure 3.9 shows the ln-permeability distribution as well as the histogram of ln-permeability values of the true model and three prior realizations. From the histograms, we can observe that ln-permeability values present a tri-modal distribution as expected. There are four producers (P1 to P4) and one injector (I1) in the field. The facies type at all five wells is sand, and these observations are used as hard data to be honored in the posterior realizations. In the true model, producers P1 and P2 are directly connected to the water injector I1 by two sand channels. P3 and I1 are connected by a loop consisting of two branch channels and the injected water has two different ways to reach P3, whereas P4 lies in the corner of the reservoir and is not connected to the injector directly.



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

Figure 3.9: True model and prior realizations (Example 1): $\ln(k)$ field (top row) and histogram (bottom row).

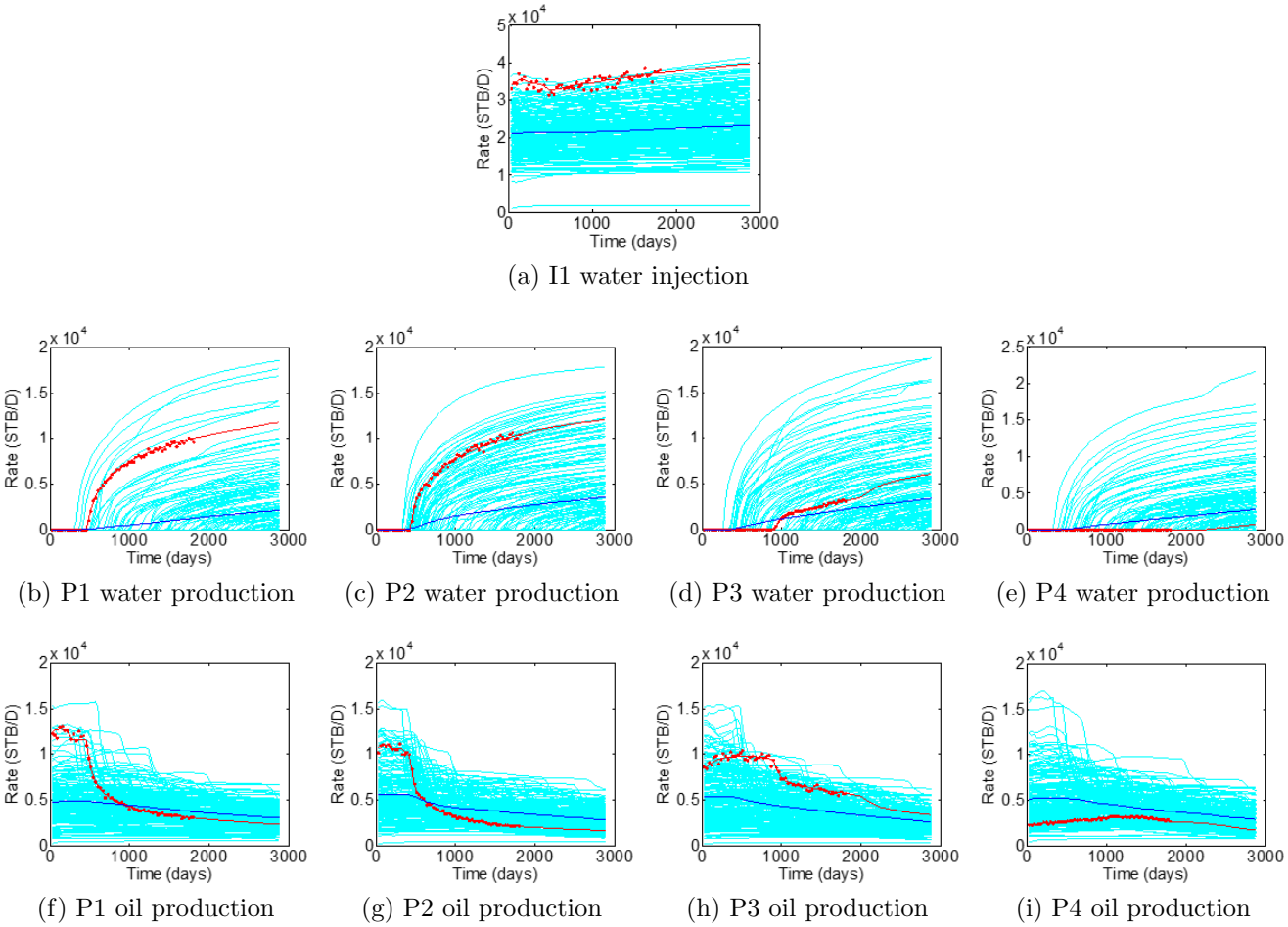
In this synthetic example, the initial reservoir pressure is 5000 psi and the water saturation is 0.2 before production. The water injection well is operated at constant bottom hole pressure (BHP) of 5500 psi and the production wells are operated at 2000 psi. The history matching period is 5 years (1800 days), followed by a 3-year (1080 days) forecast period. In this work, the subsurface flow responses are obtained by running the Eclipse reservoir simulator. The historical data consist of monthly measurements of water injection rate (q_{inj}) for the injector, and oil and water flow rate (q_{o} and q_{w}) for the producers. Therefore, the observed data vector is given by

$$\boldsymbol{d}_{\text{obs}} = \left[\boldsymbol{q}_{\text{inj}}^{\text{T}}, \boldsymbol{q}_{\text{o}}^{\text{T}}, \boldsymbol{q}_{\text{w}}^{\text{T}}, \boldsymbol{d}_{\text{hard}}^{\text{T}}\right]_{N_d \times 1}^{\text{T}}, \tag{3.33}$$

where $\boldsymbol{d}_{\text{hard}}$ denotes the hard data vector and is used to constrain the facies type at all wells. In order to generate the observed data, we perturbed the true data with Gaussian random noise and the standard deviation of measurement error is 3% for dynamic rates and 0.01 for facies type indicator at well locations.

That there is significant variability in the initial ensemble can be confirmed by the

prior production data shown in Fig. 3.10 where red curves denote the results of the true model; red dots represent the observations; light blue curves are obtained by running the prior realizations and dark blue curves represent the prior ensemble averages. Based on the results of Fig. 3.10, we can see that the prior uncertainty is very high. Note that for producer P4, the production data after water breakthrough are not included in the historical period.



(a) I1 water injection

(b) P1 water production (c) P2 water production (d) P3 water production (e) P4 water production

(f) P1 oil production (g) P2 oil production (h) P3 oil production (i) P4 oil production

Figure 3.10: Production data of prior ensemble (Example 1). True (red curves), observed data (red dots), simulated data of realizations (light blue curves), and ensemble mean (dark blue curves). History: $0 < t \leq 1800$ days, forecast: $t > 1800$ days.

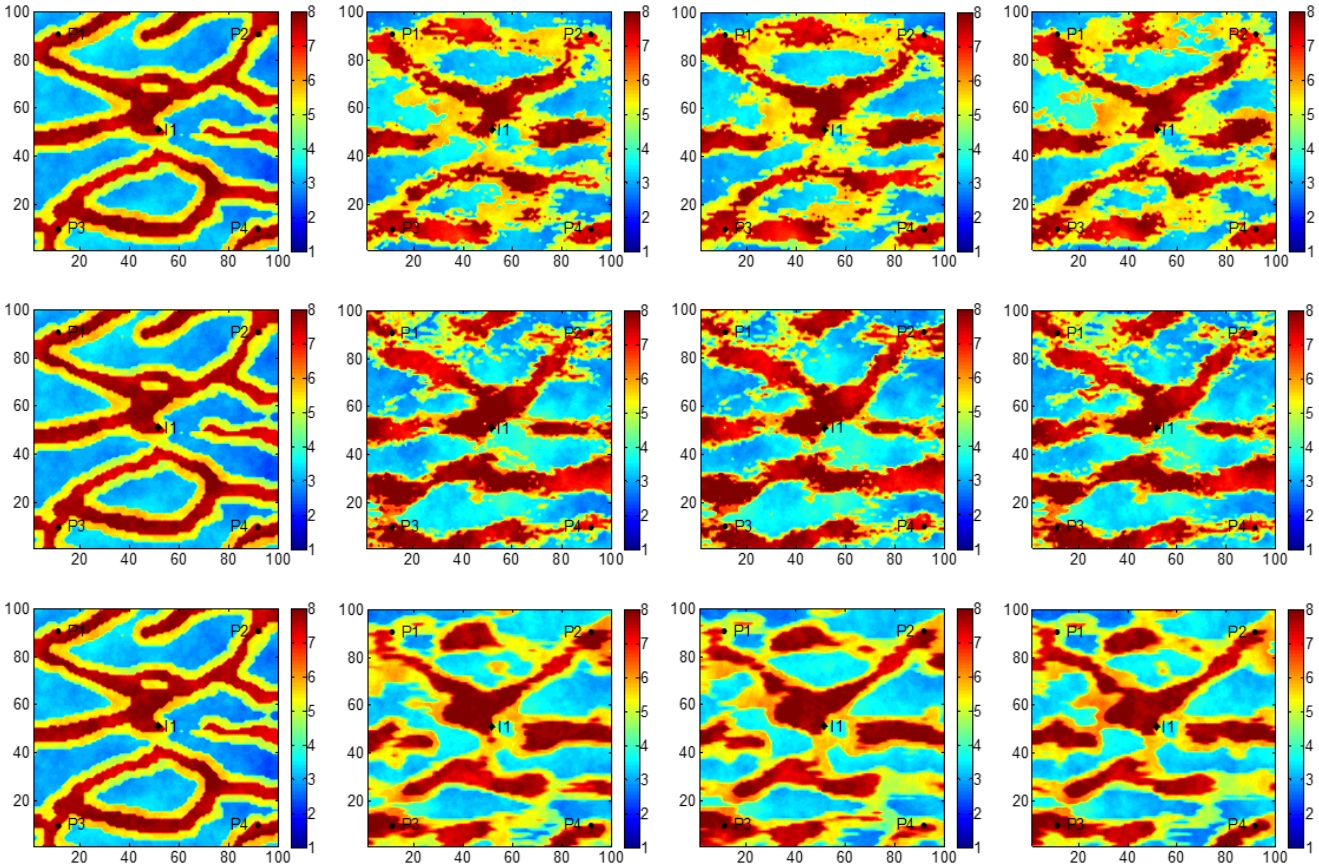
In the parameterization of this example, the retained number with the "NMF" approach is set to be $N_c = 650$ using the determination scheme introduced in Section 3.1.2 with $\xi = 0.15$; see Eq. 3.29. For the sake of fair comparison, the retained number of coefficients is also set to 650 when we implement the "PCA" and "DCT" parameterizations. The vectors of model parameters, $\boldsymbol{y}$, of different parameterization approaches are assembled according

to Eq. 3.5. The iteration number of ES-MDA-GEO algorithm, N_a, is set to 8 with all of the inflation factors calculated by the geometric method proposed by Rafiee and Reynolds (2017) as $\alpha_\ell = \{1000.0,\ 401.04,\ 160.83,\ 64.50,\ 25.87,\ 10.37,\ 4.16,\ 1.67\}$. Due to the large variation in the predicted rates from prior realizations, the inflation factors of the first several iterations are relatively large in order to ensure small updates of the model parameters.

Figures 3.11 and 3.12 show the ln-permeability field and histogram corresponding to the true model and three posterior realizations obtained with the three parameterization approaches. We can observe that the large-scale geological structures, e.g., the connections between P1, P2 and I1, are well preserved in all posterior realizations. With the "NMF" approach, the most problematic channel loop connecting P3 and I1 is almost resolved although the spatial continuity of sand channels is damaged in some areas. When we use the "PCA", the reconstructed channel loop is not complete and P4 is incorrectly connected to the loop rather than isolated in the reservoir corner as in the true model. On the other hand, the channel loop structure obtained with the "DCT" method presents smoother boundaries between different facies but suffers poor continuity at some parts. For each method, there is little variation between realizations in the posterior facies distribution. From the histograms, we note there is some smearing of the ln-permeability fields between adjacent facies. Compared with the other two parameterization methods, the proportion of each facies and the standard deviation of ln-permeability values obtained with the "NMF" approach show better approximations to those in the true model.

Figures 3.13 and 3.14 show the posterior production data after history matching. For wells P1 and P2, the history match and forecast results obtained with "NMF" and "PCA" are good. Because of the channel loop structure, the water production rate of P3 obtained with the true model presents a rapid increase around 2000 days. Since the corresponding period is not part of the historical period, the predictions for P3 are not as satisfactory as those for P1 and P2. The predicted future performance of P4 is also poor with all three parameterization approaches, which is probably due to the fact that no data after breakthrough is available.

Figure 3.15(a) shows the box plots of the root mean square error (RMSE) obtained



(a) True (b) Realization #1 (c) Realization #2 (d) Realization #3

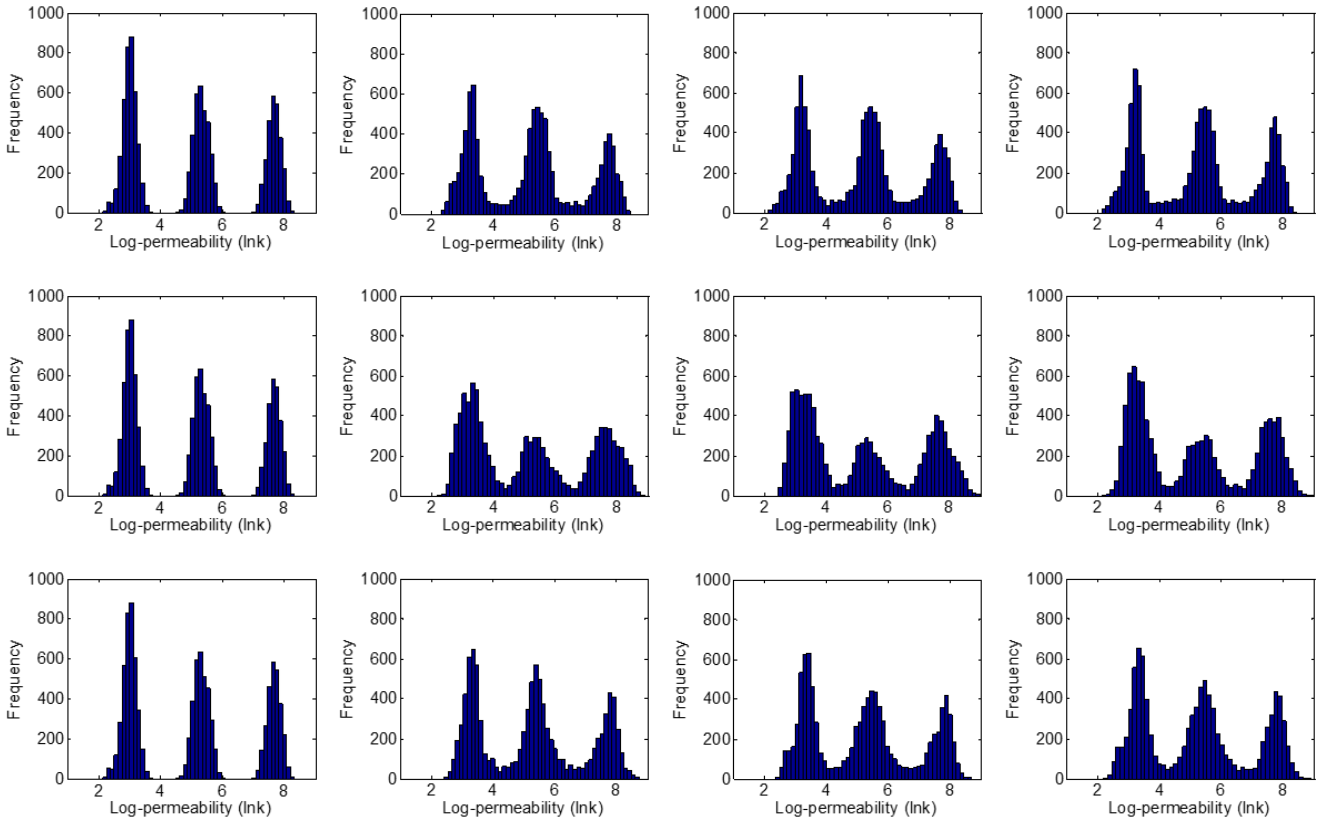
Figure 3.11: $\ln(k)$ field of the true model and posterior realizations (Example 1): NMF (1st row), PCA (2nd row) and DCT (3rd row).

with three parameterization approaches. The RMSE values are computed by

$$\text{RMSE} = \sqrt{\frac{1}{N_m} \left\| \boldsymbol{m} - \boldsymbol{m}_{\text{true}} \right\|_2^2} \,, \tag{3.34}$$

where $\boldsymbol{m}$ and $\boldsymbol{m}_{\text{true}}$ are the ln-permeability fields of the whole reservoir model. It can be seen that the posterior realizations obtained with the "PCA" give the largest RMSE values mainly because the channel loop structure is poorly resolved whereas the "NMF" gives the smallest root mean square errors. In addition, Fig. 3.15(b) presents the box plots of the normalized objective function regarding the posterior data mismatch calculated by

$$O(\boldsymbol{d}) = \frac{1}{N_d} \left(\boldsymbol{d} - \boldsymbol{d}_{\text{obs}} \right)^{\text{T}} \boldsymbol{C}_{\text{D}}^{-1} \left(\boldsymbol{d} - \boldsymbol{d}_{\text{obs}} \right) , \tag{3.35}$$



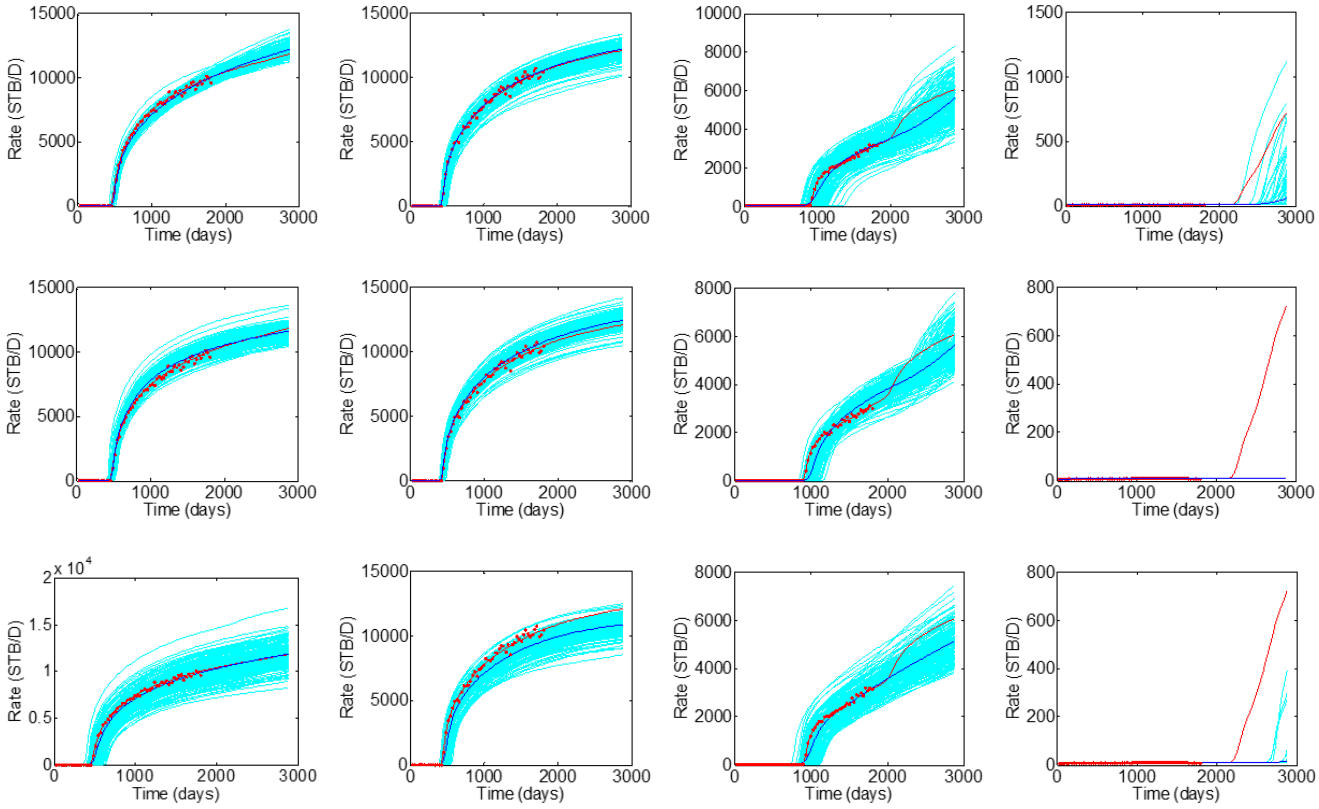
(a) True (b) Realization #1 (c) Realization #2 (d) Realization #3

Figure 3.12: Histogram of the true model and posterior realizations (Example 1): NMF (1st row), PCA (2nd row) and DCT (3rd row).

where $\boldsymbol{d}$ and $\boldsymbol{d}_{\text{obs}}$ only include production data during the historical period. Here, $\boldsymbol{d}$ is the predicted data vector which is different for each posterior realization. Since the spatial continuity of the channel loop structure is not well preserved, the data mismatch obtained with the "DCT" is the largest among three methods. For the "NMF", a normalized data match close to unity is obtained for almost all posterior realizations whereas the "PCA" and "DCT" give normalized data matches on the order of 500 or greater for about 25 percent of the posterior realizations. Overall, Fig. 3.15 clearly shows that the best approximations of the true model and the best data matches are generally achieved using the "NMF" even though the oil rates of well P4 are poorly matched.

3.2.2 Example 2: 3D Three-Facies Channelized Reservoir

The 3D reservoir model in the second example consists of five layers and each layer

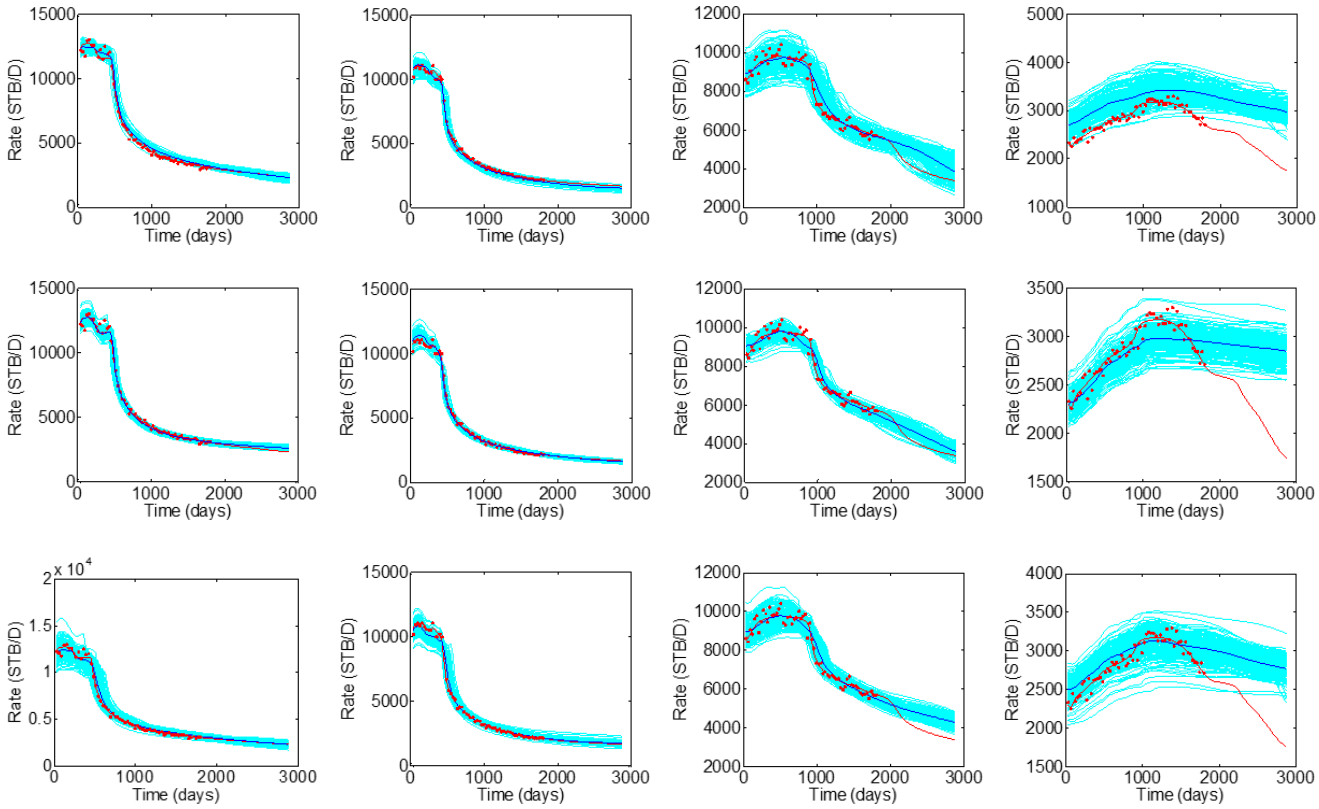


(a) P1 (b) P2 (c) P3 (d) P4

Figure 3.13: Posterior water production rates (Example 1): NMF (1st row), PCA (2nd row) and DCT (3rd row). True (red curves), observed data (red dots), simulated data of realizations (light blue curves), and ensemble mean (dark blue curves). History: $0 < t \leq 1800$ days, forecast: $t > 1800$ days.

has a 50×50 grid system with size given by 100ft $\times$ 100ft $\times$ 50ft. Due to the lack of 3D channelized training image, the object-based modeling technique is employed to generate the facies distributions given the hard data (observed facies type at wells) in Table 3.2. The prior realization of the permeability field of each facies is still obtained by the sequential Gaussian simulation technique with the same properties specified in Table 3.2. From the true permeability field shown in Fig. 3.16, we can see that the top two layers of the reservoir belong to a same geological zone which differs from the one that consists of the other three layers, and the channelized facies presents distinct geometrical patterns across two zones.

The locations of the four injectors and nine producers are shown in Fig. 3.16, all the wells are vertical and perforate all five layers. The water injectors are operated at constant BHP of 5500 psi and the producers are operated at 3000 psi. As for Example 1, we set the



(a) P1 (b) P2 (c) P3 (d) P4

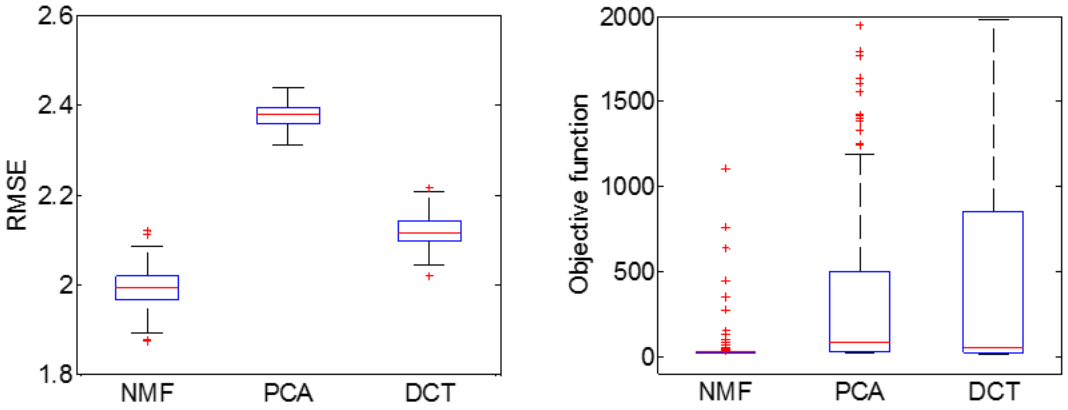
Figure 3.14: Posterior oil production rates (Example 1): NMF (1st row), PCA (2nd row) and DCT (3rd row). True (red curves), observed data (red dots), simulated data of realizations (light blue curves), and ensemble mean (dark blue curves). History: $0 < t \leq 1800$ days, forecast: $t > 1800$ days.

initial reservoir pressure and water saturation to 5000 psi and 0.2, respectively. The historical period is 300 days and the total production period is equal to 600 days. The production data consist of water injection rate (q_{inj}) for the injectors, and oil and water flow rate (q_{o} and q_{w}) for the producers. The noise level (standard deviation of measurement error) is 3% of the true data for dynamic rates and 0.01 for facies type indicator at well locations.

Table 3.2: Observed facies type at wells (Example 2)

Facies		Wells
Zone 1	Sand	I2, I3, P1, P2, P3, P5, P7
Zone 2	Sand	I1, I4, P2, P4, P5, P6, P7, P8, P9

In this example, the training ensemble consists of 1000 models and the proposed

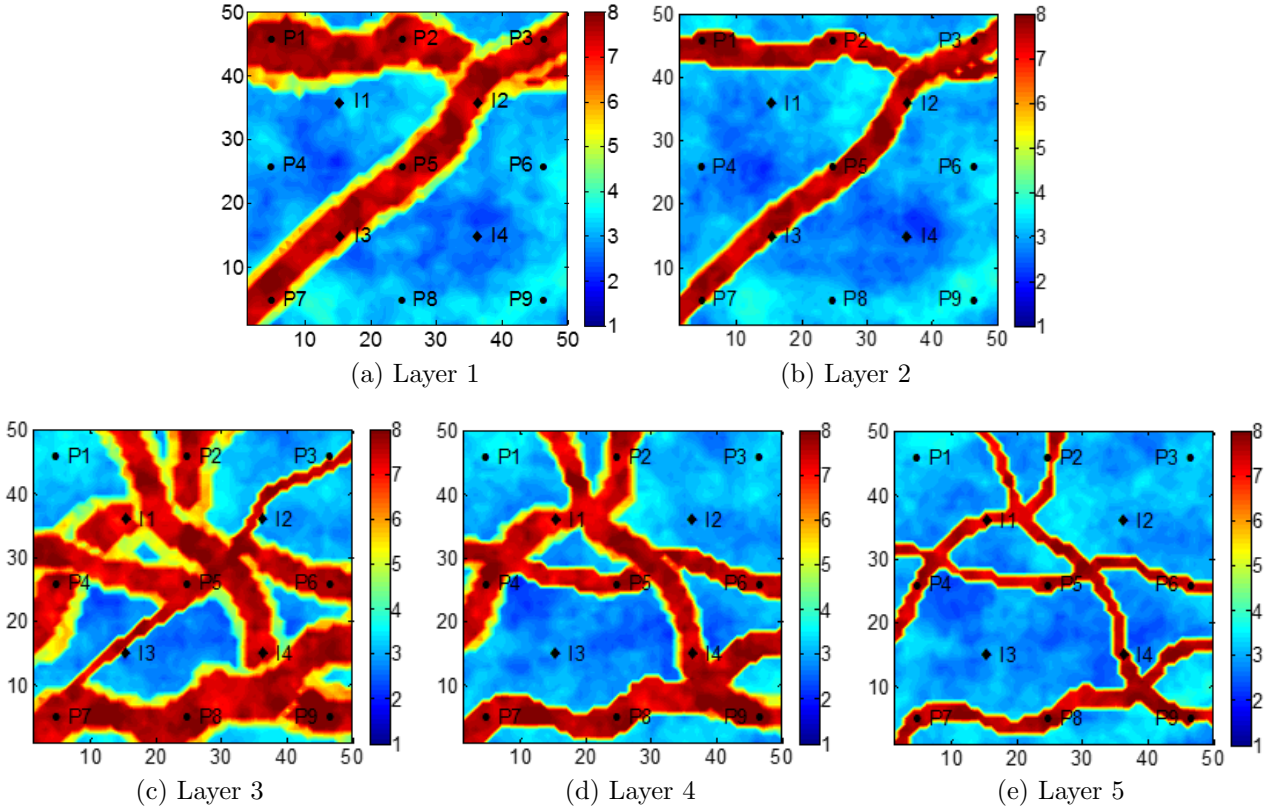


(a) RMSE of $\ln(k)$ fields (b) Posterior normalized data mismatch

Figure 3.15: Performance comparison of three parameterization approaches (Example 1). Median (central red mark), 25th and 75th percentiles (bottom and top edges of the box), extreme data points (whiskers), and outliers ("+" symbol).

history matching workflow is tested on two ensembles of different size ($N_e = 200$ and $N_e = 500$). According to the Eq. 3.29 with $\xi = 0.15$, the retained number of basis vectors for each facies field is set to be $N_c = 500$ when we implement the "NMF", "PCA" and "DCT" parameterization approaches. Figure 3.17 presents the permeability in three layers of the true model and three prior realizations. The production data generated from the 200 prior realizations are shown in Fig. 3.18 where the curves and dots have the same meanings as in Example 1. Since the wells are in the sand channels and perforate all five layers, we observe very early water breakthrough and a quick oil rate drop at the producers.

In the history matching, the iteration number of ES-MDA-GEO is $N_a = 6$ and the inflation factors are $\alpha_\ell = \{1000.0,\ 267.29,\ 71.44,\ 19.09,\ 5.11,\ 1.36\}$ which are obtained by the geometric method. From the results in Figs. 3.19, 3.20 and 3.22, we can compare three layers of posterior realizations obtained with different parameterization approaches when $N_e = 200$. It can be seen that the realizations calibrated with different approaches are all able to resolve reasonably well large-scale geological structures in layers 1 and 3 (see Figs. 3.19 and 3.20). However, the "NMF" method shows superior capacity of resolving detailed features such as the thin channels in the bottom layer (see Fig. 3.22). When a larger ensemble ($N_e = 500$) is employed, more geological variability is preserved in the



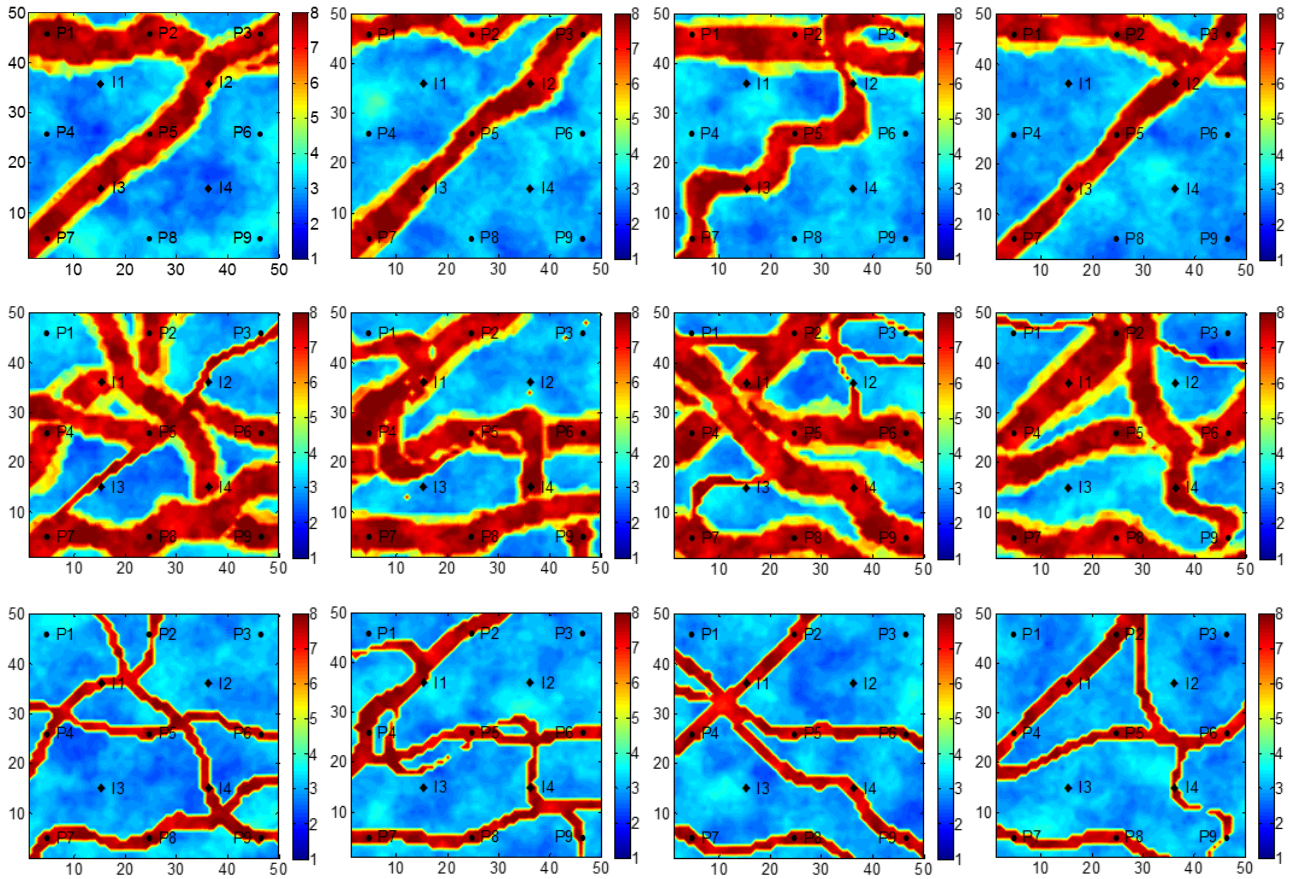
(a) Layer 1 (b) Layer 2

(c) Layer 3 (d) Layer 4 (e) Layer 5

Figure 3.16: $\ln(k)$ field of the true model (Example 2)

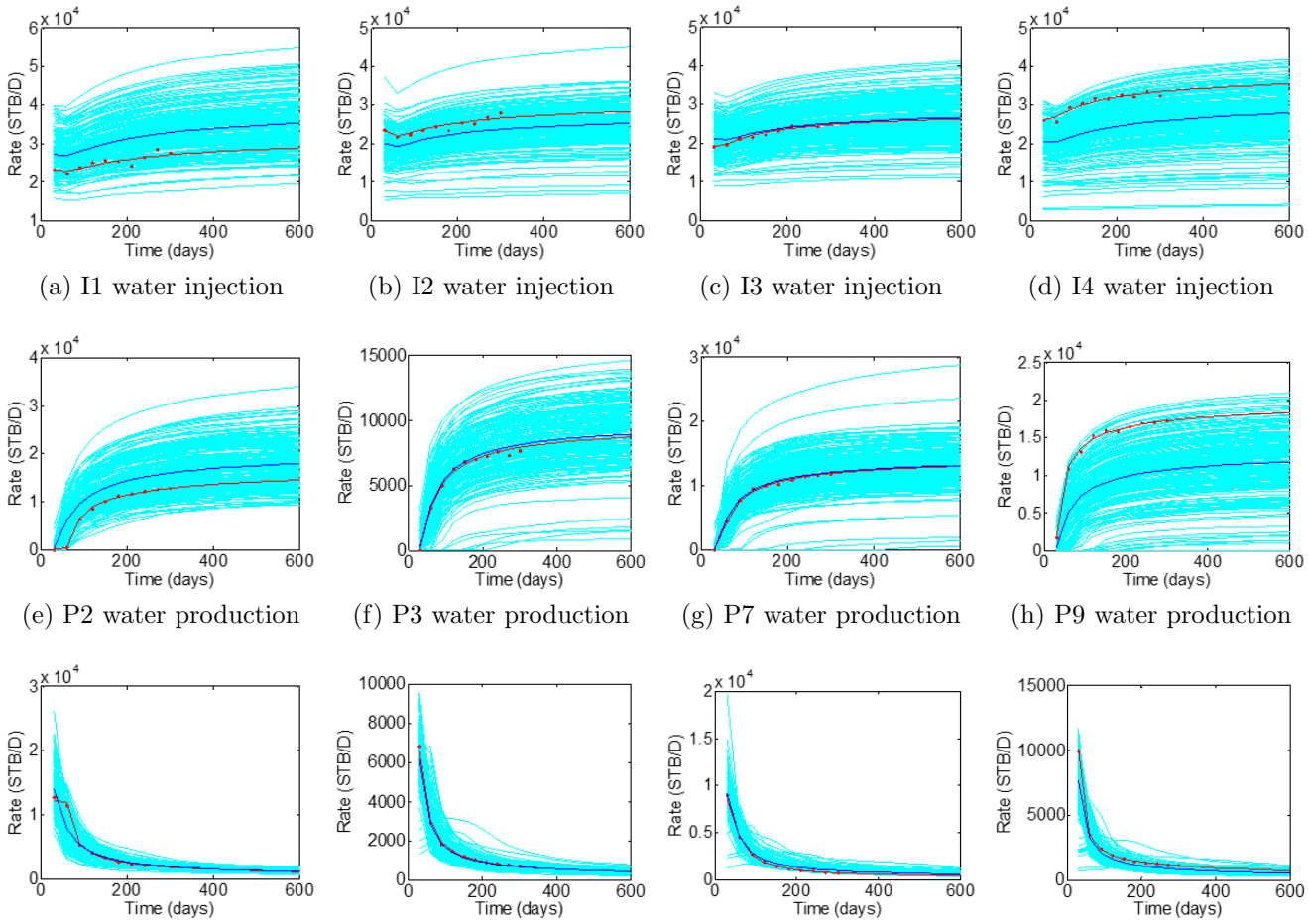
posterior realizations obtained with three parameterization approaches (see Figs. 3.21 and 3.23). Again, the geological details in the bottom layer are better resolved with the "NMF" and "PCA" methods while the continuity of channels is poorly preserved by the "DCT" method.

In Figs. 3.24 to 3.27, the match results of well production data obtained with the "NMF" and "PCA" methods are generally good and there remains more variability in the posterior realizations when $N_e = 500$. However, for the "DCT" method, we have poor data matches at well P3 when $N_e = 500$.



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

Figure 3.17: $\ln(k)$ field of the true model and prior realizations (Example 2): layer 1 (1st row), layer 3 (2nd row) and layer 5 (3rd row).

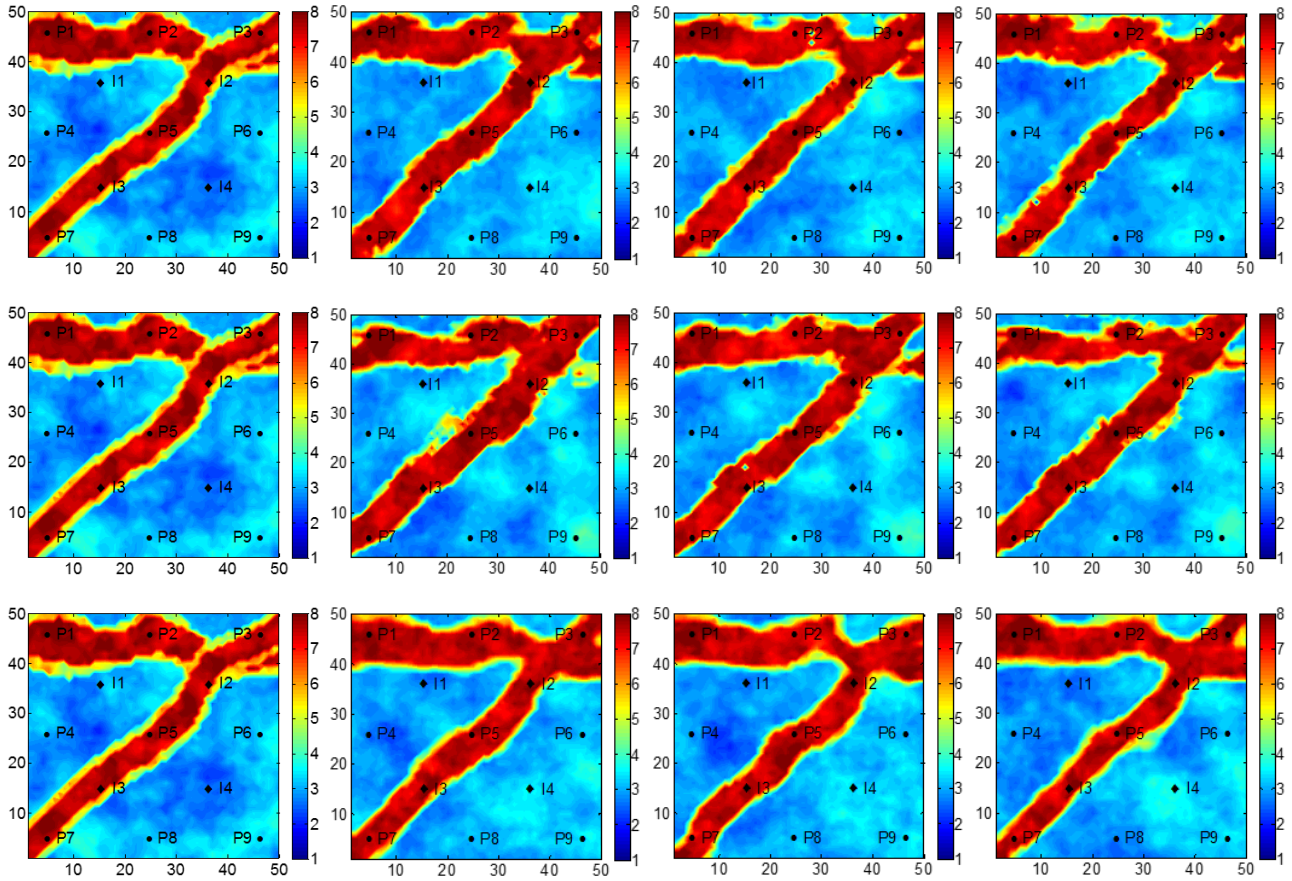


(a) I1 water injection (b) I2 water injection (c) I3 water injection (d) I4 water injection

(e) P2 water production (f) P3 water production (g) P7 water production (h) P9 water production

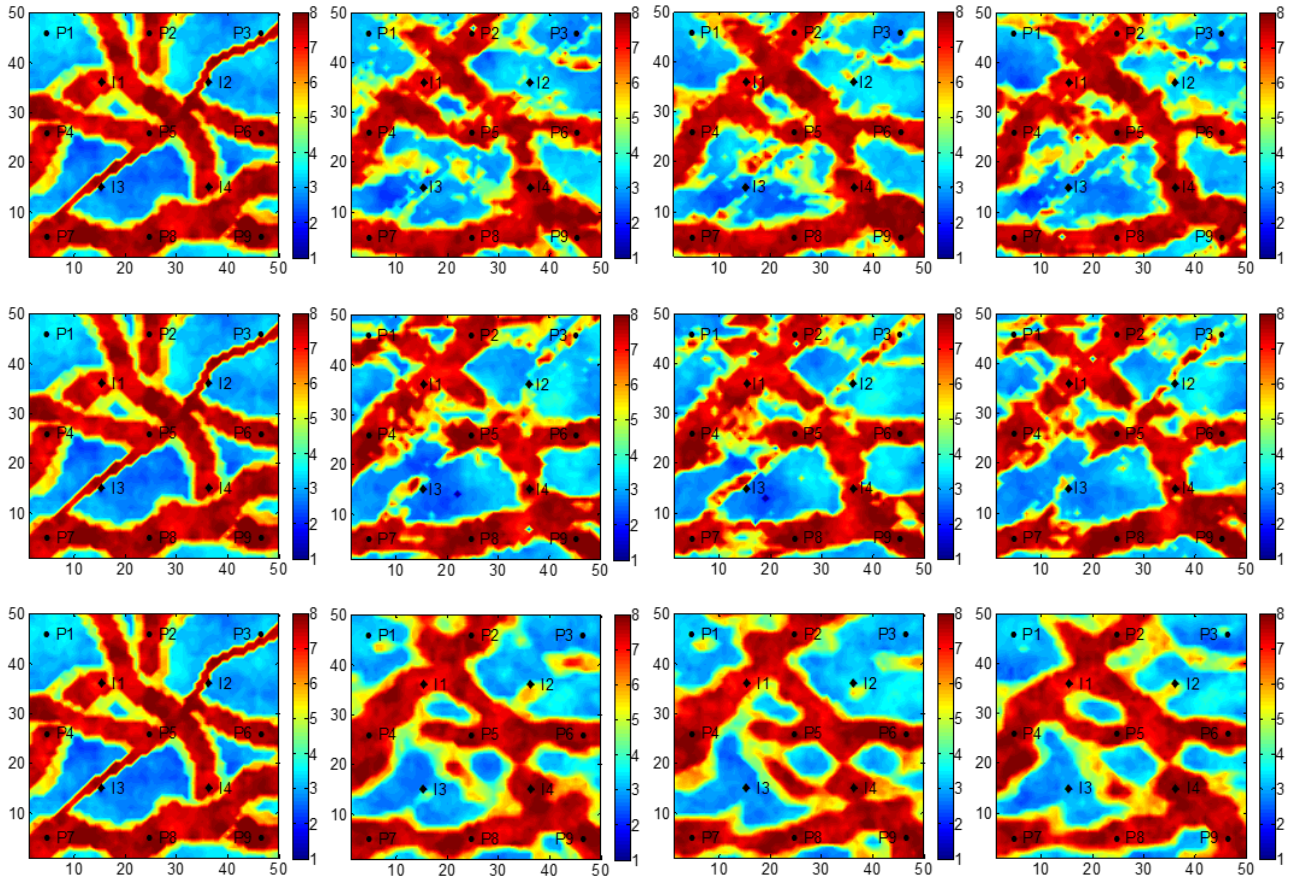
(i) P2 oil production (j) P3 oil production (k) P7 oil production (l) P9 oil production

Figure 3.18: Production data of prior ensemble (Example 2). True (red curves), observed data (red dots), simulated data of realizations (light blue curves), and ensemble mean (dark blue curves). History: $0 < t \leq 300$ days, forecast: $t > 300$ days.



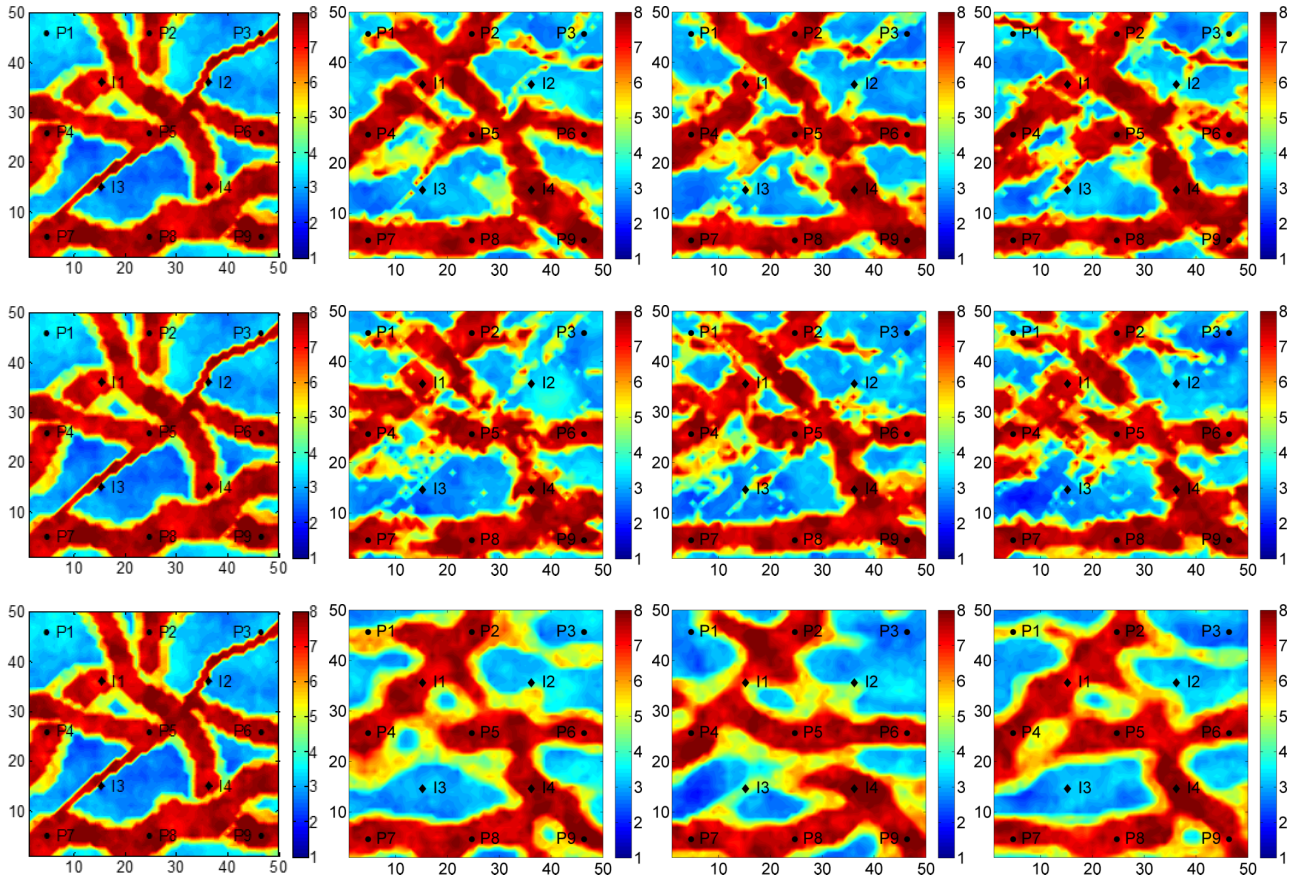
(a) True (b) Posterior model #1 (c) Posterior model #2 (d) Posterior model #3

Figure 3.19: $\ln(k)$ layer 1 of the true model and posterior realizations (Example 2, $N_e = 200$): NMF (1st row), PCA (2nd row) and DCT (3rd row).



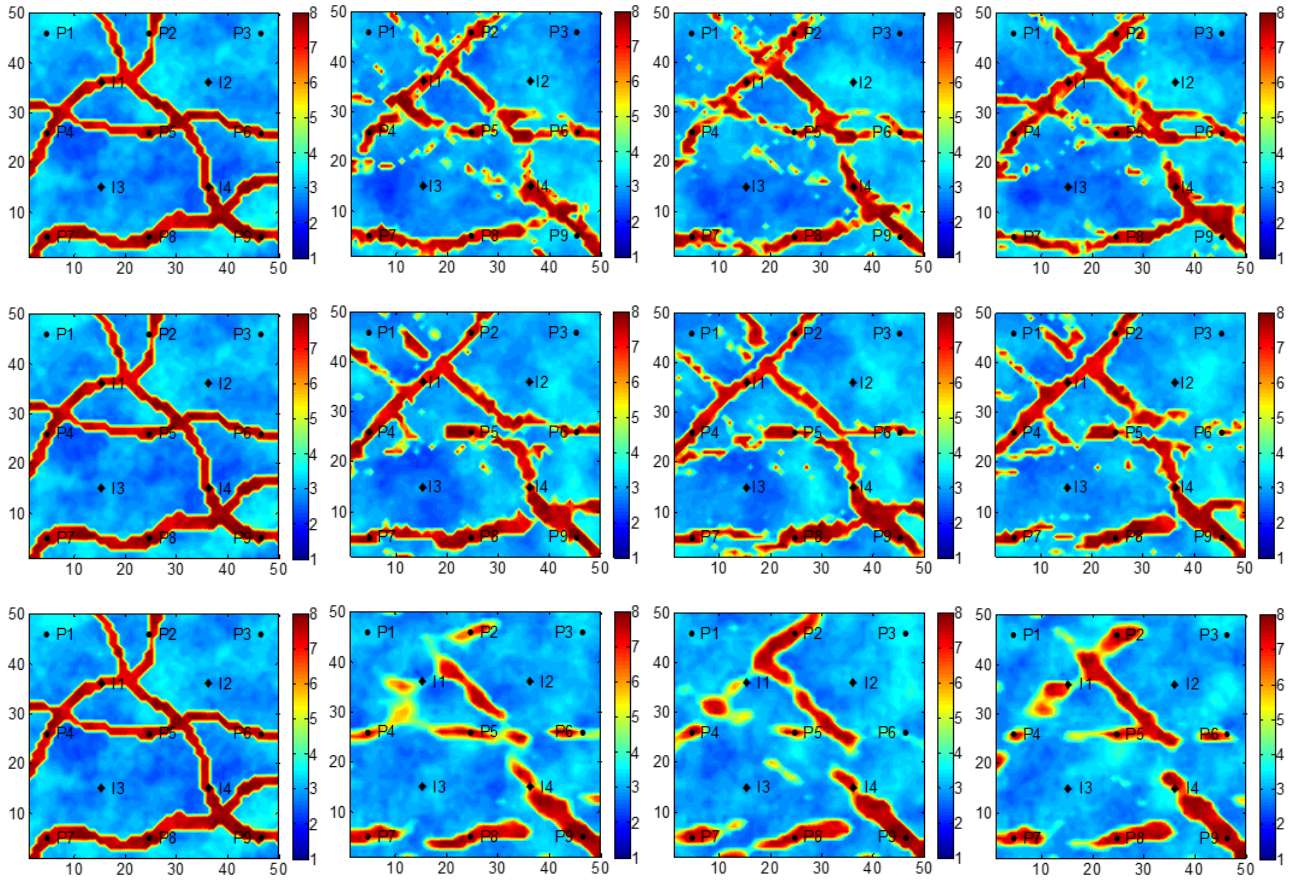
(a) True (b) Posterior model #1 (c) Posterior model #2 (d) Posterior model #3

Figure 3.20: $\ln(k)$ layer 3 of the true model and posterior realizations (Example 2, $N_e = 200$): NMF (1st row), PCA (2nd row) and DCT (3rd row).



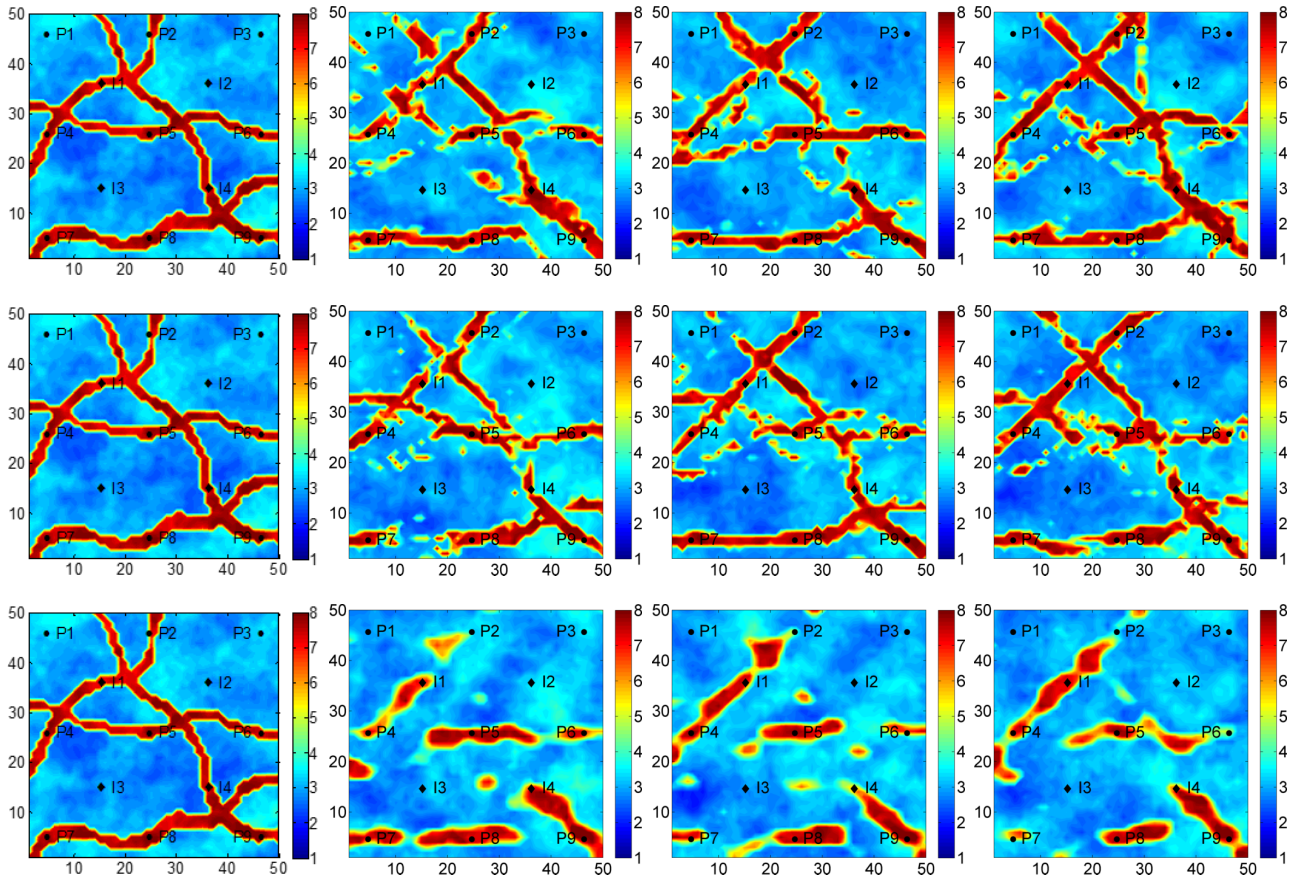
(a) True (b) Posterior model #1 (c) Posterior model #2 (d) Posterior model #3

Figure 3.21: $\ln(k)$ layer 3 of the true model and posterior realizations (Example 2, $N_e = 500$): NMF (1st row), PCA (2nd row) and DCT (3rd row).



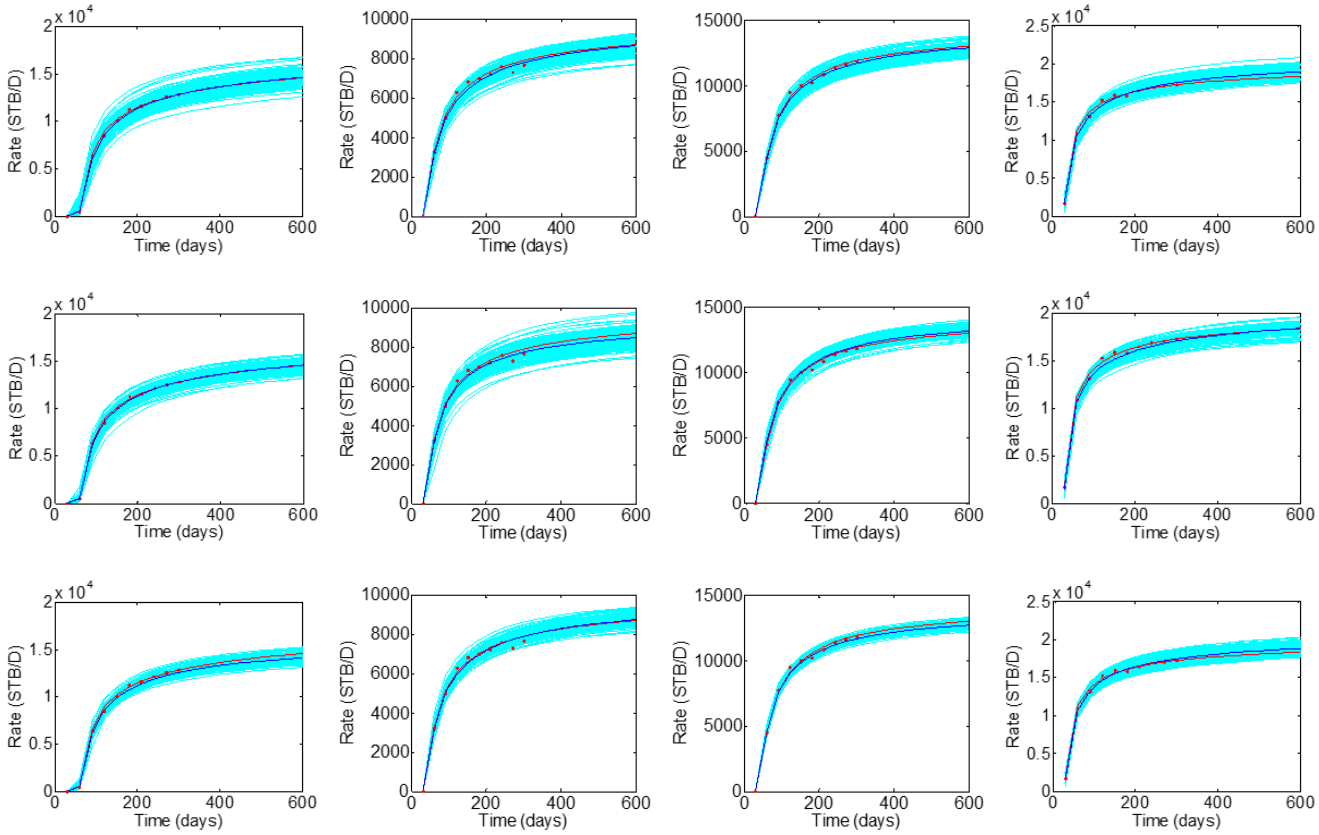
(a) True (b) Posterior model #1 (c) Posterior model #2 (d) Posterior model #3

Figure 3.22: $\ln(k)$ layer 5 of the true model and posterior realizations (Example 2, $N_e = 200$): NMF (1st row), PCA (2nd row) and DCT (3rd row).



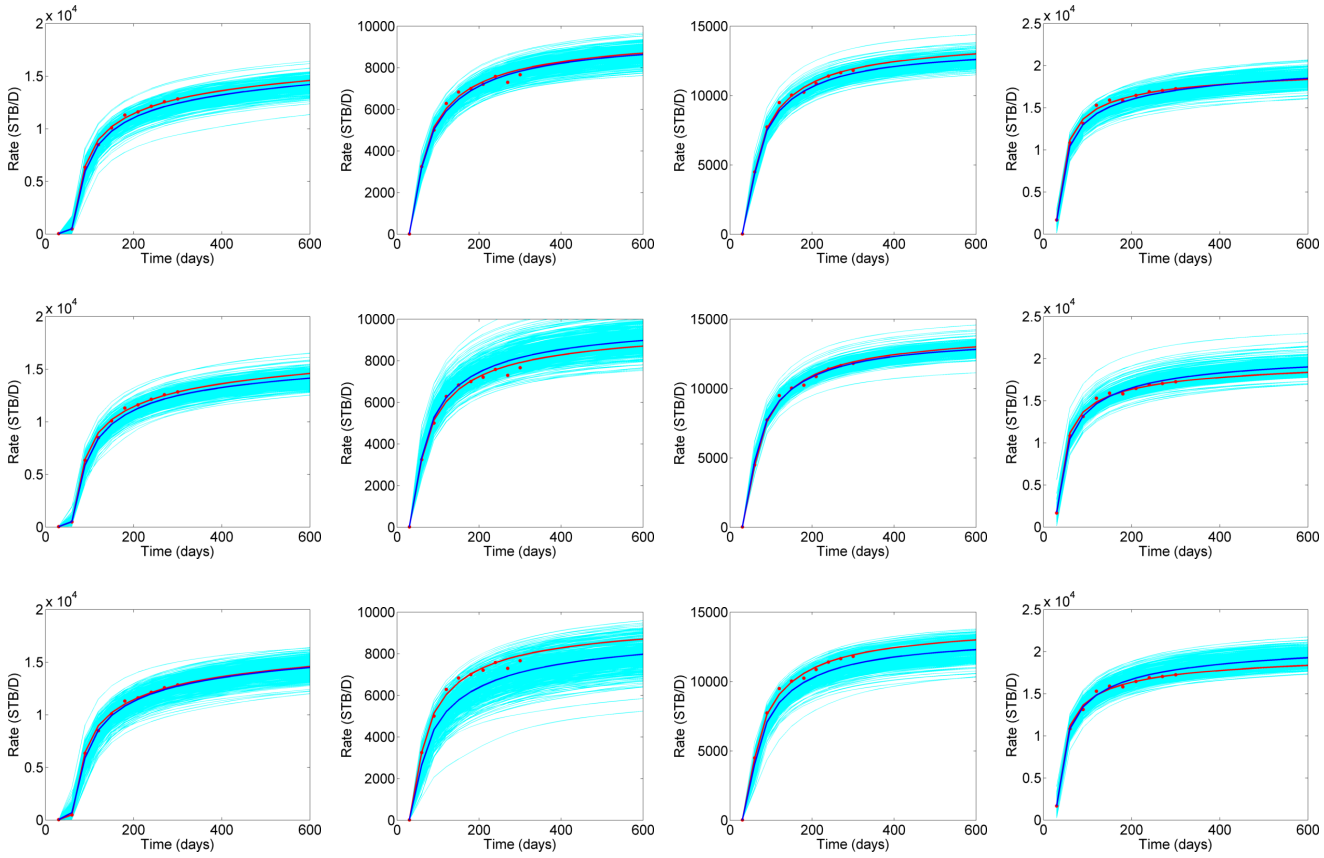
(a) True (b) Posterior model #1 (c) Posterior model #2 (d) Posterior model #3

Figure 3.23: $\ln(k)$ layer 5 of the true model and posterior realizations (Example 2, $N_e = 500$): NMF (1st row), PCA (2nd row) and DCT (3rd row).



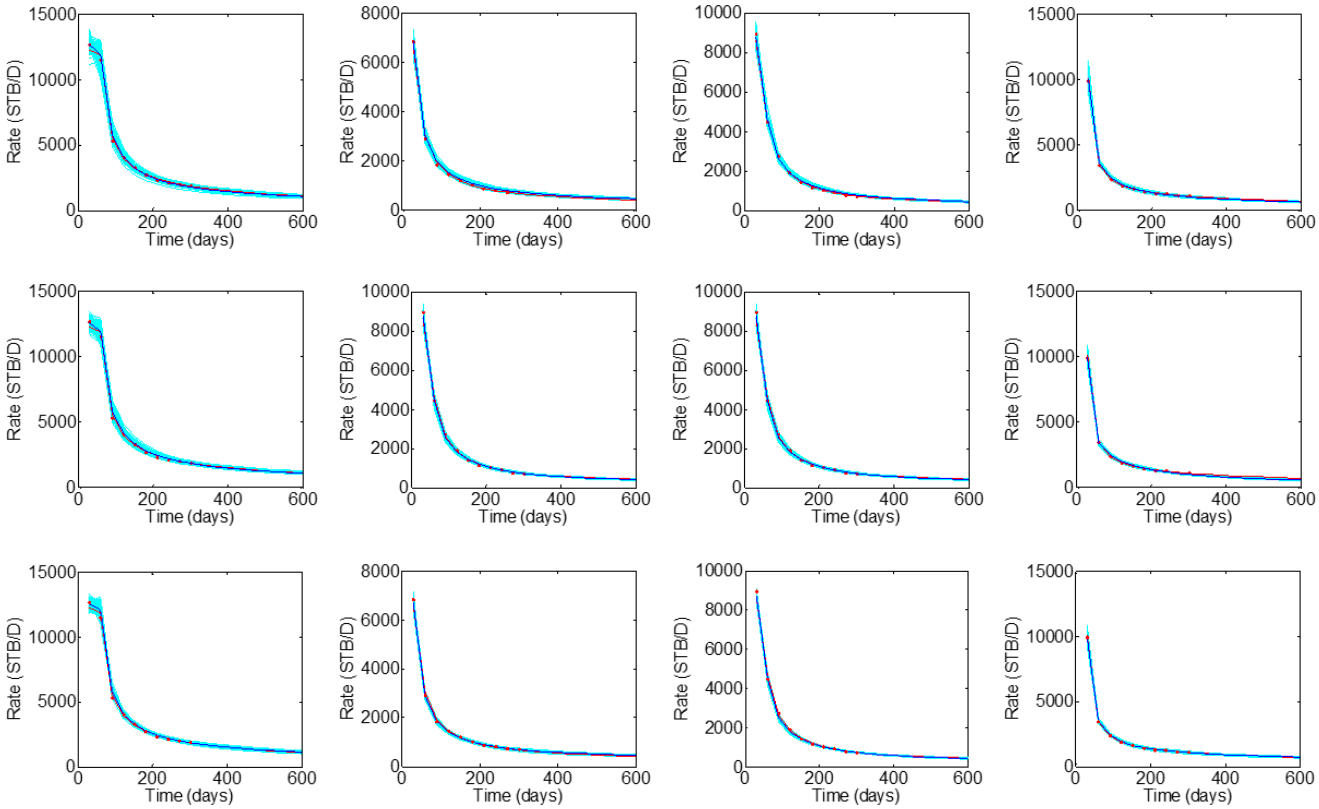
(a) P2 (b) P3 (c) P7 (d) P9

Figure 3.24: Posterior water production rates (Example 2, $N_e = 200$): NMF (1st row), PCA (2nd row) and DCT (3rd row). True (red curves), observed data (red dots), simulated data of realizations (light blue curves), and ensemble mean (dark blue curves). History: $0 < t \leq 300$ days, forecast: $t > 300$ days.



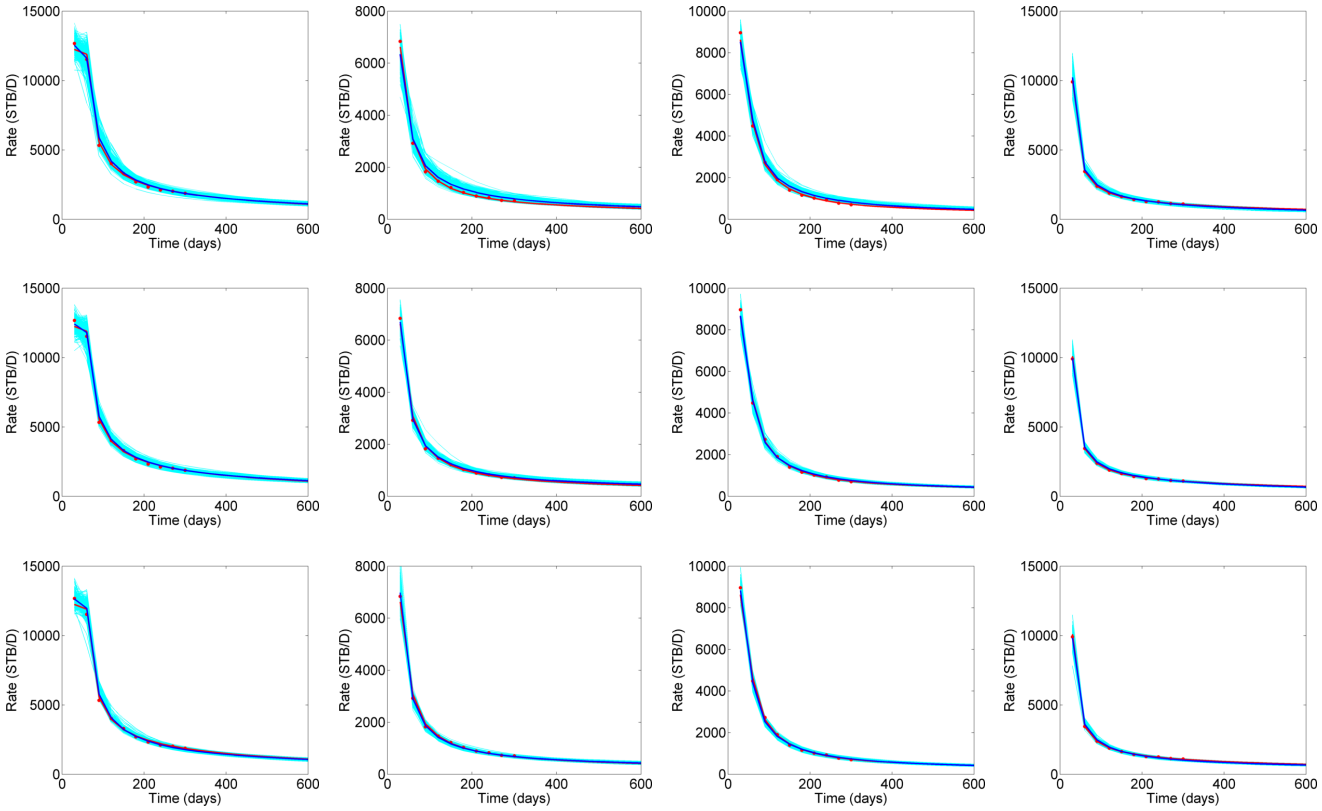
(a) P2 (b) P3 (c) P7 (d) P9

Figure 3.25: Posterior water production rates (Example 2, $N_e = 500$): NMF (1st row), PCA (2nd row) and DCT (3rd row). True (red curves), observed data (red dots), simulated data of realizations (light blue curves), and ensemble mean (dark blue curves). History: $0 < t \leq 300$ days, forecast: $t > 300$ days.



(a) P2 (b) P3 (c) P7 (d) P9

Figure 3.26: Posterior oil production rates (Example 2, $N_e = 200$): NMF (1st row), PCA (2nd row) and DCT (3rd row). True (red curves), observed data (red dots), simulated data of realizations (light blue curves), and ensemble mean (dark blue curves). History: $0 < t \leq 300$ days, forecast: $t > 300$ days.



(a) P2 (b) P3 (c) P7 (d) P9

Figure 3.27: Posterior oil production rates (Example 2, $N_e = 500$): NMF (1st row), PCA (2nd row) and DCT (3rd row). True (red curves), observed data (red dots), simulated data of realizations (light blue curves), and ensemble mean (dark blue curves). History: $0 < t \leq 300$ days, forecast: $t > 300$ days.

CHAPTER 4

PARAMETERIZATION BASED ON TENSOR AND NON-NEGATIVE TENSOR DECOMPOSITION FOR ASSISTED HISTORY MATCHING OF CHANNELIZED RESERVOIRS WITH MULTIPLE FACIES

It is important to note that all the approaches described in the previous chapters consider the geological features as a whole and do not handle different dimensions separately. Typically, current parameterization approaches deal with a matrix constructed by vectorizing or flattening the property fields (e.g., permeability or facies fields) of the multi-dimensional reservoir where flattening refers to representing the model parameters as a one-dimensional vector. However, the geologic deposition process usually leads to dominant spatial structures in one particular direction over the others. For instance, large-scale fluvial or braided channels may meander horizontally in a very complicated way but appear to be less complex in the vertical direction. Thus, it is rational to treat different dimensions separately and retain more features in the dimension of relatively larger variability. Fortunately, with recent progress in tensor factorization/decomposition (TD) methods (De Lathauwer et al., 2000a,b; Kolda and Bader, 2009), we are able to build a multi-dimensional array, a tensor, which naturally allows the extension of the matrix factorization to tensor factorization that enables the decomposition of high-dimensional data set to extract various low-dimensional features hidden in different dimensions of the data tensor.

In the reservoir community, Afra and Gildin (2013, 2016) first introduced a tensor-based technique with higher-order singular value decomposition (HOSVD) for parameterizing the permeability fields of channelized reservoirs. On the basis of their work, Sebacher and Hanea (2018) proposed to implement the normal score transform for parameterization of facies fields first then apply the HOSVD to the tensor defined by the parameter fields.

However, all these researchers only dealt with small simple 2D cases. In this chapter, the features of the tensor-based parameterization methods are demonstrated by more complex 3D examples. Moreover, our work is highly motivated by the increasing popularity of the non-negative tensor decomposition (NTD) in recent years. Although there are many successful applications of NTD in other fields (Shashua and Hazan, 2005; Cichocki et al., 2009; Xu and Yin, 2013; Vesselinov et al., 2019), no relevant work has been done to use NTD for parameterization and history matching. Therefore, we aim to not only further investigate the properties of the TD and NTD methods but also compare the performance between TD, NTD and other parameterization approaches.

The remainder of this chapter is organized as follows: we first present and investigate the basic concepts, formulations and properties of TD and NTD methods with a reconstruction experiment followed by a discussion of truncation scheme for TD and NTD. Then, the ensemble-based method for calibrating the model parameters is briefly described. The computational results of 2D and 3D synthetic examples are shown in the next section to demonstrate the performance of the proposed history matching workflow using TD and NTD methods for complex channelized reservoirs.

4.1 Methodology

4.1.1 Notations and Basic Concepts of Tensors

As the natural generalization of vector and matrix, a tensor is actually a multi-dimensional array and the order of tensor denotes the number of dimensions which are also known as modes. Therefore, an Nth-order tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is an N-dimensional array where the elements of $\mathcal{A}$ are $a_{i_1 i_2 \cdots i_n \cdots i_N}$ for $i_n = 1, 2, \cdots, I_n$ and $n = 1, 2, \cdots, N$.

An Nth-order tensor $\mathcal{A}$ is called a rank-one tensor if it can be written as the outer product of N vectors, i.e.,

$$\mathcal{A} = \boldsymbol{u}^{(1)} \otimes \boldsymbol{u}^{(2)} \otimes \cdots \otimes \boldsymbol{u}^{(N)}, \tag{4.1}$$

where the symbol "$\otimes$" represents the vector outer product (see definition in Appendix B.1).

This means that each element of the tensor is the product of the corresponding vector elements as

$$a_{i_1 i_2 \cdots i_n \cdots i_N} = u^{(1)}_{i_1} u^{(2)}_{i_2} \cdots u^{(n)}_{i_n} \cdots u^{(N)}_{i_N}, \quad \text{for } i_n = 1, 2, \cdots, I_n \text{ and } n = 1, 2, \cdots, N. \tag{4.2}$$

For a training ensemble of N_{te} reservoir models each of which is defined on a two-dimensional $N_x \times N_y$ grid system, we can easily construct a third-order tensor, $\mathcal{A} \in \mathbb{R}^{N_x \times N_y \times N_{te}}$, by stacking all models together instead of converting each 2D model to a 1D vector. Similarly, it is straightforward to have a fourth-order tensor, $\mathcal{A} \in \mathbb{R}^{N_x \times N_y \times N_z \times N_{te}}$, for 3D reservoir models each of which is defined on a three-dimensional $N_x \times N_y \times N_z$ grid system.

4.1.2 Formulations of TD

In the literature, there are two popular tensor decomposition methods that have been widely used: Canonical Polyadic (CP) decomposition (CANDECOMP/PARAFAC) (Hitchcock, 1927; Harshman, 1970; Carroll and Chang, 1970; Kiers, 2000) and Tucker decomposition (Tucker, 1963, 1964; Levin, 1965; Tucker, 1966).

With the CP method, a tensor is approximated by decomposing it into a linear combination of rank-one tensors. For the third-order tensor described above, $\mathcal{A} \in \mathbb{R}^{N_x \times N_y \times N_{te}}$, using the CP method yields

$$\mathcal{A} \approx \hat{\mathcal{A}} = \sum_{r=1}^{R} \lambda_r \boldsymbol{u}^{(1)}_r \otimes \boldsymbol{u}^{(2)}_r \otimes \boldsymbol{u}^{(3)}_r, \tag{4.3}$$

where R denotes the rank which is defined as the smallest number of rank-one tensors that generate $\mathcal{A}$ as their sum (Kruskal, 1977) and we have vectors $\boldsymbol{u}^{(1)}_r \in \mathbb{R}^{N_x}$, $\boldsymbol{u}^{(2)}_r \in \mathbb{R}^{N_y}$ and $\boldsymbol{u}^{(3)}_r \in \mathbb{R}^{N_{te}}$ for $r = 1, \cdots, R$. It is noteworthy that there is no straight algorithm to determine the rank of a specific given tensor which is actually a NP-hard problem (Håstad, 1990). Based on the approximation of Eq. 4.3, each element in $\mathcal{A} \in \mathbb{R}^{N_x \times N_y \times N_{te}}$ can be

estimated as

$$a_{ijk} \approx \sum_{r=1}^{R} \lambda_r u_{ri}^{(1)} u_{rj}^{(2)} u_{rk}^{(3)}, \quad \text{for } i = 1, \cdots, N_x,\ j = 1, \cdots, N_y,\ k = 1, \cdots, N_{te}, \tag{4.4}$$

where $u_{rm}^{(\ell)}$ denotes the mth element of the vector $u_r^{(\ell)}$ for $\ell = 1, 2, 3$.

Different from the CP method, the Tucker method decomposes a tensor into a core tensor multiplied by factor matrices of different modes and we focus on this approach in our work because of its superior characteristics (Kolda and Bader, 2009). Therefore, for a third-order tensor, $\mathcal{A} \in \mathbb{R}^{N_x \times N_y \times N_{te}}$, we have

$$\mathcal{A} \approx \hat{\mathcal{A}} = \mathcal{B} \times_1 \boldsymbol{U}^{(1)} \times_2 \boldsymbol{U}^{(2)} \times_3 \boldsymbol{U}^{(3)}, \tag{4.5}$$

where $\mathcal{B} \in \mathbb{R}^{R_1 \times R_2 \times R_3}$ denotes the core tensor; factor matrices $U^{(1)} \in \mathbb{R}^{N_x \times R_1}$, $U^{(2)} \in \mathbb{R}^{N_y \times R_2}$ and $U^{(3)} \in \mathbb{R}^{N_{te} \times R_3}$ represent the feature changes in the x direction, the y direction and the dimension of the ensemble, respectively. R_1, R_2 and R_3 are the 1-mode, 2-mode and 3-mode rank of $\mathcal{A}$, respectively. In other words, R_1, R_2 and R_3 are the column rank of factor matrix $\boldsymbol{U}^{(1)}$, $\boldsymbol{U}^{(2)}$ and $\boldsymbol{U}^{(3)}$, respectively. The n-mode rank should not be confused with the idea of rank in Eq. 4.3 and the terminology "rank" refers to the n-mode rank in this work. In Eq. 4.5, the notation, $\times_n$, represents the n-mode multiplication between a tensor and a matrix where the calculation details are given in Appendix B.2. Note that if we retain all information for each mode, then $R_1 = N_x$, $R_2 = N_y$ and $R_3 = N_{te}$. Figure 4.1 schematically depicts the tensor decomposition of Eq. 4.5.

Based on the original work of Tucker (Tucker, 1966) and subsequent contributions of De Lathauwer et al. (2000a), the Tucker decomposition can be computed by using the higher-order singular value decomposition (HOSVD) which is a generalization of standard matrix singular value decomposition (SVD). Later, De Lathauwer et al. (2000b) proposed the higher-order orthogonal iteration (HOOI) algorithm (see Appendix B.3) in order to improve the calculations of factor matrices but HOOI is more computationally expensive

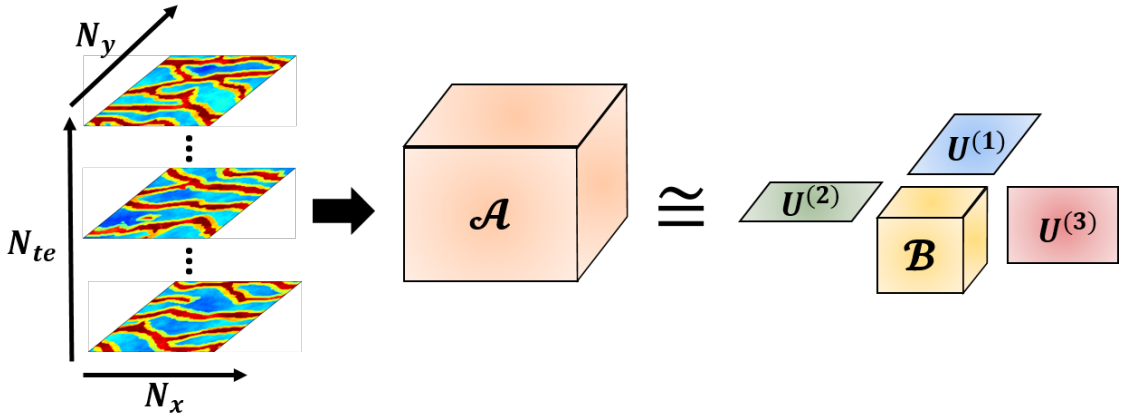


Figure 4.1: Third-order tensor construction and decomposition for an ensemble consisting of N_{te} reservoir models and each model is defined on a 2D $N_x \times N_y$ grid system.

than HOSVD (Cichocki et al., 2009). In this work, the tensor decomposition is implemented using the Tucker method with a Matlab package called "Tensorlab 3.0" (Vervliet et al., 2016a,b) where the HOSVD algorithm is available.

In fact, for a training ensemble of N_{te} reservoir models defined on a 2D $N_x \times N_y$ grid system, we can construct two types of tensors and have the following two implementation schemes of tensor decomposition:

- Third-order tensor, $\mathcal{A} \in \mathbb{R}^{N_x \times N_y \times N_{te}}$, can be decomposed into

$$\mathcal{A} \approx \hat{\mathcal{A}} = \mathcal{B} \times_1 \boldsymbol{U}^{(1)} \times_2 \boldsymbol{U}^{(2)} \times_3 \boldsymbol{U}^{(3)}, \tag{4.6}$$

  where there are one core tensor $\mathcal{B} \in \mathbb{R}^{R_1 \times R_2 \times R_3}$, and three factor matrices $\boldsymbol{U}^{(1)} \in \mathbb{R}^{N_x \times R_1}$, $\boldsymbol{U}^{(2)} \in \mathbb{R}^{N_y \times R_2}$ and $\boldsymbol{U}^{(3)} \in \mathbb{R}^{N_{te} \times R_3}$.

- Second-order tensor, $\mathcal{A} \in \mathbb{R}^{N_g \times N_{te}}$ with $N_g = N_x \times N_y$, can be decomposed into

$$\mathcal{A} \approx \hat{\mathcal{A}} = \mathcal{B} \times_1 \boldsymbol{U}^{(1)} \times_2 \boldsymbol{U}^{(2)}, \tag{4.7}$$

  where there are one core tensor $\mathcal{B} \in \mathbb{R}^{R_1 \times R_2}$, and two factor matrices $\boldsymbol{U}^{(1)} \in \mathbb{R}^{N_g \times R_1}$ and $\boldsymbol{U}^{(2)} \in \mathbb{R}^{N_{te} \times R_2}$.

Similarly, for a training ensemble of N_{te} reservoir models defined on a 3D $N_x \times N_y \times N_z$

grid system, we can construct three types of tensors and have three ways to implement the tensor decomposition as shown below.

- Fourth-order tensor, $\mathcal{A} \in \mathbb{R}^{N_x \times N_y \times N_z \times N_{te}}$, can be decomposed into

$$\mathcal{A} \approx \hat{\mathcal{A}} = \mathcal{B} \times_1 \boldsymbol{U}^{(1)} \times_2 \boldsymbol{U}^{(2)} \times_3 \boldsymbol{U}^{(3)} \times_4 \boldsymbol{U}^{(4)}, \tag{4.8}$$

  where there are one core tensor $\mathcal{B} \in \mathbb{R}^{R_1 \times R_2 \times R_3 \times R_4}$, and four factor matrices $\boldsymbol{U}^{(1)} \in \mathbb{R}^{N_x \times R_1}$, $\boldsymbol{U}^{(2)} \in \mathbb{R}^{N_y \times R_2}$, $\boldsymbol{U}^{(3)} \in \mathbb{R}^{N_z \times R_3}$ and $\boldsymbol{U}^{(4)} \in \mathbb{R}^{N_{te} \times R_4}$.

- Third-order tensor, $\mathcal{A} \in \mathbb{R}^{N_l \times N_z \times N_{te}}$ with $N_l = N_x \times N_y$, can be decomposed into

$$\mathcal{A} \approx \hat{\mathcal{A}} = \mathcal{B} \times_1 \boldsymbol{U}^{(1)} \times_2 \boldsymbol{U}^{(2)} \times_3 \boldsymbol{U}^{(3)}, \tag{4.9}$$

  where there are one core tensor $\mathcal{B} \in \mathbb{R}^{R_1 \times R_2 \times R_3}$, and three factor matrices $\boldsymbol{U}^{(1)} \in \mathbb{R}^{N_l \times R_1}$, $\boldsymbol{U}^{(2)} \in \mathbb{R}^{N_z \times R_2}$ and $\boldsymbol{U}^{(3)} \in \mathbb{R}^{N_{te} \times R_3}$. Note that there are two other variations of the third-order tensor as $\mathcal{A} \in \mathbb{R}^{N_l \times N_x \times N_{te}}$ with $N_l = N_y \times N_z$ and $\mathcal{A} \in \mathbb{R}^{N_l \times N_y \times N_{te}}$ with $N_l = N_x \times N_z$, but those variations are not considered in this work.

- Second-order tensor, $\mathcal{A} \in \mathbb{R}^{N_g \times N_{te}}$ with $N_g = N_x \times N_y \times N_z$, can be decomposed into

$$\mathcal{A} \approx \hat{\mathcal{A}} = \mathcal{B} \times_1 \boldsymbol{U}^{(1)} \times_2 \boldsymbol{U}^{(2)}, \tag{4.10}$$

  where there are one core tensor $\mathcal{B} \in \mathbb{R}^{R_1 \times R_2}$, and two factor matrices $\boldsymbol{U}^{(1)} \in \mathbb{R}^{N_g \times R_1}$ and $\boldsymbol{U}^{(2)} \in \mathbb{R}^{N_{te} \times R_2}$.

In this dissertation, the tensor construction and corresponding decomposition given by Eq. 4.8 is referred to as the "four-mode" implementation scheme. The tensor decomposition described in Eqs. 4.6 and 4.9 is called the "three-mode" implementation scheme. Similarly, Eqs. 4.7 and 4.10 define the "two-mode" implementation scheme.

One essential step before application of tensor decomposition for parameterization in history matching is to determine basis functions and calculate coefficients for each realization

in an initial ensemble of unconditional realizations of the reservoir model parameters. We take a training ensemble of N_{te} reservoir models defined on a 3D $N_x \times N_y \times N_z$ grid system as an example. If the two-mode implementation scheme is used, we first flatten every 3D reservoir model into a 1D vector and construct a second-order tensor, $\mathcal{A} \in \mathbb{R}^{N_g \times N_{te}}$, as

$$\mathcal{A} = [\boldsymbol{m}_1, \boldsymbol{m}_2, \cdots, \boldsymbol{m}_j, \cdots, \boldsymbol{m}_{N_{te}}], \tag{4.11}$$

where $N_g = N_x \times N_y \times N_z$ denotes the total number of grids and $\boldsymbol{m}_j$ is a N_g-dimensional vector containing model parameters (e.g., facies type indicator, permeability, porosity, etc.) of each gridblock of the jth ensemble member. Note in this instance, the tensor $\mathcal{A}$ is an $N_g \times N_{te}$ matrix.

By using tensor decomposition, the original tensor can be decomposed into one second-order core tensor and two factor matrices as shown in Eq. 4.10. Multiplying the core tensor, $\mathcal{B} \in \mathbb{R}^{R_1 \times R_2}$, with the second factor matrix, $\boldsymbol{U}^{(2)} \in \mathbb{R}^{N_{te} \times R_2}$, yields

$$\mathcal{A} \approx \hat{\mathcal{A}} = \mathcal{B} \times_1 \boldsymbol{U}^{(1)} \times_2 \boldsymbol{U}^{(2)} = \mathcal{C} \times_1 \boldsymbol{U}^{(1)}, \tag{4.12}$$

where $\mathcal{C} = \mathcal{B} \times_2 \boldsymbol{U}^{(2)} \in \mathbb{R}^{R_1 \times N_{te}}$ is the coefficient tensor which is actually a 2D matrix and its columns consist of coefficients for different realizations; the factor matrix, $\boldsymbol{U}^{(1)} \in \mathbb{R}^{N_g \times R_1}$, contains basis vectors in its columns. Appendix B.2 provides the procedure for computing tensor products like $\mathcal{B} \times_2 \boldsymbol{U}^{(2)}$. Therefore, each model in the original tensor, $\mathcal{A}$, can be approximated by a linear combination of basis vectors with corresponding coefficients (see Fig. 4.2).

If we use matrix symbols, $\boldsymbol{\Phi}$ and $\boldsymbol{V}$, respectively to replace $\boldsymbol{U}^{(1)}$ and $\mathcal{C}$, respectively, and introduce conventional matrix multiplication, Eq. 4.12 can be rewritten as

$$\mathcal{A} \approx \hat{\mathcal{A}} = \boldsymbol{\Phi}\boldsymbol{V}, \tag{4.13}$$

where $\hat{\mathcal{A}} \in \mathbb{R}^{N_g \times N_{te}}$ is the approximate tensor, $\boldsymbol{\Phi} \in \mathbb{R}^{N_g \times N_c}$ is the basis matrix, $\boldsymbol{V} \in \mathbb{R}^{N_c \times N_{te}}$

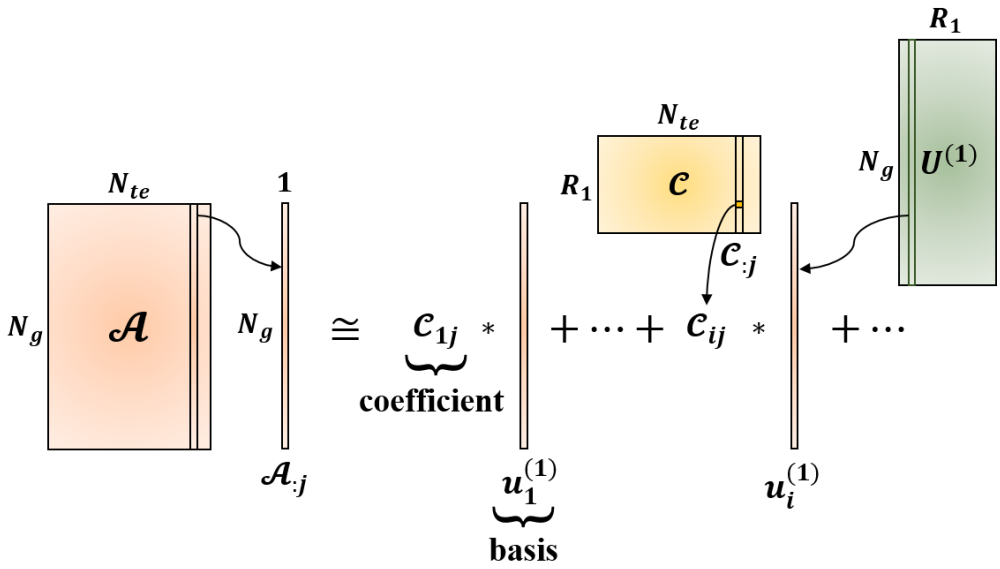


Figure 4.2: Decomposition of the second-order tensor, $\mathcal{A} \in \mathbb{R}^{N_g \times N_{te}}$ with $N_g = N_x \times N_y \times N_z$. The jth model, $\mathcal{A}_{:j}$, is estimated by a linear combination of basis vectors in $\boldsymbol{U}^{(1)}$ with the corresponding coefficients in the jth column of tensor $\mathcal{C}$.

is the coefficient matrix and $N_c = R_1$ denotes the retained number of basis functions; in particular, $\boldsymbol{\Phi} = \boldsymbol{U}^{(1)}$ and $\boldsymbol{V} = \mathcal{C}$ are given respectively by

$$\boldsymbol{\Phi} = [\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \cdots, \boldsymbol{\phi}_k, \cdots, \boldsymbol{\phi}_{N_c}], \tag{4.14}$$

$$\boldsymbol{V} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_j, \cdots, \boldsymbol{v}_{N_{te}}], \tag{4.15}$$

where $\boldsymbol{\phi}_k$ is a N_g-dimensional basis vector (i.e., $\boldsymbol{\phi}_k = [\phi_{k,1}, \phi_{k,2}, \cdots, \phi_{k,N_g}]^{\mathrm{T}}$) in the kth column of the basis matrix $\boldsymbol{\Phi}$ and $\boldsymbol{v}_j$ is a N_c-dimensional coefficient vector for reconstructing the jth ensemble member; i.e., $\boldsymbol{v}_j = [v_{j,1}, v_{j,2}, \cdots, v_{j,N_c}]^{\mathrm{T}}$. As shown in Fig. 4.2, each column of the ensemble tensor shown in Eq. 4.11 can be approximated by

$$\boldsymbol{m}_j \approx \hat{\boldsymbol{m}}_j = \sum_{k=1}^{N_c} v_{j,k}\boldsymbol{\phi}_k = \boldsymbol{\Phi}\boldsymbol{v}_j, \quad \text{for } j = 1, \cdots, N_{te}. \tag{4.16}$$

Instead of vectorizing the 3D model in the two-mode scheme, we can also construct a fourth-order tensor, $\mathcal{A} \in \mathbb{R}^{N_x \times N_y \times N_z \times N_{te}}$, for the same ensemble of 3D reservoir models. This tensor can be thought as a series of cubes (see Fig. 4.3) and each cube represents one 3D

reservoir model whose dimension is $N_x \times N_y \times N_z$. Through tensor decomposition, the tensor can be decomposed into one fourth-order core tensor and four factor matrices as shown in Eq. 4.8. The product of the core tensor, $\mathcal{B} \in \mathbb{R}^{R_1 \times R_2 \times R_3 \times R_4}$, with the last factor matrix, $\boldsymbol{U}^{(4)} \in \mathbb{R}^{N_{te} \times R_4}$, gives

$$\begin{aligned} \mathcal{A} \approx \hat{\mathcal{A}} &= \mathcal{B} \times_1 \boldsymbol{U}^{(1)} \times_2 \boldsymbol{U}^{(2)} \times_3 \boldsymbol{U}^{(3)} \times_4 \boldsymbol{U}^{(4)} \\ &= \mathcal{C} \times_1 \boldsymbol{U}^{(1)} \times_2 \boldsymbol{U}^{(2)} \times_3 \boldsymbol{U}^{(3)}, \end{aligned} \tag{4.17}$$

where $\mathcal{C} = \mathcal{B} \times_4 \boldsymbol{U}^{(4)} \in \mathbb{R}^{R_1 \times R_2 \times R_3 \times N_{te}}$ is the coefficient tensor and its cubes contain coefficients for different realizations. Thus, the pth cube (model) of tensor $\mathcal{A}$ (i.e., $\mathcal{A}_{:::p}$) can be approximated by

$$\begin{aligned} \mathcal{A}_{:::p} \approx \hat{\mathcal{A}}_{:::p} &= \sum_{i=1}^{R_1} \sum_{j=1}^{R_2} \sum_{k=1}^{R_3} \mathcal{C}_{ijkp} \boldsymbol{\Phi}_{ijk} \\ &= \sum_{i=1}^{R_1} \sum_{j=1}^{R_2} \sum_{k=1}^{R_3} \mathcal{C}_{ijkp} \boldsymbol{u}_i^{(1)} \otimes \boldsymbol{u}_j^{(2)} \otimes \boldsymbol{u}_k^{(3)}, \quad \text{for } p = 1, \cdots, N_{te}. \end{aligned} \tag{4.18}$$

where $\boldsymbol{\Phi}_{ijk}$ denotes the basis cube whose dimension is $N_x \times N_y \times N_z$ and $\mathcal{C}_{ijkp}$ is the corresponding coefficient for pth model; $\boldsymbol{u}_i^{(1)} \in \mathbb{R}^{N_x \times 1}$, $\boldsymbol{u}_j^{(2)} \in \mathbb{R}^{N_y \times 1}$ and $\boldsymbol{u}_k^{(3)} \in \mathbb{R}^{N_z \times 1}$ denote the ith, jth and kth column vector of $\boldsymbol{U}^{(1)}$, $\boldsymbol{U}^{(2)}$ and $\boldsymbol{U}^{(3)}$, respectively. The decomposition process is more clearly illustrated in Fig. 4.3.

For the other TD implementation schemes, the derivation process is similar. The basis functions are fixed during the history matching process and the coefficients of each ensemble member are calibrated as model parameters by assimilating observed data.

4.1.3 Formulations of NTD

On the basis of conventional tensor decomposition, the non-negativity constraints can be imposed in the non-negative tensor decomposition (NTD) so that the components in the resulting core tensor and factor matrices are non-negative. In order to satisfy this constraint for an Nth-order tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, we need to solve an optimization problem in order

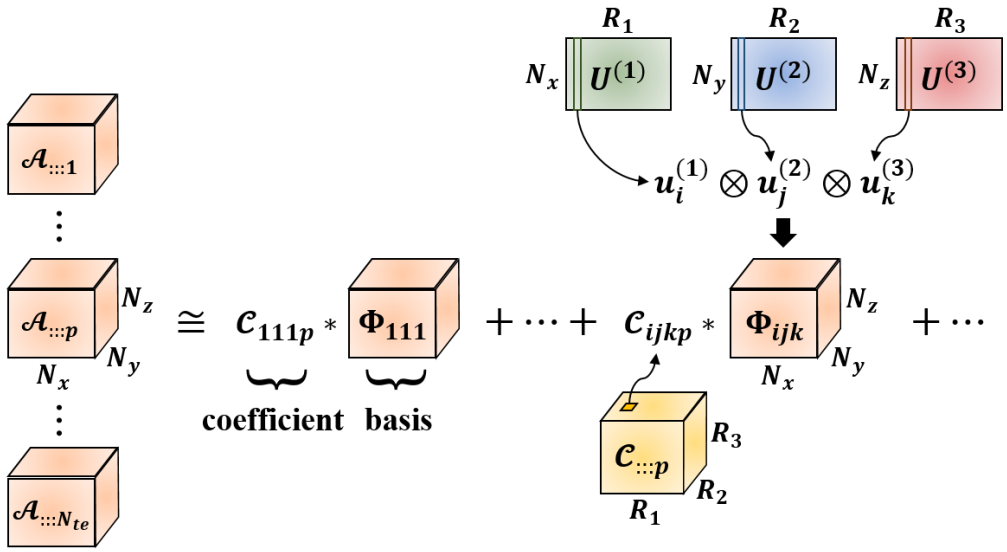


Figure 4.3: Decomposition of the fourth-order tensor, $\mathcal{A} \in \mathbb{R}^{N_x \times N_y \times N_z \times N_{te}}$. The pth model, $\mathcal{A}_{:::p}$, is estimated by a linear combination of basis cubes obtained by the outer products of column vectors from factor matrices (i.e., $\boldsymbol{U}^{(1)}$, $\boldsymbol{U}^{(2)}$ and $\boldsymbol{U}^{(3)}$), with the corresponding coefficients in the pth cube of tensor $\mathcal{C}$.

to minimize the least squares objective function given by

$$O(\hat{\mathcal{A}}) = ||\mathcal{A} - \hat{\mathcal{A}}||_{\mathrm{F}}^2 = ||\mathcal{A} - \mathcal{B} \times_1 \boldsymbol{U}^{(1)} \times_2 \boldsymbol{U}^{(2)} \cdots \times_N \boldsymbol{U}^{(N)}||_{\mathrm{F}}^2, \tag{4.19}$$

where the subscript "F" denotes the Frobenius norm. With the multiplicative update rules proposed by Lee and Seung (2001) who developed the non-negative matrix factorization (NMF) approach, we can update the nth factor matrix, $\boldsymbol{U}^{(n)}$, by

$$\boldsymbol{U}_{\ell+1}^{(n)} = \boldsymbol{U}_{\ell}^{(n)} \circledast \frac{\mathcal{A}_{(n)}^{\ell} (\mathcal{B}_{(n)}^{\ell})^{\mathrm{T}}}{\boldsymbol{U}_{\ell}^{(n)} \mathcal{B}_{(n)}^{\ell} (\mathcal{B}_{(n)}^{\ell})^{\mathrm{T}}}, \tag{4.20}$$

where ℓ is the iteration index, the symbol "$\circledast$" denotes the Hadamard product (see Appendix B.4) and $\boldsymbol{U}_{\ell}^{(n)}$ has the dimension of $I_n \times R_n$ where R_n is the rank of the nth mode and if all information is retained, we have the full rank, i.e., $R_n = I_n$. $\mathcal{A}_{(n)}^{\ell}$ is the n-mode matricization of the Tucker model calculated by

$$\mathcal{A}_{(n)}^{\ell} = \boldsymbol{U}_{\ell}^{(n)} \mathcal{B}_{(n)}^{\ell}, \tag{4.21}$$

where $\mathcal{B}^{\ell}_{(n)}$ is computed in the following way:

$$\mathcal{B}^{\ell}_{(n)} = \mathcal{B}^{\ell} \times_1 \boldsymbol{U}^{(1)}_{\ell} \cdots \times_{n-1} \boldsymbol{U}^{(n-1)}_{\ell} \times_{n+1} \boldsymbol{U}^{(n+1)}_{\ell} \cdots \times_N \boldsymbol{U}^{(N)}_{\ell}. \tag{4.22}$$

The other factor matrices can be easily updated in the same way by unfolding the Tucker model into associated modes. After all the factor matrices, $\boldsymbol{U}^{(n)}$, for $n = 1, \cdots, N$, are updated, the core tensor is updated as

$$\mathcal{B}^{\ell+1} = \mathcal{B}^{\ell} \circledast \frac{\mathcal{A} \times_1 \left(\boldsymbol{U}^{(1)}_{\ell+1}\right)^{\mathrm{T}} \cdots \times_N \left(\boldsymbol{U}^{(N)}_{\ell+1}\right)^{\mathrm{T}}}{\mathcal{B}^{\ell} \times_1 \left(\boldsymbol{U}^{(1)}_{\ell+1}\right)^{\mathrm{T}} \boldsymbol{U}^{(1)}_{\ell+1} \cdots \times_N \left(\boldsymbol{U}^{(N)}_{\ell+1}\right)^{\mathrm{T}} \boldsymbol{U}^{(N)}_{\ell+1}}. \tag{4.23}$$

With Eqs. 4.20 and 4.23, the updates of factor matrices and core tensor are guaranteed to be positive and the updating procedure is iteratively conducted until the discrepancy between $\mathcal{A}$ and $\hat{\mathcal{A}}$ reaches the stopping criteria. The complete derivations can be found in Kim and Choi (2007) and Lee and Seung (2001) where it is proved that the application of multiplicative update rules is guaranteed to find at least a locally optimal solution of the problem defined in Eq. 4.19. Note that although the update rules are similar, the NMF approach factorizes the ensemble matrix into the basis matrix and coefficient matrix directly, which yields different basis functions compared to those obtained by the NTD method. Since there is no HOSVD algorithm (De Lathauwer et al., 2000a) for NTD in Tensorlab 3.0, we employ a Python package called "TensorLy" (Kossaifi et al., 2019) to implement NTD using the HOOI algorithm (De Lathauwer et al., 2000b) in this work; see Appendix B.3.

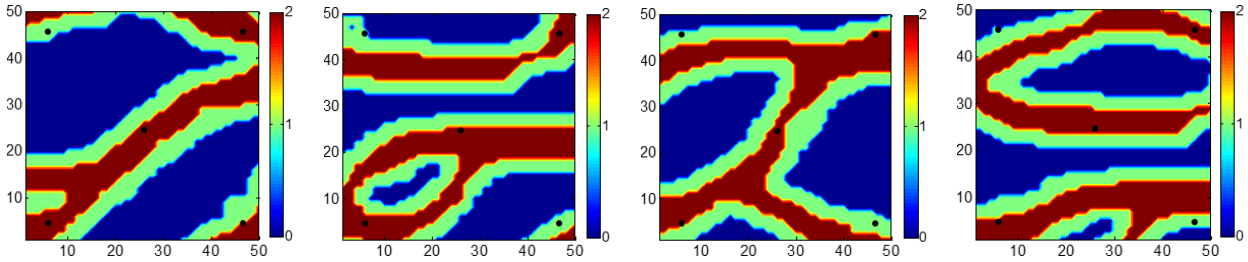
4.1.4 Reconstruction Experiment

In order to assess and illustrate the parameterization performance of the TD and NTD methods, a reconstruction experiment was conducted with a training ensemble of 200 models ($N_{te} = 200$). Each ensemble member is a 2D channelized reservoir containing three distinct facies (see Table 4.1). The reservoir model is based on a 50×50 grid system so that $N_x = 50$, $N_y = 50$ and $N_g = 2500$. The facies models are generated by using the multi-point statistics (MPS) algorithm and four different models are presented in Fig. 4.4 where the

black dots show the locations of five wells. The two wells in the top left and bottom right corners of the reservoir are in the levee facies while the other three wells are drilled in the sand channels. We assume that the facies type at these wells are observed in advance and used as hard data in the reservoir modeling process.

Table 4.1: Facies

Facies	Indicator	Color
Sand	2	Red
Levee	1	Green
Shale	0	Blue



(a) Model #1 (b) Model #2 (c) Model #3 (d) Model #4

Figure 4.4: Original facies models. Black dots represent the well locations and the facies type observations are used as hard data in the model generation process.

For the purpose of comparison, we also tested the NMF method and compared the results with those obtained with the TD and NTD. In this experiment, the total retained number of basis functions is arbitrarily set to be $N_c = 100$ which means the dimensionality of each facies model is reduced from 2500 to 100. To achieve this goal, we designed nine implementation schemes of parameterization with the TD, NTD and NMF methods; see the detailed descriptions in Table 4.2. We do not compress the dimension of ensemble in this experiment which leads to $N_c = R_1$ for the two-mode implementation schemes and $N_c = R_1 \times R_2$ for the three-mode implementation schemes. The retained number of basis functions for different dimensions shown in Table 4.2 are determined heuristically but a truncation scheme for TD and NTD is presented in next section. The last column in Table 4.2 gives the computing time, t^{para}, used to complete the parameterization. Note that t^{para} only

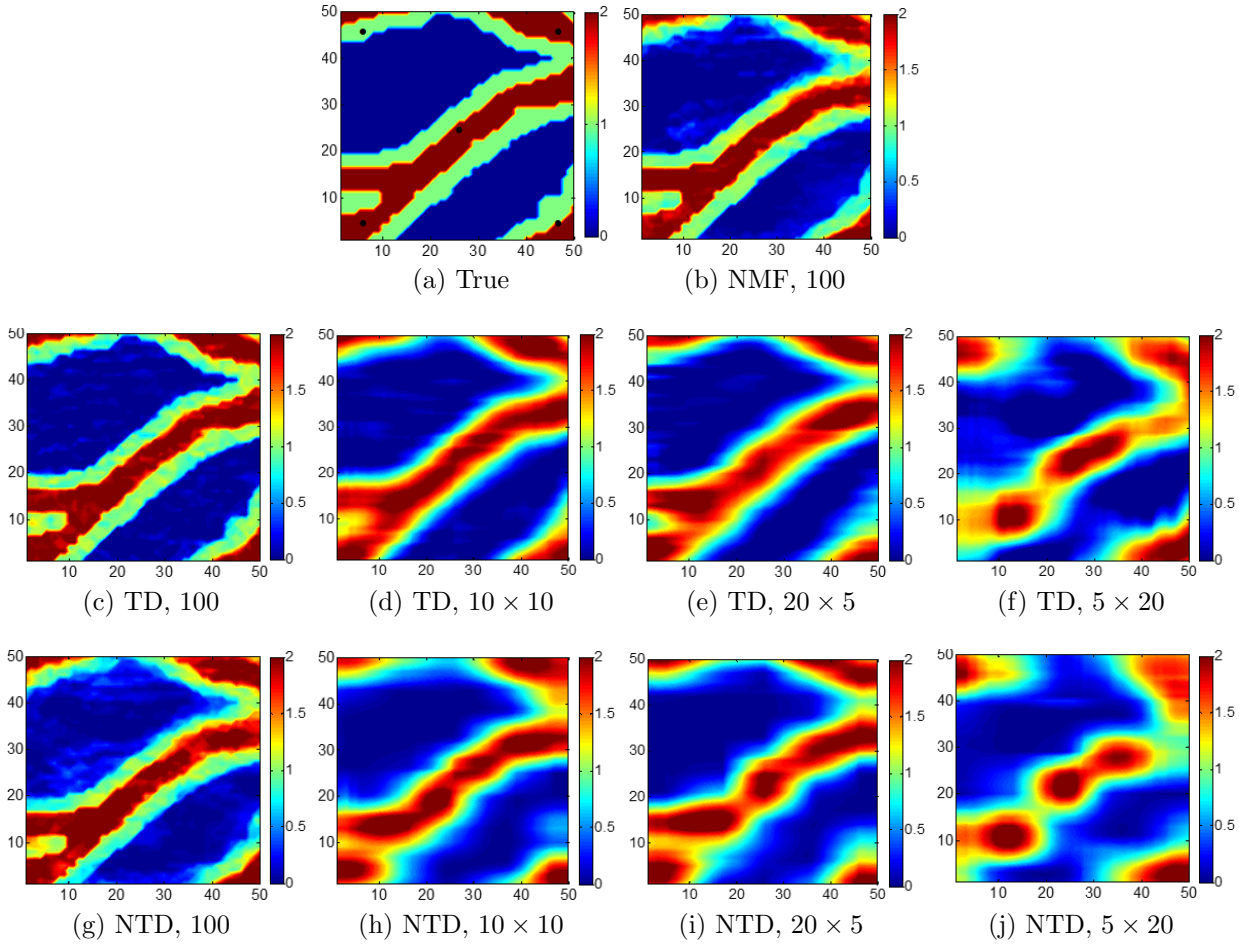
represents the cost of computing the core tensor and factor matrices from the original tensor. The CPU in the computer is Intel(R) Xeon(R) E5-1620 v2 (10M Cache, 3.70 GHz) with 16.0 GB RAM. It is shown that in this experiment the computing time of NTD is about 10 times than that of TD when the implementation scheme is the same and also indicates that the two-mode scheme takes more time than the multi-mode schemes to complete decomposition.

Table 4.2: Parameterization schemes ($N_{te} = 200$)

Method	Ensemble tensor	Scheme	Retained number	Name	t^{para}
TD	$\mathcal{A} \in \mathbb{R}^{N_g \times N_{te}}$	2-mode	$R_1 = 100, R_2 = N_{te}$	TD, 100	0.32s
	$\mathcal{A} \in \mathbb{R}^{N_x \times N_y \times N_{te}}$	3-mode	$R_1 = 10, R_2 = 10, R_3 = N_{te}$	TD, 10×10	0.16s
	$\mathcal{A} \in \mathbb{R}^{N_x \times N_y \times N_{te}}$	3-mode	$R_1 = 20, R_2 = 5, R_3 = N_{te}$	TD, 20×5	0.21s
	$\mathcal{A} \in \mathbb{R}^{N_x \times N_y \times N_{te}}$	3-mode	$R_1 = 5, R_2 = 20, R_3 = N_{te}$	TD, 5×20	0.21s
NTD	$\mathcal{A} \in \mathbb{R}^{N_g \times N_{te}}$	2-mode	$R_1 = 100, R_2 = N_{te}$	NTD, 100	2.06s
	$\mathcal{A} \in \mathbb{R}^{N_x \times N_y \times N_{te}}$	3-mode	$R_1 = 10, R_2 = 10, R_3 = N_{te}$	NTD, 10×10	1.79s
	$\mathcal{A} \in \mathbb{R}^{N_x \times N_y \times N_{te}}$	3-mode	$R_1 = 20, R_2 = 5, R_3 = N_{te}$	NTD, 20×5	1.92s
	$\mathcal{A} \in \mathbb{R}^{N_x \times N_y \times N_{te}}$	3-mode	$R_1 = 5, R_2 = 20, R_3 = N_{te}$	NTD, 5×20	1.88s
NMF	$\mathcal{A} \in \mathbb{R}^{N_g \times N_{te}}$	/	$N_c = 100$	NMF, 100	1.22s

With the training ensemble, we can construct a second-order tensor, $\mathcal{A} \in \mathbb{R}^{N_g \times N_{te}}$ with $N_g = N_x \times N_y$, for two-mode TD and NTD, and a third-order tensor, $\mathcal{A} \in \mathbb{R}^{N_x \times N_y \times N_{te}}$, for three-mode TD and NTD. By using Eqs. 4.6 and 4.7, the original tensors are decomposed into core tensors and factor matrices, and the products of core tensors and factor matrices lead to reconstructed models. Note that there are continuous values not discrete facies indicators in the reconstructed models. Figure. 4.5 and Figure. 4.6 each show a different true facies model and the corresponding reconstructed models obtained by applying nine parameterization schemes to the training ensemble. From the results, it is shown that the "TD, 100", "NTD, 100" and "NMF, 100" methods are able to yield well-reproduced models where the large-scale channels are almost reproduced completely. When the three-mode schemes are adopted, the TD and NTD methods extract the geological features in the x and y directions separately. For instance, the orientation of fluvial channels in the second

true model (see Fig. 4.6) is basically along the x direction. Thus, the reproduced structures obtained with the "TD, 20×5" and "NTD, 20×5" schemes where more features in the x direction are retained display enhanced spatial continuity in the x direction compared to the "5×20" results. This observation shows the flexibility of the tensor-based parameterization approaches and this characteristic is supposed to be useful when it comes to the geological systems of strong spatial orientation.



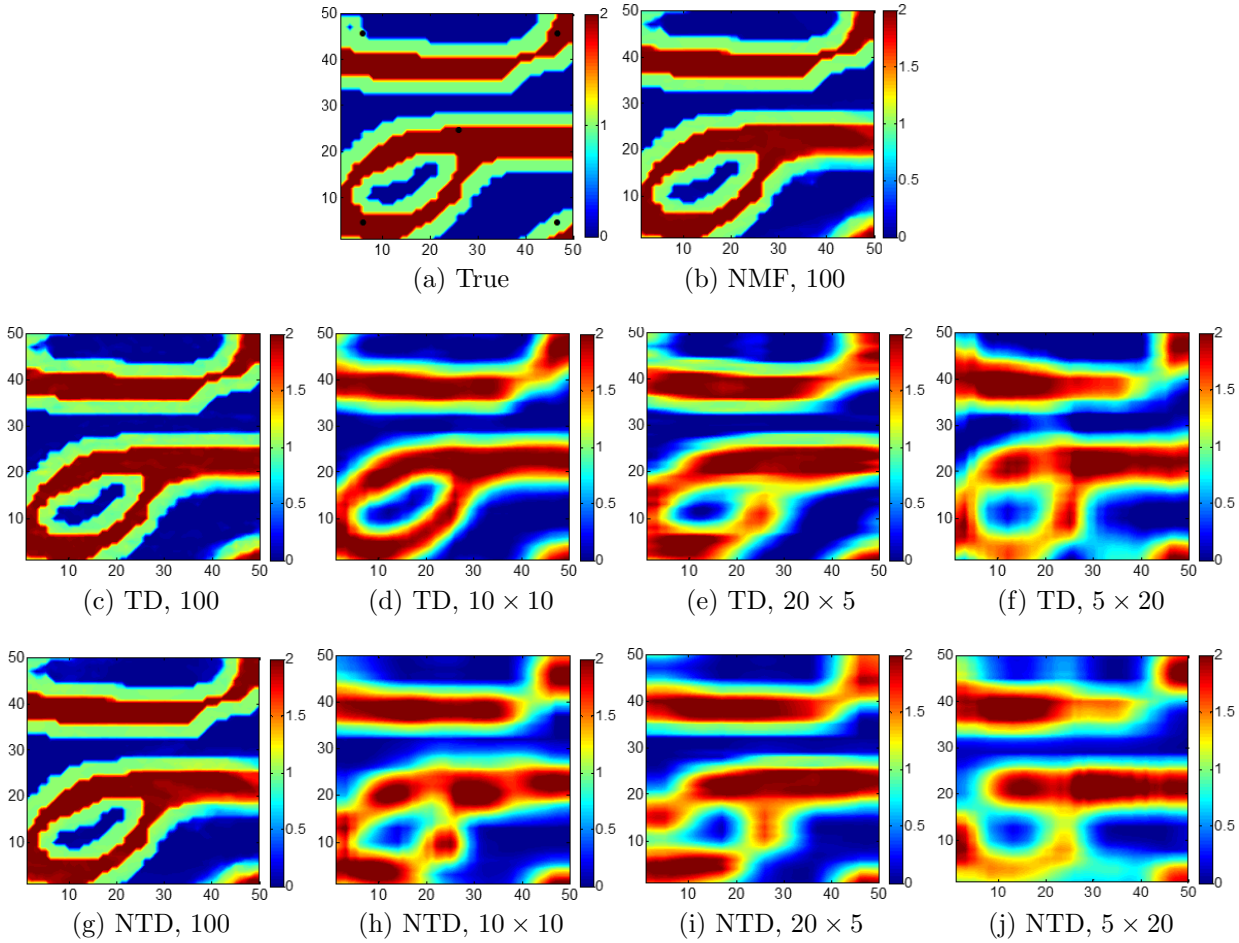
(a) True (b) NMF, 100

(c) TD, 100 (d) TD, 10×10 (e) TD, 20×5 (f) TD, 5×20

(g) NTD, 100 (h) NTD, 10×10 (i) NTD, 20×5 (j) NTD, 5×20

Figure 4.5: Comparison of reconstructed facies model #1

From the retained basis functions displayed in Figs. 4.7 and 4.8, the behaviors of different tensor-based parameterizations are illustrated more clearly in these figures. Among the total 100 retained basis functions, we selected two bases (i.e., the first and the last one) for visualization. With the TD method, the retained leading bases (e.g., basis #1) presents large-scale structures but the features become more trivial in the non-leading bases (e.g.,



(a) True (b) NMF, 100

(c) TD, 100 (d) TD, 10×10 (e) TD, 20×5 (f) TD, 5×20

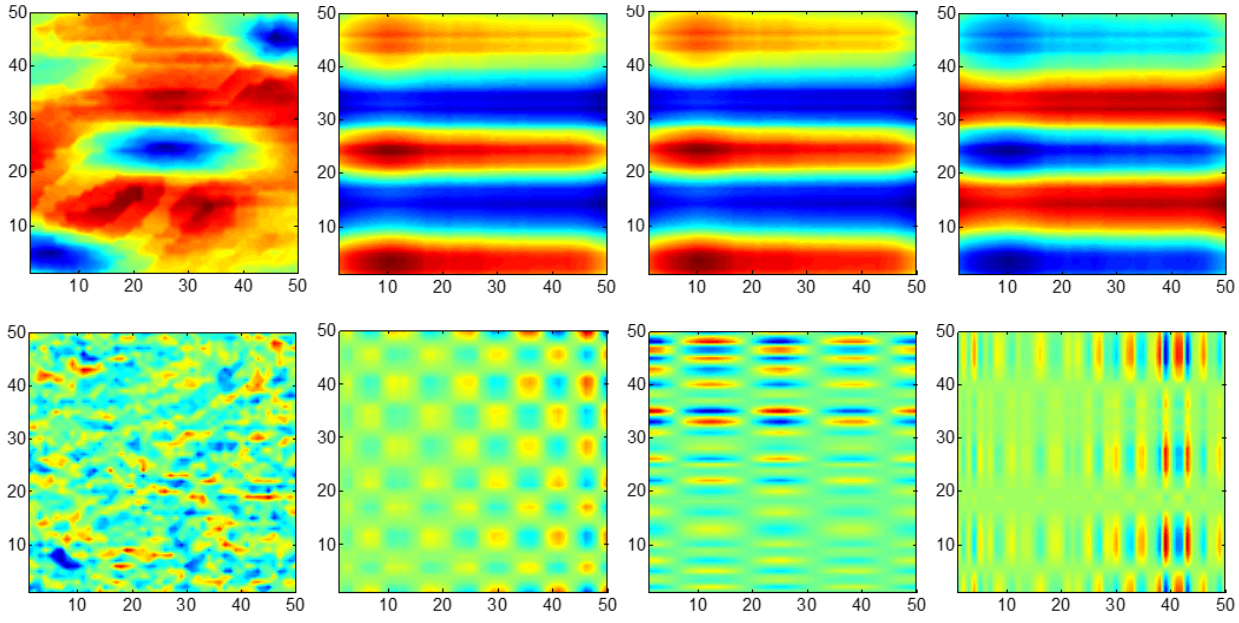
(g) NTD, 100 (h) NTD, 10×10 (i) NTD, 20×5 (j) NTD, 5×20

Figure 4.6: Comparison of reconstructed facies model #2

basis #100). Moreover, it can be seen that when different model dimensions are handled individually by the three-mode implementation scheme, the retained bases show dominant features in the direction where more features are retained. On the contrary, in the basis images obtained by the NTD method (see Fig. 4.8), the dark blue pixels are approximately zero and all the retained bases appear to extract localized features of similar size. Intuitively, one might expect localized basis functions to be superior to global ones for constructing a good approximation of any model.

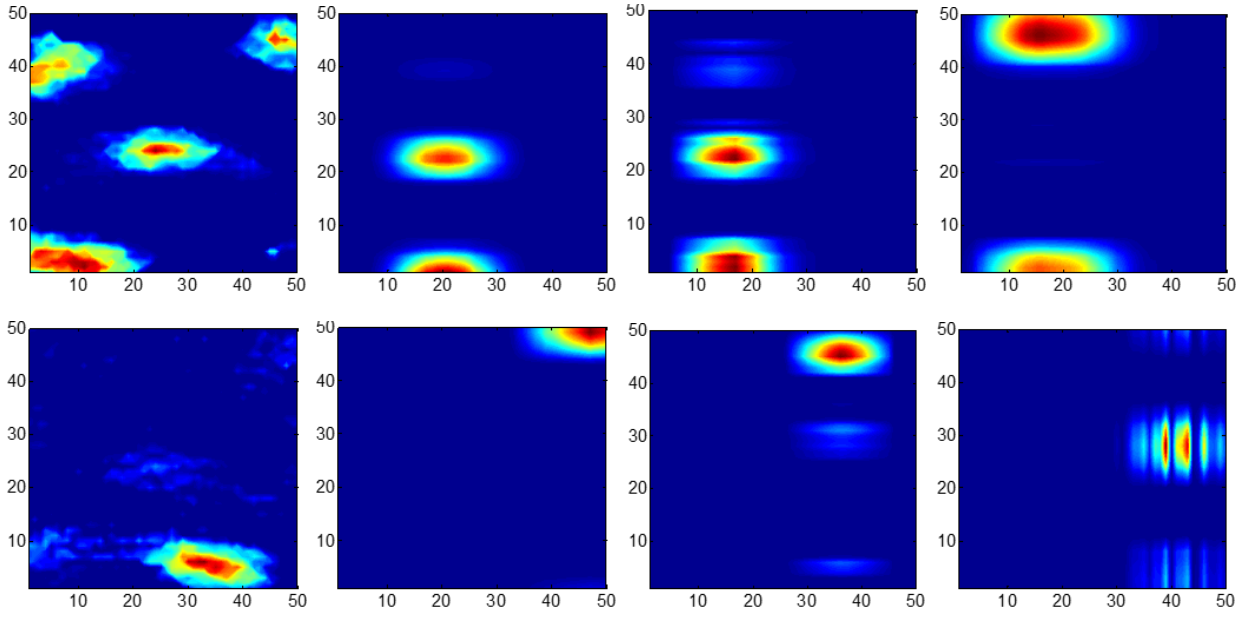
To quantitatively evaluate the reconstruction performance of different parameterizations, the reconstructed model mismatch is calculated by

$$O(\hat{\mathcal{A}}) = \left\| \mathcal{A} - \hat{\mathcal{A}} \right\|_{\mathrm{F}}, \tag{4.24}$$



(a) TD, 100 (b) TD, 10×10 (c) TD, 20×5 (d) TD, 5×20

Figure 4.7: Basis functions retained by the TD method: basis #1 (top row) and basis #100 (bottom row).



(a) NTD, 100 (b) NTD, 10×10 (c) NTD, 20×5 (d) NTD, 5×20

Figure 4.8: Basis functions retained by the NTD method: basis #1 (top row) and basis #100 (bottom row).

where the subscript "F" denotes the Frobenius norm. For the ensemble of 2D reservoir models, the tensor is constructed to be $\mathcal{A} \in \mathbb{R}^{N_g \times N_{te}}$ and $\mathcal{A} \in \mathbb{R}^{N_x \times N_y \times N_{te}}$ with two-mode

and three-mode scheme, respectively. In Fig. 4.9, the "TD, 100" parameterization is shown to give the smallest model mismatch, and the "NTD, 100" and "NMF, 100" also yield good results. With the same implementation scheme, we found that the TD method usually obtains better reconstructed models than the NTD method.

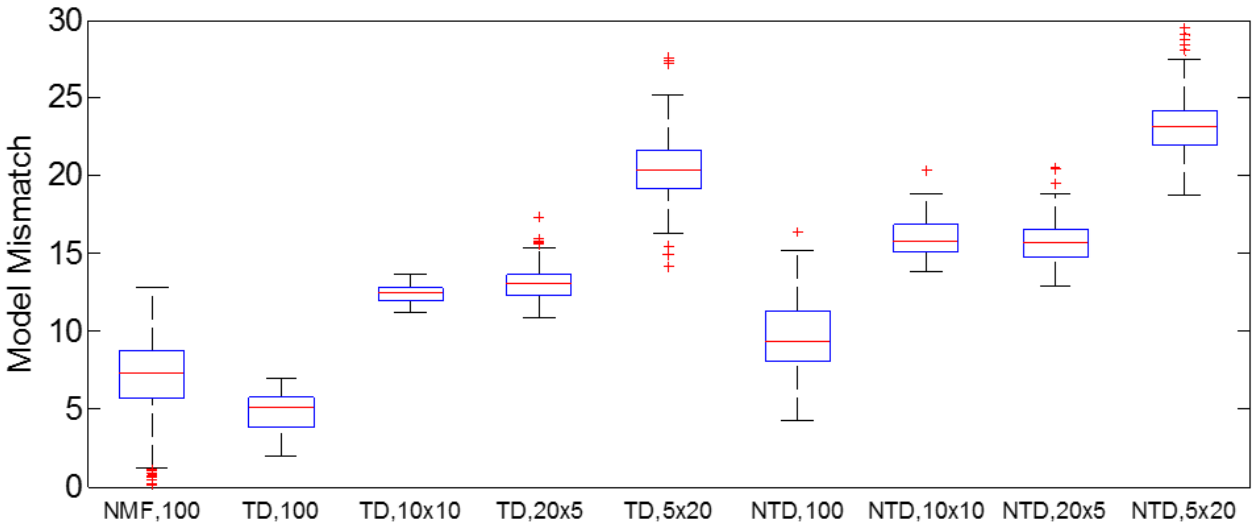


Figure 4.9: Model mismatch for reconstruction results. Median (central red mark), 25th and 75th percentiles (bottom and top edges of the box), extreme data points (whiskers), and outliers ("+" symbol).

Based on the presented results and discussion, it appears that the two-mode implementation scheme may be the best option when the retained number of basis functions is the same. But one must keep in mind that the three-mode TD and NTD takes less computational time to complete than the two-mode TD and NTD (see the computational cost in Table 4.2). Moreover, at this point, we have arbitrarily chosen the ranks in the three-mode implementations. In addition, if the computational cost of the two-mode and three-mode implementations is forced to be identical, we are able to retain more basis functions with the three-mode implementation scheme and may obtain better reconstructed models than those obtained with the two-mode scheme. One should also note that the computational cost of computing the basis elements is on the order of a few seconds which is negligible compared to the cost of running the reservoir simulation a few hundred times to perform history matching where each run of the simulation takes on the order of one minute for the two-dimensional problem and on the order of two minutes for the more complicated problem

considered later.

4.1.5 Truncation Scheme

As stated previously, for a training ensemble of N_{te} reservoir models defined on a 3D $N_x \times N_y \times N_z$ grid system, it is straightforward to construct a fourth-order tensor, $\mathcal{A} \in \mathbb{R}^{N_x \times N_y \times N_z \times N_{te}}$. If we retain all information for each mode, a full tensor decomposition can be obtained using HOSVD algorithm:

$$\mathcal{A} \approx \hat{\mathcal{A}} = \mathcal{B} \times_1 \boldsymbol{U}^{(1)} \times_2 \boldsymbol{U}^{(2)} \times_3 \boldsymbol{U}^{(3)} \times_4 \boldsymbol{U}^{(4)}, \tag{4.25}$$

where $\mathcal{B} \in \mathbb{R}^{R_1 \times R_2 \times R_3 \times R_4}$ is the core tensor; $U^{(1)} \in \mathbb{R}^{N_x \times R_1}$, $U^{(2)} \in \mathbb{R}^{N_y \times R_2}$, $U^{(3)} \in \mathbb{R}^{N_z \times R_3}$ and $U^{(4)} \in \mathbb{R}^{N_{te} \times R_4}$ are the factor matrices, and since all features are retained, $R_1 = N_x$, $R_2 = N_y$, $R_3 = N_z$ and $R_4 = N_{te}$.

If we take the 1st mode (x dimension) as an example, the Frobenius norms of the "slices" along the 1st mode of the core tensor $\mathcal{B}$ can be calculated by

$$\lambda_j^{(1)} = \left\| \mathcal{B}_{j:::} \right\|_{\mathrm{F}}, \quad \text{for } j = 1, \cdots, N_x, \tag{4.26}$$

where $\mathcal{B}_{j:::}$ denotes the jth "slice" along the 1st mode which is replaced by $\mathcal{B}_{:j::}$, $\mathcal{B}_{::j:}$ and $\mathcal{B}_{:::j}$ for the 2nd, 3rd and 4th modes, respectively.

As a generalization of standard SVD of matrix where the singular values decreases on the diagonal, the $\lambda_j^{(1)}$ presents a decreasing order with respect to the Frobenius norm as

$$\lambda_{\max}^{(1)} = \lambda_1^{(1)} \geq \lambda_2^{(1)} \geq \cdots \geq \lambda_{N_x}^{(1)} = \lambda_{\min}^{(1)} \geq 0. \tag{4.27}$$

Therefore, we can determine the rank of the 1st mode, R_1, by choosing the smallest positive integer, R_1, that satisfies

$$\sum_{j=1}^{R_1} \lambda_j^{(1)} \geq \zeta \sum_{j=1}^{N_x} \lambda_j^{(1)}, \tag{4.28}$$

where ζ is a real number not greater than one, and the choice of ζ determines the truncation level of features in the 1st mode. It is obvious that more features are retained with larger ζ. Through the same procedure, we can obtain R_2, R_3 and R_4 for the other modes of the fourth-order tensor.

The approximation quality of the original tensor, $\mathcal{A}$, can be quantified by the normalized model mismatch defined by

$$\xi = \frac{\left\| \mathcal{A} - \hat{\mathcal{A}} \right\|_{\mathrm{F}}}{\|\mathcal{A}\|_{\mathrm{F}}}, \tag{4.29}$$

where the subscript "F" denotes the Frobenius norm and $\hat{\mathcal{A}}$ is the approximate tensor obtained by HOSVD (see Eq. 4.25) given the ranks of different modes, i.e., R_1, R_2, R_3 and R_4. Since ζ is commonly smaller than one, the tensor is compressed and we can calculate the model reduction ratio by

$$\eta = \frac{\mathrm{size}(\mathcal{B})}{\mathrm{size}(\mathcal{A})}, \tag{4.30}$$

where $\mathrm{size}(\mathcal{A}) = N_x \times N_y \times N_z \times N_{te}$ and $\mathrm{size}(\mathcal{B}) = R_1 \times R_2 \times R_3 \times R_4$.

In order to balance the approximation quality and model reduction ratio, the value of ζ is set to 0.9 by default in this work and the normalized model mismatch, ξ, is calculated to check if it is greater than 0.2. If so, we increase the value of ζ until ξ is smaller than 0.2. For the same ensemble of 3D reservoir models, we can also construct a third-order tensor $\mathcal{A} \in \mathbb{R}^{N_l \times N_z \times N_{te}}$ with $N_l = N_x \times N_y$ or a second-order tensor $\mathcal{A} \in \mathbb{R}^{N_g \times N_{te}}$ with $N_g = N_x \times N_y \times N_z$, and use the same procedure to determine the ranks of different modes when implementing the tensor decomposition.

The proposed truncation scheme is tested with the training ensemble of 200 models ($N_x = 50$, $N_y = 50$, $N_{te} = 200$) used in the previous reconstruction experiment (see Section 4.1.4). A third-order tensor, $\mathcal{A} \in \mathbb{R}^{N_x \times N_y \times N_{te}}$, is constructed and the truncation scheme is applied with three different values of ζ (0.85, 0.9 and 0.95). The obtained ranks of three modes, normalized model mismatch ξ, model reduction ratio η and computing time

of parameterization t^{para} are given in the top three rows of Table 4.3. It is clear that as ζ increases, the model mismatch decreases and the model reduction ratio increases. For the reconstruction experiment presented previously where we determined R_1, R_2 and R_3 in a heuristic way, the corresponding ξ, η and t^{para} are shown in the bottom three rows of Table 4.3 for comparison. We can see that although the dimensionality of tensor is reduced significantly ($\eta = 0.04$), the model mismatch is larger than 0.2 which is not desirable. Therefore, the developed truncation scheme provides an efficient way to determine proper ranks of different modes given a specific tensor. Although the results of Table 4.3 suggest we can apply truncated TD and NTD, we will postpone the application of the truncated scheme in the history matching workflow until we consider more realistic three-dimensional problems with heterogeneous permeability fields.

Table 4.3: Application of truncation scheme

Method	ζ	ξ	η	R_1	R_2	R_3	t^{para}
TD	0.95	0.08	0.51	42	40	152	1.21s
	0.9	0.12	0.26	35	31	122	0.22s
	0.85	0.15	0.13	28	24	98	0.16s
TD	/	0.21	0.04	10	10	200	0.06s
	/	0.23	0.04	20	5	200	0.11s
	/	0.35	0.04	5	20	200	0.05s

4.1.6 Data Assimilation and Post-Processing

In this work, we calibrate the facies and the petrophysical properties (i.e., permeability) within facies simultaneously so that the vector of model parameters, $\boldsymbol{y}$, is given by

$$\boldsymbol{y} = [\boldsymbol{m}_{\text{facies}}^{\text{T}}, \boldsymbol{m}_{\text{perm}}^{\text{T}}]^{\text{T}}, \tag{4.31}$$

where $\boldsymbol{m}_{\text{facies}}$ and $\boldsymbol{m}_{\text{perm}}$ represents the model parameters associated with the facies field

and the permeability within each facies, respectively. Specifically, we define

$$\boldsymbol{m}_{\text{facies}} = \boldsymbol{v}_{\text{facies}}, \tag{4.32}$$

where $\boldsymbol{v}_{\text{facies}}$ denotes the coefficient vector of facies field obtained by TD and NTD. If the permeability within facies is homogeneous, we define

$$\boldsymbol{m}_{\text{perm}} = [k_{\text{sand}}, k_{\text{levee}}, k_{\text{shale}}]^{\text{T}}, \tag{4.33}$$

where k_{sand}, k_{levee} and k_{shale} denote the natural logarithm of the permeability ($\ln(k)$) within sand, levee and shale facies, respectively. On the other hand, if the permeability within facies is heterogeneous, we define

$$\boldsymbol{m}_{\text{perm}} = [\boldsymbol{v}_{\text{sand}}^{\text{T}}, \boldsymbol{v}_{\text{levee}}^{\text{T}}, \boldsymbol{v}_{\text{shale}}^{\text{T}}]^{\text{T}}, \tag{4.34}$$

where $\boldsymbol{v}_{\text{sand}}$, $\boldsymbol{v}_{\text{levee}}$ and $\boldsymbol{v}_{\text{shale}}$ are the parameterized coefficient vectors corresponding to the $\ln(k)$ field of each facies, i.e., $\boldsymbol{m}_{\text{sand}}$, $\boldsymbol{m}_{\text{levee}}$ and $\boldsymbol{m}_{\text{shale}}$, respectively. Note that in this work we apply the two-mode implementation scheme of TD and NTD to the $\ln(k)$ fields of facies with $N_c = N_{te}$. The specific expression of $\boldsymbol{y}$ is given at the beginning of computational examples.

In order to honor historical production data by adjusting the model parameters, the ensemble smoother with multiple data assimilation (ES-MDA) algorithm (Emerick and Reynolds, 2012, 2013a) is employed. In the original ES-MDA, same set of observed data are assimilated N_a times but with the measurement error covariance matrix $\boldsymbol{C}_{\text{D}}$ replaced by $\alpha_\ell \boldsymbol{C}_{\text{D}}$ at the ℓth assimilation step where the inflation factor $\alpha_\ell \geq 1$ and

$$\sum_{\ell=1}^{N_a} \frac{1}{\alpha_\ell} = 1. \tag{4.35}$$

For the linear Gaussian case, the constraint in Eq. 4.35 ensures that ES-MDA does

the sampling correctly (Emerick and Reynolds, 2012, 2013a). As in previous chapter, we employ the ES-MDA-GEO (Rafiee and Reynolds, 2017) where the "GEO" refers to the fact that after the largest inflation factor α_1 is chosen, all other factors are calculated by

$$\alpha_{\ell+1} = \beta\alpha_\ell, \quad \text{for } \ell = 1, \cdots, N_a - 1, \tag{4.36}$$

where the factor β is solved in the domain of $(0, 1)$ by

$$\frac{1-\beta^{N_a}}{1-\beta} = \alpha_1, \tag{4.37}$$

so that Eq. 4.35 is satisfied. As in Rafiee and Reynolds (2017), the inflation factor, α, is chosen based on the discrepancy principle which was first advocated by Iglesias (2015) to determine inflation factors in an iterative ensemble smoother algorithm. However, here we also require that $\alpha_1 \leq 1000$. A brief derivation of update equations in the ES-MDA-GEO is given in Appendix A.1.

It should be noted that after one iteration of ES-MDA-GEO, the updated facies fields usually contain continuous values rather than categorical type indicators. Therefore, a post-processing procedure is adapted to implement the discrete mapping and the corresponding details can be found in Section 2.3.2 of Chapter 2 and Section 3.1.3 of Chapter 3. Based on our proposed methodology, the complete history matching workflow is shown in Fig. 4.10. Note that since we use an iterative ensemble smoother algorithm and the training ensemble for parameterization is typically very large (e.g., $N_{te} = 1000$), we randomly select N_e realizations (i.e., $N_e < N_{te}$) as the prior ensemble for history matching in order to reduce the computational cost.

4.2 Computational Results and Discussion

Two synthetic examples are designed and tested in this section in order to assess the performance of the history matching workflow combining the TD and NTD parameterization methods with ensemble-based data assimilation algorithm. As we have established in Chap-

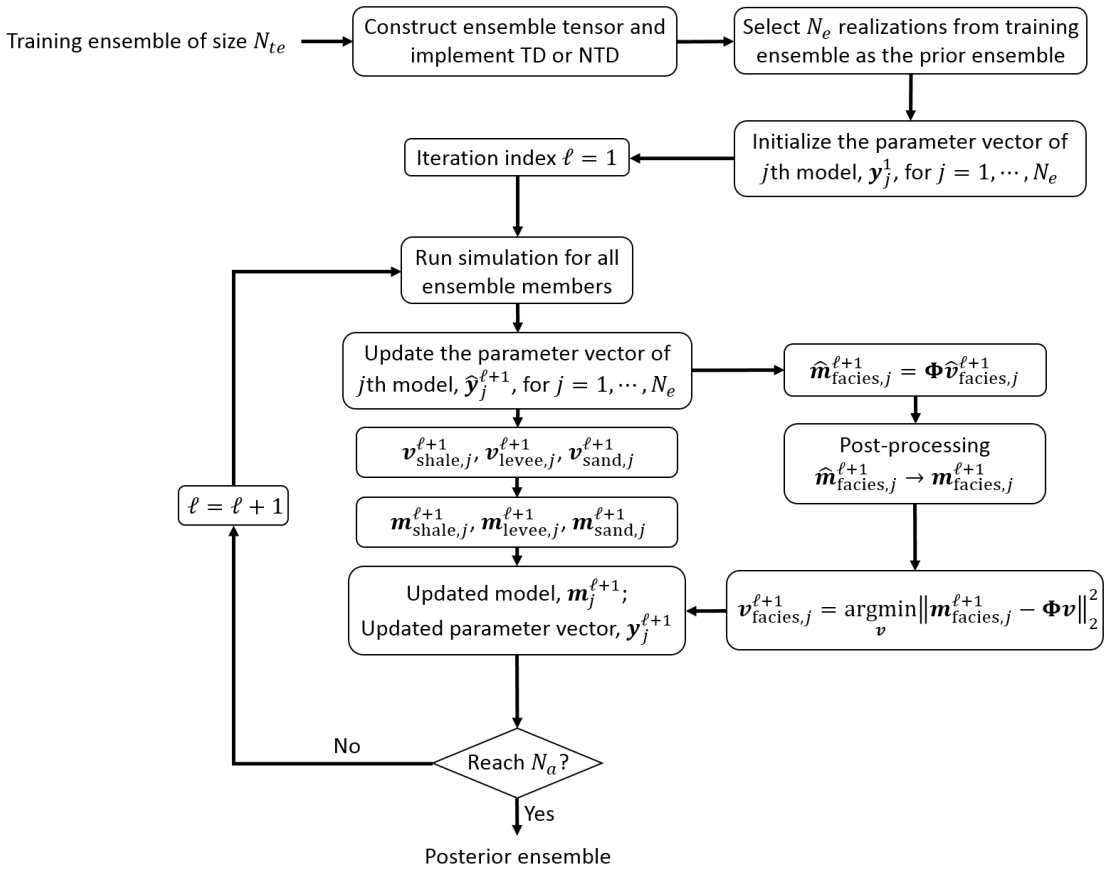


Figure 4.10: History matching workflow

ter 3 that parameterization with the non-negative matrix factorization (NMF) outperforms parameterization with the discrete cosine transform (DCT) and with the principal component analysis (PCA), we compare results obtained with the TD and NTD only with those obtained with the NMF here. The first example considers a 2D three-facies channelized reservoir and primarily aims to compare different implementation schemes of TD and NTD with NMF. In the second example, we consider a 3D reservoir example which has ten layers and two vertically distinct geological zones with complex distributed fluvial channels. Three different cases are designed for the purposes of comparing the performance of TD, NTD and NMF, and more importantly, testing the truncation scheme for TD and NTD proposed in Section 4.1.5.

4.2.1 Example 1: 2D Three-Facies Channelized Reservoir

In the first example, we consider a three-facies (shale, levee and sand), two-dimensional

fluvial reservoir model which is defined on a 100×100 grid system. The size of each grid is 100ft $\times$ 100ft with a constant thickness of 50 ft. Given the observed facies type at wells as hard data, the MPS algorithm (Strebelle, 2002) is used to generate the ensemble of the facies field with the pre-prepared training image (see Fig. 4.11). As is commonly done, we assume that the natural logarithm of the permeability (i.e., $\ln(k)$) follows Gaussian distribution with the mean and standard deviation specified separately for each facies. In this work, the prior $\ln(k)$ fields are obtained by the sequential Gaussian simulation technique where an isotropic spherical variogram is used with correlation range equal to ten times the width of a grid. The detailed properties of facies are given in Table 4.4.

As stated in Section 4.1.6, since the permeability within facies is heterogeneous, the vector of model parameters in this example is written as

$$\boldsymbol{y} = [\boldsymbol{v}_{\text{facies}}^{\text{T}}, \boldsymbol{v}_{\text{sand}}^{\text{T}}, \boldsymbol{v}_{\text{levee}}^{\text{T}}, \boldsymbol{v}_{\text{shale}}^{\text{T}}]^{\text{T}}, \tag{4.38}$$

where $\boldsymbol{v}_{\text{facies}}$ denotes the coefficient vector of facies field obtained by parameterization; $\boldsymbol{v}_{\text{sand}}$, $\boldsymbol{v}_{\text{levee}}$ and $\boldsymbol{v}_{\text{shale}}$ are the parameterized coefficient vectors corresponding to the $\ln(k)$ field of each facies, respectively.

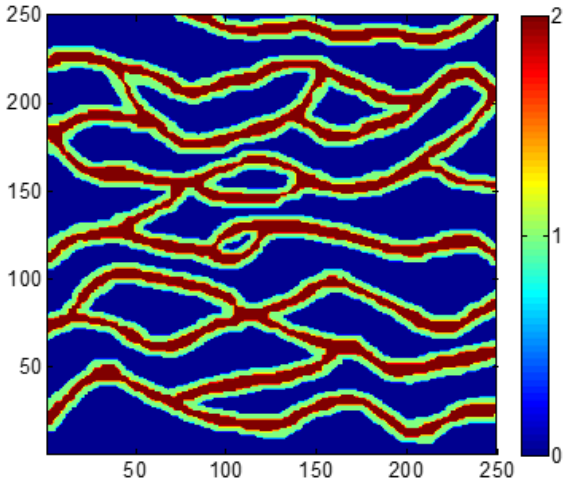


Figure 4.11: Training image (Example 1)

The primary objective of the first example is to compare and investigate the performance of different implementation schemes (i.e., two-mode, three-mode, etc.) of TD and

Table 4.4: Properties of facies

Facies	Indicator	Color	$\overline{k}$, mD	$\overline{\ln(k)}$	$\sigma(\ln(k))$
Sand	2	Red	2000	7.601	0.3
Levee	1	Green	200	5.298	0.3
Shale	0	Blue	20	2.996	0.3

NTD parameterization methods. Thus, for the 2D reservoir model where $N_x = N_y = 100$ and $N_g = 10,000$, we generated four training ensembles for facies, k_{sand}, k_{levee} and k_{shale}, respectively, and each ensemble consists of 1000 models ($N_{te} = 1000$). For the TD parameterization method, four implementation schemes and corresponding computing time are summarized in Table 4.5 and we also include the NMF method for the sake of comparison.

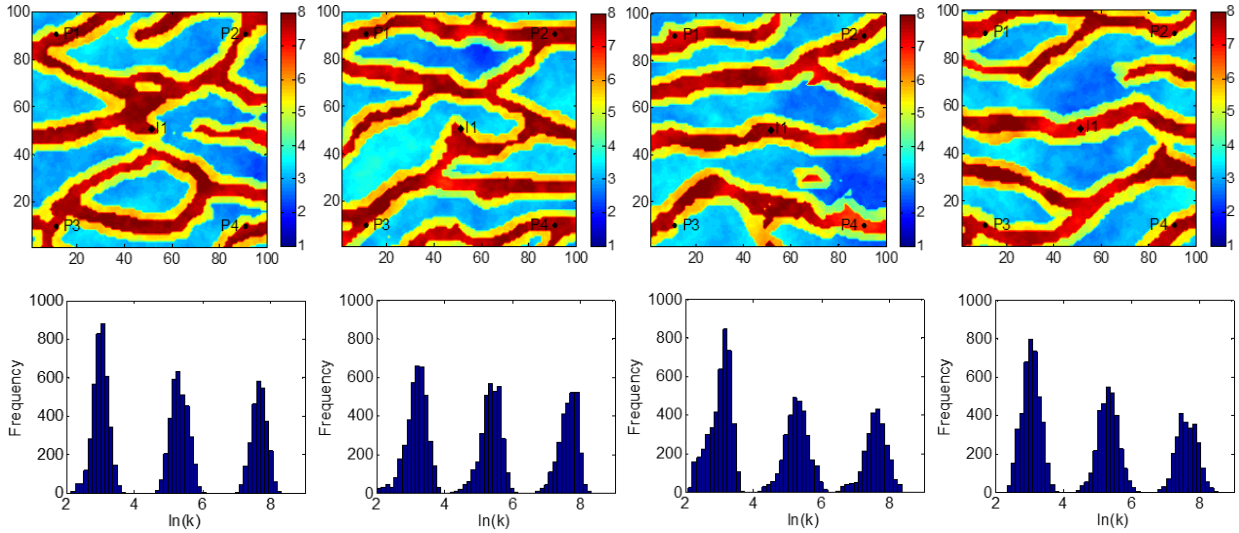
Table 4.5: Parameterization schemes (Example 1, $N_{te} = 1000$)

Method	Ensemble tensor	Scheme	Retained number	Name	t^{para}
TD	$\mathcal{A} \in \mathbb{R}^{N_g \times N_{te}}$	2-mode	$R_1 = 400, R_2 = N_{te}$	TD, 400	8.59s
	$\mathcal{A} \in \mathbb{R}^{N_x \times N_y \times N_{te}}$	3-mode	$R_1 = 20, R_2 = 20, R_3 = N_{te}$	TD, 20×20	6.71s
	$\mathcal{A} \in \mathbb{R}^{N_x \times N_y \times N_{te}}$	3-mode	$R_1 = 40, R_2 = 10, R_3 = N_{te}$	TD, 40×10	7.81s
	$\mathcal{A} \in \mathbb{R}^{N_x \times N_y \times N_{te}}$	3-mode	$R_1 = 10, R_2 = 40, R_3 = N_{te}$	TD, 10×40	6.95s
NMF	$\mathcal{A} \in \mathbb{R}^{N_g \times N_{te}}$	/	$N_c = 400$	NMF, 400	43.89s

In this example the truncation scheme is not employed and we only focus on extracting features along the spatial directions so the dimension of ensemble is not compressed when implementing TD. Therefore, for each realization in the training ensemble, the total number of coefficients corresponding to the retained basis functions obtained by the above parameterization methods is the same as $N_c = 400$ which means we obtained a model reduction from 10,000 parameters to 400 parameters.

In history matching, we randomly selected 200 realizations from the training data set as the prior ensemble, i.e., $N_e = 200$. Figure 4.12 shows the $\ln(k)$ field as well as the corresponding histogram of the true model and three prior realizations. Through the histograms, we can observe that the $\ln(k)$ values present a tri-modal distribution as expected.

There are four producers (P1 to P4) and one injector (I1) deployed in a five-spot pattern in the reservoir. The facies type at all five wells is observed to be sand, and these observations are used as hard data in the initial geological modeling and subsequent history matching process. In the true model, producers P1 and P2 are directly connected to the water injector I1 by two sand channels. On the other hand, P3 and I1 are connected by a loop structure consisting of two branch channels so the injected water has two different paths to reach P3, whereas P4 lies in the corner of the reservoir and is not connected to the injector directly.



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

Figure 4.12: True model and prior realizations (Example 1, $N_e = 200$): $\ln(k)$ field (top row) and histogram (bottom row).

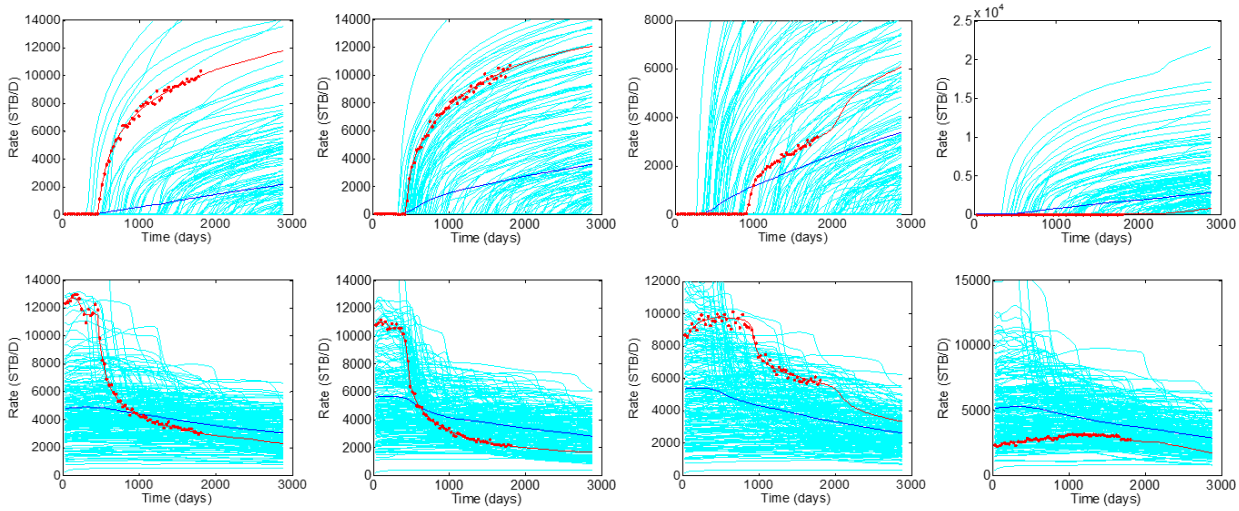
The initial reservoir pressure is 5000 psi and the water saturation is 0.2 before production. The water injection well is operated at a constant bottom hole pressure (BHP) of 5500 psi and the production wells are operated at a BHP of 2000 psi. The history matching period is 5 years (1800 days), followed by a 3-year (1080 days) forecast period. The subsurface flow responses are obtained by running the Eclipse reservoir simulator. The historical data consist of monthly measurements of water injection rate (q_{inj}) at the injector, and oil and water flow rate (q_{o} and q_{w}) at the producers. With available hard data information and

facies proportions of the prior ensemble, we can construct the observed data vector as

$$\boldsymbol{d}_{\text{obs}} = \left[\boldsymbol{q}_{\text{inj}}^{\text{T}}, \boldsymbol{q}_{\text{o}}^{\text{T}}, \boldsymbol{q}_{\text{w}}^{\text{T}}, \boldsymbol{d}_{\text{hard}}^{\text{T}}, \boldsymbol{d}_{\text{fp}}^{\text{T}}\right]_{N_d \times 1}^{\text{T}}, \tag{4.39}$$

where $\boldsymbol{d}_{\text{hard}}$ denotes the vector of hard data which is honored to constrain the facies type at wells and $\boldsymbol{d}_{\text{fp}}$ contains average facies proportions calculated from the prior ensemble. Typically, the observed data are generated by perturbing the synthetic true data with Gaussian random noise. In this example, the standard deviation of measurement error is set equal to 3% of the noise-free true data for dynamic production rates, 0.01 for hard data at wells and 5% of the prior mean for facies proportions.

That there is significant variability in the initial ensemble can be illustrated by the prior production data shown in Fig. 4.13. In this and similar figures, the red dots are the observed data collected during the historical period and the red curve is the noise-free true data of the entire production period. Each light blue curve represents the predicted data obtained by running the reservoir simulation with one specific realization while the dark blue curve depicts the average results of the whole ensemble.



(a) P1 (b) P2 (c) P3 (c) P4

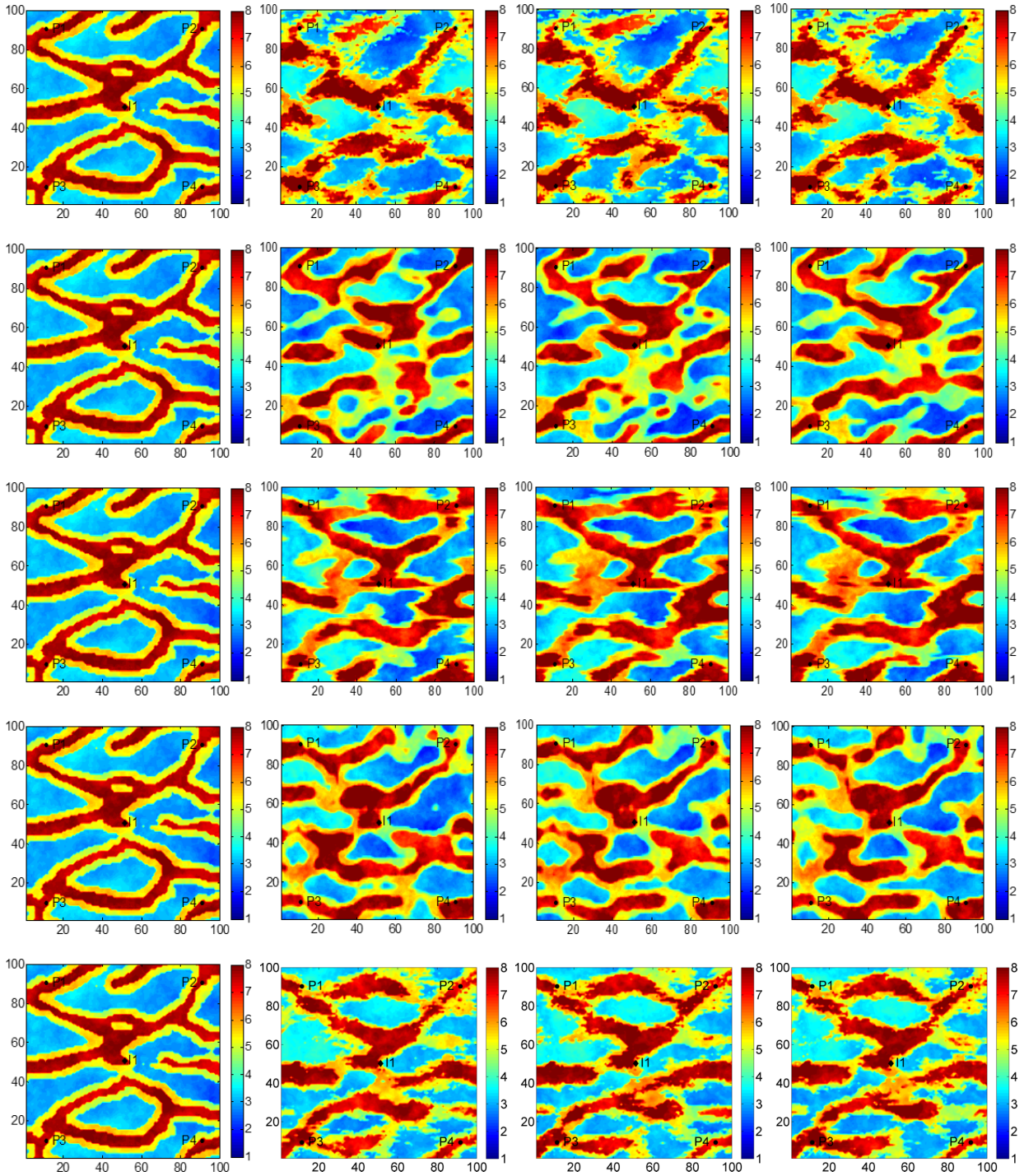
Figure 4.13: Prior production rates (Example 1, $N_e = 200$): water rate (top row) and oil rate (bottom row). True (red curves), observed data (red dots), simulated data of realizations (light blue curves), and ensemble mean (dark blue curves). History: $0 < t \leq 1800$ days, forecast: $t > 1800$ days.

The number of data assimilation steps of the ES-MDA-GEO algorithm, N_a, is set to 8 with the inflation factors calculated by the geometric method proposed by Rafiee and Reynolds (2017) as $\alpha_\ell = \{1000.0,\ 401.4,\ 160.8,\ 64.5,\ 25.9,\ 10.4,\ 4.2,\ 1.7\}$. In Fig. 4.14, we compare the $\ln(k)$ fields of the true model and posterior realizations obtained with the four implementation schemes of TD and the NMF method. It is shown that some large-scale geological structures, such as the channels connecting P1, P2 and I1, are approximately reproduced in all posterior realizations. With the "TD, 400" and "NMF, 400", the most challenging channel loop connecting P3 and I1 is almost resolved although the structural continuity is marred in some areas. However, it is more difficult to recognize the clear shape of the channel loop in the posterior realizations obtained with the "TD, 20×20" implementation scheme. Since the spatial orientation of the fluvial channels in the reservoir is generally in the x direction, more retained features in that direction using the "TD, 40×10" implementation scheme therefore yields better preserved large-scale structures along the x direction while there occurs undesirable small channel loops along the y direction in the realizations updated with the "TD, 10×40" implementation scheme.

From the posterior production data shown in Figs. 4.15 and 4.16, it can be seen that for wells P1 and P2, the history data matches and production forecast are best using the "TD, 400", "TD, 10×40" and "NMF, 400" methods. Because of the channel loop structure, the water production rate of P3 obtained with the true model presents a rapid increase around 2000 days. Since the corresponding period is not part of the well production history, the predictions for P3 are not as good as those for P1 and P2. For well P4, the history data are reasonably matched with different parameterization methods but it is difficult to obtain accurate production forecast since the well did not experience water breakthrough during the historical period.

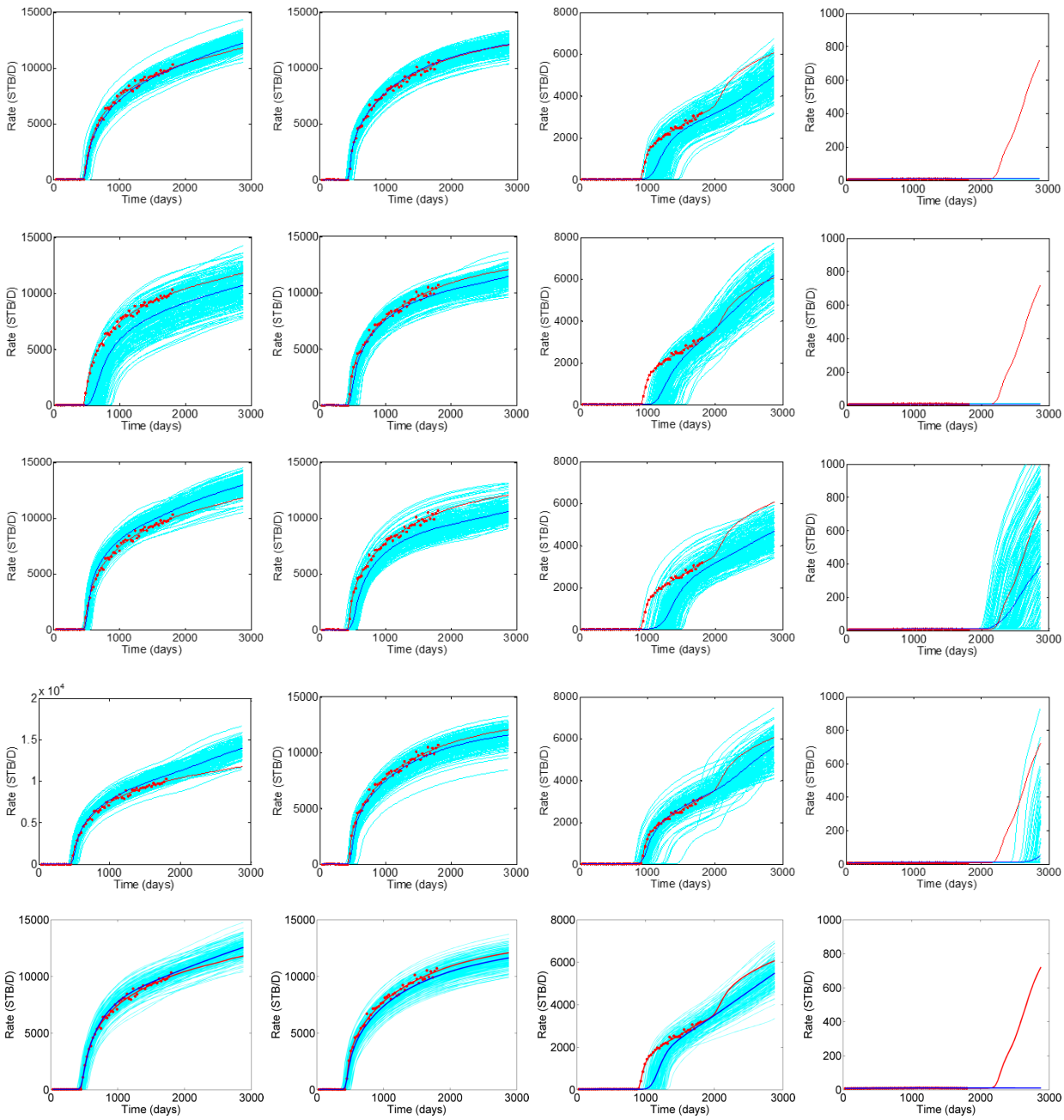
Figure 4.17 presents the box plots of the normalized data mismatch defined by

$$O(\boldsymbol{d}) = \frac{1}{N_d} \left(\boldsymbol{d} - \boldsymbol{d}_{\text{obs}}\right)^{\text{T}} \boldsymbol{C}_{\text{D}}^{-1} \left(\boldsymbol{d} - \boldsymbol{d}_{\text{obs}}\right), \tag{4.40}$$



(a) True (b) Posterior model #1 (c) Posterior model #2 (d) Posterior model #3

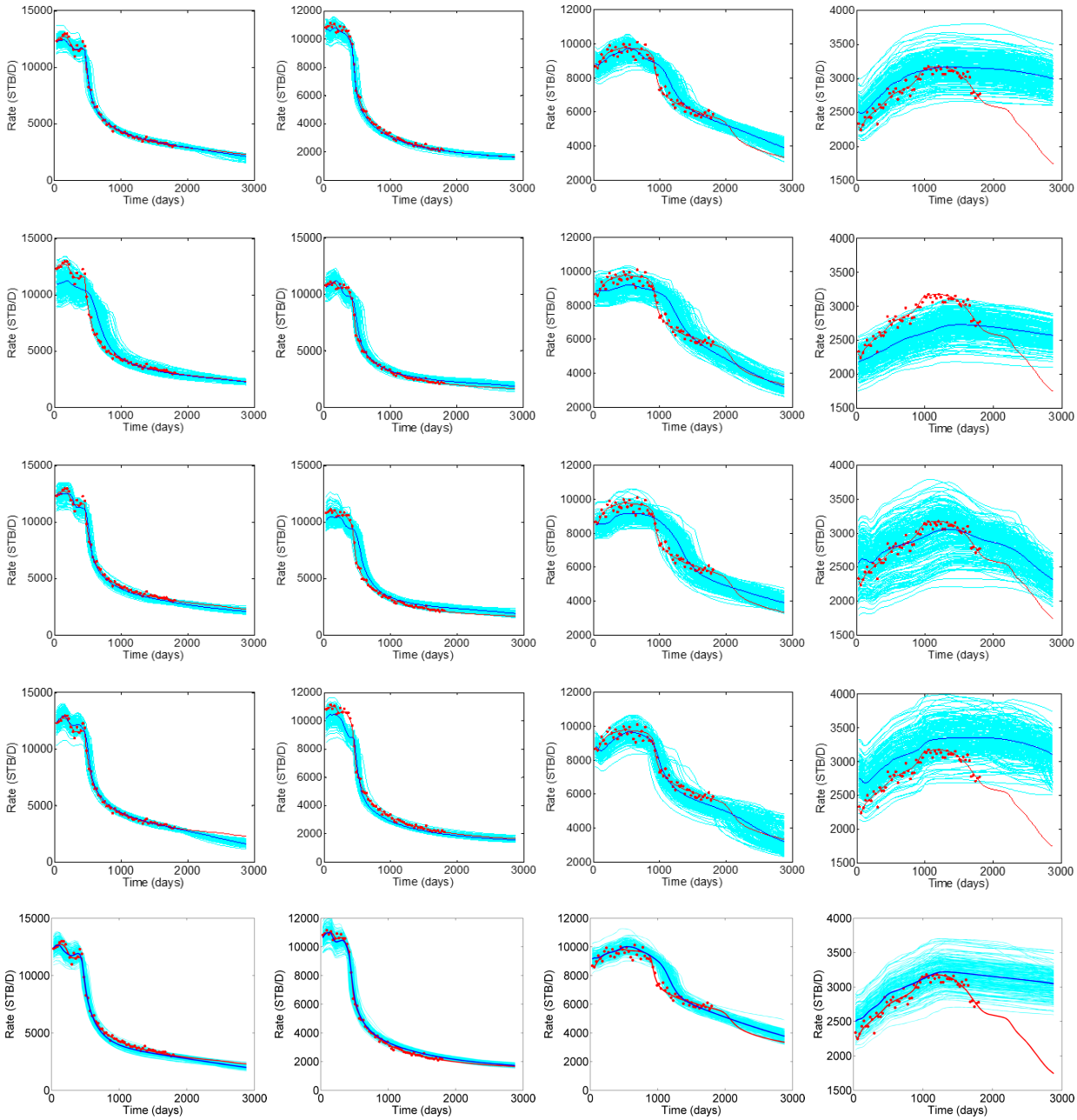
Figure 4.14: $\ln(k)$ field of the true model and posterior realizations (Example 1, $N_e = 200$): TD, 400 (1st row); TD, 20×20 (2nd row); TD, 40×10 (3rd row); TD, 10×40 (4th row); NMF, 400 (5th row).



(a) P1 (b) P2 (c) P3 (d) P4

Figure 4.15: Posterior water rates (Example 1, $N_e = 200$): TD, 400 (1st row); TD, 20×20 (2nd row); TD, 40×10 (3rd row); TD, 10×40 (4th row); NMF, 400 (5th row). True (red curves), observed data (red dots), simulated data of realizations (light blue curves), and ensemble mean (dark blue curves). History: $0 < t \leq 1800$ days, forecast: $t > 1800$ days.

where $\boldsymbol{d}$ and $\boldsymbol{d}_{\text{obs}}$ only include production data during the historical period. Here, $\boldsymbol{d}$ is the predicted data vector which is different for each posterior realization. We can see that the



(a) P1 (b) P2 (c) P3 (d) P4

Figure 4.16: Posterior oil rates (Example 1, $N_e = 200$): TD, 400 (1st row); TD, 20×20 (2nd row); TD, 40×10 (3rd row); TD, 10×40 (4th row); NMF, 400 (5th row). True (red curves), observed data (red dots), simulated data of realizations (light blue curves), and ensemble mean (dark blue curves). History: $0 < t \leq 1800$ days, forecast: $t > 1800$ days.

data matches obtained with the "TD, 400" and "TD, 10×40" are better than the results obtained with the other implementation schemes of TD. However, we do not achieve really

good data matches since the average normalized objective function is not close to 1 for any case. This reflects the difficulty of history matching data from reservoirs with multiple facies and complex oriented geological features. Table 4.6 shows the total computing time, t^{total}, used to complete each case and N_w denotes the number of wells in the reservoir.

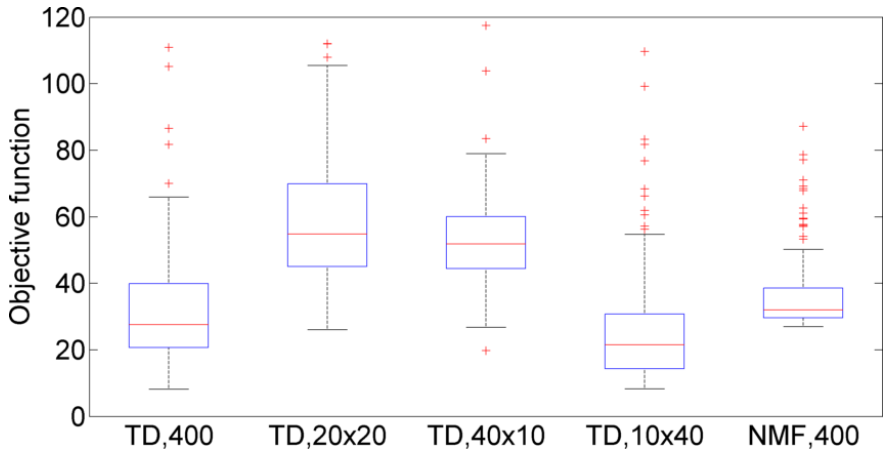


Figure 4.17: Normalized data mismatch (Example 1, $N_e = 200$). Median (central red mark), 25th and 75th percentiles (bottom and top edges of the box), extreme data points (whiskers), and outliers ("+" symbol).

Table 4.6: Computational cost (Example 1)

Method	N_e	N_a	N_w	t^{total}
TD, 400	200	8	5	2h 15m 20s
TD, 20×20	200	8	5	2h 21m 16s
TD, 40×10	200	8	5	2h 27m 56s
TD, 10×40	200	8	5	2h 35m 16s
NMF, 400	200	8	5	2h 22m 37s

4.2.2 Example 2: 3D Three-Facies Channelized Reservoir

In Example 2, we consider a 3D three-facies channelized reservoir which consists of ten layers in the vertical direction and each layer has a 50×50 grid system. The size of each grid is 100ft $\times$ 100ft $\times$ 50ft. There are 4 water injection wells and 9 production wells deployed in a five-spot pattern in the reservoir where the ten layers are fully penetrated by these vertical wells and are fully perforated.

The facies distributions of the synthetic true model and prior realizations are generated using the object-based modeling tool available in Petrel given the hard data information (see Table 4.7) and parameter settings (see Table 4.8). Note that the symbol "/" in Table 4.8 indicates the dimensionless parameters and the width and thickness of the levee are set as a fraction of the channel width and thickness, respectively.

Table 4.7: Hard data information at wells (Example 2)

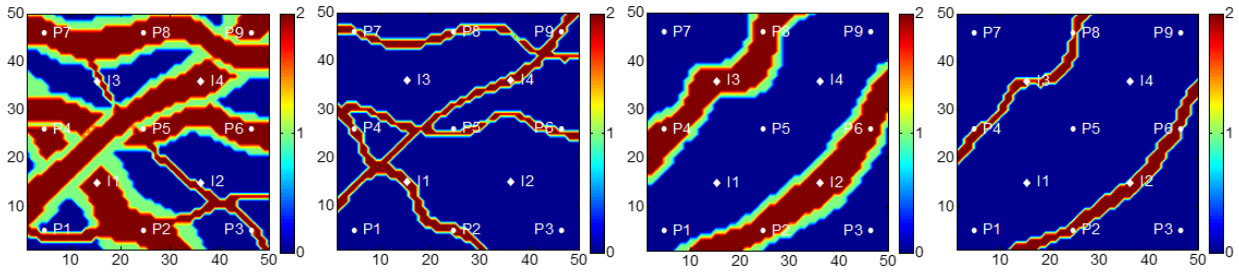
Zone	Facies	Wells
1	Sand	I1, I4, P2, P4, P5, P6, P7, P8, P9
2	Sand	I2, I3, P2, P4, P6, P8

Table 4.8: Parameter settings for object-based modeling (Example 2)

Parameter	Notation	Unit	Distribution	Mean (μ)	STD (σ)
orientation	θ	degrees	Gaussian	90	45
amplitude	a	ft	Gaussian	500	100
wavelength	λ	ft	Gaussian	3000	500
sinuosity	s	/	Gaussian	0.2	0.05
width of sand channel	w_{sand}	ft	Gaussian	600	50
thickness of sand channel	h_{sand}	ft	Gaussian	50	5
width of levee	w_{levee}	/	Gaussian	0.35	0.05
thickness of levee	h_{levee}	/	Gaussian	0.7	0.05

From the facies fields of the true model with well locations shown in Fig. 4.18, we can see that the ten layers are grouped into two distinct geological zones. To be specific, layer 1 and 6 are the top and bottom layer of the top zone, and layer 7 and 10 are the top and bottom layer of the bottom zone. The geological objects in the reservoir are described in Table 4.9. Since different sets of hard data are honored in the process of object-based modeling, the spatial patterns of channelized facies vary significantly between two zones while the geological structures are quite similar throughout different layers within the same zone except the channels become thinner as depth increases. Therefore, maintaining the

distinct geological features of two zones and resolving the detailed structures at the same time pose additional challenges to the history matching work.



(a) Layer 1 (b) Layer 6 (c) Layer 7 (d) Layer 10

Figure 4.18: Facies fields of the true model (Example 2)

Table 4.9: Geological objects in three-facies channelized reservoir (Example 2)

Object	Facies	Reservoir quality	Perm & Poro
Channel	Sand	Good	High
Levee	Levee	Moderate	Moderate
Floodplain	Shale	Poor	Low

In this example, we designed and tested three different cases (see the descriptions in Table 4.10) in order to further assess the proposed history matching workflow with more complex 3D reservoir models. Note that there are only 4 injectors and 5 producers out of the original 13 wells in Case 2.3 and we also change the well name of producers correspondingly. The water injectors and producers operate under constant BHP of 4000 psi and 2500 psi, respectively. The initial reservoir pressure and water saturation, respectively, are set to 5000 psi and 0.2 for every grid in the model. The porosity is assumed to be known and set to 0.2 for all facies. The period of collecting historical data is 300 days which is followed by an additional 300-day forecast period in order to evaluate the predictive ability of the updated models. The observed data include measurements at 30-day intervals of the water injection rate (q_{inj}) at the injectors, and oil and water production rate (q_{o} and q_{w}) at the producers. Since the hard data information and prior facies proportions are both available in this example, the observed data vector is given by Eq. 4.39 and the measurement noise

levels of various types of data are also identical to those specified in Example 1. The ES-MDA-GEO algorithm is used to assimilate the observed data and the iteration number, N_a, is set to 8 with the inflation factors obtained through the geometric scheme as $\alpha_\ell = \{1000.0,\ 401.4,\ 160.8,\ 64.5,\ 25.9,\ 10.4,\ 4.2,\ 1.7\}$. It is noteworthy that in Case 2.2 and Case 2.3, we used the truncation scheme (see Section 4.1.5) to obtain ranks of different modes when implementing the truncated TD and NTD, while the ranks are still determined heuristically in Case 2.1 for the purpose of fair comparison between TD, NTD and NMF.

Table 4.10: Descriptions of four cases (Example 2)

Case	Permeability	Wells	N_e	Method	Scheme
2.1	Homogeneous	4 Inj., 9 Prod.	200	TD, NTD, NMF	2-mode
2.2	Heterogeneous	4 Inj., 9 Prod.	200, 500	TD, NTD	2-mode, 4-mode
2.3	Heterogeneous	4 Inj., 5 Prod.	500	TD, NTD	2-mode, 4-mode

Case 2.1: Homogeneous Permeability, 13 Wells:

In the first case, we consider calibrating both the facies distributions and the permeability values simultaneously. Although the permeability within facies is homogeneous, the true values are unknown and we assume the $\ln(k)$ value of each facies throughout the whole ensemble follow Gaussian distributions with the mean values and standard deviations given in Table 4.4. Thus, the vector of model parameters is given by

$$\boldsymbol{y} = [\boldsymbol{v}_{\text{facies}}^{\text{T}}, k_{\text{sand}}, k_{\text{levee}}, k_{\text{shale}}]^{\text{T}}, \tag{4.41}$$

where k_{sand}, k_{levee} and k_{shale} denote the natural logarithm of the permeability within sand, levee and shale facies, respectively. In order to demonstrate the properties of TD and NTD methods, the vector of coefficients, $\boldsymbol{v}_{\text{facies}}$, is obtained by applying three different approaches (i.e., TD, NTD and NMF) to the same training ensemble ($N_{te} = 1000$) with an identical retained number of basis functions, $N_c = 1000$, for each realization. Note that we used the two-mode implementation scheme of TD and NTD methods in this case by vectorizing the

3D reservoir model first. Table 4.11 provides the details of three parameterization schemes and it is shown that the computing time of NTD is approximately 10 times than that of TD.

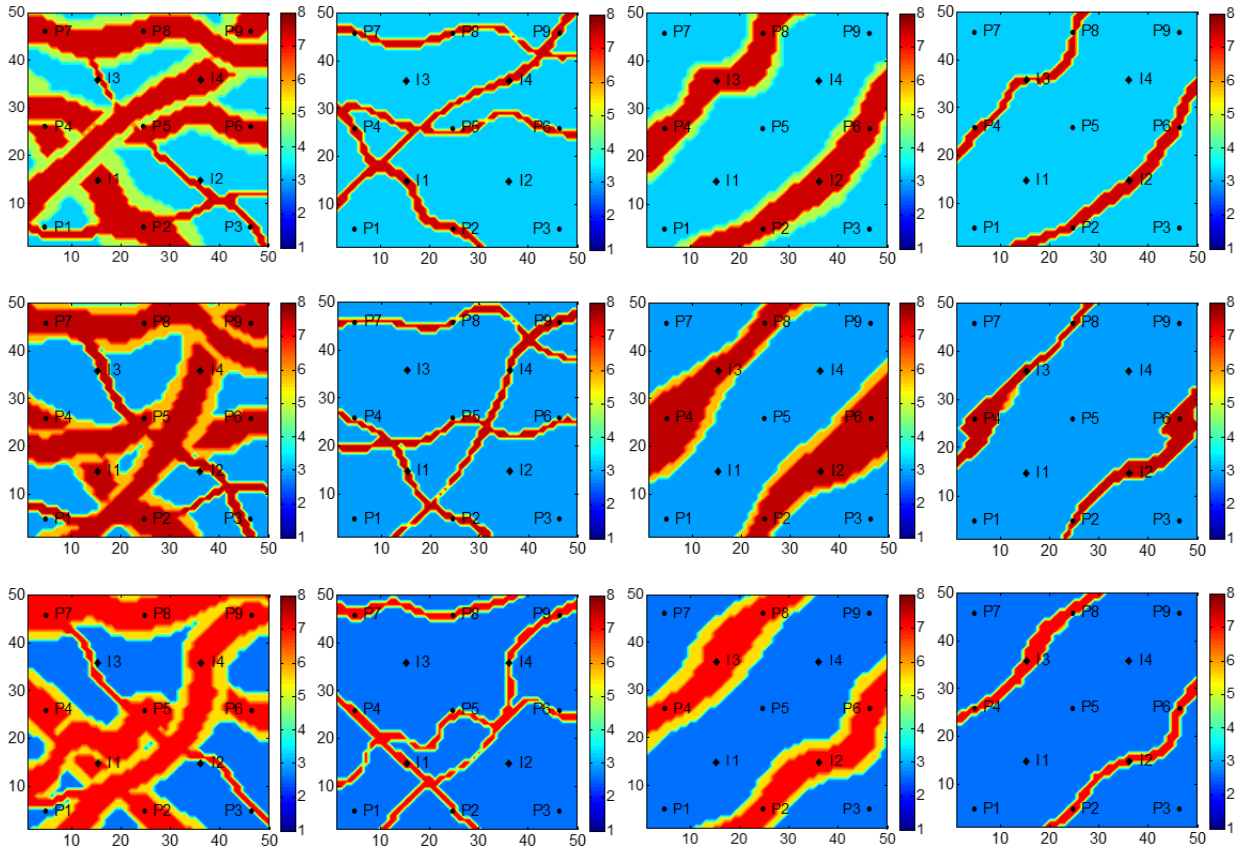
Table 4.11: Parameterization schemes (Case 2.1, $N_{te} = 1000$)

Method	Ensemble tensor	Scheme	Retained number	Name	t^{para}
TD	$\mathcal{A} \in \mathbb{R}^{N_g \times N_{te}}$	2-mode	$R_1 = 1000, R_2 = N_{te}$	TD, 1000	32.72s
NTD	$\mathcal{A} \in \mathbb{R}^{N_g \times N_{te}}$	2-mode	$R_1 = 1000, R_2 = N_{te}$	NTD, 1000	373.36s
NMF	$\mathcal{A} \in \mathbb{R}^{N_g \times N_{te}}$	/	$N_c = 1000$	NMF, 1000	181.57s

The size of history matching ensemble is 200 ($N_e = 200$) and the prior geological uncertainties not only come from the facies distributions but also from the permeability variation within individual facies. The true model and two prior realizations are shown in Fig. 4.19.

From the first rows in Figs. 4.20 and 4.21, it can seen that the prior water and oil production rates display large uncertainties. The historical period lasts for 300 days and the time period, $300 < t < 600$ days, is used to test the uncertainty quantification of the production predictions. It is noteworthy that the two producers, P3 and P7, are in the corners of the reservoir, and we did not observe water breakthrough at these two wells during the 300-day historical period, which makes it more difficult to resolve the structural details around these wells and obtain accurate predictions. Based on the second to fourth rows in Figs. 4.20 and 4.21, it is clearly shown that the prior uncertainties have been significantly reduced through the history matching process with three parameterization methods. Although NTD is the only method where the ensemble of predictions bound the results from the truth at every well, the history matches and predictions obtained from the three parameterization methods are not radically different. Note that the water production rate at well P3 is very low and has a negligible influence on the total data mismatch. Figure 4.22 shows, in the box plot format, a quantitative evaluation of the normalized data mismatch defined by Eq. 4.40. The results indicate that the best data match is obtained with the NTD method.

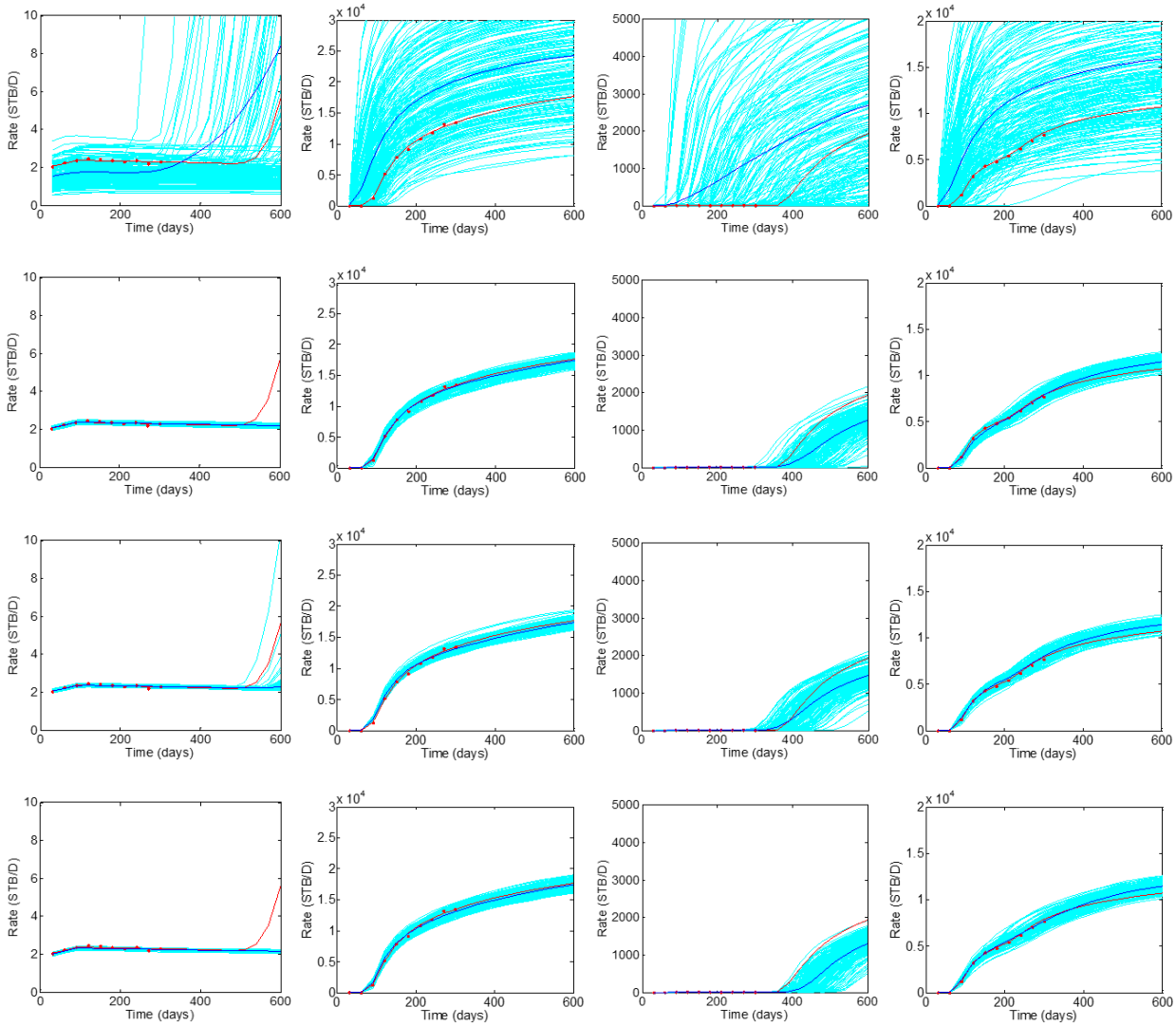
In Figs. 4.23 and 4.24, we can see the $\ln(k)$ fields of the true model and two posterior



(a) Layer 1 (b) Layer 6 (c) Layer 7 (d) Layer 10

Figure 4.19: $\ln(k)$ fields of the true model and two prior realizations (Case 2.1, $N_e = 200$): true model (1st row), prior model #1 (2nd row) and prior model #2 (3rd row).

realizations updated with each of the three methods (i.e., TD, NTD and NMF). Note that the updated realizations correspond to the unconditional realizations in Fig. 4.19. Although the original facies distributions are different in the prior realizations, we see more similar geological structures in the calibrated models and posterior realizations capture the main structural features of the truth. One should bear in mind however that the availability of facies observation at the thirteen wells enhance our ability to produce a good characterization of the true model. Since it is difficult to determine the spatial relationships between the parameterized variables and observed data, we did not employ any distance-based localization techniques in this work to mitigate the spurious sampling errors and ensemble collapse. For this 3D complex reservoir model, reasonable variability can be still maintained with a relatively small ensemble of 200 realizations although we will see later that increasing the

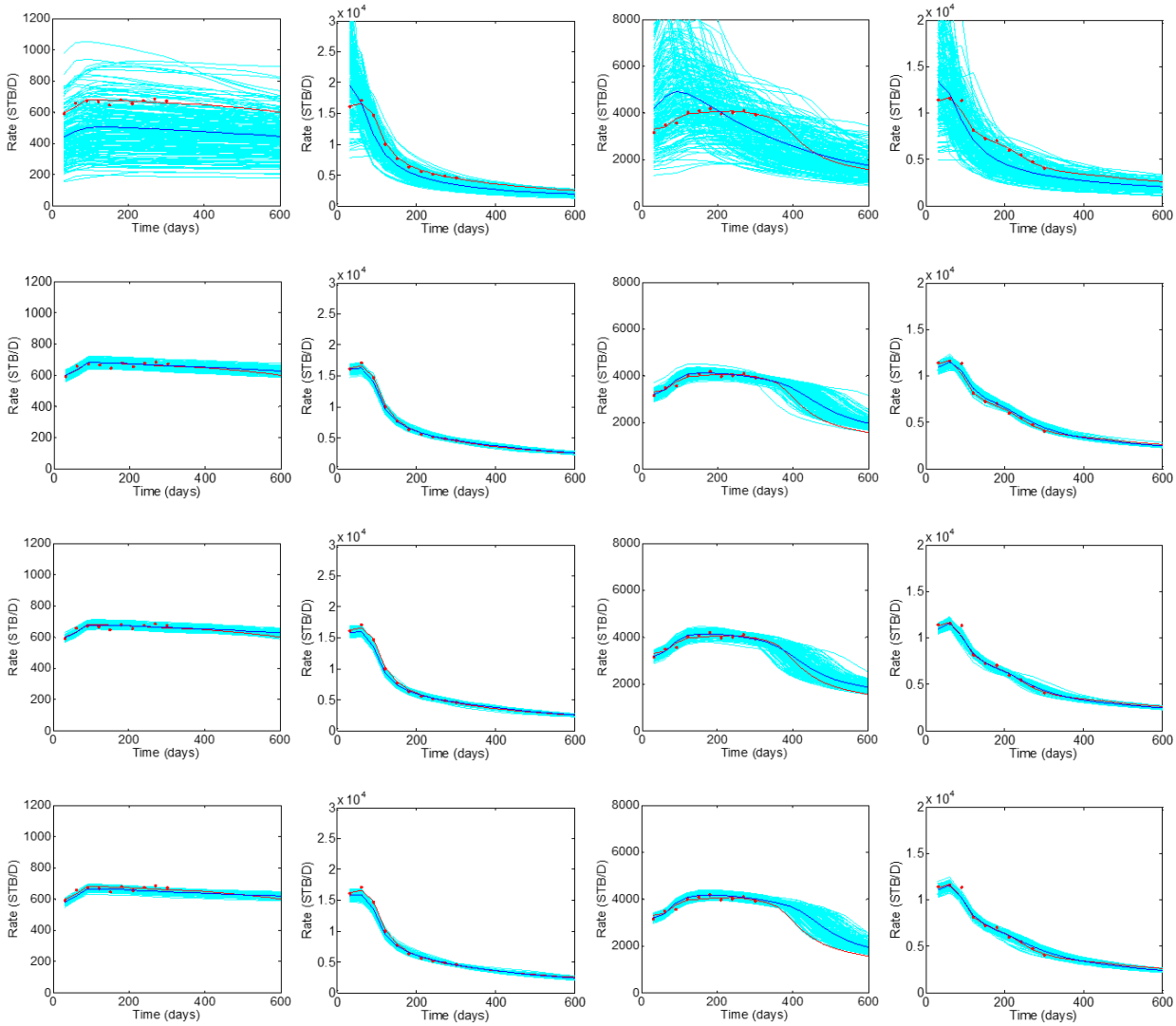


(a) P3 (b) P4 (c) P7 (d) P8

Figure 4.20: Prior and posterior water rates (Case 2.1, $N_e = 200$): prior (1st row); posterior, TD, 1000 (2nd row); posterior, NTD, 1000 (3rd row); posterior, NMF, 1000 (4th row). True (red curves), observed data (red dots), simulated data of realizations (light blue curves), and ensemble mean (dark blue curves). History: $0 < t \leq 300$ days, forecast: $t > 300$ days.

ensemble size increases the variability (uncertainty) in the posterior models.

It is interesting and straightforward to compare the probability maps of facies in the posterior ensembles. In Fig. 4.25, the black pixels in the first row correspond to gridblocks occupied by the channel facies in the true reservoir model and we aim to resolve these points correctly through the history matching process although to do so would suggest that data is sufficient to resolve the true model which is never the case. Each colored pixel (black, red



(a) P3 (b) P4 (c) P7 (d) P8

Figure 4.21: Prior and posterior oil rates (Case 2.1, $N_e = 200$): prior (1st row); posterior, TD, 1000 (2nd row); posterior, NTD, 1000 (3rd row); posterior, NMF, 1000 (4th row). True (red curves), observed data (red dots), simulated data of realizations (light blue curves), and ensemble mean (dark blue curves). History: $0 < t \leq 300$ days, forecast: $t > 300$ days.

or blue) in the figures of the other four rows indicates that every realization of the ensemble has channel facies at that specific grid, i.e., based on the ensemble of posterior realizations, the probability of each of these gridblocks being occupied by the channel facies is 1.0 (100%) so there is no uncertainty in the facies marginal probability distribution at these locations.

Note that there are very few black pixels in the prior probability maps and they only exist around the wells where we have hard data (see Table 4.7). This observation illustrates

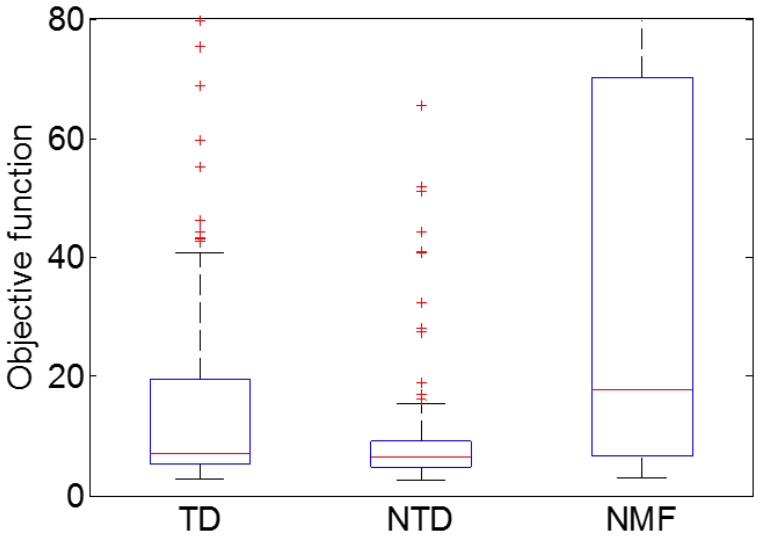
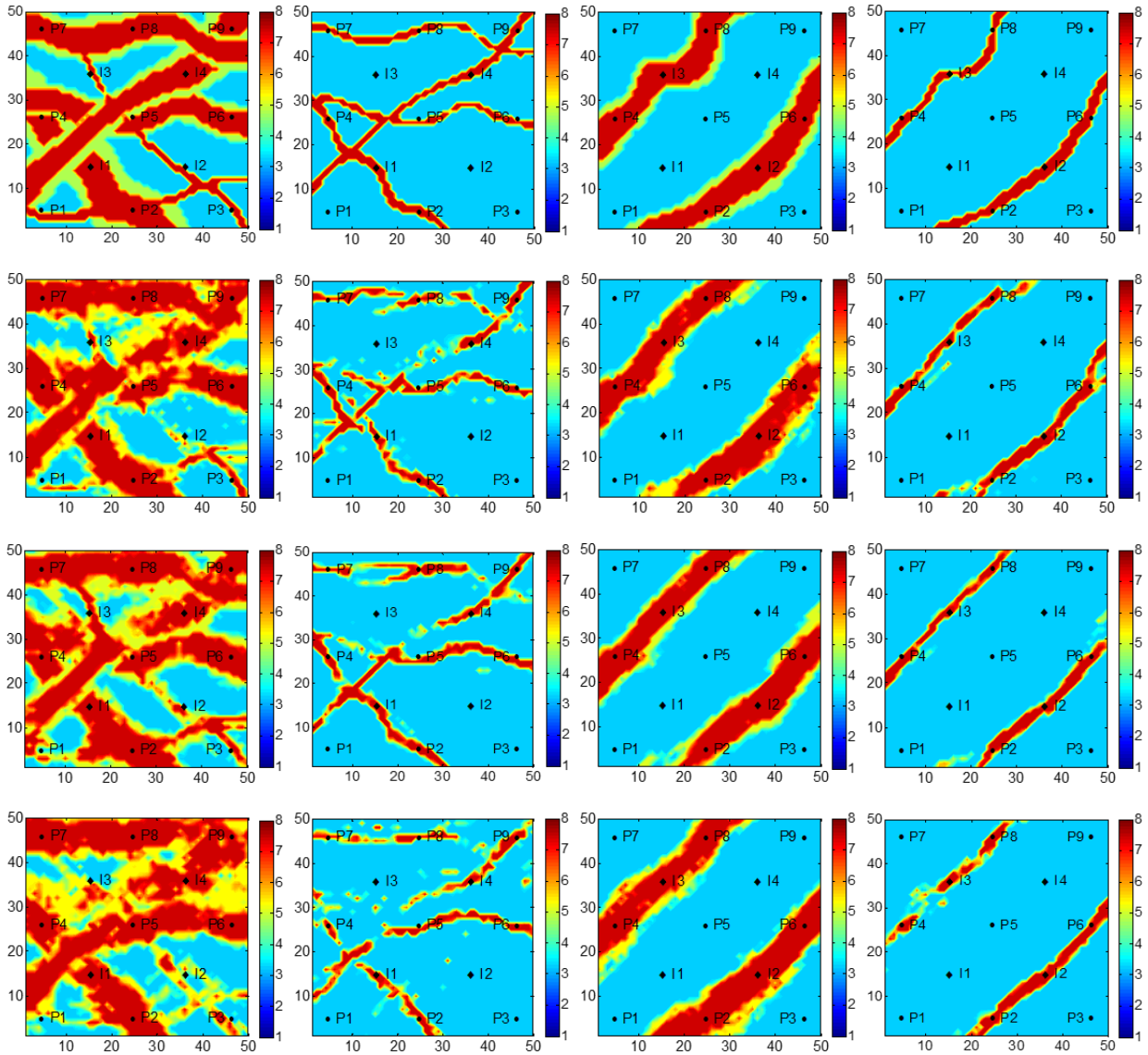


Figure 4.22: Normalized data mismatch (Case 2.1, $N_e = 200$). Median (central red mark), 25th and 75th percentiles (bottom and top edges of the box), extreme data points (whiskers), and outliers ("+" symbol).

the conditioning capability of the object-based modeling technique, and more importantly shows the large uncertainties in the prior ensemble. In the posterior probability maps obtained with the TD, NTD and NMF methods (see the third to fifth rows in Fig. 4.25), each red point (a gridblock) means that the true model has the channel facies in this gridblock and the channel facies is correctly resolved at the corresponding gridblock in all posterior models. On the other hand, each blue point (a gridblock) indicates that all posterior realizations have the channel facies in this gridblock but in the true model this gridblock is not occupied by the channel facies. Compared to the prior probability maps in the second row, we can see that there are a large number of correctly resolved gridblocks with channel facies (red points) in the posterior ensembles obtained with the three methods and the number of points incorrectly resolved to be the channel facies with 100% probability are very rare. The results of Fig. 4.25 clearly indicate that NTD is better at correctly resolving the distribution of the channel facies than the TD and NMF methods.

In order to quantitatively evaluate the quality and spatial continuity of resolved geological structures, we count the number of gridblocks where the channel facies is correctly resolved (red gridblocks in Fig. 4.25) and denote this number by N_{gc}^{good}. We also count the number of gridblocks where the posterior distribution indicates the probability of the chan-

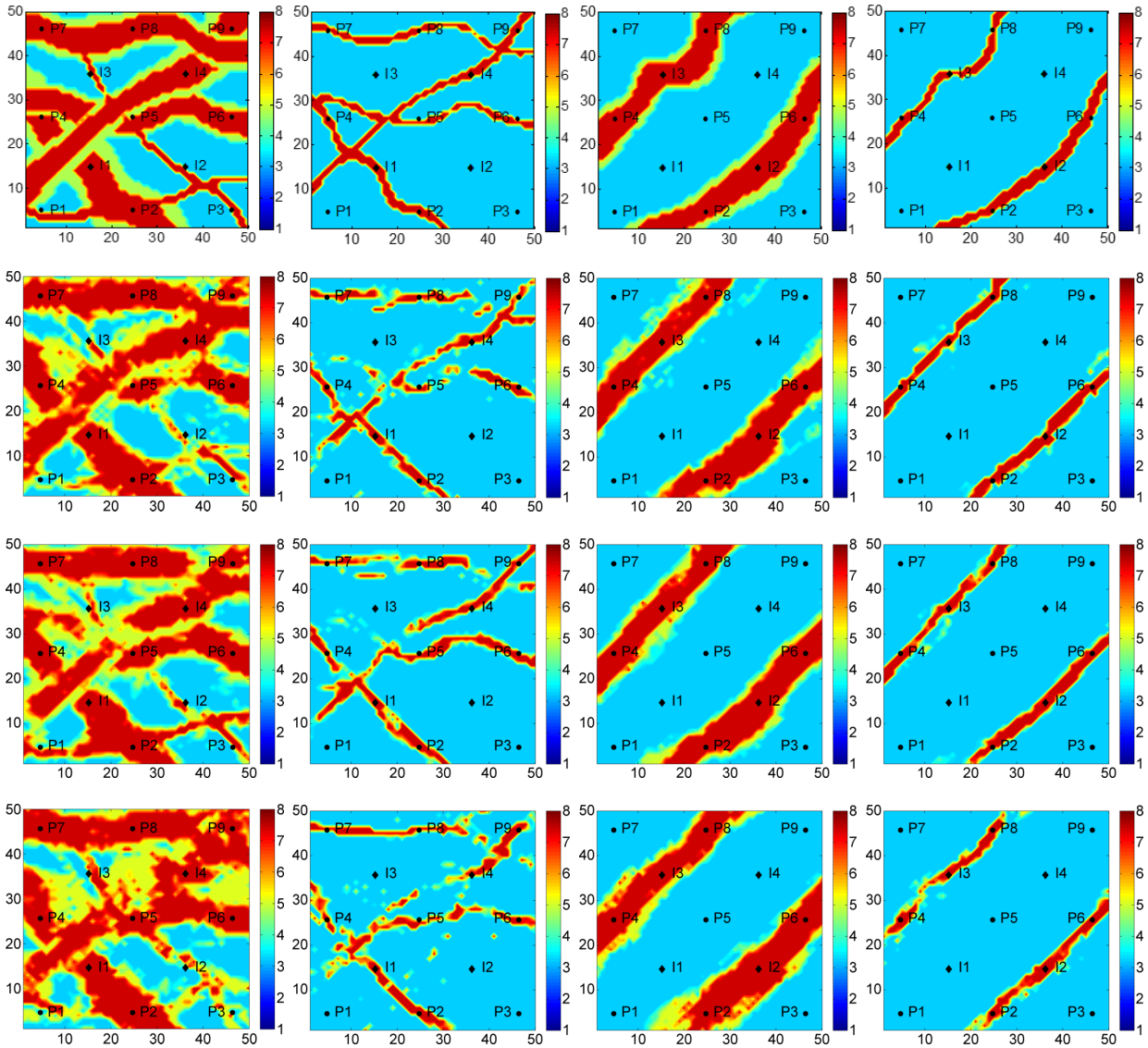


(a) Layer 1 (b) Layer 6 (c) Layer 7 (d) Layer 10

Figure 4.23: $\ln(k)$ fields of the true model and posterior realization #1 (Case 2.1, $N_e = 200$): true model (1st row); TD, 1000 (2nd row); NTD, 1000 (3rd row); NMF, 1000 (4th row).

nel facies is 100% but the corresponding gridblocks in the true model are not occupied by channel (blue gridblocks in Fig. 4.25) and denote this number by N_{gc}^{bad}. We let $N_{gc}^{\text{posterior}}$ denote the number of gridblocks that are occupied by channel facies in all 200 posterior realizations so that

$$N_{gc}^{\text{posterior}} = N_{gc}^{\text{good}} + N_{gc}^{\text{bad}}. \tag{4.42}$$



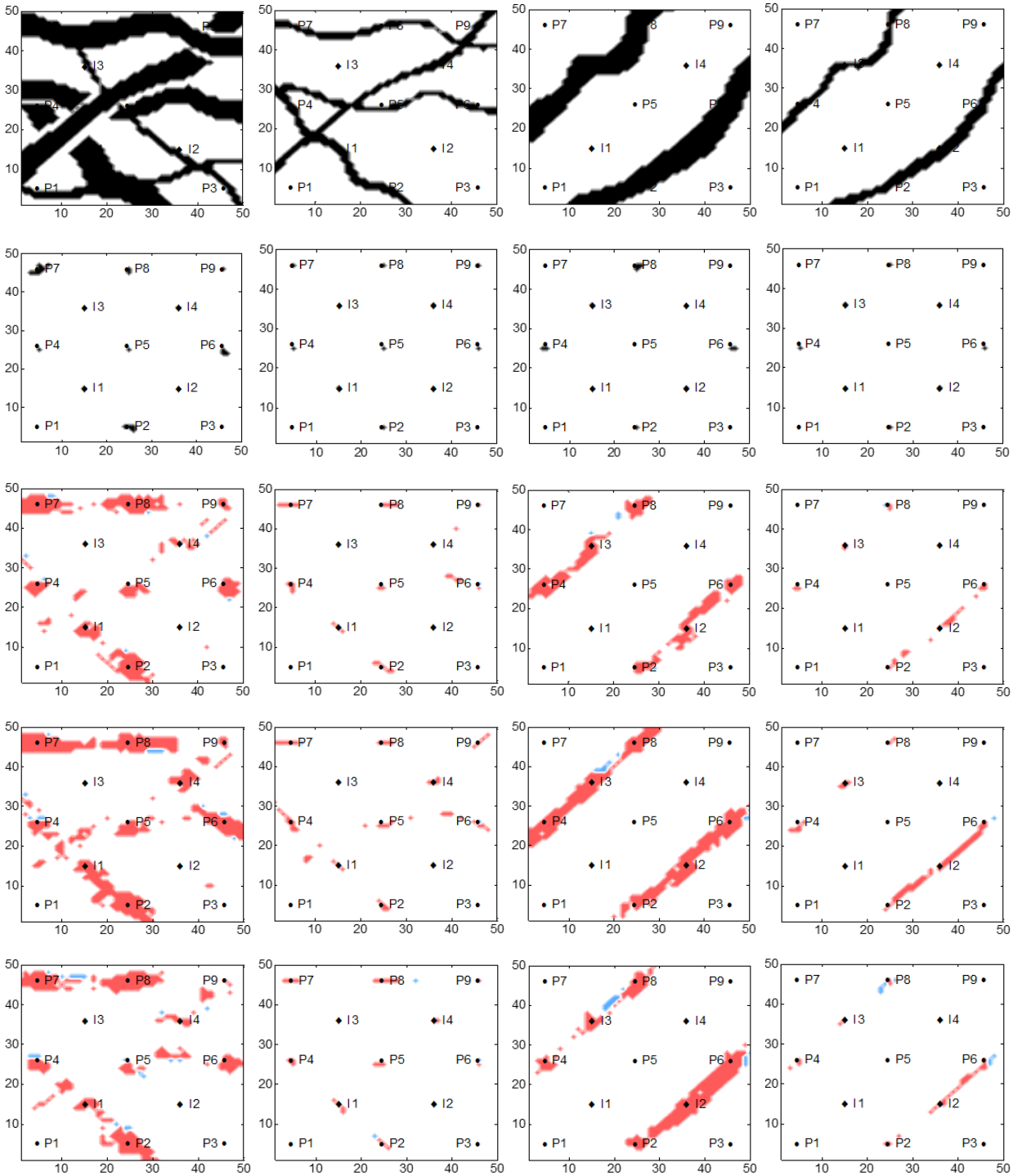
(a) Layer 1 (b) Layer 6 (c) Layer 7 (d) Layer 10

Figure 4.24: $\ln(k)$ fields of the true model and posterior realization #2 (Case 2.1, $N_e = 200$): true model (1st row); TD, 1000 (2nd row); NTD, 1000 (3rd row); NMF, 1000 (4th row).

Let N_{gc}^{true} denote the number of gridblocks in the true model that are occupied by the channel facies. Thus, the true distribution of channel facies is completely resolved if and only if

$$N_{gc}^{\text{true}} = N_{gc}^{\text{posterior}} = N_{gc}^{\text{good}}. \tag{4.43}$$

In addition, $R^{\text{good}} = N_{gc}^{\text{good}}/N_{gc}^{\text{posterior}}$ denotes the fraction of "resolved gridblocks" that are resolved correctly to be channel facies and $R^{\text{bad}} = N_{gc}^{\text{bad}}/N_{gc}^{\text{posterior}}$ is the fraction of channel



(a) Layer 1 (b) Layer 6 (c) Layer 7 (d) Layer 10

Figure 4.25: 100% probability maps of sand channels (Case 2.1, $N_e = 200$): true (1st row); prior (2nd row); posterior, TD, 1000 (3rd row); posterior, NTD, 1000 (4th row); posterior, NMF, 1000 (5th row). In the bottom three rows, all gridblocks that are not white indicate that all posterior realizations ($N_e = 200$) have the channel facies in that gridblock.

gridblocks resolved incorrectly.

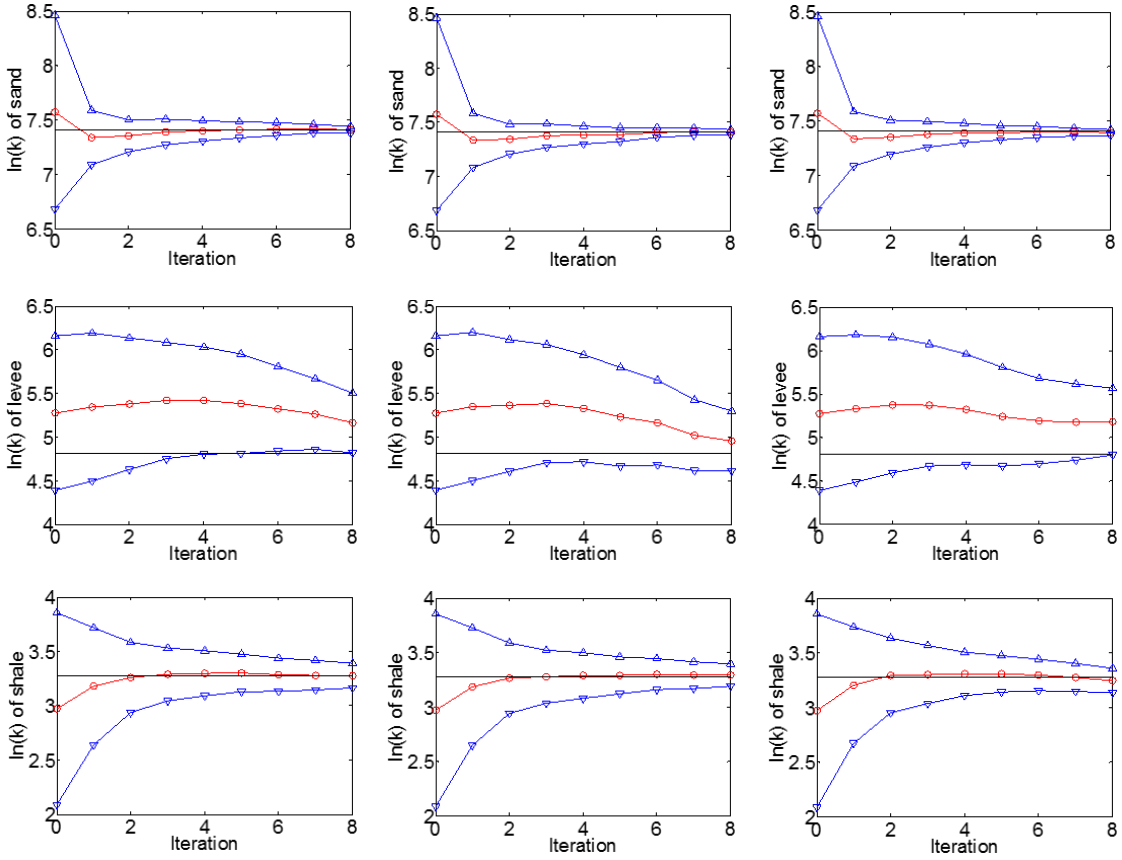
The resulting statistics based on the 100% probability maps in Fig. 4.25 are given in Table 4.12. It is shown that 2109 out of the 6104 gridblocks that are occupied by the channel facies in the true model are identified as such with 100% probability by the posterior probability maps obtained from the posterior realizations generated using the NTD parameterization. Whereas with the TD and NMF methods, there are only 1295 and 1493 such gridblocks, respectively. Moreover, with the NTD, over 96% of the gridblocks that are identified as containing the channel with 100% probability are actually occupied by the channel facies in the true model. This proportion is very close to the 98.2% obtained with the TD and higher than the 92.3% obtained with the NMF. Therefore, the statistics indicate that the NTD resolves the true distribution of the channel facies better than the TD and NMF, and mainly for this reason, we think the NTD is superior to the TD and NMF for parameterization and uncertainty characterization of the facies distribution.

Table 4.12: Statistical indicators based on 100% probability maps (Case 2.1)

Method	N_{gc}^{true}	$N_{gc}^{\text{posterior}}$	N_{gc}^{good}	N_{gc}^{bad}	$R^{\text{good}}(\%)$	$R^{\text{bad}}(\%)$
TD	6104	1295	1271	24	98.2%	1.8%
NTD	6104	2109	2036	73	96.5%	3.5%
NMF	6104	1493	1378	115	92.3%	7.7%

Besides the correctly resolved facies distributions, good data matches in this case also require reasonable approximations of petrophysical properties within facies. We can see the evolution of estimated permeability values during the history matching process with the ES-MDA-GEO step in Fig. 4.26 where the x axis is the iteration step index and "0" corresponds to the prior ensemble. In each plot, the black horizontal straight line represents the true value, the red curves represent the ensemble mean of the calibrated permeability and the two blue curves are the ensemble mean ± 3 ensemble standard deviation respectively, which gives a characterization of the uncertainty in the estimated variables. Since the wells in this reservoir are drilled either in sand channels or on shale plains, the permeability values within

sand and shale facies converge rapidly to the true values with different parameterization methods. However, the permeability within the levee facies is more difficult to estimate according to the results in Fig. 4.26. We can see similar computing time of running this case with three parameterization methods in Table 4.13.



(a) TD, 1000 (b) NTD, 1000 (c) NMF, 1000

Figure 4.26: Evolution of estimated permeability values as a function of the ES-MDA step (Case 2.1, $N_e = 200$): k_{sand} (1st row), k_{levee} (2nd row) and k_{shale} (3rd row). True (black horizontal straight lines), ensemble mean (red curves), and ensemble mean ± 3 ensemble standard deviation (blue curves).

Table 4.13: Computational cost (Case 2.1)

Method	N_e	N_a	N_w	t^{total}
TD, 1000	200	8	13	1h 56m 54s
NTD, 1000	200	8	13	1h 55m 12s
NMF, 1000	200	8	13	1h 50m 12s

Case 2.2: Heterogeneous Permeability, 13 Wells:

In the second case, the heterogeneous permeability within each facies leads to a more realistic reservoir example and the $\ln(k)$ values follow Gaussian distributions with the mean and standard deviation specified for each facies (see Table 4.4). The sequential Gaussian simulation technique is used to generate the $\ln(k)$ fields with an isotropic spherical variogram and the correlation range equals to ten times the width of one grid. Similar to Example 1, we construct the vector of model parameters as

$$\boldsymbol{y} = [\boldsymbol{v}_{\text{facies}}^{\text{T}}, \boldsymbol{v}_{\text{sand}}^{\text{T}}, \boldsymbol{v}_{\text{levee}}^{\text{T}}, \boldsymbol{v}_{\text{shale}}^{\text{T}}]^{\text{T}}, \tag{4.44}$$

where $\boldsymbol{v}_{\text{sand}}$, $\boldsymbol{v}_{\text{levee}}$ and $\boldsymbol{v}_{\text{shale}}$ are the coefficient vectors obtained by applying the parameterization methods to the training ensemble of whole permeability field of each facies. Consequently, we have four training ensembles, one for facies and one for each of the permeability fields, k_{sand}, k_{levee} and k_{shale}, respectively. The size of each training ensemble is still 1000 ($N_{te} = 1000$) in Case 2.2.

In this case, we implement the truncated TD and NTD methods, and compare their performance with those of the TD and NTD without truncation in order to investigate the usefulness of higher-order SVD (HOSVD) for obtaining reduced-order parameterizations by truncation. Note that the truncation scheme (see Section 4.1.5) is only applied to the tensor of facies fields. Specifically, for the training ensemble consisting of 1000 facies models ($N_x = 50$, $N_y = 50$, $N_z = 10$, $N_{te} = 1000$), we can construct a fourth-order tensor, $\mathcal{A} \in \mathbb{R}^{N_x \times N_y \times N_z \times N_{te}}$, and implement the HOSVD algorithm to do the full tensor decomposition. Then, the truncation scheme is applied with different values of threshold ζ (0.9, 0.95 and 0.99) and the results are given in Table 4.14 including the ranks of four modes, the normalized model mismatch ξ, the model reduction ratio η and the computing time of parameterization t^{para}.

It is shown that although the model mismatch ξ is only 0.05 when $\zeta = 0.99$, we retain almost all information in different modes. When $\zeta = 0.9$, we reduce the size of original

Table 4.14: Truncated four-mode TD

Method	Tensor	Scheme	ζ	ξ	η	R_1	R_2	R_3	R_4	t^{para}
TD	$\mathcal{A} \in \mathbb{R}^{N_x \times N_y \times N_z \times N_{te}}$	4-mode	0.9	0.19	0.33	37	36	8	778	28.4s
			0.95	0.14	0.59	43	43	9	882	41.0s
			0.99	0.05	0.94	49	49	10	975	45.8s

tensor by two-thirds ($\eta = 0.33$) and the model mismatch ξ is smaller than 0.2, which seems to achieve a good balance between the approximation quality and model reduction. Similarly, we can vectorize the 3D facies fields first and construct a second-order tensor, $\mathcal{A} \in \mathbb{R}^{N_g \times N_{te}}$ with $N_g = N_x \times N_y \times N_z$. By applying the truncation scheme with HOSVD, we can obtain the truncated two-mode TD with different values of ζ (0.9, 0.95 and 0.99) and the results are given in Table 4.15. It is shown that when $\zeta = 0.9$, the number of model parameters is dramatically reduced ($\eta = 0.02$) and the approximation quality of original tensor is also desirable ($\xi = 0.1$).

Table 4.15: Truncated two-mode TD

Method	Tensor	Scheme	ζ	ξ	η	R_1	R_2	t^{para}
TD	$\mathcal{A} \in \mathbb{R}^{N_g \times N_{te}}$	2-mode	0.9	0.1	0.02	778	778	10.62s
			0.95	0.07	0.03	882	882	12.48s
			0.99	0.03	0.04	975	975	17.25s

Therefore, we implement the truncated two-mode and four-mode TD with the ranks associated with $\zeta = 0.9$, and the truncated NTD methods have identical ranks of different modes since there is no available HOSVD algorithm for NTD. The details of truncated parameterization methods are given in Table 4.16. For the purpose of comparison, we also design four parameterization schemes of TD and NTD without truncation and the ranks of different modes are determined heuristically (see Table 4.17). It is shown that when the ranks of different modes are the same, the computing time of NTD is about 5 times than that of TD when the four-mode scheme is employed, and it takes about 10 times longer to complete two-mode NTD than two-mode TD.

Table 4.16: Truncated TD and NTD ($N_{te} = 1000$)

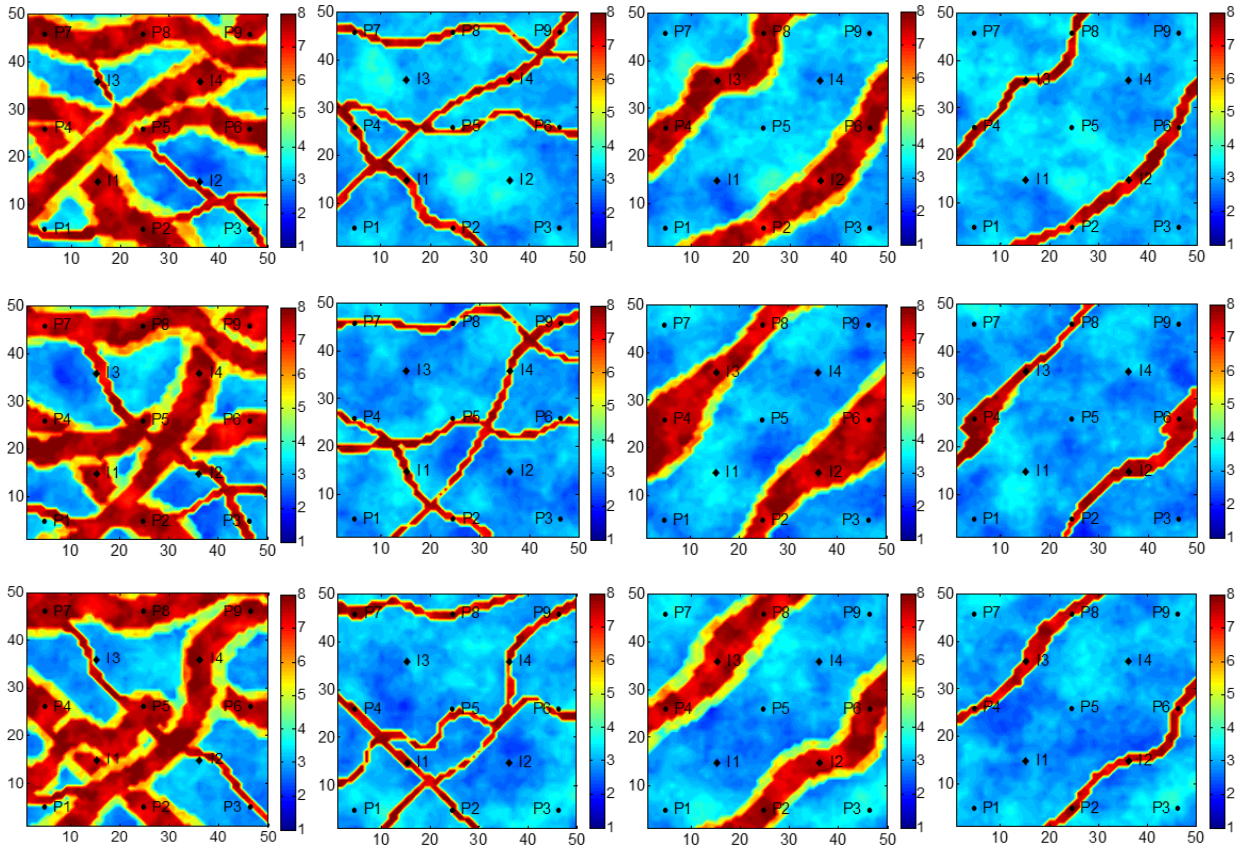
Method	Scheme	Retained number	Name	t^{para}
TD	2-mode	$R_1 = 778, R_2 = 778$	TD, 778	10.62s
NTD	2-mode	$R_1 = 778, R_2 = 778$	NTD, 778	158.34s
TD	4-mode	$R_1 = 37, R_2 = 36, R_3 = 8, R_4 = 778$	TD, $37 \times 36 \times 8 \times 778$	28.43s
NTD	4-mode	$R_1 = 37, R_2 = 36, R_3 = 8, R_4 = 778$	NTD, $37 \times 36 \times 8 \times 778$	119.51s

Table 4.17: TD and NTD without truncation ($N_{te} = 1000$)

Method	Scheme	Retained number	Name	t^{para}
TD	2-mode	$R_1 = 1000, R_2 = 1000$	TD, 1000	32.72s
NTD	2-mode	$R_1 = 1000, R_2 = 1000$	NTD, 1000	373.36s
TD	4-mode	$R_1 = 20, R_2 = 20, R_3 = 5, R_4 = 1000$	TD, $20 \times 20 \times 5 \times 1000$	16.29s
NTD	4-mode	$R_1 = 20, R_2 = 20, R_3 = 5, R_4 = 1000$	NTD, $20 \times 20 \times 5 \times 1000$	82.54s

In this case, a large ensemble of 500 realizations ($N_e = 500$) is employed for history matching. Figure 4.27 clearly shows the underlying complexity associated with facies distributions and heterogeneous permeability in the true model and prior realizations. The high geological uncertainty associated with the prior ensemble can be further illustrated through the prior production data shown in Fig. 4.28.

The prior production data and history matches obtained with eight parameterization schemes (see Tables 4.16 and 4.17) are shown in Figs. 4.29 to 4.32. It is clear that the large prior uncertainty has been reduced significantly with different parameterization methods. Through history matching, we are able to obtain good history data matches for all wells except the water production rate at P3 which did not experience water breakthrough during the historical period. With same implementation scheme (e.g., "TD, 778" and "NTD, 778"), the NTD method seems to slightly outperform the TD method in terms of history data match and production forecast. Besides, the truncated TD and NTD methods perform similarly compared to the TD and NTD without truncation. Moreover, the data matches obtained with two-mode TD and NTD methods appear to be slightly better than those obtained with four-mode TD and NTD methods. Especially for well P3, the history water rates can be

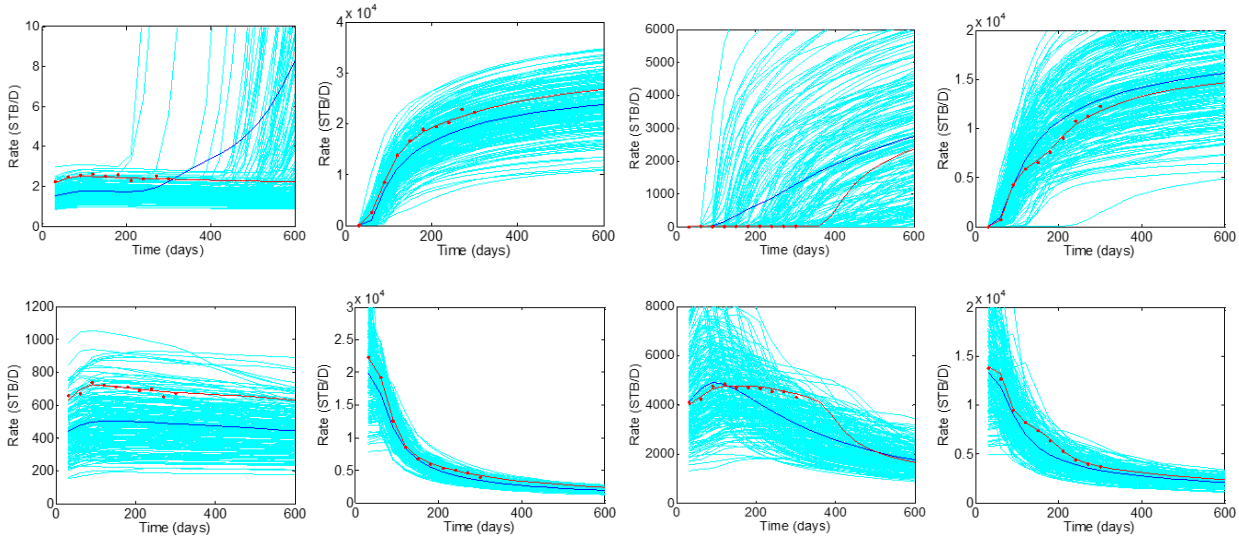


(a) Layer 1 (b) Layer 6 (c) Layer 7 (d) Layer 10

Figure 4.27: $\ln(k)$ fields of the true model and two prior realizations (Case 2.2, $N_e = 500$): true model (1st row), prior model #1 (2nd row) and prior model #2 (3rd row).

reasonably matched with "TD, 1000" and "NTD, 1000", but there are obviously early water breakthrough in many posterior realizations obtained with "TD, $20 \times 20 \times 5 \times 1000$" and "NTD, $20 \times 20 \times 5 \times 1000$". However, one should keep in mind that when the total retained number of basis functions is the same, the computational cost of implementing two-mode TD and NTD are much higher than those of implementing four-mode TD and NTD.

The $\ln(k)$ fields of the true model and two posterior realizations shown in Figs. 4.33 to 4.36 suggest that the major features of the facies distributions of the true model are resolved reasonably well with different parameterization methods. Although more geological details in layer 6 can be captured by the two-mode TD and NTD methods, the channel distributions look more rough than those obtained with the four-mode TD and NTD methods probably because we retain more high frequency features when implementing the two-mode

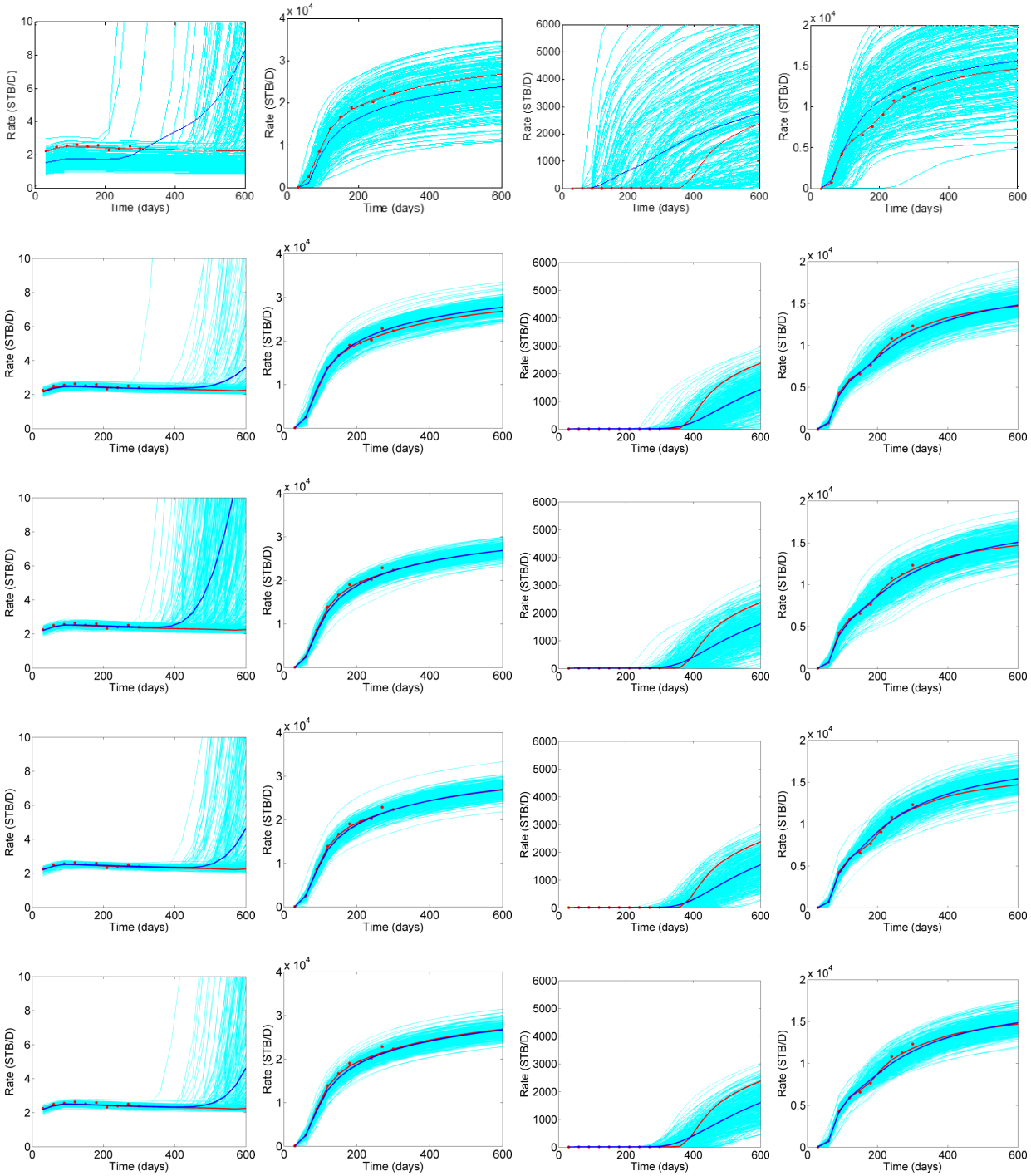


(a) P3 (b) P4 (c) P7 (d) P8

Figure 4.28: Prior production data (Case 2.2, $N_e = 500$): water rates (1st row) and oil rates (2nd row). True (red curves), observed data (red dots), simulated data of realizations (light blue curves), and ensemble mean (dark blue curves). History: $0 < t \leq 300$ days, forecast: $t > 300$ days.

scheme. Besides, there is not much difference between the posterior realizations calibrated with the truncated TD and NTD methods with those updated with the TD and NTD without truncation. Through comparing the two posterior realizations obtained with same parameterization method, we can see that there still remains geological variability in the posterior ensembles when $N_e = 500$.

Figures 4.37 and 4.38 show the 100% and 90% probability maps based on the posterior realizations obtained with four-mode TD and NTD methods. In the 100% probability maps, each red point means that the gridblock is occupied by channel facies in the true model and it is correctly resolved to be channel facies in all posterior models. On the other hand, each blue point indicates that the gridblock has channel facies in all posterior realizations but in the true model this gridblock is not occupied by channel facies. The 90% probability map is defined similarly. With the large ensemble of 500 models, the 100% probability maps of sand channels obtained with the "TD, $20 \times 20 \times 5 \times 1000$" and "TD, $37 \times 36 \times 8 \times 778$" give desirable uncertainty characterization of the posterior facies distributions since only the gridblocks at or very near the well locations where the facies type is observed are resolved to



(a) P3 (b) P4 (c) P7 (d) P8

Figure 4.29: Prior and posterior water rates (Case 2.2, $N_e = 500$): prior (1st row); posterior, TD, 1000 (2nd row); posterior, TD, 778 (3rd row); posterior, NTD, 1000 (4th row); posterior, NTD, 778 (5th row). True (red curves), observed data (red dots), simulated data of realizations (light blue curves), and ensemble mean (dark blue curves). History: $0 < t \leq 300$ days, forecast: $t > 300$ days.

be channel facies through history matching. Although there are more red points in the 100% probability maps obtained with the "NTD, $20\times20\times5\times1000$" and "NTD, $37\times36\times8\times778$", these gridblocks are correctly resolved to be channel facies and we can see the blue points are very rare. In the 90% probability maps (see Fig. 4.37), it is shown that the correctly resolved channel gridblocks (red points) are far more than the gridblocks that are incorrectly resolved to be channel facies (blue points) based on the posterior realizations obtained with different parameterization methods.

According to the 100% and 90% probability maps obtained with eight parameterization schemes, the statistical indicators are calculated and given in Tables 4.18 and 4.19. It is shown that we are able to resolve slightly more channel gridblocks correctly with the NTD method than the TD method. Moreover, compared to the truncated TD and NTD methods, the posterior realizations obtained with the TD and NTD methods without truncation give more correctly resolved channel gridblocks.

Table 4.18: Statistical indicators based on 100% and 90% probability maps obtained with two-mode TD and NTD (Case 2.2)

Method	N_e	N_w	Prob.(%)	N_{gc}^{true}	$N_{gc}^{\text{posterior}}$	N_{gc}^{good}	$R^{\text{good}}(\%)$
TD, 1000	500	13	100%	6104	107	107	100.0%
TD, 1000	500	13	90%	6104	1891	1764	93.3%
TD, 778	500	13	100%	6104	103	103	100.0%
TD, 778	500	13	90%	6104	1623	1554	95.7%
NTD, 1000	500	13	100%	6104	266	264	99.2%
NTD, 1000	500	13	90%	6104	2311	2180	94.3%
NTD, 778	500	13	100%	6104	151	151	100.0%
NTD, 778	500	13	90%	6104	2080	1956	94.1%

Besides the gridblocks that are resolved to be channel facies with 100% or 90% probability level based on the posterior realizations, we may also consider the gridblocks that are occupied by channel facies in the true model but are not resolved through history matching. Figure 4.39 shows the 0% probability maps of sand channels obtained with four-mode TD

Table 4.19: Statistical indicators based on 100% and 90% probability maps obtained with four-mode TD and NTD (Case 2.2)

Method	N_e	N_w	Prob.(%)	N_{gc}^{true}	$N_{gc}^{\text{posterior}}$	N_{gc}^{good}	$R^{\text{good}}(\%)$
TD, $20 \times 20 \times 5 \times 1000$	500	13	100%	6104	226	226	100.0%
TD, $20 \times 20 \times 5 \times 1000$	500	13	90%	6104	2135	2004	93.9%
TD, $37 \times 36 \times 8 \times 778$	500	13	100%	6104	114	114	100.0%
TD, $37 \times 36 \times 8 \times 778$	500	13	90%	6104	1795	1706	95.1%
NTD, $20 \times 20 \times 5 \times 1000$	500	13	100%	6104	616	613	99.5%
NTD, $20 \times 20 \times 5 \times 1000$	500	13	90%	6104	2628	2415	91.9%
NTD, $37 \times 36 \times 8 \times 778$	500	13	100%	6104	576	572	99.3%
NTD, $37 \times 36 \times 8 \times 778$	500	13	90%	6104	2664	2440	91.6%

and NTD methods. In the bottow four rows of Fig. 4.39, each blue point indicates that although the gridblock has channel facies in the true model, no posterior realization has channel facies in that particular gridblock, i.e., the posterior probability of that gridblock being occupied by channel facies is 0. We can see that the blue points in the 0% probability maps of sand channels obtained with the "TD, $20\times20\times5\times1000$" and "TD, $37\times36\times8\times778$" are very rare. On the other hand, there are small number of blue points in the 0% probability maps obtained with the "NTD, $20 \times 20 \times 5 \times 1000$" and "NTD, $37 \times 36 \times 8 \times 778$". Although more channel gridblocks are not resolved through history matching based on the 10% probability maps of sand channels (see Fig. 4.40), most of the blue points distribute along the boundaries of channels and are away from the wells, so they are expected to have less influence on the well production. Moreover, the probability maps obtained by the TD and NTD methods with or without truncation look very similar.

For the purpose of quantitative evaluation, we count the number of true channel gridblocks that not effectively resolved (blue points in Figs. 4.39 and 4.40) and denote this number by N_{gc}^{missed}. Again, let N_{gc}^{true} denote the number of gridblocks in the true model that are occupied by the channel facies. Thus, the fraction of channel gridblocks in the true model

that are not resolved through history matching is defined by

$$R^{\text{missed}} = N_{gc}^{\text{missed}} / N_{gc}^{\text{true}}. \tag{4.45}$$

With Eq. 4.45, we calculate the statistical indicators based on the 0% and 10% probability maps and summarize the results in Tables 4.20 and 4.21. It is shown that the R^{missed} is less than 3% based on the 0% probability maps since there is still geological variability in the posterior realizations with the large ensemble of 500 models. When it comes to the 10% probability maps, the proportion of true channel gridblocks that are not resolved with the NTD methods is higher than that with the TD methods when same implementation scheme is used. However, the unresolved channel gridblocks seem to have small impact on the quality of history data matches and uncertainty quantification of production predictions based on the results shown in Figs. 4.30 and 4.32. Compared to the results obtained with the TD and NTD methods without truncation, the proportion of true channel gridblocks that are not resolved through history matching is lower when the truncated TD and NTD methods are employed.

Table 4.20: Statistical indicators based on 0% and 10% probability maps obtained with two-mode TD and NTD (Case 2.2)

Method	N_e	N_w	Prob.(%)	N_{gc}^{true}	N_{gc}^{missed}	R^{missed}(%)
TD, 1000	500	13	0%	6104	10	0.16%
TD, 1000	500	13	10%	6104	507	8.31%
TD, 778	500	13	0%	6104	1	0.02%
TD, 778	500	13	10%	6104	504	8.26%
NTD, 1000	500	13	0%	6104	2	0.03%
NTD, 1000	500	13	10%	6104	572	9.37%
NTD, 778	500	13	0%	6104	2	0.03%
NTD, 778	500	13	10%	6104	536	8.78%

Figure 4.41 gives the normalized data mismatch calculated by Eq. 4.40. It is shown that the data match results obtained with the truncated TD and NTD methods are slightly

Table 4.21: Statistical indicators based on 0% and 10% probability maps obtained with four-mode TD and NTD (Case 2.2)

Method	N_e	N_w	Prob.(%)	N_{gc}^{true}	N_{gc}^{missed}	$R^{\text{missed}}(\%)$
TD, $20 \times 20 \times 5 \times 1000$	500	13	0%	6104	22	0.36%
TD, $20 \times 20 \times 5 \times 1000$	500	13	10%	6104	756	12.39%
TD, $37 \times 36 \times 8 \times 778$	500	13	0%	6104	4	0.07%
TD, $37 \times 36 \times 8 \times 778$	500	13	10%	6104	598	9.79%
NTD, $20 \times 20 \times 5 \times 1000$	500	13	0%	6104	156	2.56%
NTD, $20 \times 20 \times 5 \times 1000$	500	13	10%	6104	1102	18.05%
NTD, $37 \times 36 \times 8 \times 778$	500	13	0%	6104	139	2.28%
NTD, $37 \times 36 \times 8 \times 778$	500	13	10%	6104	940	15.39%

better than those obtained with the methods without truncation. Although the two-mode methods seem to yield smaller data mismatch values than those associated with the four-mode methods, one should note that it takes much more time to compute the two-mode TD and NTD.

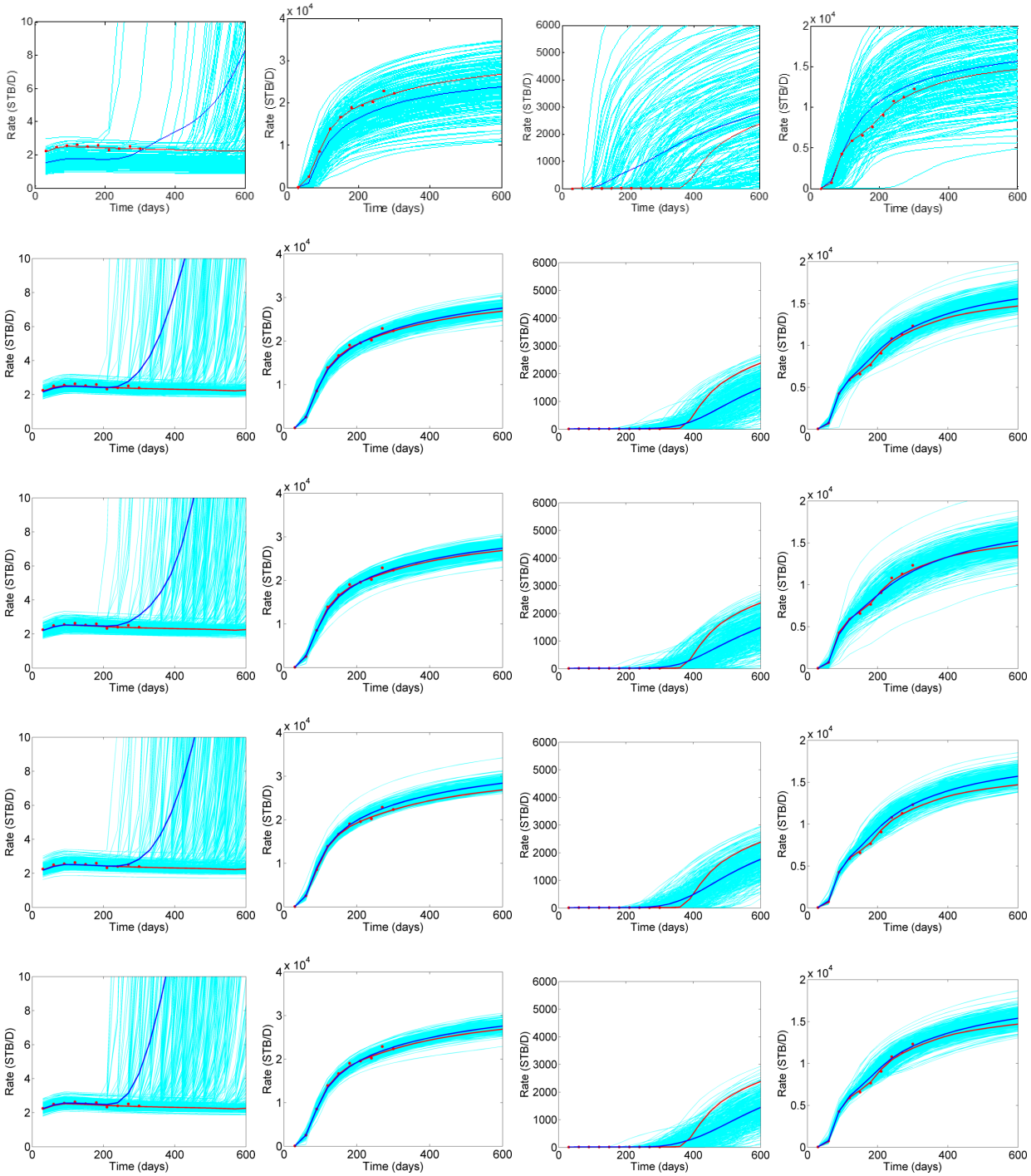
In Case 2.2, we also investigate the effect of ensemble size on the performance of history matching. To be specific, two ensembles of different size ($N_e = 200$ and $N_e = 500$) are parameterized by the "TD, 1000" and "NTD, 1000" methods, respectively, and calibrated by assimilating the same set of observed data. Figures 4.42 and 4.43 show the posterior production data obtained with the "TD, 1000" and "NTD, 1000" when $N_e = 200$ and $N_e = 500$. We can see that the posterior ensembles tend to collapse when only 200 realizations are used for history matching and the corresponding production predictions appear to be more biased (see column (c) in Fig. 4.42).

Comparing the last two rows of 100% probability maps shown in Fig. 4.44, we see that when a large ensemble ($N_e = 500$) is used, only gridblocks at or very near well gridblocks where the channel facies is observed are occupied by the channel facies in all 500 posterior realizations, i.e., posterior uncertainty in the distribution of the channel facies appears to be much greater when $N_e = 500$ than when $N_e = 200$ (see statistical indicators in Table 4.22).

However, the probability maps when $N_e = 500$ give better characterization of uncertainty than the $N_e = 200$ case because the probability of a gridblock being occupied by a particular facies should be 1.0 (100%) only if the facies is observed there. To clarify this, we show in Fig. 4.45 the true channel distribution and the 0.9 (90%) and 0.8 (80%) probability maps for the $N_e = 500$ case with the "NTD, 1000" method. In the 90% probability map, a gridblock is colored red if and only if 90% (450) of the posterior realizations have the channel facies in that gridblock. The 80% probability map is defined similarly. Note that the 90% and 80% probability maps show more geological continuity and look similar to the true map of the channel distributions shown in the first row of Fig. 4.45. In addition, the box plot of the normalized data mismatch corresponding to the $N_e = 200$ and $N_e = 500$ cases is shown in Fig. 4.46. The computing time of running cases is given in Table 4.23.

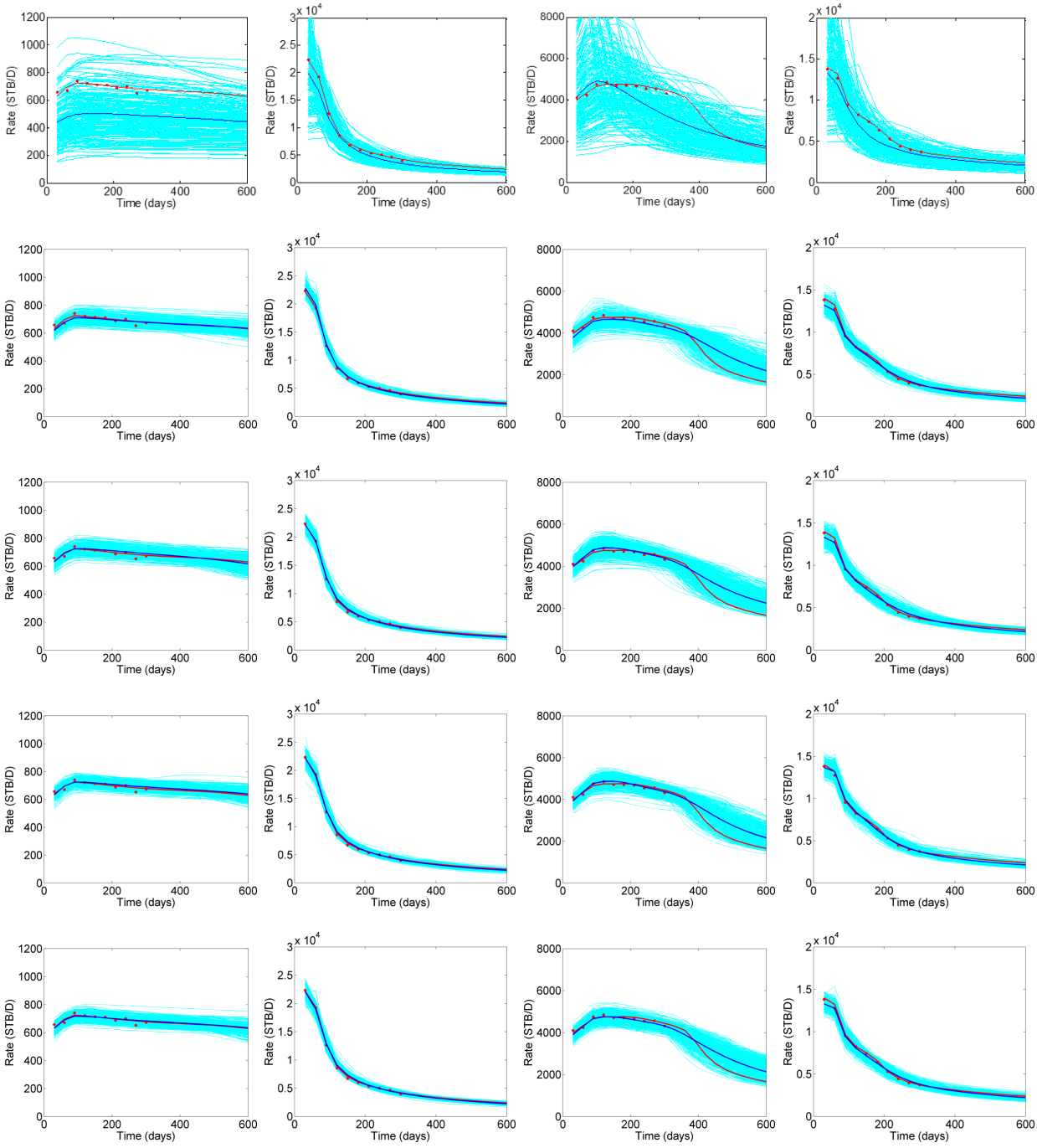
Table 4.22: Statistical indicators based on 100% probability maps (Case 2.2)

Method	N_e	N_{gc}^{true}	$N_{gc}^{\text{posterior}}$	N_{gc}^{good}	N_{gc}^{bad}	$R^{\text{good}}(\%)$	$R^{\text{bad}}(\%)$
TD, 1000	200	6104	1602	1513	89	94.4%	5.6%
TD, 1000	500	6104	128	127	1	99.2%	0.8%
NTD, 1000	200	6104	2244	2069	175	92.2%	7.8%
NTD,1000	500	6104	244	242	2	99.2%	0.8%



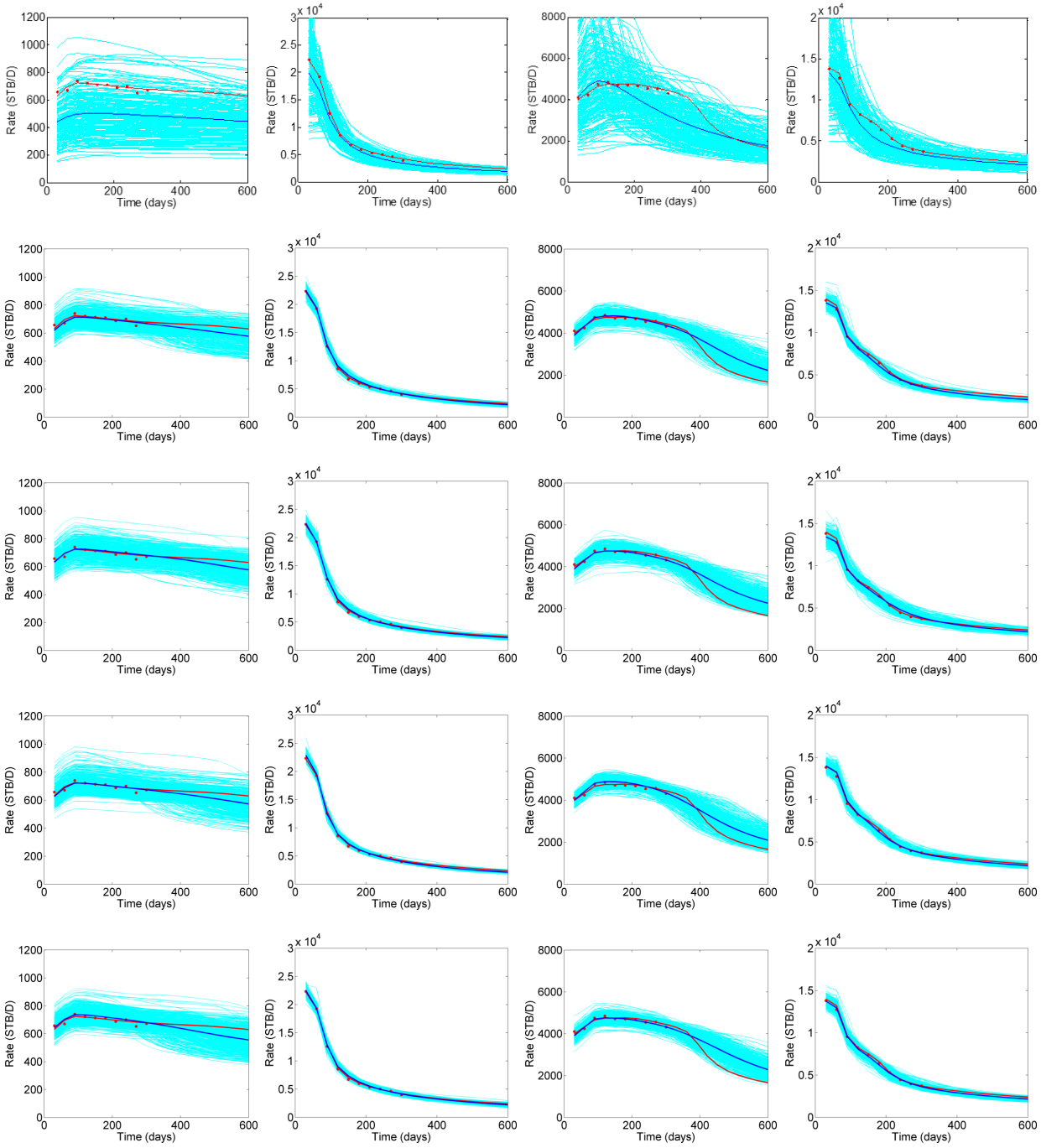
(a) P3 (b) P4 (c) P7 (d) P8

Figure 4.30: Prior and posterior water rates (Case 2.2, $N_e = 500$): prior (1st row); posterior, TD, $20 \times 20 \times 5 \times 1000$ (2nd row); posterior, TD, $37 \times 36 \times 8 \times 778$ (3rd row); posterior, NTD, $20 \times 20 \times 5 \times 1000$ (4th row); posterior, NTD, $37 \times 36 \times 8 \times 778$ (5th row). True (red curves), observed data (red dots), simulated data of realizations (light blue curves), and ensemble mean (dark blue curves). History: $0 < t \leq 300$ days, forecast: $t > 300$ days.



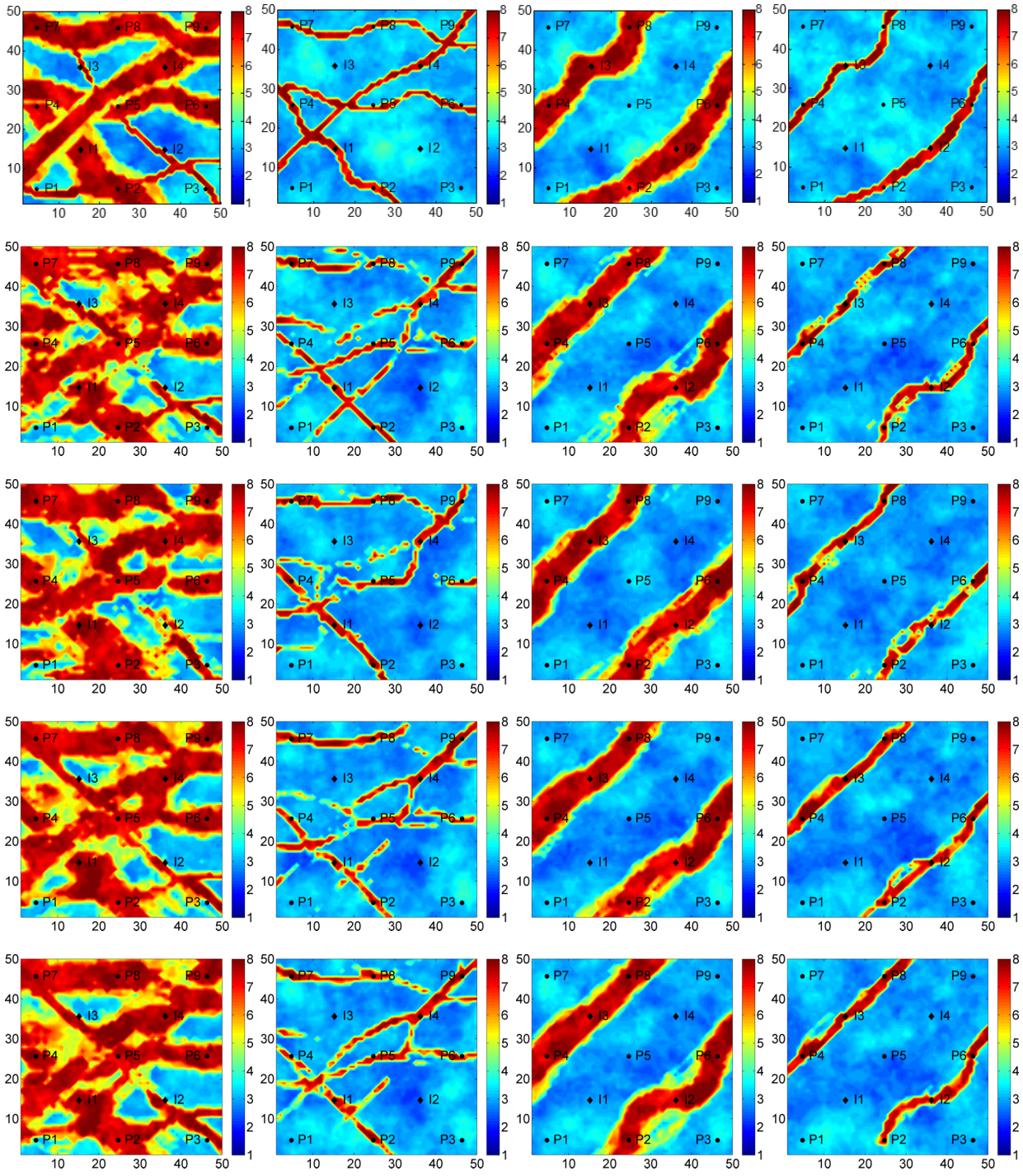
(a) P3 (b) P4 (c) P7 (d) P8

Figure 4.31: Prior and posterior oil rates (Case 2.2, $N_e = 500$): prior (1st row); posterior, TD, 1000 (2nd row); posterior, TD, 778 (3rd row); posterior, NTD, 1000 (4th row); posterior, NTD, 778 (5th row). True (red curves), observed data (red dots), simulated data of realizations (light blue curves), and ensemble mean (dark blue curves). History: $0 < t \leq 300$ days, forecast: $t > 300$ days.



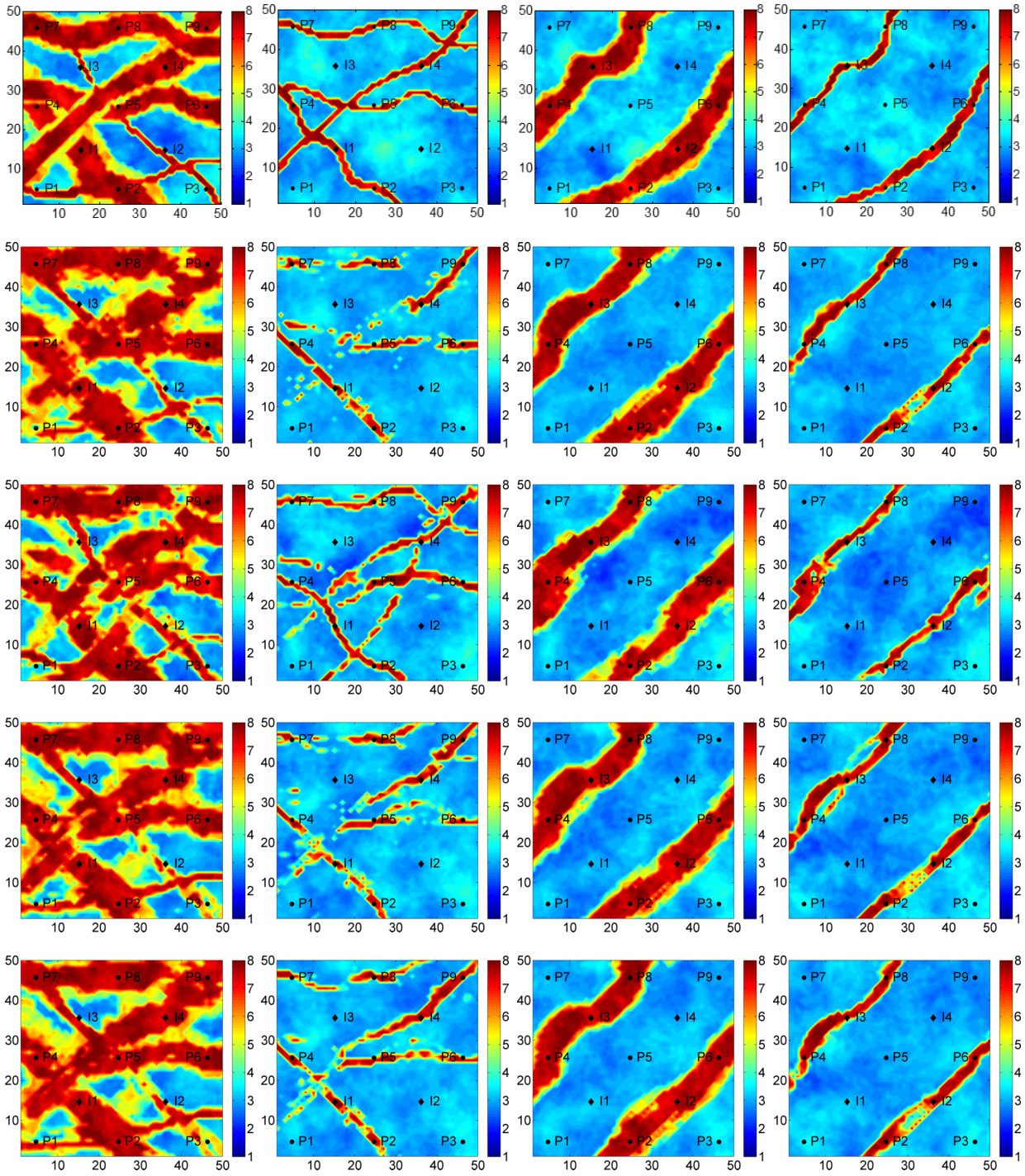
(a) P3 (b) P4 (c) P7 (d) P8

Figure 4.32: Prior and posterior oil rates (Case 2.2, $N_e = 500$): prior (1st row); posterior, TD, $20 \times 20 \times 5 \times 1000$ (2nd row); posterior, TD, $37 \times 36 \times 8 \times 778$ (3rd row); posterior, NTD, $20 \times 20 \times 5 \times 1000$ (4th row); posterior, NTD, $37 \times 36 \times 8 \times 778$ (5th row). True (red curves), observed data (red dots), simulated data of realizations (light blue curves), and ensemble mean (dark blue curves). History: $0 < t \leq 300$ days, forecast: $t > 300$ days.



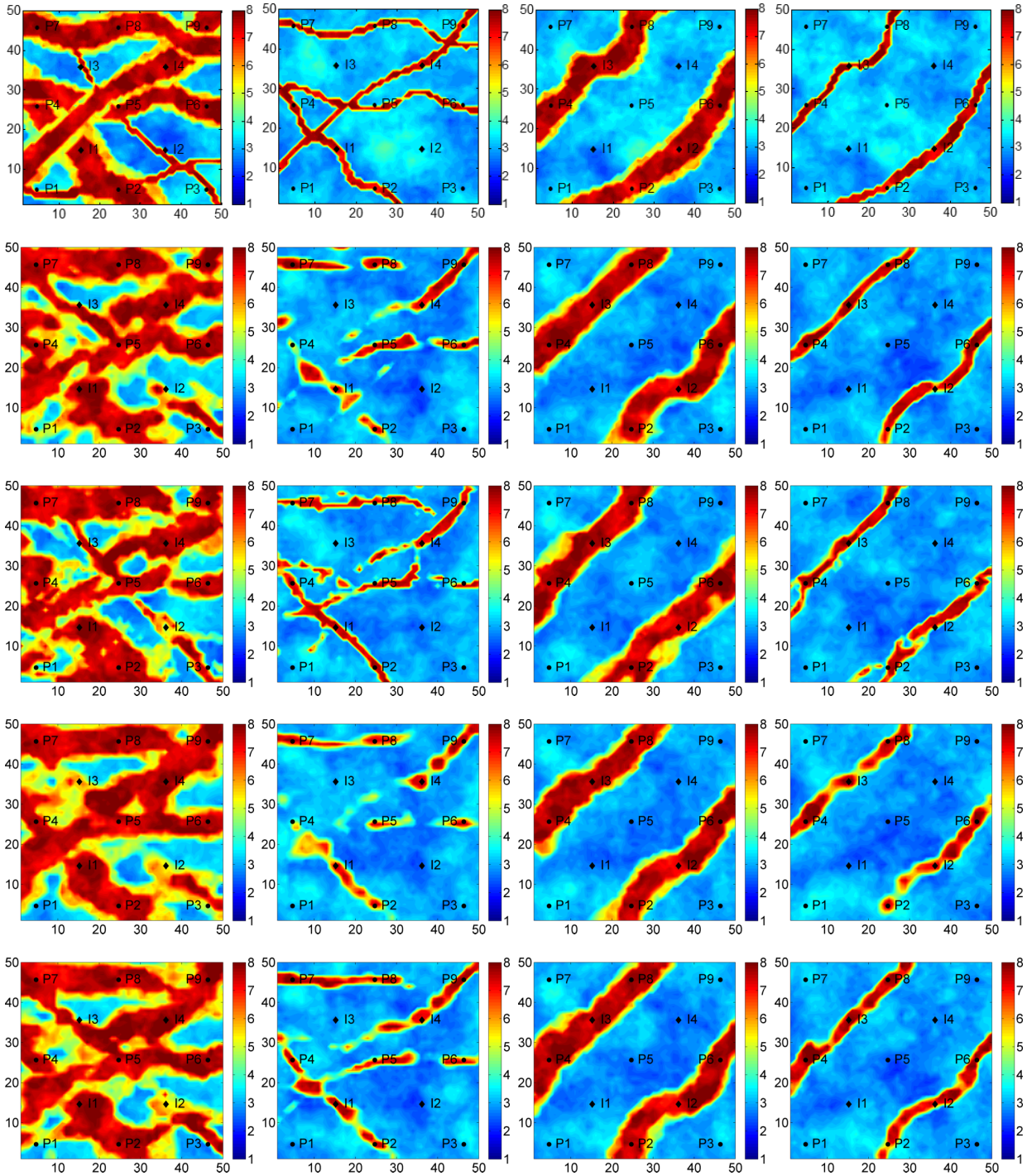
(a) Layer 1 (b) Layer 6 (c) Layer 7 (d) Layer 10

Figure 4.33: $\ln(k)$ fields of the true model and posterior realization #1 (Case 2.2, $N_e = 500$): true model (1st row); TD, 1000 (2nd row); TD, 778 (3rd row); NTD, 1000 (4th row); NTD, 778 (5th row).



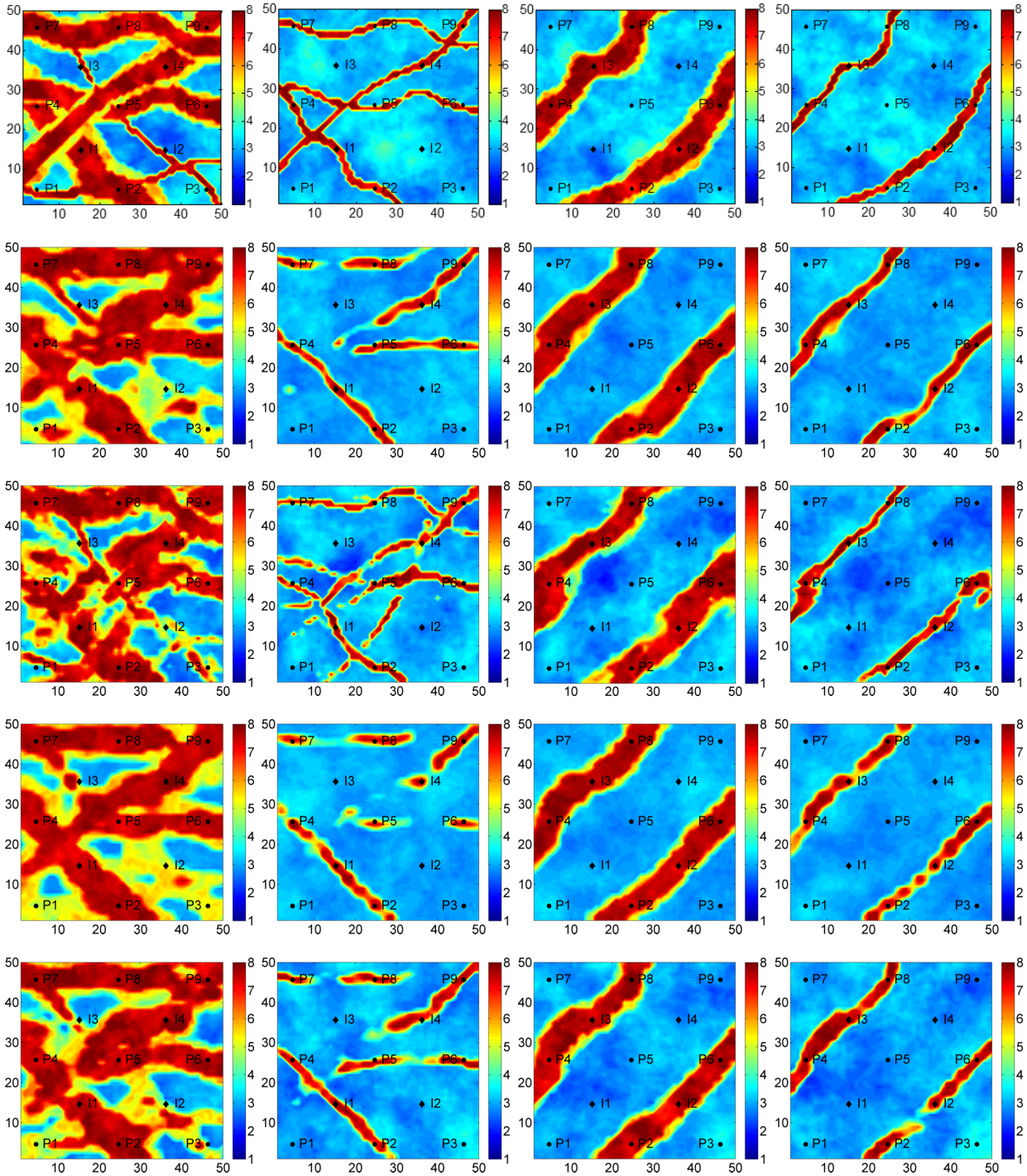
(a) Layer 1 (b) Layer 6 (c) Layer 7 (d) Layer 10

Figure 4.34: $\ln(k)$ fields of the true model and posterior realization #2 (Case 2.2, $N_e = 500$): true model (1st row); TD, 1000 (2nd row); TD, 778 (3rd row); NTD, 1000 (4th row); NTD, 778 (5th row).



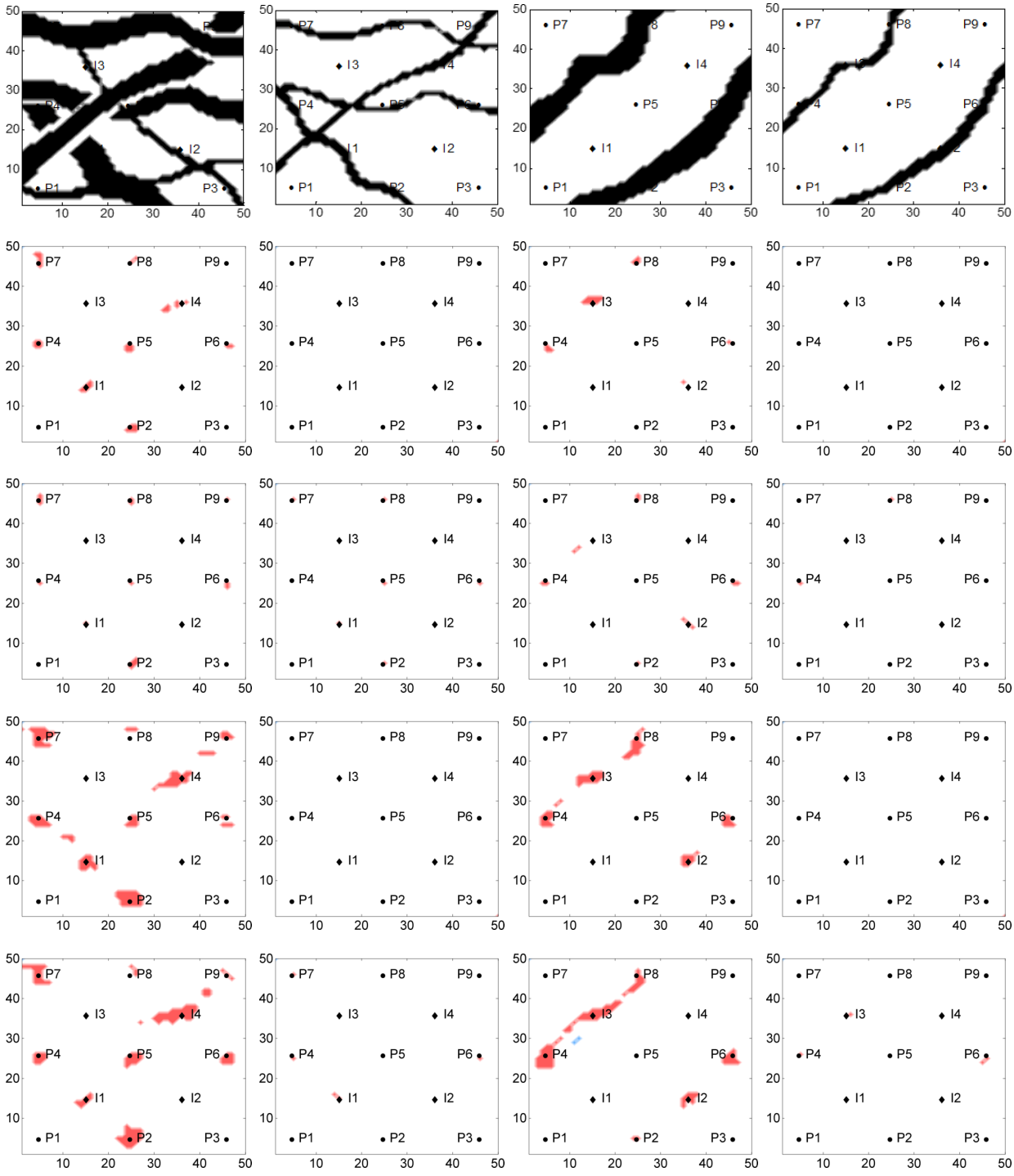
(a) Layer 1 (b) Layer 6 (c) Layer 7 (d) Layer 10

Figure 4.35: $\ln(k)$ fields of the true model and posterior realization #1 (Case 2.2, $N_e = 500$): true model (1st row); TD, $20 \times 20 \times 5 \times 1000$ (2nd row); TD, $37 \times 36 \times 8 \times 778$ (3rd row); NTD, $20 \times 20 \times 5 \times 1000$ (4th row); NTD, $37 \times 36 \times 8 \times 778$ (5th row).



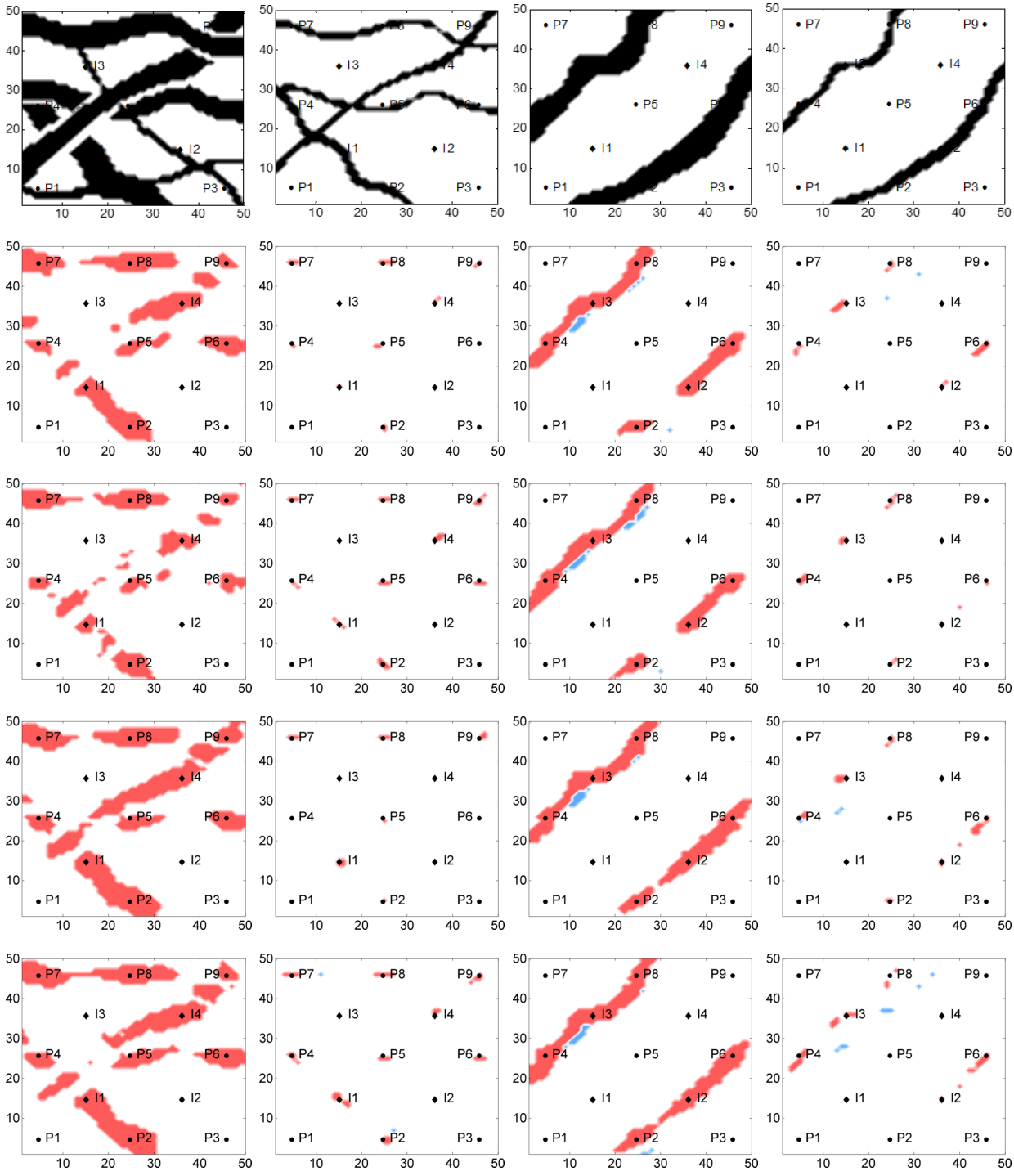
(a) Layer 1 (b) Layer 6 (c) Layer 7 (d) Layer 10

Figure 4.36: $\ln(k)$ fields of the true model and posterior realization #2 (Case 2.2, $N_e = 500$): true model (1st row); TD, $20 \times 20 \times 5 \times 1000$ (2nd row); TD, $37 \times 36 \times 8 \times 778$ (3rd row); NTD, $20 \times 20 \times 5 \times 1000$ (4th row); NTD, $37 \times 36 \times 8 \times 778$ (5th row).



(a) Layer 1 (b) Layer 6 (c) Layer 7 (d) Layer 10

Figure 4.37: 100% probability maps of sand channels (Case 2.2, $N_e = 500$): true (1st row); posterior, TD, $20 \times 20 \times 5 \times 1000$ (2nd row); posterior, TD, $37 \times 36 \times 8 \times 778$ (3rd row); posterior, NTD, $20 \times 20 \times 5 \times 1000$ (4th row); posterior, NTD, $37 \times 36 \times 8 \times 778$ (5th row).

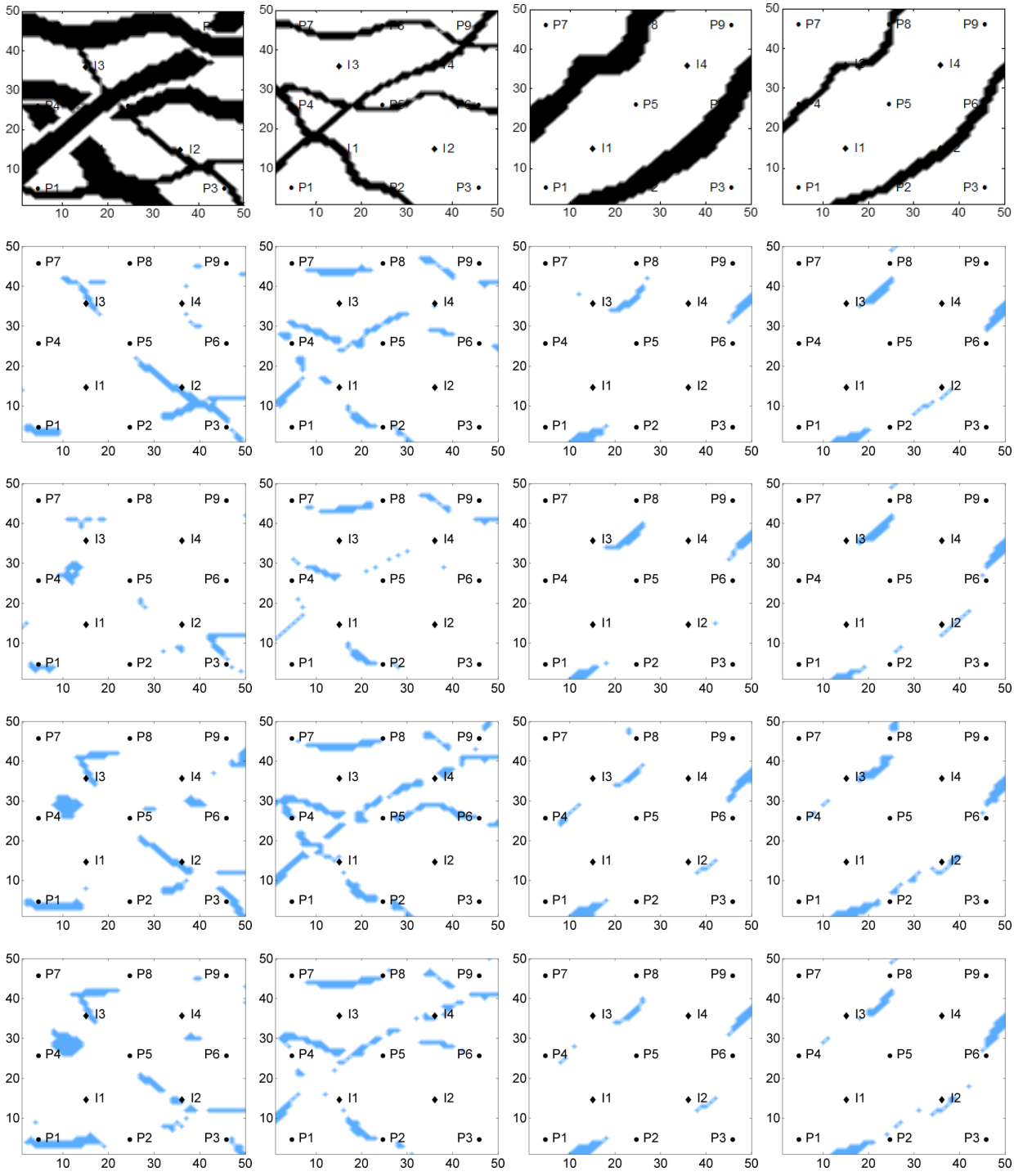


(a) Layer 1 (b) Layer 6 (c) Layer 7 (d) Layer 10

Figure 4.38: 90% probability maps of sand channels (Case 2.2, $N_e = 500$): true (1st row); posterior, TD, $20 \times 20 \times 5 \times 1000$ (2nd row); posterior, TD, $37 \times 36 \times 8 \times 778$ (3rd row); posterior, NTD, $20 \times 20 \times 5 \times 1000$ (4th row); posterior, NTD, $37 \times 36 \times 8 \times 778$ (5th row).

(a) Layer 1 (b) Layer 6 (c) Layer 7 (d) Layer 10

Figure 4.39: 0% probability maps of sand channels (Case 2.2, $N_e = 500$): true (1st row); posterior, TD, $20 \times 20 \times 5 \times 1000$ (2nd row); posterior, TD, $37 \times 36 \times 8 \times 778$ (3rd row); posterior, NTD, $20 \times 20 \times 5 \times 1000$ (4th row); posterior, NTD, $37 \times 36 \times 8 \times 778$ (5th row).



(a) Layer 1 (b) Layer 6 (c) Layer 7 (d) Layer 10

Figure 4.40: 10% probability maps of sand channels (Case 2.2, $N_e = 500$): true (1st row); posterior, TD, $20 \times 20 \times 5 \times 1000$ (2nd row); posterior, TD, $37 \times 36 \times 8 \times 778$ (3rd row); posterior, NTD, $20 \times 20 \times 5 \times 1000$ (4th row); posterior, NTD, $37 \times 36 \times 8 \times 778$ (5th row).

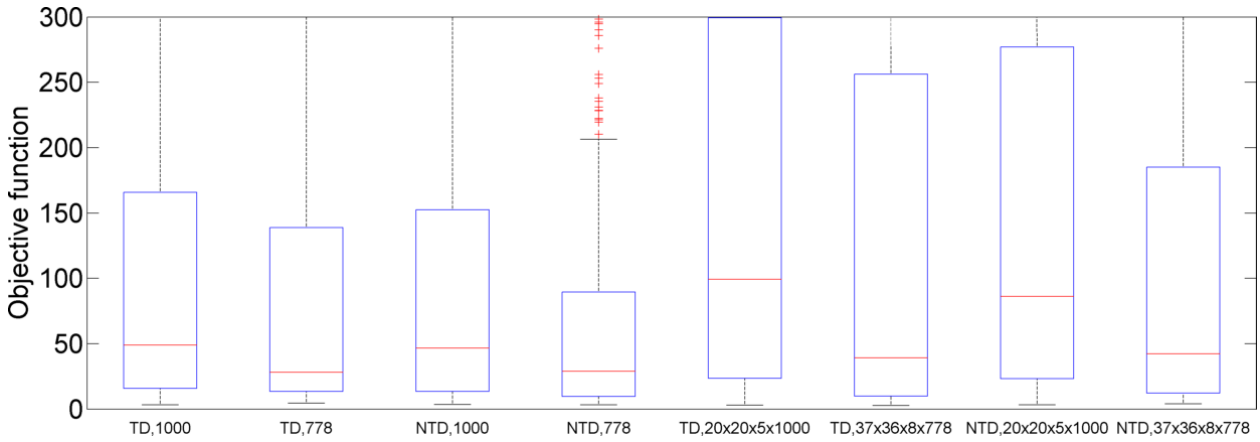
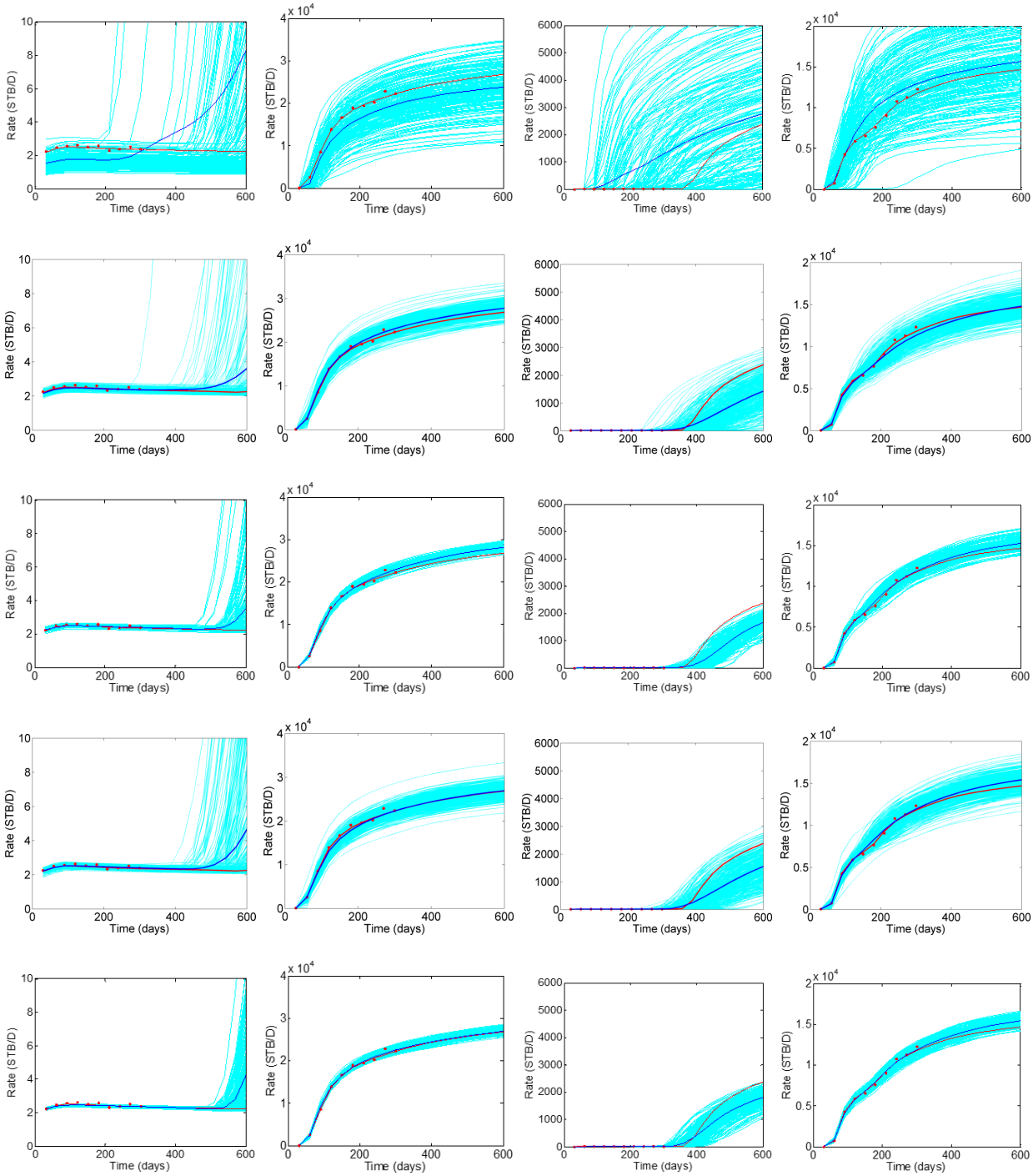
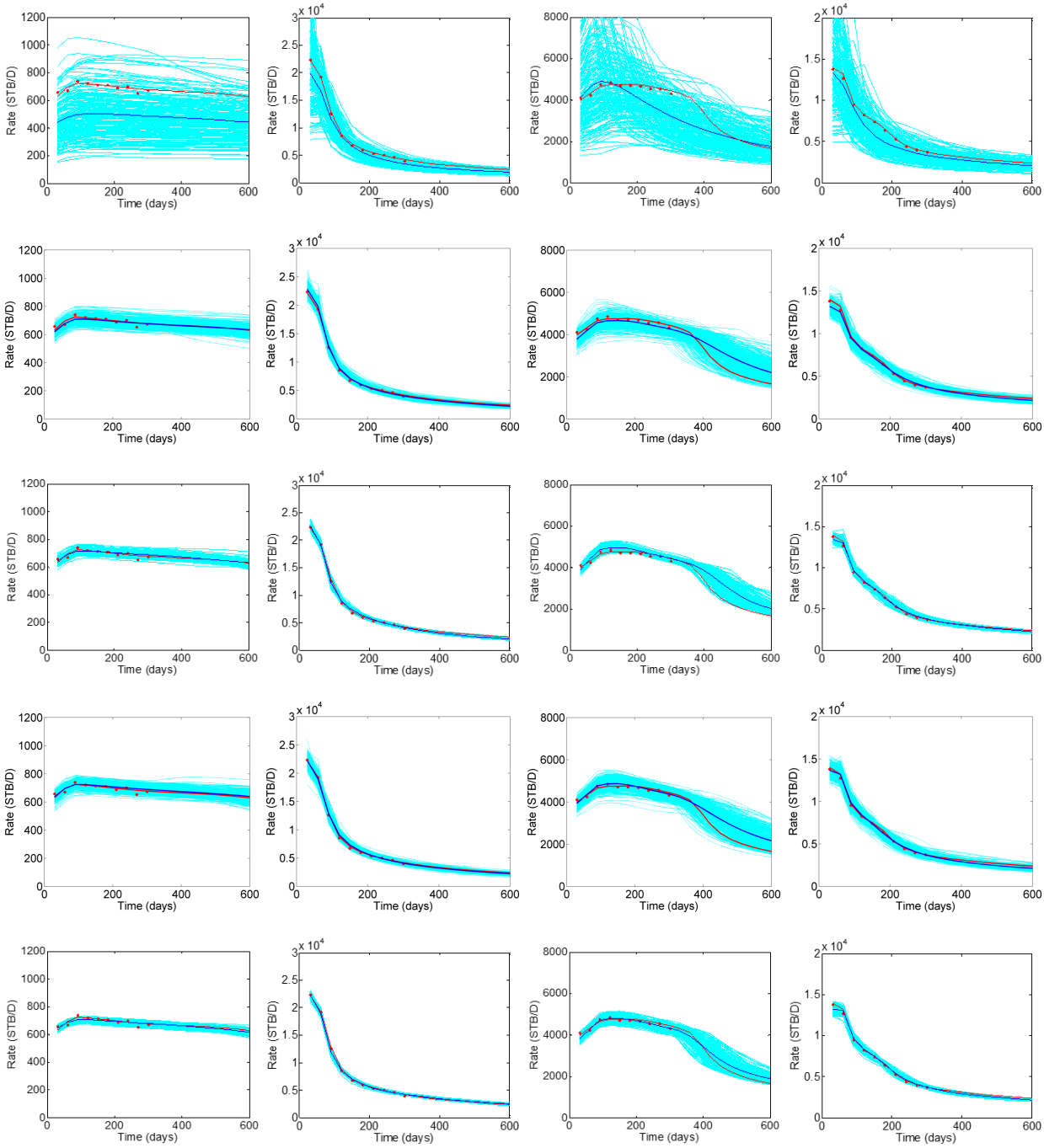


Figure 4.41: Normalized data mismatch (Case 2.2, $N_e = 500$). Median (central red mark), 25th and 75th percentiles (bottom and top edges of the box), extreme data points (whiskers), and outliers ("+" symbol).



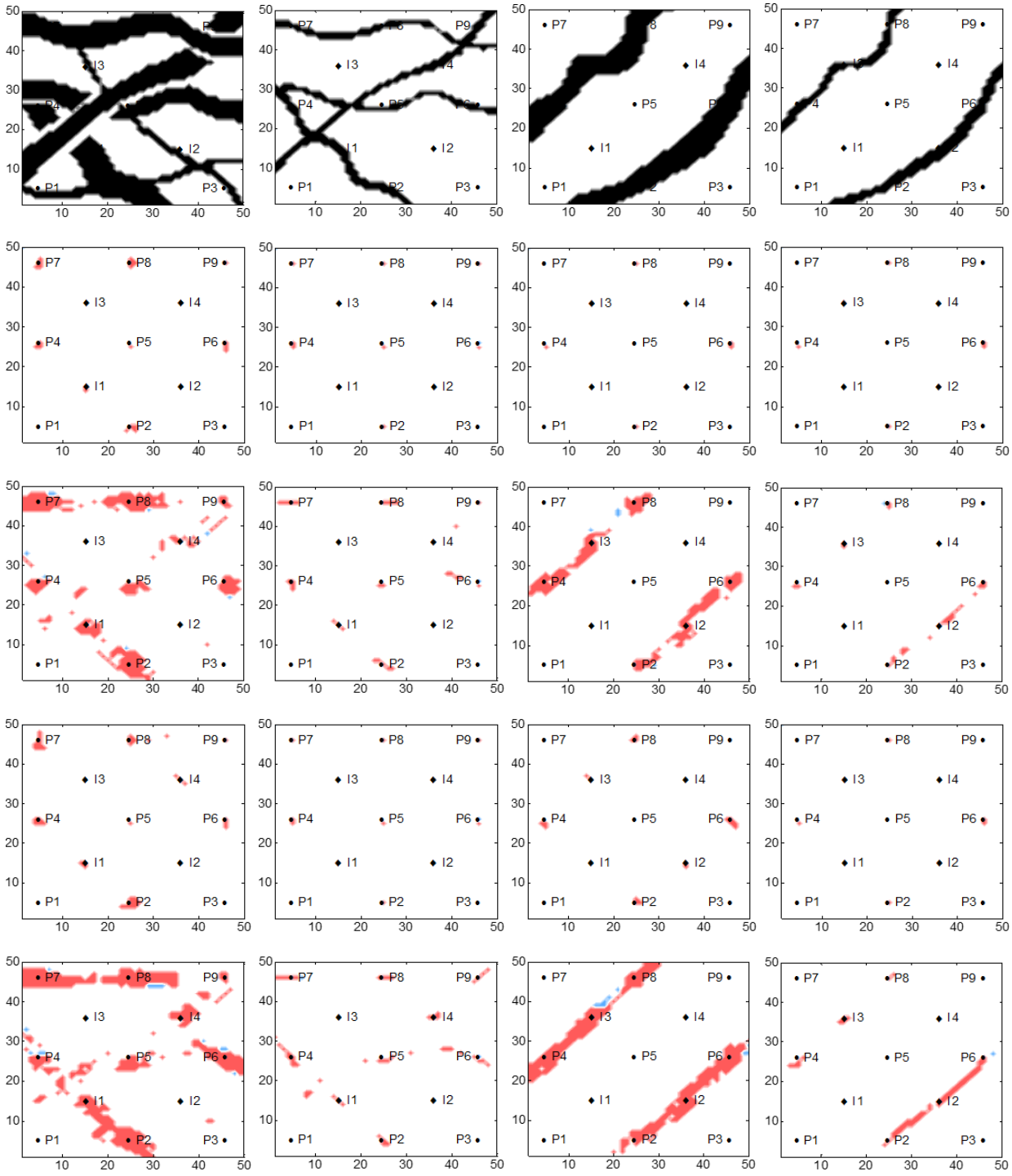
(a) P3 (b) P4 (c) P7 (d) P8

Figure 4.42: Prior and posterior water rates (Case 2.2): prior (1st row); posterior, TD, 1000, $N_e = 500$ (2nd row); posterior, TD, 1000, $N_e = 200$ (3rd row); posterior, NTD, 1000, $N_e = 500$ (4th row); posterior, NTD, 1000, $N_e = 200$ (5th row). True (red curves), observed data (red dots), simulated data of realizations (light blue curves), and ensemble mean (dark blue curves). History: $0 < t \leq 300$ days, forecast: $t > 300$ days.



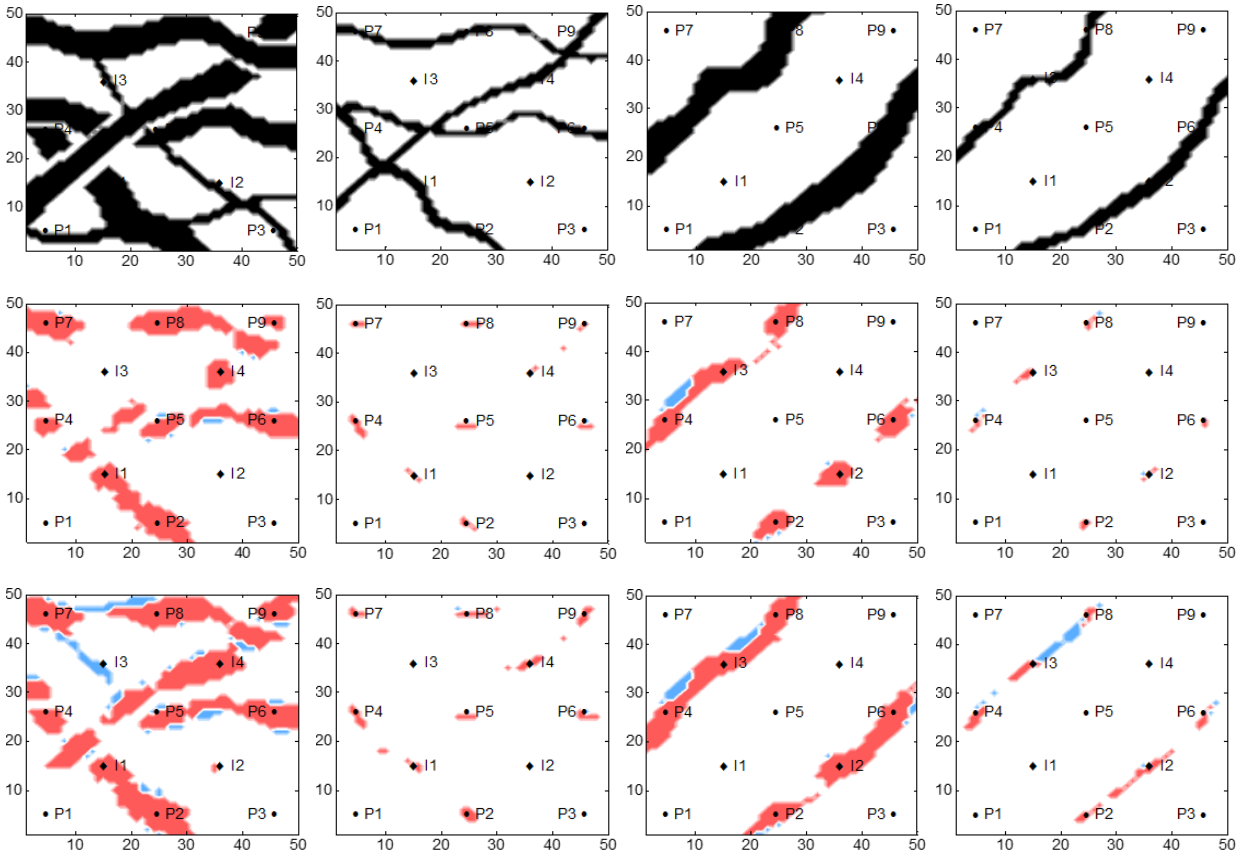
(a) P3 (b) P4 (c) P7 (d) P8

Figure 4.43: Prior and posterior oil rates (Case 2.2): prior (1st row); posterior, TD, 1000, $N_e = 500$ (2nd row); posterior, TD, 1000, $N_e = 200$ (3rd row); posterior, NTD, 1000, $N_e = 500$ (4th row); posterior, NTD, 1000, $N_e = 200$ (5th row). True (red curves), observed data (red dots), simulated data of realizations (light blue curves), and ensemble mean (dark blue curves). History: $0 < t \leq 300$ days, forecast: $t > 300$ days.



(a) Layer 1 (b) Layer 6 (c) Layer 7 (d) Layer 10

Figure 4.44: 100% probability maps of sand channels (Case 2.2): true (1st row); posterior, TD, 1000, $N_e = 500$ (2nd row); posterior, TD, 1000, $N_e = 200$ (3rd row); posterior, NTD, 1000, $N_e = 500$ (4th row); posterior, NTD, 1000, $N_e = 200$ (5th row).



(a) Layer 1 (b) Layer 6 (c) Layer 7 (d) Layer 10

Figure 4.45: 90% and 80% probability maps of sand channels (Case 2.2, $N_e = 500$): true (1st row); posterior, NTD, 1000, 90% prob. (2nd row); posterior, NTD, 1000, 80% prob. (3rd row).

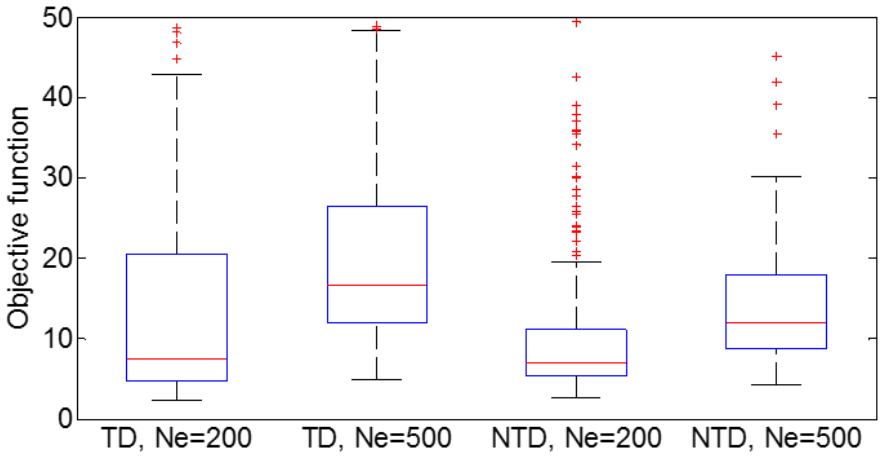


Figure 4.46: Normalized data mismatch (Case 2.2): "TD" denotes the "TD, 1000" method and "NTD" denotes the "NTD, 1000" method. Median (central red mark), 25th and 75th percentiles (bottom and top edges of the box), extreme data points (whiskers), and outliers ("+" symbol).

Table 4.23: Computational cost (Case 2.2)

Method	N_e	N_a	N_w	t^{total}
TD, 1000	200	8	13	3h 18m 37s
TD, 1000	500	8	13	7h 40m 47s
TD, 778	500	8	13	7h 26m 7s
TD, $20 \times 20 \times 5 \times 1000$	500	8	13	7h 5m 36s
TD, $37 \times 36 \times 8 \times 778$	500	8	13	7h 12m 23s
NTD, 1000	200	8	13	3h 20m 12s
NTD, 1000	500	8	13	7h 35m 24s
NTD, 778	500	8	13	7h 16m 29s
NTD, $20 \times 20 \times 5 \times 1000$	500	8	13	7h 3m 21s
NTD, $37 \times 36 \times 8 \times 778$	500	8	13	7h 23m 7s

Case 2.3: Heterogeneous Permeability, 9 Wells:

In previous case, we comprehensively compare the truncated two-mode and four-mode TD and NTD methods with the TD and NTD without truncation in terms of data match, uncertainty characterization of channel distributions and computational cost. To further demonstrate the findings obtained from previous case, we reduce the original number of wells from 13 to 9 in Case 2.3 and redo the case with the eight parameterization schemes (see Tables 4.16 and 4.17).

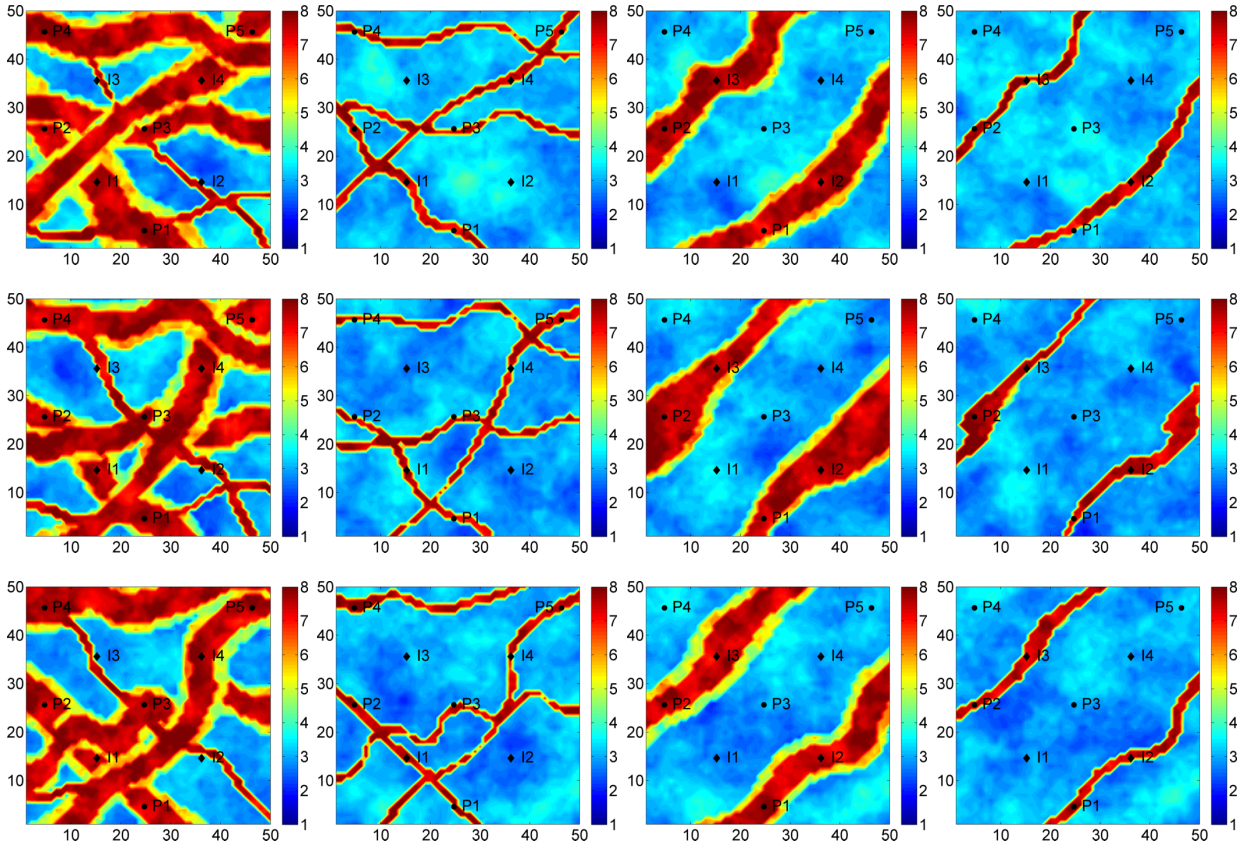
We still calibrate the facies distributions and heterogeneous permeability within facies simultaneously so the vector of model parameters is defined by

$$\boldsymbol{y} = [\boldsymbol{v}_{\text{facies}}^{\text{T}}, \boldsymbol{v}_{\text{sand}}^{\text{T}}, \boldsymbol{v}_{\text{levee}}^{\text{T}}, \boldsymbol{v}_{\text{shale}}^{\text{T}}]^{\text{T}}, \tag{4.46}$$

where $\boldsymbol{v}_{\text{facies}}$, $\boldsymbol{v}_{\text{sand}}$, $\boldsymbol{v}_{\text{levee}}$ and $\boldsymbol{v}_{\text{shale}}$ are the coefficient vectors corresponding to the facies fields and permeability fields of three facies, respectively. Note that the four training ensembles (i.e., one for facies and one for each of the permeability fields, k_{sand}, k_{levee} and k_{shale}, respectively) used for parameterization are exact the same as those in Case 2.2 and the size of each training ensemble is still 1000 ($N_{te} = 1000$). Besides, all other settings of history matching are the same as those in Case 2.2

The ensemble used for history matching consists of 500 heterogeneous reservoir models ($N_e = 500$) in this case and the true model is the same as the one in Case 2.2. As mentioned above, we only keep 4 injectors and 5 producers out of the original 13 wells in the reservoir, which poses additional challenges to the history matching work. From Fig. 4.47, we can see the locations of 9 wells, and the $\ln(k)$ fields of the true model and two prior realizations present high geological complexity associated with the facies distributions and heterogeneous permeability.

The large geological uncertainty of the prior ensemble can be illustrated from the prior production data shown in Fig. 4.48. Figures 4.49 and 4.50 present the posterior oil production rates obtained with the TD and NTD methods with and without truncation,



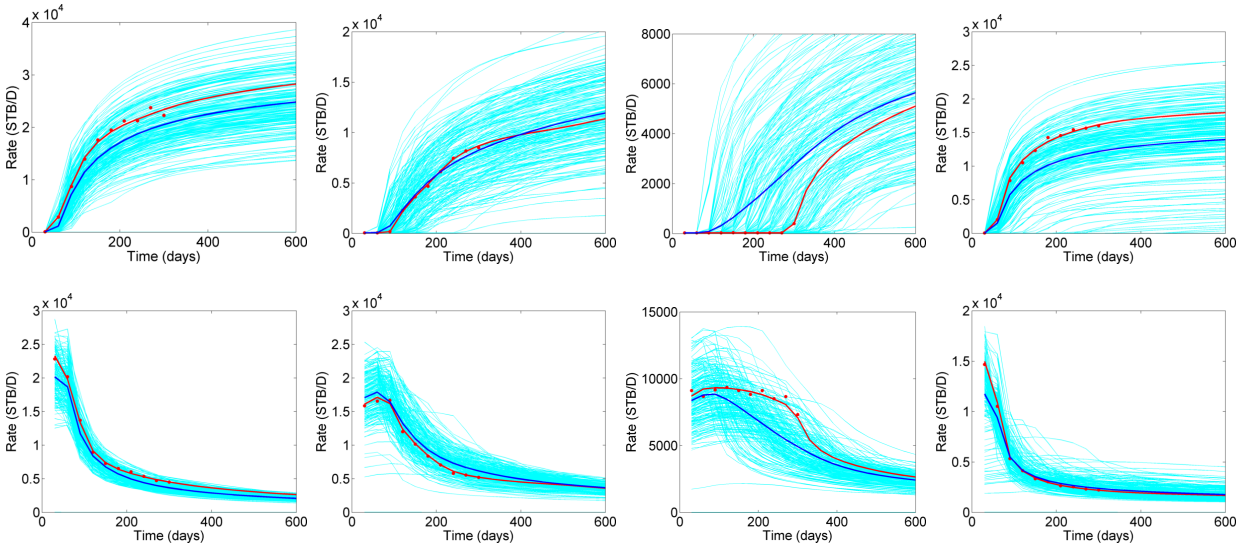
(a) Layer 1 (b) Layer 6 (c) Layer 7 (d) Layer 10

Figure 4.47: $\ln(k)$ fields of the true model and two prior realizations (Case 2.3, $N_e = 500$): true model (1st row), prior model #1 (2nd row) and prior model #2 (3rd row).

and it can be seen that the prior uncertainty has been reduced significantly through history matching. We are able to obtain desirable history data matches and reasonable production forecasts at most wells except for P4 where the oil rate predictions tend to be biased with different methods. The truncated TD and NTD perform similarly compared to the methods without truncation where we determine the ranks of different modes heuristically.

Figures 4.51 and 4.52 show the $\ln(k)$ fields of the true model and two posterior realizations obtained with the four-mode TD and NTD with and without truncation. We can see the key geological features in the true model are resolved through history matching but the continuity of thin channels in layer 6 is marred in most posterior realizations except for the first one obtained with the "TD, $37 \times 36 \times 8 \times 778$" method.

We present the 90% and 10% probability maps based on the posterior realizations

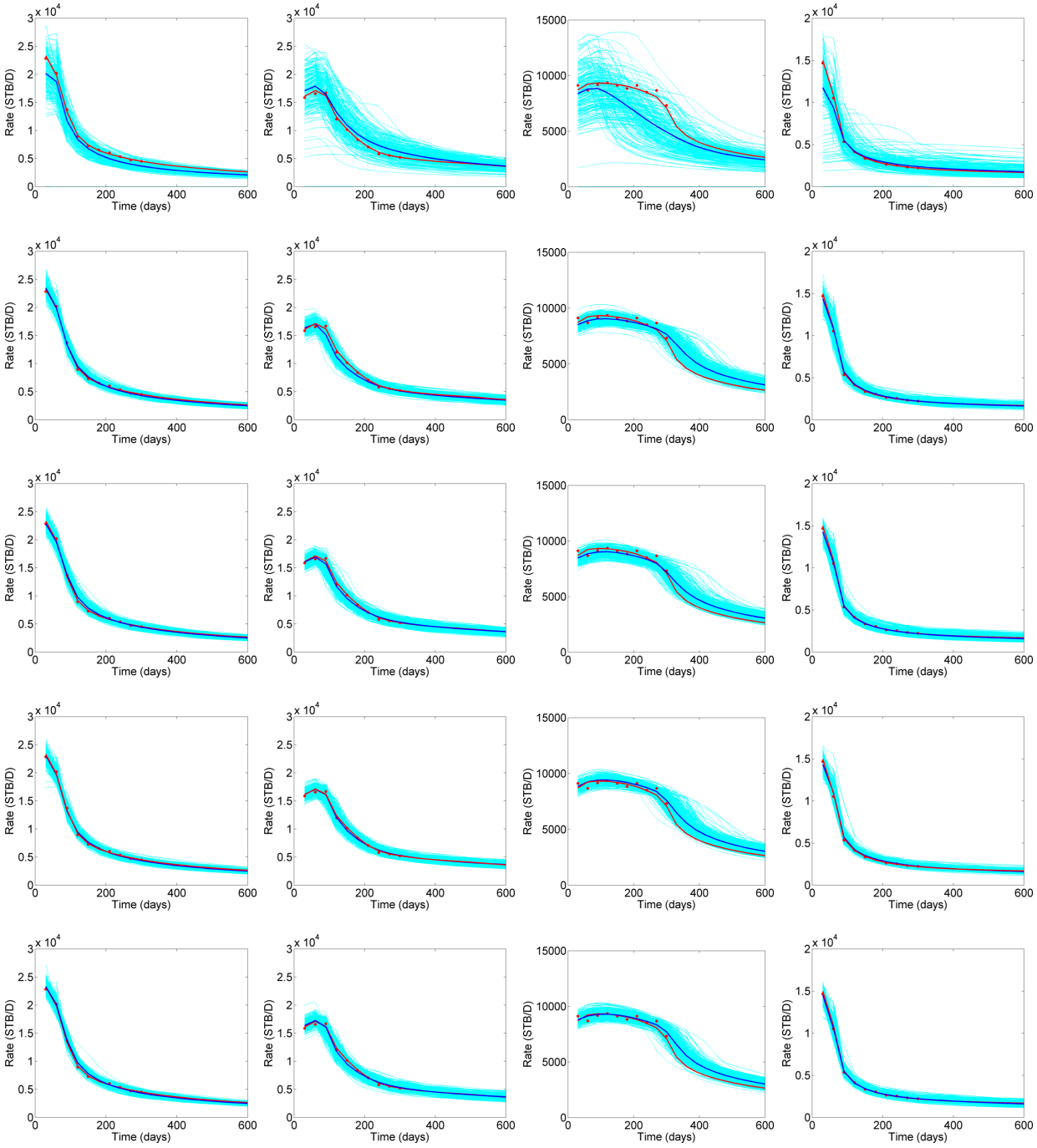


(a) P2 (b) P3 (c) P4 (d) P5

Figure 4.48: Prior production data (Case 2.3, $N_e = 500$): water rates (1st row) and oil rates (2nd row). True (red curves), observed data (red dots), simulated data of realizations (light blue curves), and ensemble mean (dark blue curves). History: $0 < t \leq 300$ days, forecast: $t > 300$ days.

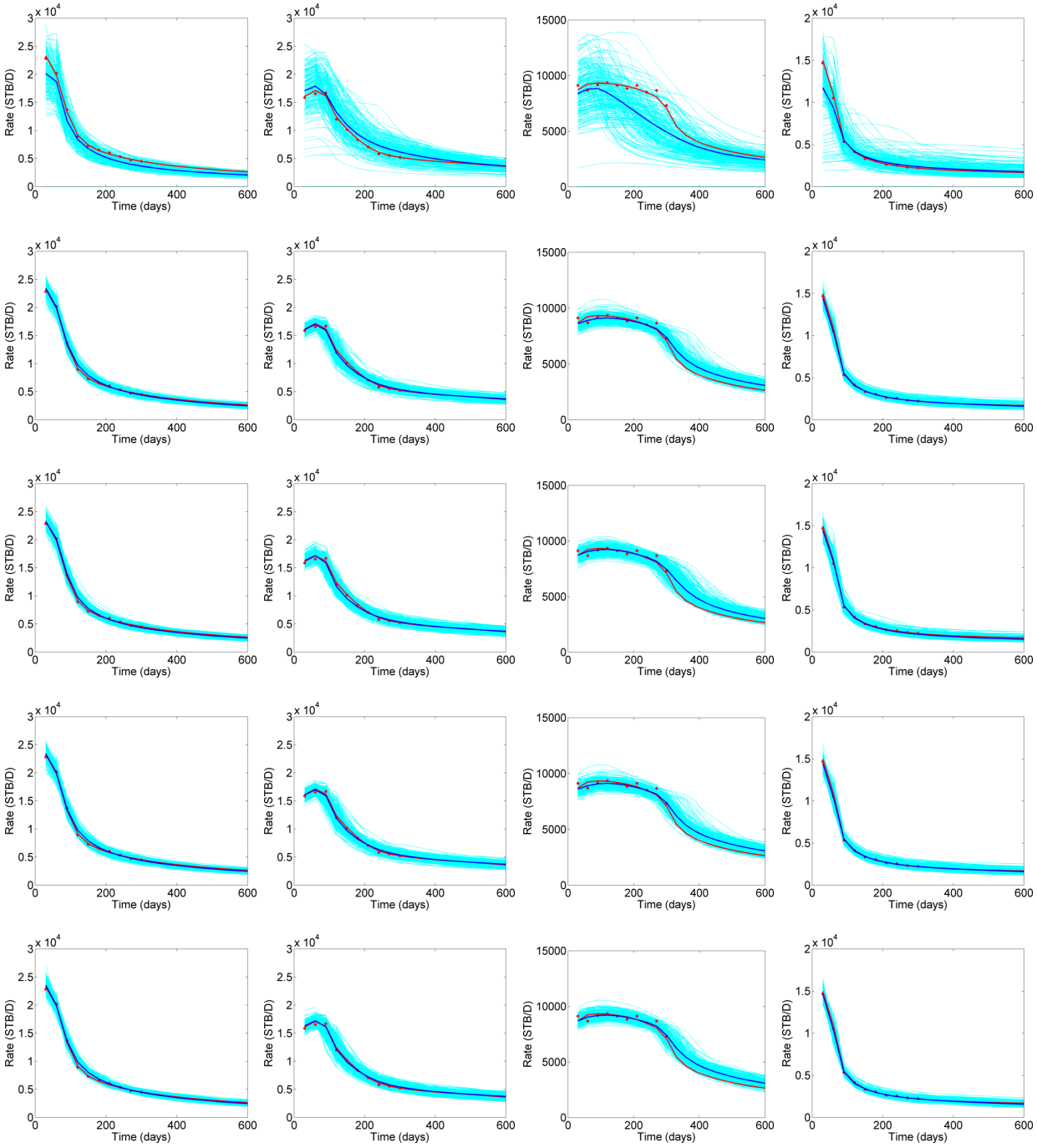
obtained with four-mode TD and NTD methods with and without truncation in Figs. 4.53 and 4.54. It is shown that the correctly resolved channel gridblocks (red points) are far more than the gridblocks that are incorrectly resolved to be channel facies (blue points) in all 90% probability maps. The spatial continuity of channels is slightly enhanced by the "NTD, $20 \times 20 \times 5 \times 1000$" and "NTD, $37 \times 36 \times 8 \times 778$" methods but based on the 10% probability maps, the NTD methods also give more blue points which represent the true channel gridblocks that are not resolved in 90% posterior realizations. Through visual comparison, the TD and NTD methods with and without truncation perform similarly in terms of uncertainty characterization of channel distributions. The statistical indicators calculated based on the 100%, 90%, 10% and 0% probability maps obtained with two-mode and four-mode TD and NTD methods with and without truncation are given in Tables 4.24 to 4.27.

Figure 4.55 shows the normalized data mismatch of Case 2.3 calculated by Eq. 4.40 and we can see the data match results obtained with the truncated TD and NTD methods are slightly better than those obtained with the methods without truncation.



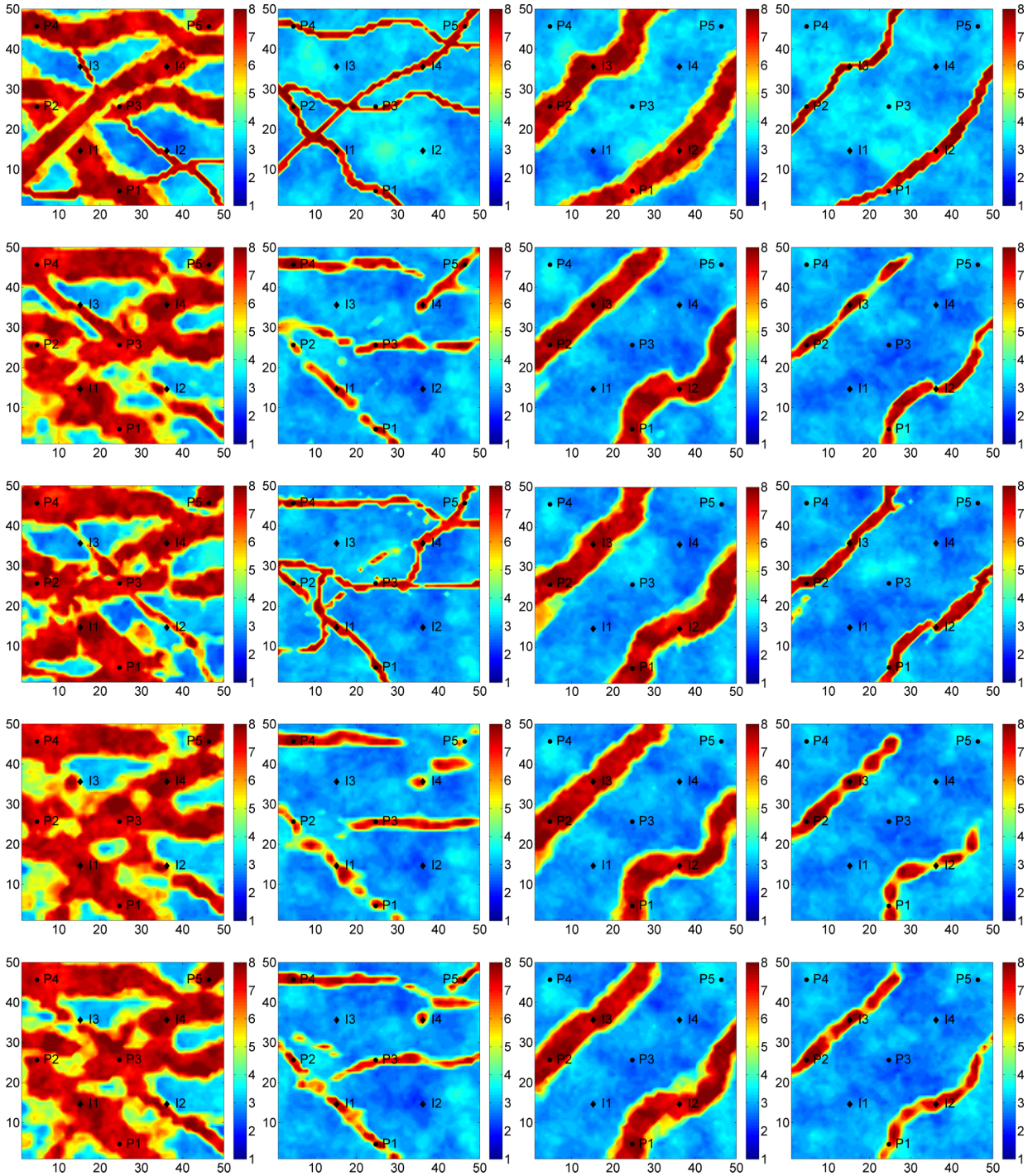
(a) P2 (b) P3 (c) P4 (d) P5

Figure 4.49: Prior and posterior oil rates (Case 2.3, $N_e = 500$): prior (1st row); posterior, TD, 1000 (2nd row); posterior, TD, 778 (3rd row); posterior, NTD, 1000 (4th row); posterior, NTD, 778 (5th row). True (red curves), observed data (red dots), simulated data of realizations (light blue curves), and ensemble mean (dark blue curves). History: $0 < t \leq 300$ days, forecast: $t > 300$ days.



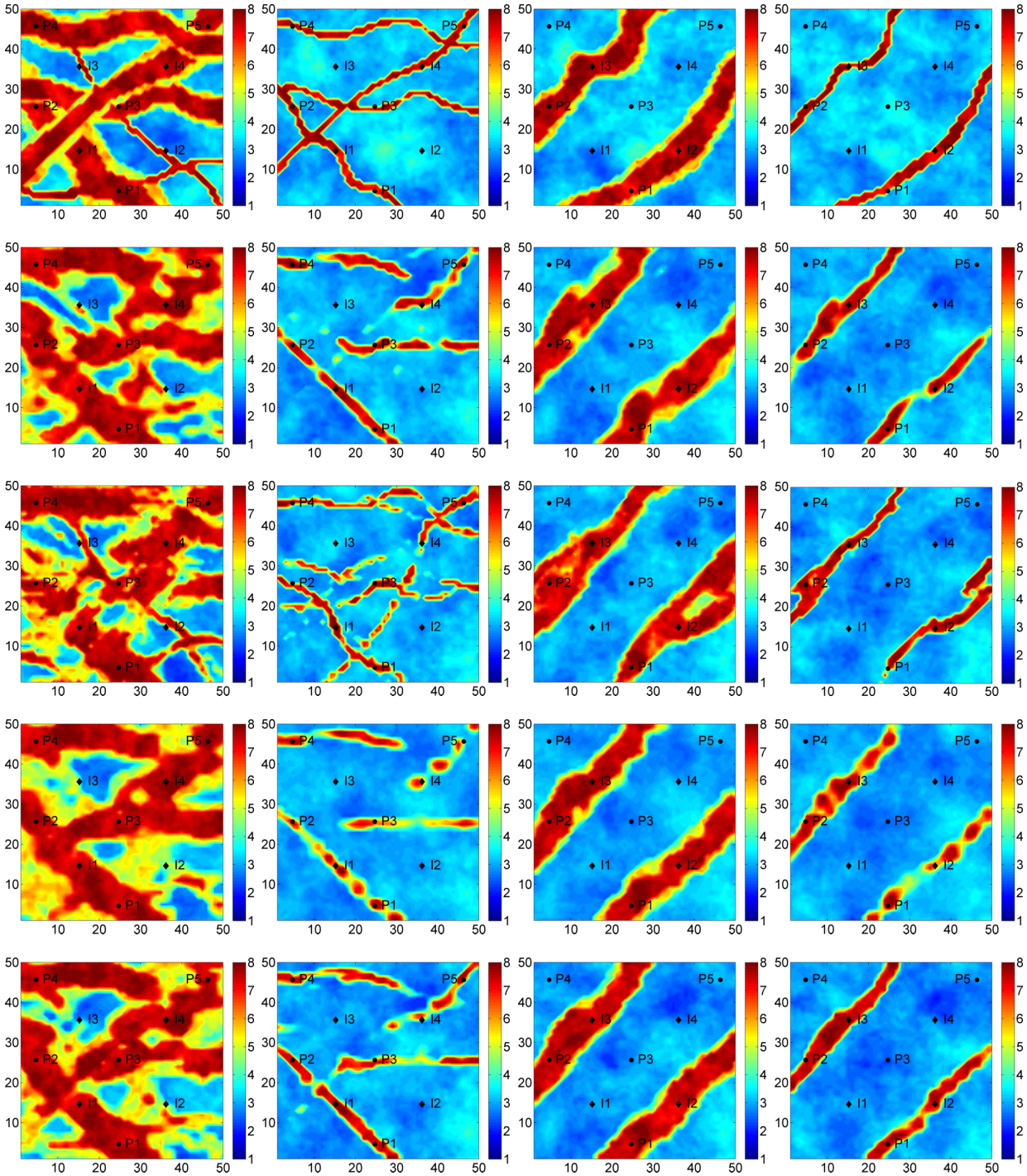
(a) P2 (b) P3 (c) P4 (d) P5

Figure 4.50: Prior and posterior oil rates (Case 2.3, $N_e = 500$): prior (1st row); posterior, TD, $20 \times 20 \times 5 \times 1000$ (2nd row); posterior, TD, $37 \times 36 \times 8 \times 778$ (3rd row); posterior, NTD, $20 \times 20 \times 5 \times 1000$ (4th row); posterior, NTD, $37 \times 36 \times 8 \times 778$ (5th row). True (red curves), observed data (red dots), simulated data of realizations (light blue curves), and ensemble mean (dark blue curves). History: $0 < t \leq 300$ days, forecast: $t > 300$ days.



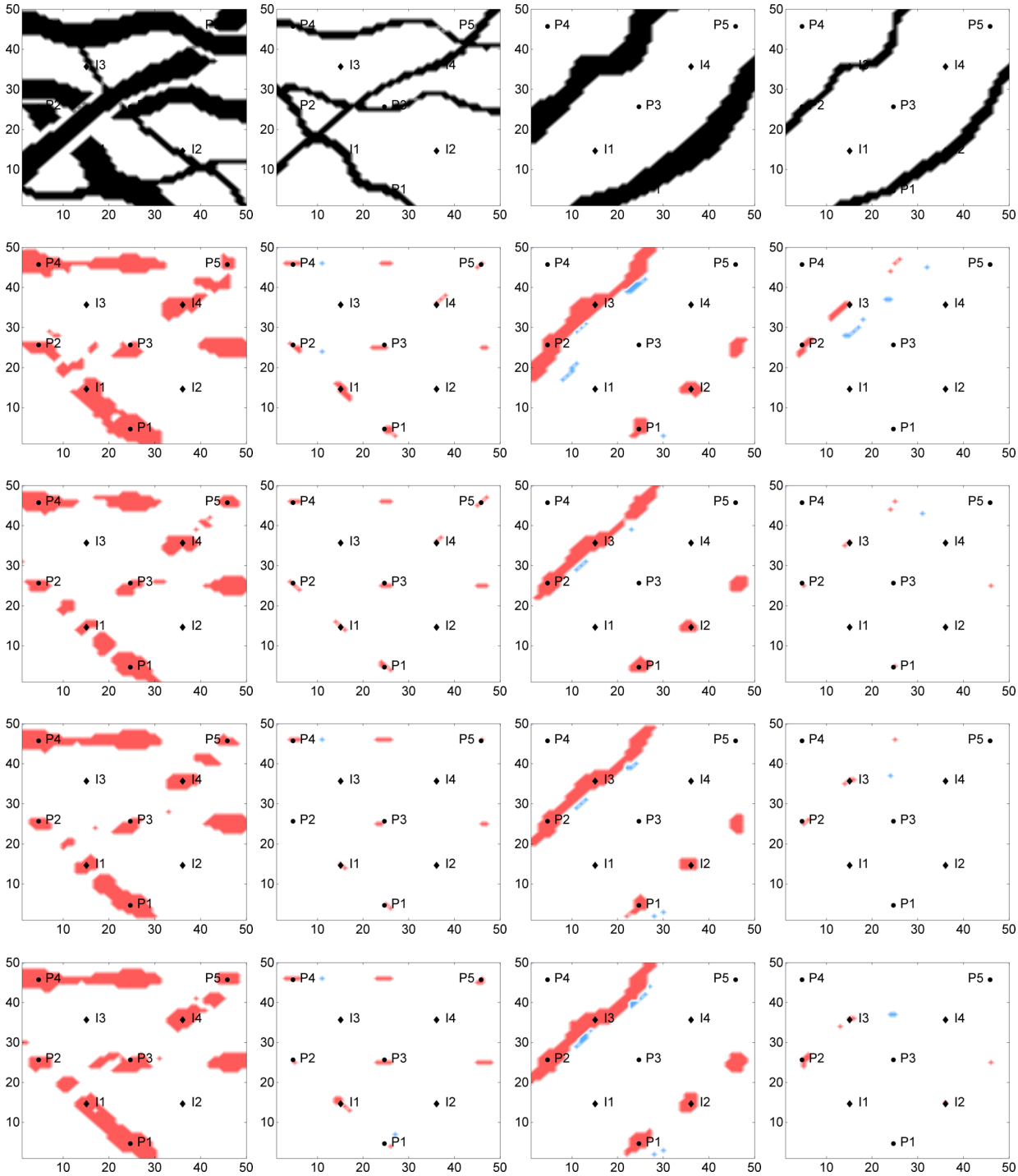
(a) Layer 1 (b) Layer 6 (c) Layer 7 (d) Layer 10

Figure 4.51: $\ln(k)$ fields of the true model and posterior realization #1 (Case 2.3, $N_e = 500$): true model (1st row); TD, $20 \times 20 \times 5 \times 1000$ (2nd row); TD, $37 \times 36 \times 8 \times 778$ (3rd row); NTD, $20 \times 20 \times 5 \times 1000$ (4th row); NTD, $37 \times 36 \times 8 \times 778$ (5th row).



(a) Layer 1 (b) Layer 6 (c) Layer 7 (d) Layer 10

Figure 4.52: $\ln(k)$ fields of the true model and posterior realization #2 (Case 2.3, $N_e = 500$): true model (1st row); TD, $20 \times 20 \times 5 \times 1000$ (2nd row); TD, $37 \times 36 \times 8 \times 778$ (3rd row); NTD, $20 \times 20 \times 5 \times 1000$ (4th row); NTD, $37 \times 36 \times 8 \times 778$ (5th row).



(a) Layer 1 (b) Layer 6 (c) Layer 7 (d) Layer 10

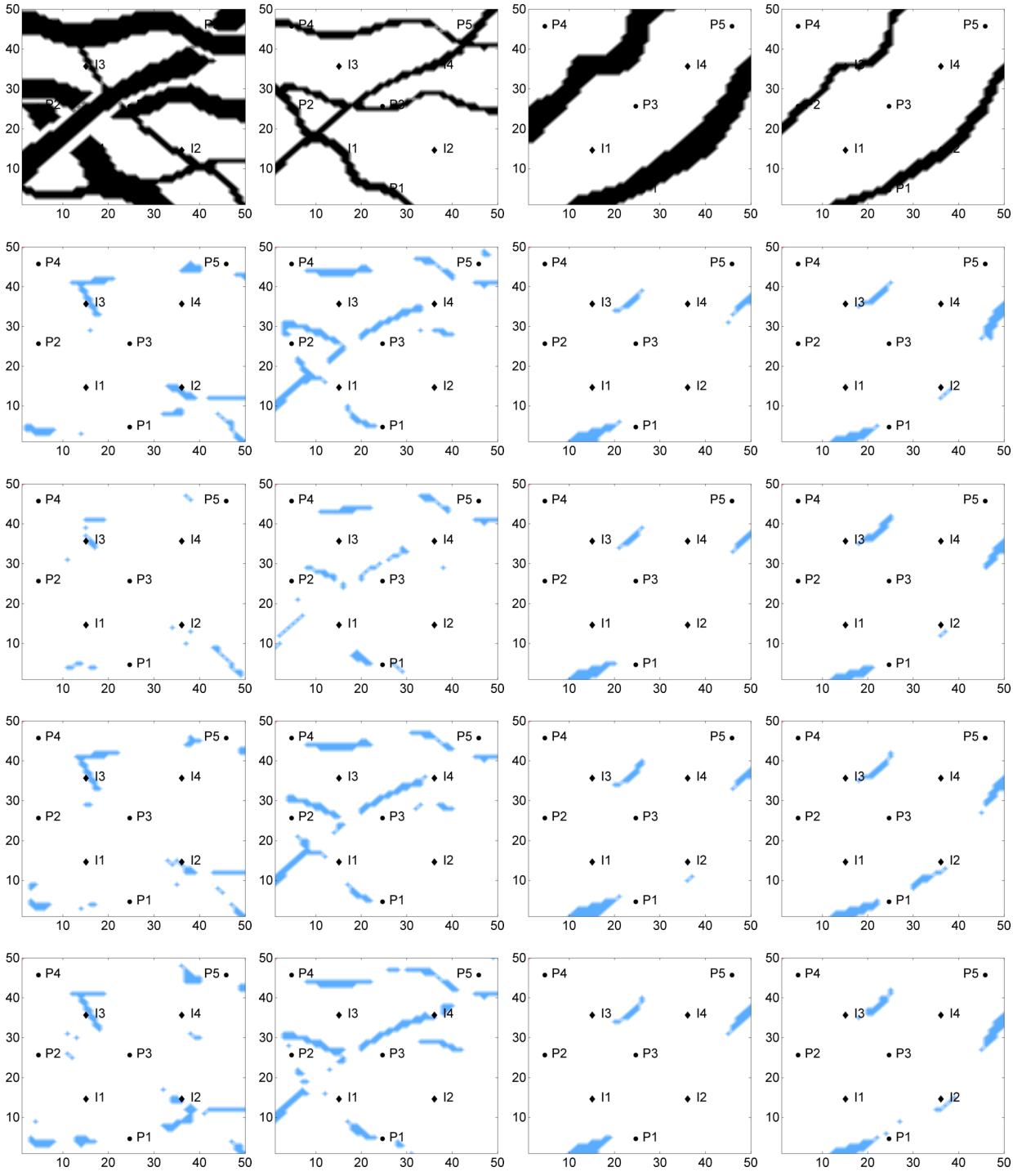
Figure 4.53: 90% probability maps of sand channels (Case 2.3, $N_e = 500$): true (1st row); posterior, TD, $20 \times 20 \times 5 \times 1000$ (2nd row); posterior, TD, $37 \times 36 \times 8 \times 778$ (3rd row); posterior, NTD, $20 \times 20 \times 5 \times 1000$ (4th row); posterior, NTD, $37 \times 36 \times 8 \times 778$ (5th row).

Table 4.24: Statistical indicators based on 100% and 90% probability maps obtained with two-mode TD and NTD (Case 2.3)

Method	N_e	N_w	Prob.(%)	N_{gc}^{true}	$N_{gc}^{\text{posterior}}$	N_{gc}^{good}	$R^{\text{good}}(\%)$
TD, 1000	500	9	100%	6104	101	101	100.0%
TD, 1000	500	9	90%	6104	1546	1442	93.3%
TD, 778	500	9	100%	6104	97	97	100.0%
TD, 778	500	9	90%	6104	1540	1459	94.7%
NTD, 1000	500	9	100%	6104	111	111	100.0%
NTD, 1000	500	9	90%	6104	1553	1443	92.9%
NTD, 778	500	9	100%	6104	199	199	100.0%
NTD, 778	500	9	90%	6104	1771	1631	92.1%

Table 4.25: Statistical indicators based on 100% and 90% probability maps obtained with four-mode TD and NTD (Case 2.3)

Method	N_e	N_w	Prob.(%)	N_{gc}^{true}	$N_{gc}^{\text{posterior}}$	N_{gc}^{good}	$R^{\text{good}}(\%)$
TD, $20 \times 20 \times 5 \times 1000$	500	9	100%	6104	90	90	100.0%
TD, $20 \times 20 \times 5 \times 1000$	500	9	90%	6104	1612	1483	91.9%
TD, $37 \times 36 \times 8 \times 778$	500	9	100%	6104	87	87	100.0%
TD, $37 \times 36 \times 8 \times 778$	500	9	90%	6104	1545	1470	95.1%
NTD, $20 \times 20 \times 5 \times 1000$	500	9	100%	6104	382	380	99.5%
NTD, $20 \times 20 \times 5 \times 1000$	500	9	90%	6104	2009	1785	88.9%
NTD, $37 \times 36 \times 8 \times 778$	500	9	100%	6104	303	301	99.3%
NTD, $37 \times 36 \times 8 \times 778$	500	9	90%	6104	1878	1688	89.9%



(a) Layer 1 (b) Layer 6 (c) Layer 7 (d) Layer 10

Figure 4.54: 10% probability maps of sand channels (Case 2.3, $N_e = 500$): true (1st row); posterior, TD, $20 \times 20 \times 5 \times 1000$ (2nd row); posterior, TD, $37 \times 36 \times 8 \times 778$ (3rd row); posterior, NTD, $20 \times 20 \times 5 \times 1000$ (4th row); posterior, NTD, $37 \times 36 \times 8 \times 778$ (5th row).

Table 4.26: Statistical indicators based on 0% and 10% probability maps obtained with two-mode TD and NTD (Case 2.3)

Method	N_e	N_w	Prob.(%)	N_{gc}^{true}	N_{gc}^{missed}	$R^{\text{missed}}(\%)$
TD, 1000	500	9	0%	6104	0	0.00%
TD, 1000	500	9	10%	6104	401	6.57%
TD, 778	500	9	0%	6104	0	0.00%
TD, 778	500	9	10%	6104	362	5.93%
NTD, 1000	500	9	0%	6104	0	0.00%
NTD, 1000	500	9	10%	6104	540	8.85%
NTD, 778	500	9	0%	6104	8	0.13%
NTD, 778	500	9	10%	6104	518	8.49%

Table 4.27: Statistical indicators based on 0% and 10% probability maps obtained with four-mode TD and NTD (Case 2.3)

Method	N_e	N_w	Prob.(%)	N_{gc}^{true}	N_{gc}^{missed}	$R^{\text{missed}}(\%)$
TD, $20 \times 20 \times 5 \times 1000$	500	9	0%	6104	12	0.19%
TD, $20 \times 20 \times 5 \times 1000$	500	9	10%	6104	686	11.24%
TD, $37 \times 36 \times 8 \times 778$	500	9	0%	6104	2	0.03%
TD, $37 \times 36 \times 8 \times 778$	500	9	10%	6104	468	7.67%
NTD, $20 \times 20 \times 5 \times 1000$	500	9	0%	6104	90	1.47%
NTD, $20 \times 20 \times 5 \times 1000$	500	9	10%	6104	1017	16.66%
NTD, $37 \times 36 \times 8 \times 778$	500	9	0%	6104	67	1.09%
NTD, $37 \times 36 \times 8 \times 778$	500	9	10%	6104	773	12.66%

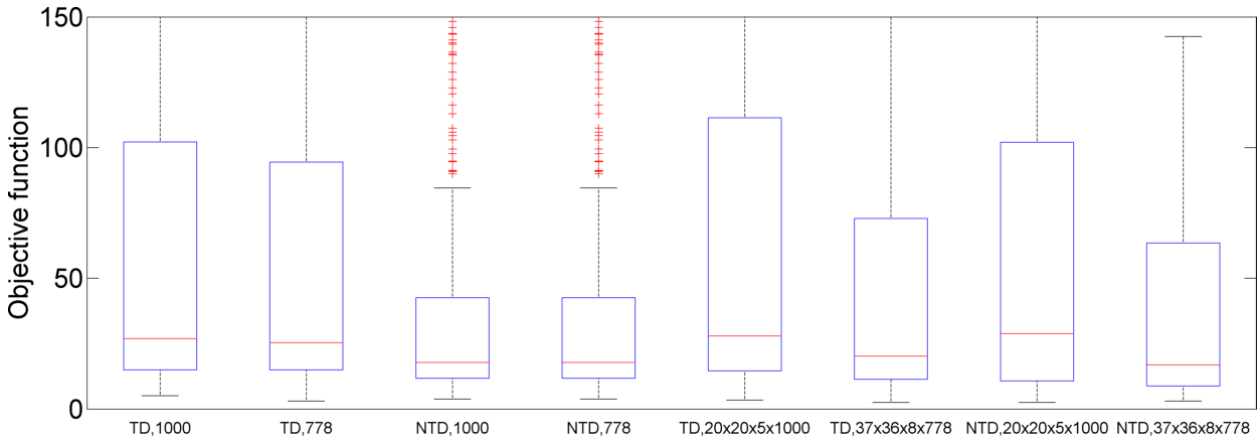


Figure 4.55: Normalized data mismatch (Case 2.3, $N_e = 500$). Median (central red mark), 25th and 75th percentiles (bottom and top edges of the box), extreme data points (whiskers), and outliers ("+" symbol).

CHAPTER 5

ASSISTED HISTORY MATCHING OF CHANNELIZED RESERVOIRS USING THE OBJECT-BASED MODEL IN PETREL

In Chapters 2 to 4, we proposed and investigated three different parameterization approaches for history matching of complex geological reservoirs. With the help of a post-processing procedure, large-scale geological structures can be resolved reasonably well in the posterior realizations but sometimes the spatial continuity of channelized facies is not completely preserved. To completely preserve the spatial continuity of channel facies distribution represented by object-based modeling, it seems that one should simply adjust by history matching the parameters used to generate a realization of a channelized system by object-based modeling. Although this seems like an obvious approach, it turns out that extremely small changes in some of the parameters used in object-based modeling with Petrel lead to an extremely large change in the facies distribution, i.e., the object-based realization is not a continuous or stable function of the parameters. Thus, what we believed would be a good approach for history matching of object-based models failed. This is somewhat surprising because in a far simple setting, Zhang et al. (2002) used parameters that define a channel structure successfully in history matching. In particular, Zhang et al. (2002) conditioned a stochastic channel embedded in a background facies to observed data at wells. In their work, the model parameters consist of geometric parameters that describe the shape, size, location of the channel, as well as the permeability and porosity in the channel and non-channel facies. This disabling limitation of the work is that it is not clear how to generalize the stochastic model to a practical case which involves multiple facies and complex distributions of multiple channels.

Although the multi-point statistics (MPS) is a popular modeling technique used in

such integrated approach (Caers and Hoffman, 2006; Jafapour and Khodabakhsi, 2011; Tavakoli et al., 2014; Le et al., 2015b; Sebacher et al., 2016), it still remains challenging to obtain a suitable and reliable training image for various types of geological environments. Besides, the MPS is highly demanding on the conditioning capability of soft data. Even if a probability map is given its maximum allowable weights when input to standard MPS algorithm to regenerate the channel, one may still obtain a channel structure which is obviously inconsistent with the probability map. Therefore, it seems preferable to employ the object-based simulation technique which appears to be more straightforward and reliable as the modeling tool.

The remainder of this chapter is organized as follows: we first briefly review the parameters for the object-based model in Petrel. In the next section, the ensemble-based method for calibrating the model parameters is briefly presented. Then, we provide the sensitivity analysis of the model parameters with different perturbation sizes. The computational results of three synthetic cases are shown in the subsequent section to demonstrate the performance of the integrated history matching workflow with object-based modeling method for complex channelized reservoirs. Since the proposed methodology is only partially successful for specific cases, further discussion is given in the last part to explain the probable reasons.

5.1 Methodology

5.1.1 Adaptive Channel Model in Petrel

In this chapter, we use the object-based modeling method to generate prior channelized reservoir models for history matching. In the corresponding module of Petrel, three main types of geologic objects can be characterized and their basic properties are summarized in Table 5.1.

As the largest structures of directional significance with preservation potential, the channels usually are characterized by a sinuous geometry and the levees deposit along the

Table 5.1: Objects in the adaptive channel model of Petrel

Object	Facies	Reservoir quality	Perm & Poro
Channel	Sand	Good	High
Levee	Levee	Moderate	Moderate
Floodplain	Shale	Poor	Low

channel margins. The low permeable background floodplain (usually shale) is viewed as the matrix within which the channels and levees are embedded. The spatial distribution and geological properties of channels and levees are described by eight stochastic parameters (orientation, amplitude, wavelength, sinuosity, width of channel, thickness of channel, width of levee and thickness of levee) and a particular realization is generated starting from a random seed in Petrel. There are three basic ways in Petrel to set these parameters in order to generate desired models:

- Channel layout (shown in Fig. 5.1): orientation θ (unit: degrees), amplitude a (unit: ft), wavelength λ (unit: ft) and sinuosity s (unit: /). Note that the symbol "/" indicates that the parameter is dimensionless. The orientation is used to specify the average azimuth of channels. The amplitude defines the average transversal distance covered during one wavelength and the wavelength is the average distance between two consecutive same handed channel turns; see Fig. 5.1. The sinuosity describes the width uniformity of sand facies along the channel and more detailed explanations about this concept are given later on.

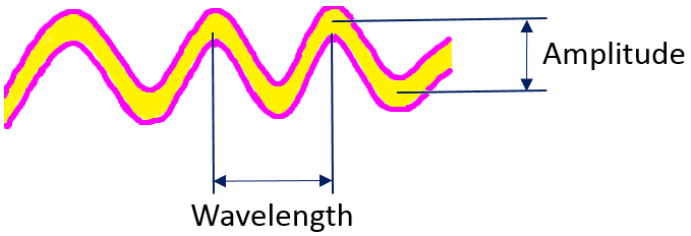


Figure 5.1: Channel layout schematic: sand facies (yellow) and levee facies (pink).

- Channel cross-section (shown in Fig. 5.2): width w_{sand} (unit: ft) and thickness h_{sand} (unit: ft) of channel (sand) facies. The cross-section design of an idealized channel

model in most commercial software packages is based on width and thickness, following a perfect symmetrical half lobe. In general, the width and thickness denotes the average channel width and average channel thickness or depth, respectively.

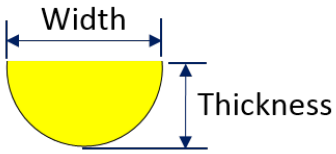


Figure 5.2: Cross-section schematic of channel: sand facies (yellow).

- Levee cross-section (shown in Fig. 5.3): (relative) width w_{levee} (unit: /) and (relative) thickness h_{levee} (unit: /) of levee facies. Similarly for levees, the shape is also a half lobe that is located in the upper part surrounding the channel. The levee width denotes the average levee width on each side of the channel as a fraction of the channel width. The thickness of levee is also defined as a fraction of channel thickness in Petrel.

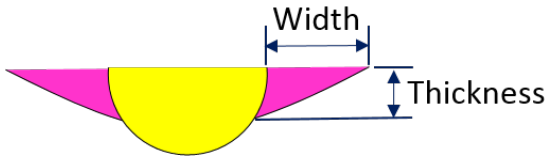


Figure 5.3: Cross-section schematic of levee: sand facies (yellow) and levee facies (pink).

It is noteworthy that the eight parameters introduced above could follow any specific statistical distribution (e.g., deterministic, uniform, triangular, normal or truncated-normal) with given attributes. For example, one can keep a parameter fixed at a given value by specifying it with a deterministic input. Once the values of the parameters are set, a sequence of building blocks are inserted with a channel/levee part following the given parameters, comprising a distribution function and the random seed which is used to set the starting point for the channel/levee insertion. If well logs or other hard data information are available and required to be honored in the geological modeling, the first channel block is positioned to honor a given well until it reaches facies observation at another well. Thereafter, certain building blocks are interactively inserted following horizontal and areal search schemes in

order to match the hard data at all the wells. The same procedure is followed for the levee facies. After all well data have been honored, the geological objects are distributed either by random or by following trends, until all the facies volume proportions are satisfied.

In this work, we assume the parameters for object-based modeling are subject to normal distribution with the mean and standard deviation (STD) given in Table 5.2. Before discussing data assimilation algorithm, we further investigate the effect of sinuosity and the random seed on the outcome of object-based modeling. In the adaptive channel model in Petrel, the sinuosity parameter controls how contorted the resulting channel is. Its value is restricted to be within 0 and 1, and very high values of this parameter (close to 1) result in shapes that can be too contorted to represent a real channel while lower values can be used to produce channels with braided or meandering characteristics. Therefore, the most useful range is suggested to be $[0.1, 0.4]$. Here, we generate a series of 2D channelized reservoir models defined on 50×50 grid system with three facies (see the facies indicators and color schemes in Table 5.3) by feeding different sinuosity values into Petrel while keeping the other parameters and the random seed fixed. Figure 5.4 shows the facies fields obtained with eight different sinuosity values ranging from 0.05 to 0.4 with all other parameters fixed at their mean values given in Table 5.2. It can be seen that as the sinuosity increases, the shape of the channels becomes more nonuniform, and the width of sand/levee facies varies more obviously along the same channel, especially when the value of sinuosity is greater than 0.3. But if we set the sinuosity to be smaller than 0.2, it is hard to tell the difference of channels as sinuosity changes.

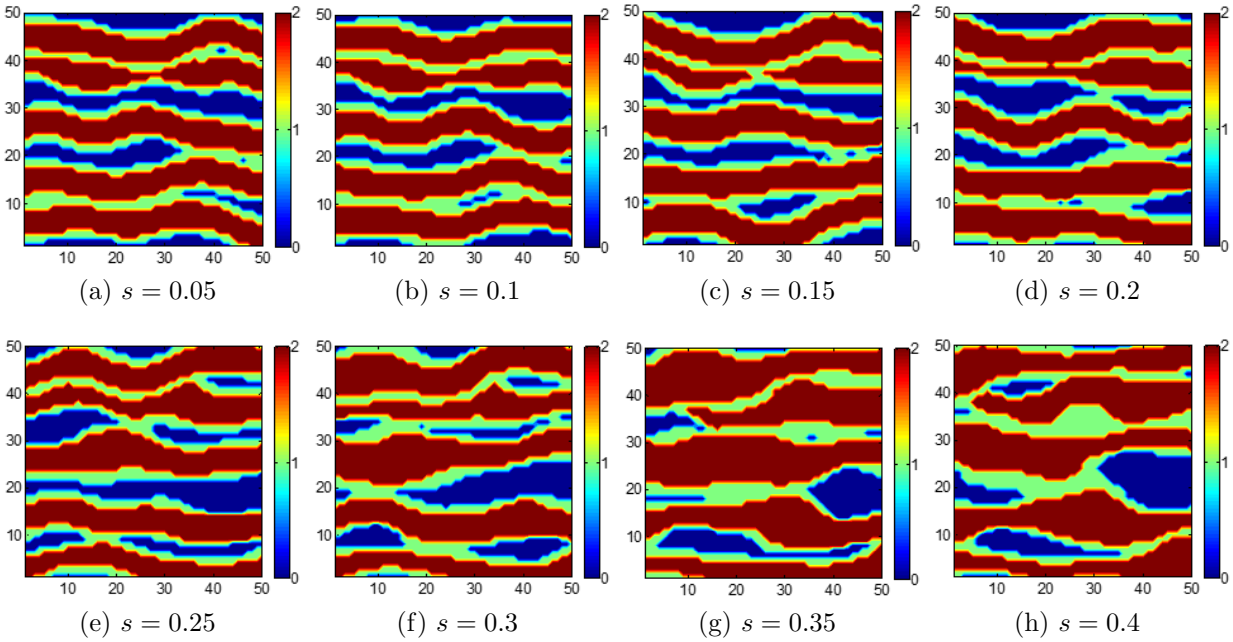
We also investigate the effect of the random seed on the resulting facies fields by varying the random seed r and fixing all other geological parameters at their mean values given in Table 5.2. The results are shown in Fig. 5.5 and represent the impact of the initial channel/levee insertion on the final facies distribution. Due to its effect and non-geological meaning, the random seed is excluded from the model parameters of history matching by assigning it a constant value in Petrel. This ensures that the change in the predicted data obtained by varying a parameter will not be simply due to a change in the random seed.

Table 5.2: Parameters for object-based modeling

Name	Notation	Unit	Mean (μ)	STD (σ)
orientation	θ	degrees	90	10
amplitude	a	ft	500	50
wavelength	λ	ft	3000	300
sinuosity	s	/	0.3	0.05
width of sand channel	w_{sand}	ft	600	50
thickness of sand channel	h_{sand}	ft	25	5
width of levee	w_{levee}	/	0.35	0.05
thickness of levee	h_{levee}	/	0.7	0.05

Table 5.3: Facies indicators and color schemes

Facies	Indicator	Color
Shale	0	Blue
Levee	1	Green
Sand	2	Red



(a) $s = 0.05$ (b) $s = 0.1$ (c) $s = 0.15$ (d) $s = 0.2$

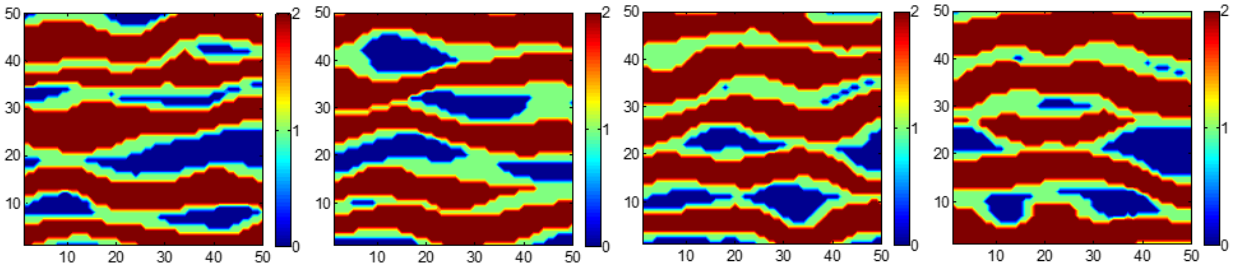
(e) $s = 0.25$ (f) $s = 0.3$ (g) $s = 0.35$ (h) $s = 0.4$

Figure 5.4: Effect of sinuosity value with all other parameters held fixed at their mean values given in Table 5.2

Therefore, the vector of parameters for object-based modeling, $\boldsymbol{m}_{\text{obj}}$, can be written as

$$\boldsymbol{m}_{\text{obj}} = [\theta, a, \lambda, s, w_{\text{sand}}, h_{\text{sand}}, w_{\text{levee}}, h_{\text{levee}}]^{\text{T}}. \tag{5.1}$$

It should be mentioned that since we assume homogeneous petrophysical properties (e.g., permeability) within each facies, the property fields for reservoir simulator are obtained by assigning specific permeability value to each gridblock according to its facies type. Consequently, the permeability values that are updated through history matching are not needed in the object-based modeling.



(a) $r = 100$ (b) $r = 200$ (c) $r = 300$ (d) $r = 400$

Figure 5.5: Effect of random seed value with all other parameters held fixed at their mean values given in Table 5.2

5.1.2 Data Assimilation Algorithm

To calibrate the spatial channel/levee distribution and petrophysical properties within facies, the vector of model parameters in history matching, $\boldsymbol{m}$, is defined as

$$\boldsymbol{m} = [\theta, a, \lambda, s, w_{\text{sand}}, h_{\text{sand}}, w_{\text{levee}}, h_{\text{levee}}, k_{\text{sand}}, k_{\text{levee}}, k_{\text{shale}}]^{\text{T}}, \tag{5.2}$$

where $k_{\text{sand}}, k_{\text{levee}}$ and k_{shale} denotes the log-permeability values of the sand, levee and shale facies, respectively. The statistical features of the model parameters are presented in Table 5.4.

Regarding the data assimilation method, we still use the ensemble smoother with multiple data assimilation (ES-MDA) algorithm developed by Emerick and Reynolds (2012, 2013a) and the inflation factors are calculated by the geometric method proposed by Rafiee and Reynolds (2017). The details of the ES-MDA-GEO algorithm can be found in Appendix A.1. After each data assimilation step, the parameters updated by ES-MDA-GEO are provided to the object-based method in Petrel in order to generate a new ensemble of

Table 5.4: Model parameters in history matching

Name	Notation	Unit	Mean (μ)	STD (σ)
orientation	θ	degrees	90	10
amplitude	a	ft	500	50
wavelength	λ	ft	3000	300
sinuosity	s	/	0.3	0.05
width of sand channel	w_{sand}	ft	600	50
thickness of sand channel	h_{sand}	ft	25	5
width of levee	w_{levee}	/	0.35	0.05
thickness of levee	h_{levee}	/	0.7	0.05
log-permeability of sand	k_{sand}	mD	ln(2000)	0.3
log-permeability of levee	k_{levee}	mD	ln(200)	0.3
log-permeability of shale	k_{shale}	mD	ln(20)	0.3

realizations for next iteration step.

5.2 Sensitivity Analysis

Although ES-MDA does not explicitly estimate and use a gradient, a gradient approximation is implicitly represented in the dimensionless sensitivity matrix, $\boldsymbol{G}_D^{\ell}$ (defined in Eq. A.6); see Reynolds et al. (2006). To do the sensitivity analysis, we first define the normalized data mismatch by

$$O(\boldsymbol{m}_i) = \frac{1}{N_d} \left\| \boldsymbol{d}_i(\boldsymbol{m}_i) - \boldsymbol{d}_{\text{base}}(\boldsymbol{m}_{\text{base}}) \right\|_2 , \tag{5.3}$$

where $\boldsymbol{m}_{\text{base}}$ is a N_m-dimensional vector defined in Eq. 5.2 and every parameter in $\boldsymbol{m}_{\text{base}}$ takes the mean value given in Table 5.4 as its base case. To obtain the perturbed vector of model parameters, $\boldsymbol{m}_i$, we have

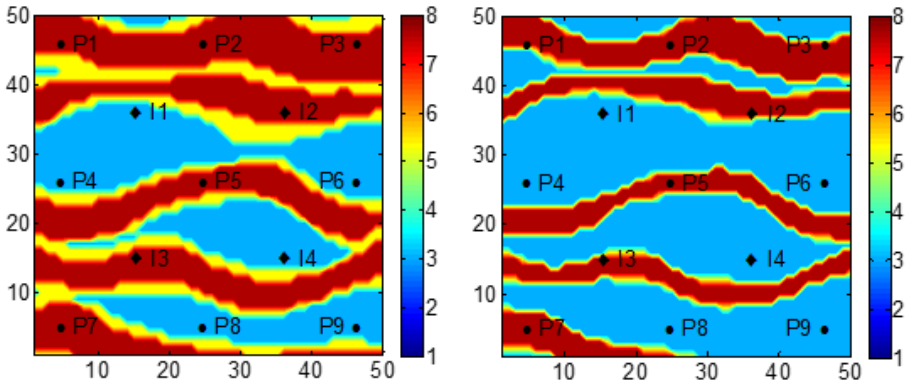
$$\boldsymbol{m}_i = [m_{\text{base},1}, \cdots, m_{\text{base},i} + \delta m_i, \cdots, m_{\text{base},N_m}] , \quad \text{with } \delta m_i = \beta m_{\text{base},i}, \tag{5.4}$$

where $m_{\text{base},i}$ is the ith parameter in the vector $\boldsymbol{m}_{\text{base}}$; β denotes the perturbation size which

is set to be 0.1%, 0.5%, 1% and 5% of the base value in this analysis. Therefore, we can analyze the sensitivity of production data to the model parameters using the finite difference approximation as

$$\frac{\partial O(\boldsymbol{m}_i)}{\partial \boldsymbol{m}_i} = \frac{\frac{1}{N_d}\left\|\boldsymbol{d}_i(\boldsymbol{m}_i) - \boldsymbol{d}_{\text{base}}(\boldsymbol{m}_{\text{base}})\right\|_2}{\delta m_i}. \tag{5.5}$$

In Fig. 5.6, the base model generated with $\boldsymbol{m}_{\text{base}}$ is the three-facies (shale, levee and sand) channelized reservoir consisting of two layers shown in Fig. 5.6 where this model honors the hard data information given in Table 5.5.



(a) Layer 1 (b) Layer 2

Figure 5.6: $\ln(k)$ fields of the base case

Table 5.5: Hard data at wells

	I1	I2	I3	I4	P1	P2	P3	P4	P5	P6	P7	P8	P9
Layer 1	/	sand	sand	/	sand	sand	sand	/	sand	/	sand	/	/
Layer 2	/	sand	sand	/	sand	sand	sand	/	sand	/	sand	/	/

Each layer in the base model is defined on a 50×50 grid system with gridblock size of 100ft $\times$ 100ft $\times$ 20ft. We can see that since the thickness of the channel is set equal to the mean value of 25 ft which is greater than the height of one reservoir layer, the two layers actually belong to the same geological system. That is to say, the channels in the second layer are actually the lower parts of the channels in the top layer. There are four injectors (I1 to I4) and nine producers (P1 to P9) arranged in a five-spot pattern in the reservoir. All vertical wells perforate the two layers and operate at a specified flowing bottom hole pressure

(BHP). To be specific, the injectors and producers, respectively, operate at a constant BHP of 5500 psi and 3000 psi. The initial pressure of the reservoir is 5000 psi and the initial water saturation is 0.2. The production period is 600 days and the predicted data consist of the water injection rate (q_{inj}) at the injectors and the oil and water production rate (q_{o} and q_{w}) at the producers. The data are collected every 30 days during the production period in order to form the data vector given by

$$\boldsymbol{d} = \left[\boldsymbol{q}_{\text{inj}}^{\text{T}}, \boldsymbol{q}_{\text{o}}^{\text{T}}, \boldsymbol{q}_{\text{w}}^{\text{T}}\right]_{N_d \times 1}^{\text{T}}, \tag{5.6}$$

and $\boldsymbol{d}_{\text{base}}$ and $\boldsymbol{d}_i$ in Eq. 5.5 denote the predicted data corresponding to $\boldsymbol{m}_{\text{base}}$ and $\boldsymbol{m}_i$, respectively.

Using Eq. 5.4, the perturbation is applied to the parameter directly so this is referred to as the "no scaling" scheme. However, we can observe that the magnitude of various parameters differs from each other significantly which can give rise to a poorly scaled problem and lead to relatively poor estimates of some of the derivatives involved in the gradient. In an attempt to mitigate this issue, we map all the parameters to the range of $[0, 1]$ and repeat the same sensitivity analysis in order to compare the performance of the two schemes. For the ith element of the vector $\boldsymbol{m}$, m_i, it is scaled to the range of $[0, 1]$ by using

$$\hat{m}_i = \frac{m_i - m_i^{\text{low}}}{m_i^{\text{up}} - m_i^{\text{low}}}, \tag{5.7}$$

where $m_i^{\text{up}} = \mu_i + 3\sigma_i$ and $m_i^{\text{low}} = \mu_i - 3\sigma_i$, which are calculated based on the mean and standard deviation given in Table 5.4. Consequently, Eqs. 5.5 and 5.4 can be rewritten as

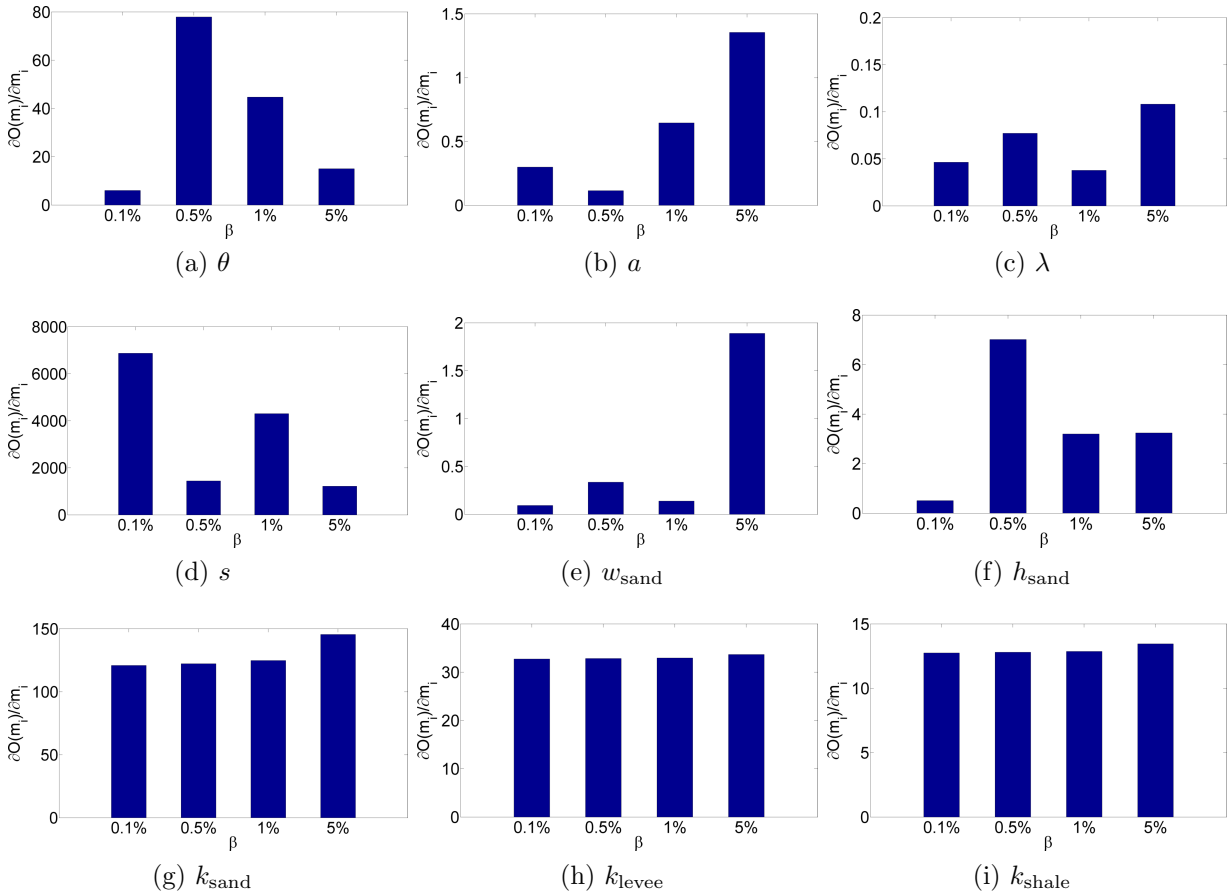
$$\frac{\partial O(\hat{\boldsymbol{m}}_i)}{\partial \hat{\boldsymbol{m}}_i} = \frac{\frac{1}{N_d}\left\|\boldsymbol{d}_i(\hat{\boldsymbol{m}}_i) - \boldsymbol{d}_{\text{base}}(\hat{\boldsymbol{m}}_{\text{base}})\right\|_2}{\delta \hat{m}_i}, \tag{5.8}$$

$$\hat{\boldsymbol{m}}_i = \left[\hat{m}_{\text{base},1}, \cdots, \hat{m}_{\text{base},i} + \delta \hat{m}_i, \cdots, \hat{m}_{\text{base},N_m}\right], \quad \text{with } \delta \hat{m}_i = \beta \hat{m}_{\text{base},i}, \tag{5.9}$$

where the scaled vectors, $\hat{\boldsymbol{m}}_i$ and $\hat{\boldsymbol{m}}_{\text{base}}$, are transformed back to the original physical space

before used to obtain $\boldsymbol{d}_i$ and $\boldsymbol{d}_{\text{base}}$ in Eq. 5.8.

The results of sensitivity analysis for different model parameter are shown in Figs. 5.7 and 5.8 where the x-axis represents the perturbation size β and the y-axis denotes the sensitivity value calculated by Eq. 5.5 or 5.8. Without scaling, we can see that the production data is extremely sensitive to the sinuosity while the impacts of other important parameters (e.g., orientation, amplitude, wavelength, width and thickness of sand channel) are almost insignificant, which is not reasonable from a geological perspective. When we implement the "$[0, 1]$ scaling" scheme, the large difference between sensitivities to various parameters is dramatically alleviated. It is shown that the sensitivities to the permeability values of three facies remain very stable with different perturbation sizes. However, the derivatives with respect to other parameters vary significantly as the perturbation size changes.



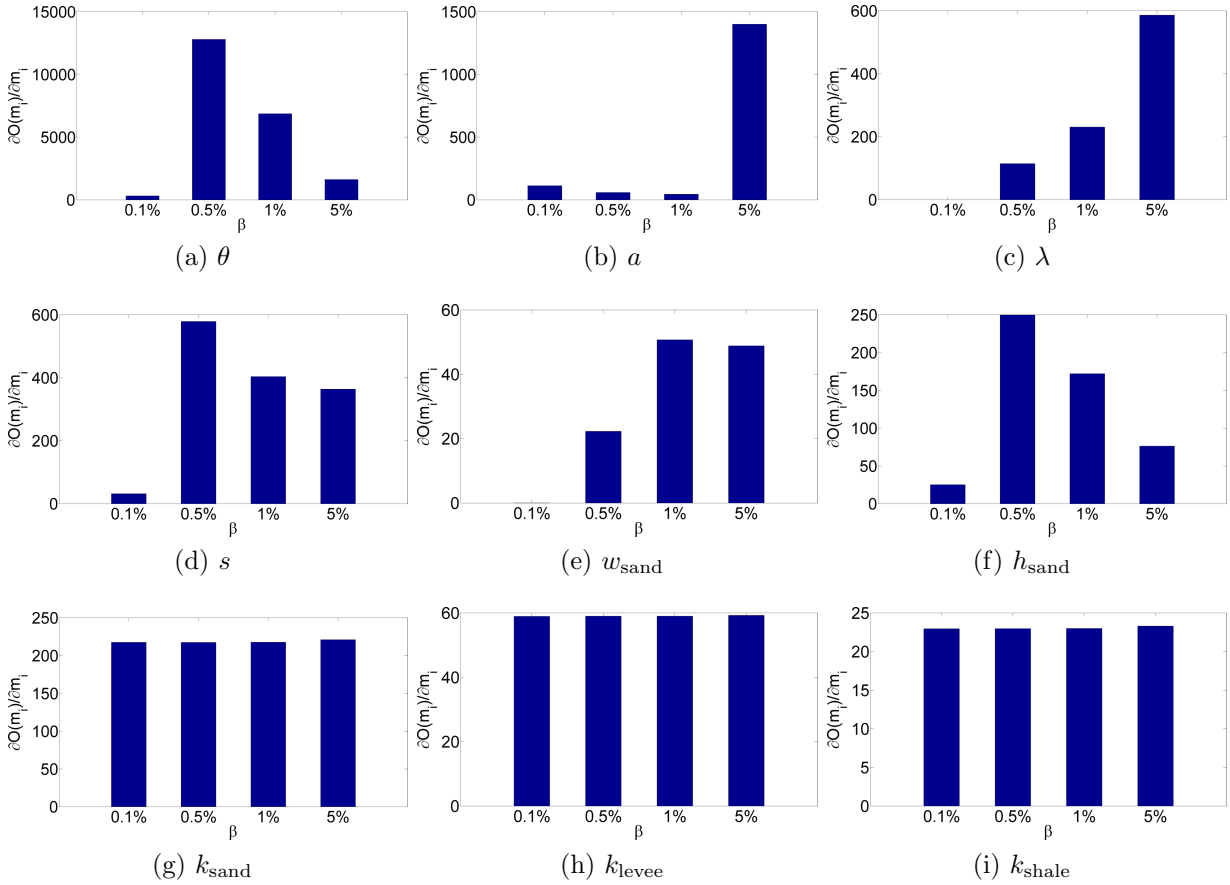
(a) θ (b) a (c) λ

(d) s (e) w_{sand} (f) h_{sand}

(g) k_{sand} (h) k_{levee} (i) k_{shale}

Figure 5.7: Sensitivity analysis for different model parameters without scaling

It is noteworthy that we only perturb one parameter at a time with the values of other



(a) θ (b) a (c) λ

(d) s (e) w_{sand} (f) h_{sand}

(g) k_{sand} (h) k_{levee} (i) k_{shale}

Figure 5.8: Sensitivity analysis for different model parameters with scaling

parameters held fixed at their base values. The observations from Figs. 5.7 and 5.8 can be quantitatively demonstrated by the data given in Table 5.6. Therefore, it may be helpful to employ the "$[0, 1]$ scaling" scheme in order to improve the approximation of gradient in history matching by mapping different model parameters to the same range.

5.3 Computational Results and Discussion

As stated in the beginning of this chapter, what we thought would be a promising approach for history matching of object-based models only worked partially well. In fact, we attempted to history match all eleven parameters shown in Eq. 5.2 at first but the proposed workflow failed. Then we tried other cases with different combinations of parameters in order to obtain a understanding of why the method fails. The posterior ensembles of some cases (e.g., a, λ, s, k_{sand}, k_{levee}, k_{shale}) converge to incorrect solutions with poor history matches,

Table 5.6: Results of sensitivity analysis

	No scaling				$[0,1]$ scaling			
	0.1%	0.5%	1%	5%	0.1%	0.5%	1%	5%
θ	6.08	77.86	44.72	15.05	323.65	12798.50	6875.38	1630.23
a	0.30	0.12	0.65	1.35	113.71	60.03	46.04	1400.40
λ	0.05	0.08	0.04	0.11	0.07	114.75	230.86	586.03
s	6869.84	1446.78	4305.71	1222.26	32.17	578.55	403.80	364.01
w_{sand}	0.09	0.34	0.14	1.89	0.07	22.29	50.72	48.83
h_{sand}	0.51	7.01	3.20	3.24	25.37	249.77	172.09	76.39
w_{levee}	36.24	82.25	197.65	122.25	0.07	5.07	28.75	35.05
h_{levee}	18.12	37.76	32.18	49.66	0.00	5.07	26.35	9.01
k_{sand}	120.90	122.23	124.77	145.47	217.58	217.41	217.66	220.94
k_{levee}	32.74	32.83	32.94	33.66	58.91	58.96	58.97	59.24
k_{shale}	12.74	12.80	12.86	13.45	22.95	22.96	22.98	23.29

but for other specific combinations of parameters (e.g., a, λ, k_{sand}, k_{levee}, k_{shale}; s, k_{sand}, k_{levee}, k_{shale}), we can obtain good data matches and the geological structures in the true model are resolved well through history matching. In the remainder of this section, the results of one failed case and two successful cases are presented in order to comprehensively illustrate the performance of the proposed methodology and discuss the probable reason causing the failure.

The reservoir model in the three synthetic cases is similar to the one in the base case. A three-facies (shale, levee and sand) two-layer channelized reservoir model is defined on a $50 \times 50 \times 2$ grid system. The size of each gridblock is 100ft $\times$ 100ft with a constant thickness of 20 ft. There are four injectors (I1 to I4) and nine producers (P1 to P9) deployed in a five-spot pattern in the reservoir. All wells are controlled by bottom hole pressure (BHP). To be specific, the injectors and producers, respectively, operate at a constant BHP of 5500 psi and 3000 psi. The initial pressure of the reservoir is 5000 psi and the initial water saturation is 0.2 which is equal to irreducible water saturation.

Given the hard data of facies type observation at well locations (see Table 5.5), the facies distribution of the true model and 400 prior realizations ($N_e = 400$) are generated

by feeding the sampled parameters to the object-based modeling module in Petrel. Since we assume the natural logarithm of permeability ($\ln(k)$) within facies is homogeneous, the permeability of every gridblock is directly assigned according to its facies type.

It is customary to apply so-called localization schemes to ensemble methods for the purpose of reducing sampling errors associated with small ensemble. In retrospect, the distance-based localization appears to be the most adopted scheme for ensemble-based data assimilation and has been worked successfully in many practical reservoir characterization problems. However, a prerequisite in using distance-based localization is that both model parameters and observed data have associated physical locations, which is not satisfied regarding the parameters for object-based modeling in this work. As a result of this, one cannot apply the distance-based localization to this kind of problem. Although several non-distance-based localization schemes have been proposed recently, here we increase the degree of freedom in data assimilation by simply using a large prior ensemble of size $N_e = 400$.

The history matching period is 600 days followed by the prediction period of 300 days. The subsurface flow responses are obtained by running the Eclipse reservoir simulator. The production data consist of the water injection rate (q_{inj}) at the injectors and the oil and water production rate (q_{o} and q_{w}) at the producers. The observed data are collected every 30 days during the history matching period in order to form the data vector given by

$$\boldsymbol{d}_{\text{obs}} = \left[\boldsymbol{q}_{\text{inj}}^{\text{T}}, \boldsymbol{q}_{\text{o}}^{\text{T}}, \boldsymbol{q}_{\text{w}}^{\text{T}}\right]_{N_d \times 1}^{\text{T}}. \tag{5.10}$$

In the synthetic cases, the observed data are obtained by adding Gaussian random noise to the noise-free data predicted by the true model. The covariance matrix of the observation errors, $\boldsymbol{C}_{\text{D}}$, is given by

$$\boldsymbol{C}_{\text{D}} = \begin{bmatrix} \sigma_1^2 & & \\ & \ddots & \\ & & \sigma_{N_d}^2 \end{bmatrix}_{N_d \times N_d}, \tag{5.11}$$

where the standard deviation of measurement error, σ, is equal to 3% of the true data for the injection and production rates.

The ES-MDA-GEO algorithm is employed to update the vector of model parameters, $\boldsymbol{m}$, by assimilating the observed data. The iteration number, N_a, is set to 6 with the inflation factors, $\alpha_\ell = \{1000.0,\ 267.3,\ 71.4,\ 19.1,\ 5.1,\ 1.4\}$, calculated by the geometric method (Rafiee and Reynolds, 2017).

5.3.1 Case 1: a, λ, s, k_{sand}, k_{levee}, k_{shale}

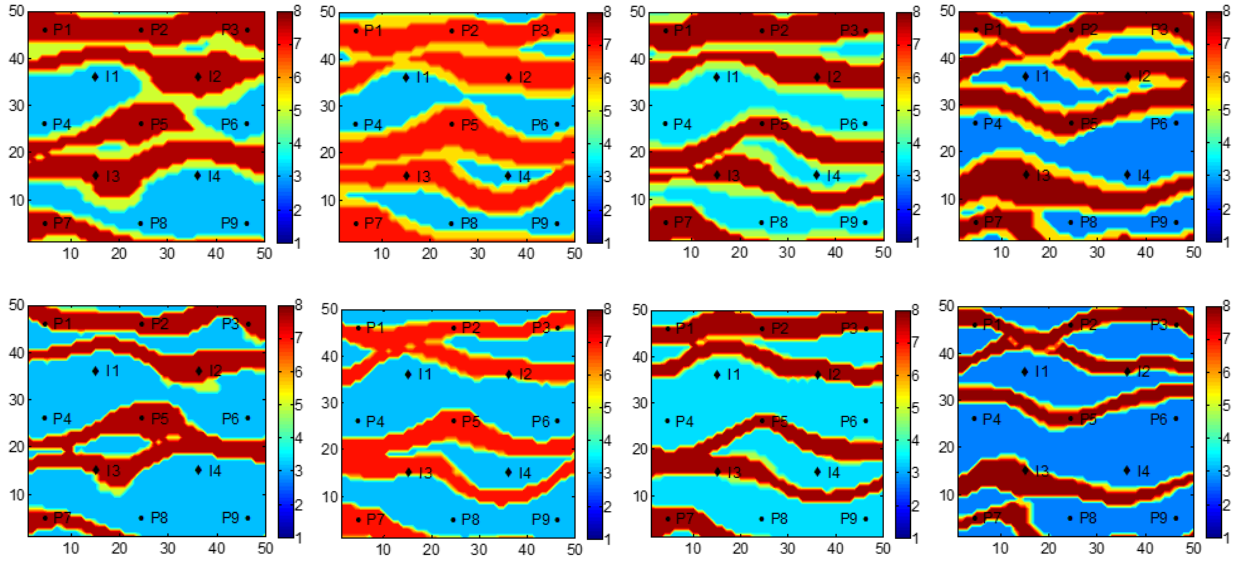
In the first case, we start from a combination of six model parameters including amplitude (a), wavelength (λ), sinuosity (s) and three permeability values (k_{sand}, k_{levee} and k_{shale}); thus, the model vector, $\boldsymbol{m}$, is given by

$$\boldsymbol{m} = \left[a, \lambda, s, k_{\text{sand}}, k_{\text{levee}}, k_{\text{shale}}\right]^{\text{T}}, \tag{5.12}$$

where each of the above parameters follows the Gaussian distribution with the mean and standard deviation given in Table 5.4 throughout the true model and whole prior ensemble. The other parameters that are essential for the object-based modeling (see Eq. 5.1) but not included in the model parameters of history matching are fixed equal to the mean values, which is the same case in the true model.

Figure 5.9 shows the $\ln(k)$ fields of the true model and three unconditional realizations. The diversity of the prior realizations can be illustrated by the production data generated with the prior ensemble (see Fig. 5.10). In Fig. 5.10 and similar figures, the red curves denote the results of the true model; the red dots represent the observed data; the light blue curves are obtained with prior realizations and the dark blue curves are the ensemble average of prior production data. We can see that the prior uncertainty is quite high due to the diverse distributions of channels in the prior ensemble as well as the uncertainty associated with the petrophysical properties.

For this case, we are not able to resolve the correct geological structures and petrophysical properties within three facies through history matching (see Fig. 5.11). From



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

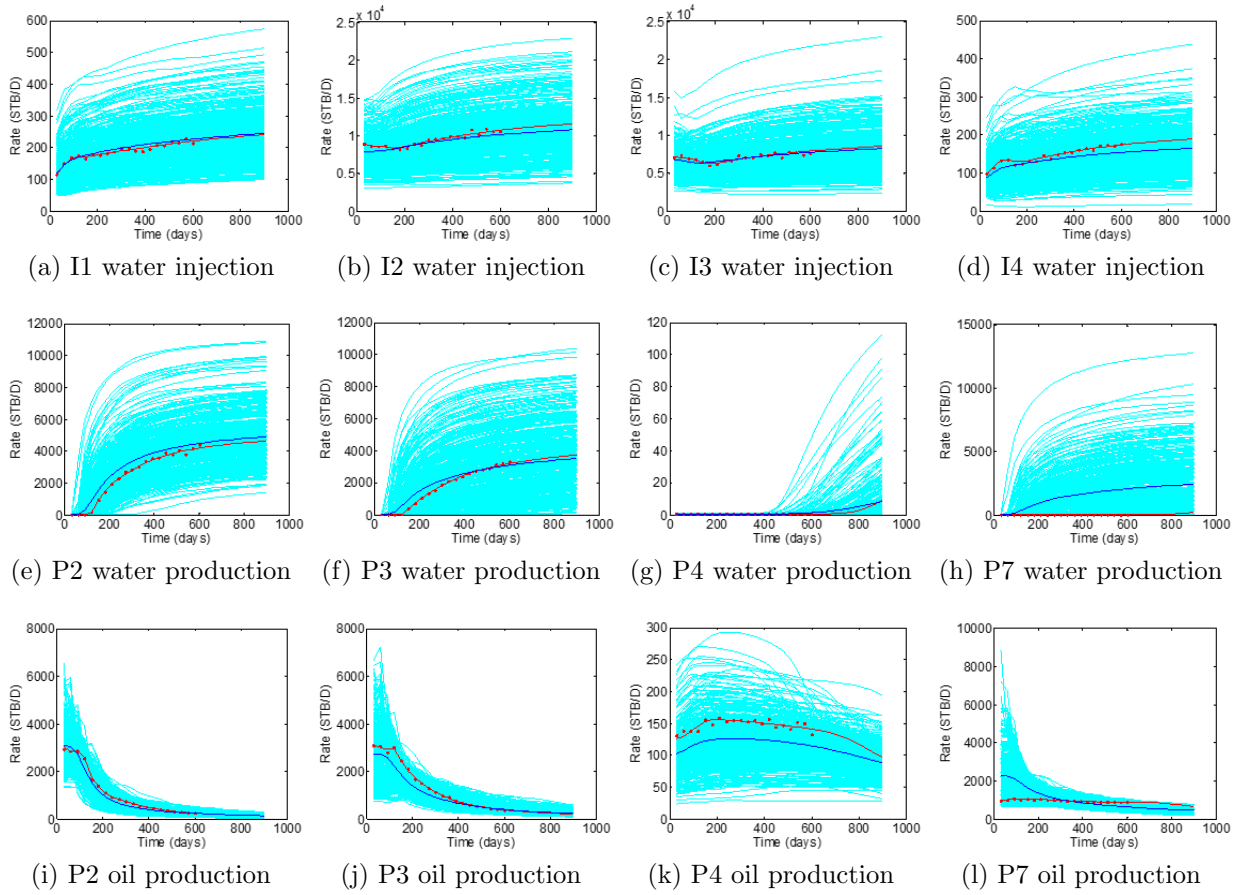
Figure 5.9: True model and three prior realizations (Case 1: a, λ, s, k_{sand}, k_{levee}, k_{shale}; $N_e = 400$): layer 1 (1st row) and layer 2 (2nd row).

Figs. 5.11 and 5.12, the posterior ensemble is shown to collapse to an incorrect model giving poor data matches at most wells. The uncertainty in the posterior realizations still remains at a high level, which is clearly shown in the box plots of normalized data mismatch (see Fig. 5.13) calculated by

$$O(\boldsymbol{d}) = \frac{1}{N_d} \left(\boldsymbol{d} - \boldsymbol{d}_{\text{obs}}\right)^{\text{T}} \boldsymbol{C}_{\text{D}}^{-1} \left(\boldsymbol{d} - \boldsymbol{d}_{\text{obs}}\right), \tag{5.13}$$

where $\boldsymbol{d}$ and $\boldsymbol{d}_{\text{obs}}$ only include the data during the historical period.

The evolution of model parameters during history matching are shown in Fig. 5.14 where the x-axis is the data assimilation step index where "0" corresponds to the prior ensemble. In Fig. 5.14 and similar figures, a dark horizontal straight line represents the true value, a red curve through circles is the ensemble mean of the calibrated parameter and two blue curves through triangles are the ensemble mean ± 3 ensemble standard deviation respectively, which gives a characterization of the uncertainty in the estimated parameters. It is shown in Fig. 5.14 that the model parameters are poorly estimated for this case.



(a) I1 water injection (b) I2 water injection (c) I3 water injection (d) I4 water injection

(e) P2 water production (f) P3 water production (g) P4 water production (h) P7 water production

(i) P2 oil production (j) P3 oil production (k) P4 oil production (l) P7 oil production

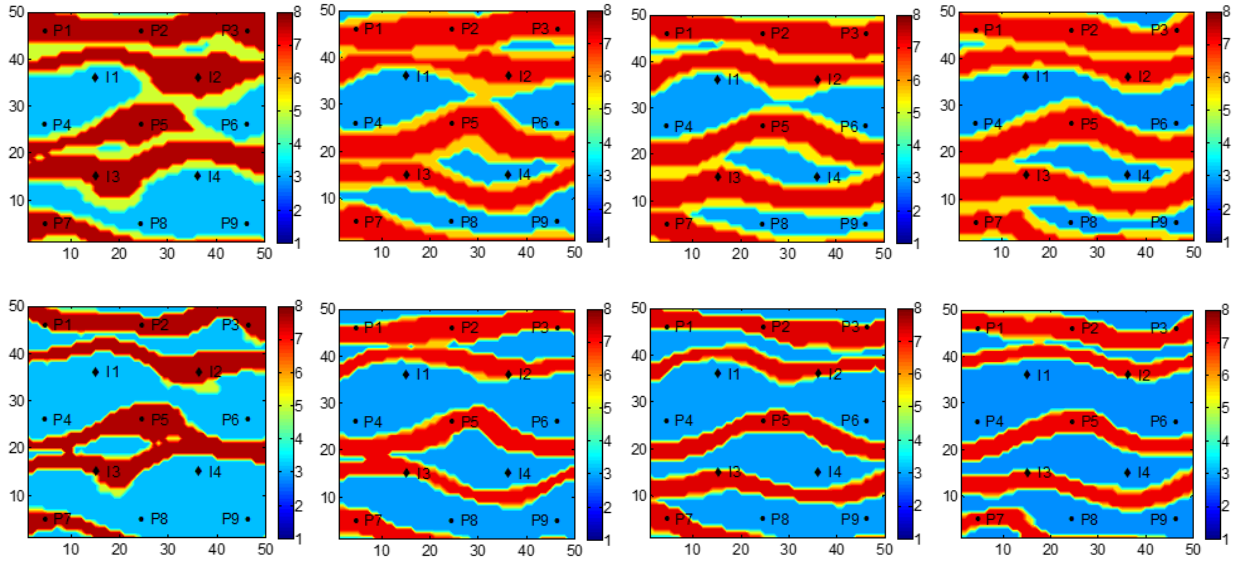
Figure 5.10: Production data of prior ensemble (Case 1: a, λ, s, k_{sand}, k_{levee}, k_{shale}; $N_e = 400$). True (red curves), observed data (red dots), simulated data of realizations (light blue curves), and ensemble mean (dark blue curves). History: $0 < t \leq 600$ days, forecast: $t > 600$ days.

5.3.2 Case 2: a, λ, k_{sand}, k_{levee}, k_{shale}

In the second case, we reduce the number of model parameters by removing the sinuosity (s) from the combination of parameters in Case 1. The resulting model vector, $\boldsymbol{m}$, is given by

$$\boldsymbol{m} = \left[a, \lambda, k_{\text{sand}}, k_{\text{levee}}, k_{\text{shale}}\right]^{\text{T}}. \tag{5.14}$$

Same as Case 1, the five parameters in Eq. 5.14 are sampled with the mean and standard deviation given in Table 5.4 throughout the true model and prior realizations. Figure 5.15 shows the $\ln(k)$ fields of the true model and three unconditional realizations where there is large variability in the facies distributions. The high uncertainty in the prior



(a) True (b) Posterior model #1 (c) Posterior model #2 (d) Posterior model #3

Figure 5.11: True model and three posterior realizations (Case 1: a, λ, s, k_{sand}, k_{levee}, k_{shale}; $N_e = 400$): layer 1 (1st row) and layer 2 (2nd row).

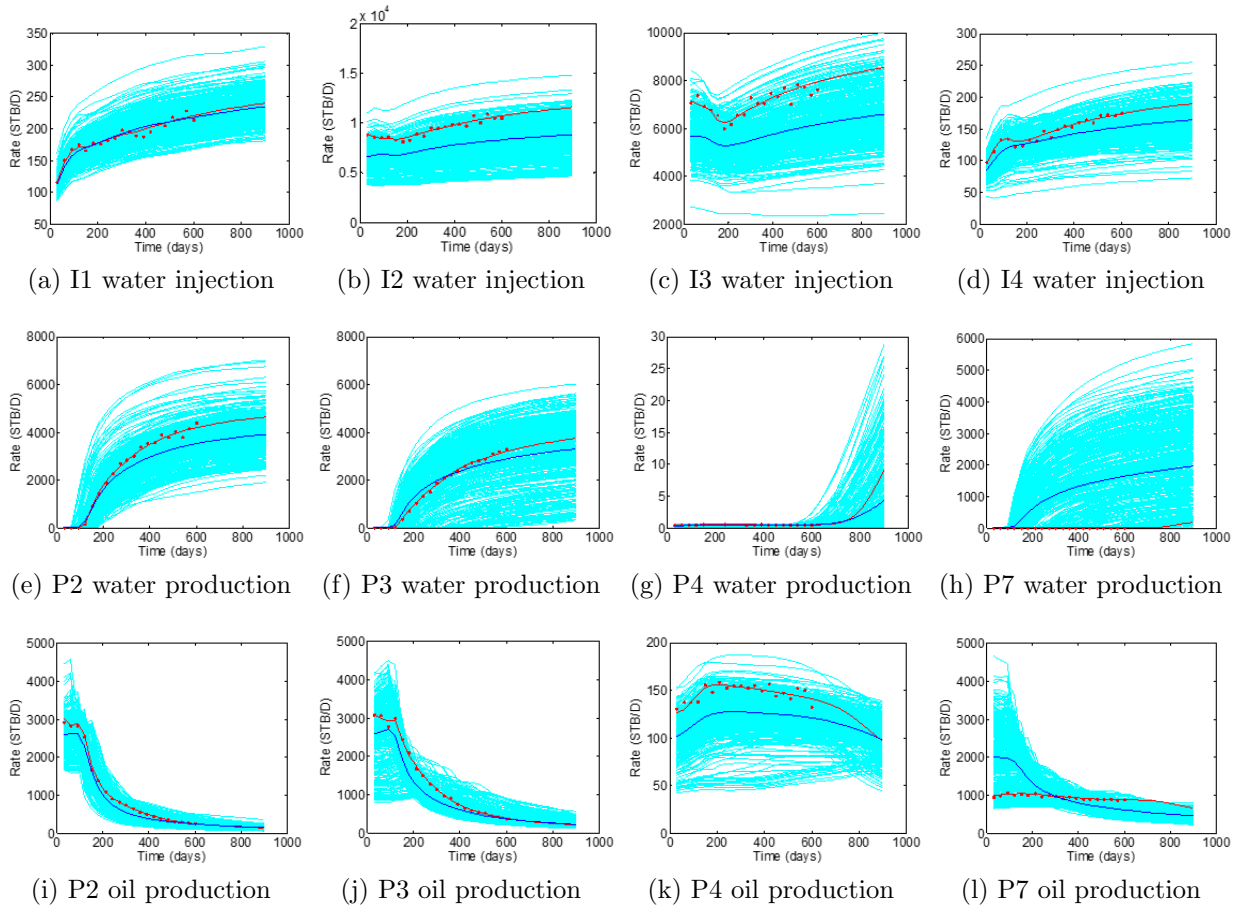
realizations is illustrated by the prior production data shown in Fig. 5.16.

Figure 5.17 compares the $\ln(k)$ fields of the true model with those of three posterior realizations. We can see that the posterior realizations converge to a model visually identical to the truth by the end of history matching. Although the spatial distributions of sand channels are almost the same as the true model, there is slight variability when it comes to the distribution and permeability value of the levee facies.

Through history matching, the prior uncertainty is significantly reduced and we are able to obtain good data matches and reasonable production forecast (see Fig. 5.18). The quantitative data mismatch is calculated by Eq. 5.13 and the changes along with the data assimilation step are shown in Fig. 5.19.

From Fig. 5.20, we can see that good estimates of the amplitude, wavelength and the permeability values of three facies are obtained in this case meanwhile the prior uncertainty has been reduced significantly. There still remains small variability in the calibrated parameters after the last iteration step of data assimilation.

5.3.3 Case 3: s, k_{sand}, k_{levee}, k_{shale}



(a) I1 water injection (b) I2 water injection (c) I3 water injection (d) I4 water injection

(e) P2 water production (f) P3 water production (g) P4 water production (h) P7 water production

(i) P2 oil production (j) P3 oil production (k) P4 oil production (l) P7 oil production

Figure 5.12: Production data of posterior ensemble (Case 1: a, λ, s, k_{sand}, k_{levee}, k_{shale}; $N_e = 400$). True (red curves), observed data (red dots), simulated data of realizations (light blue curves), and ensemble mean (dark blue curves). History: $0 < t \leq 600$ days, forecast: $t > 600$ days.

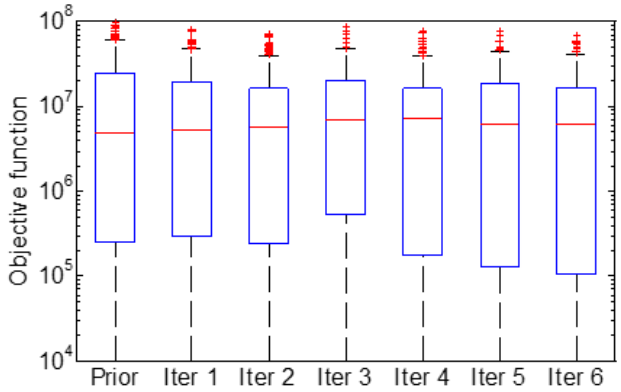
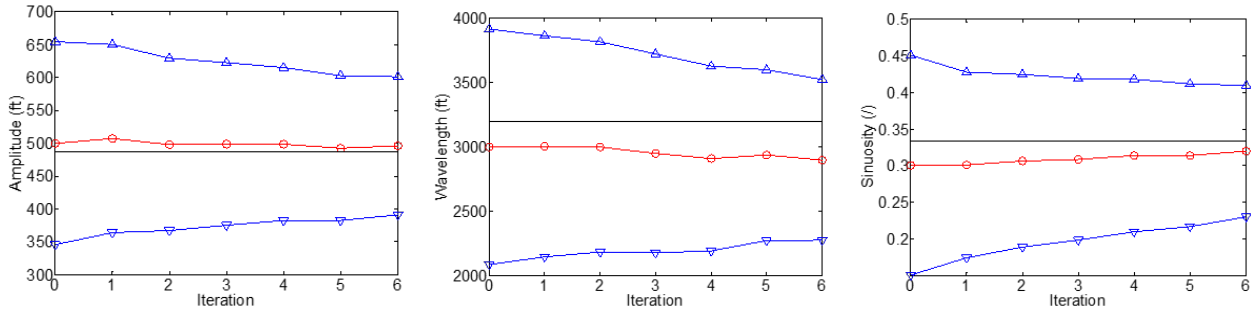


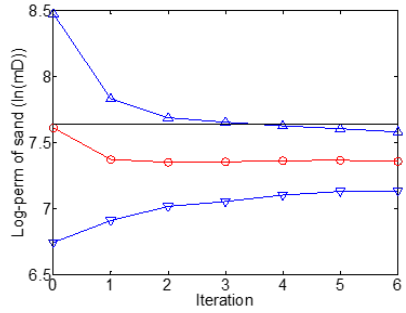
Figure 5.13: Normalized data mismatch (Case 1: a, λ, s, k_{sand}, k_{levee}, k_{shale}; $N_e = 400$). Median (central red mark), 25th and 75th percentiles (bottom and top edges of the box), extreme data points (whiskers), and outliers ("+" symbol).



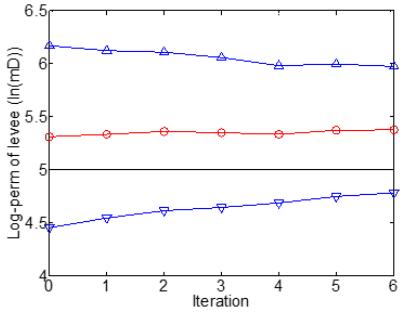
(a) Amplitude

(b) Wavelength

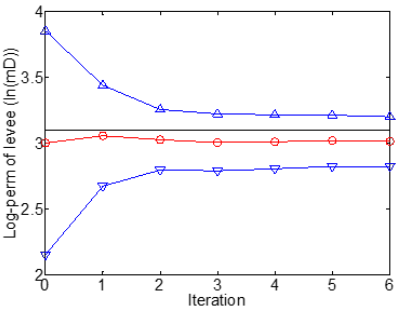
(b) Sinuosity



(c) k_{sand}

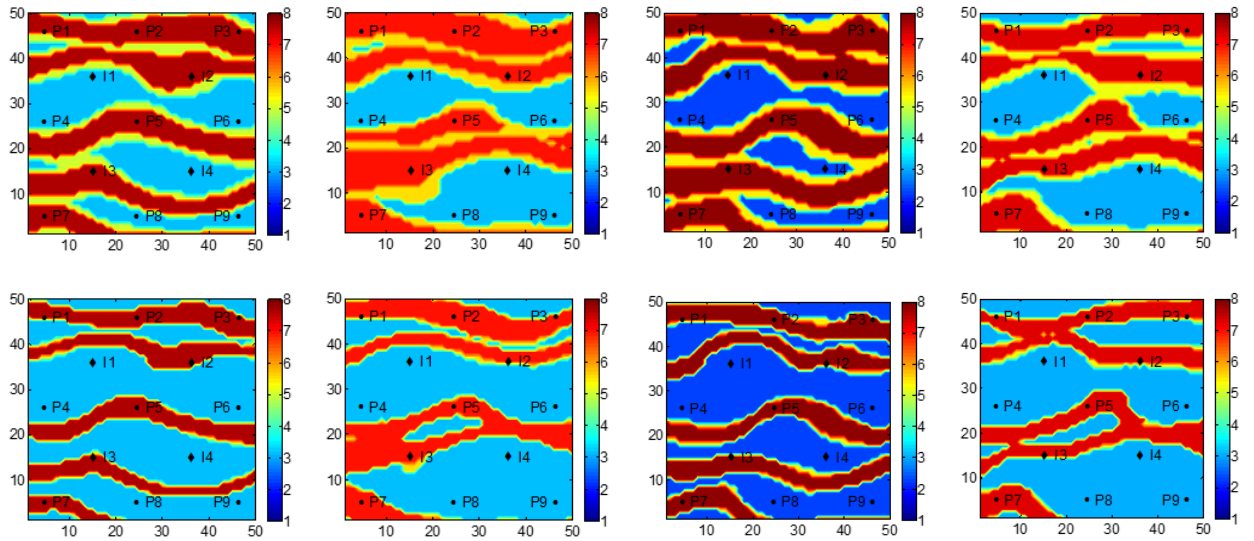


(d) k_{levee}



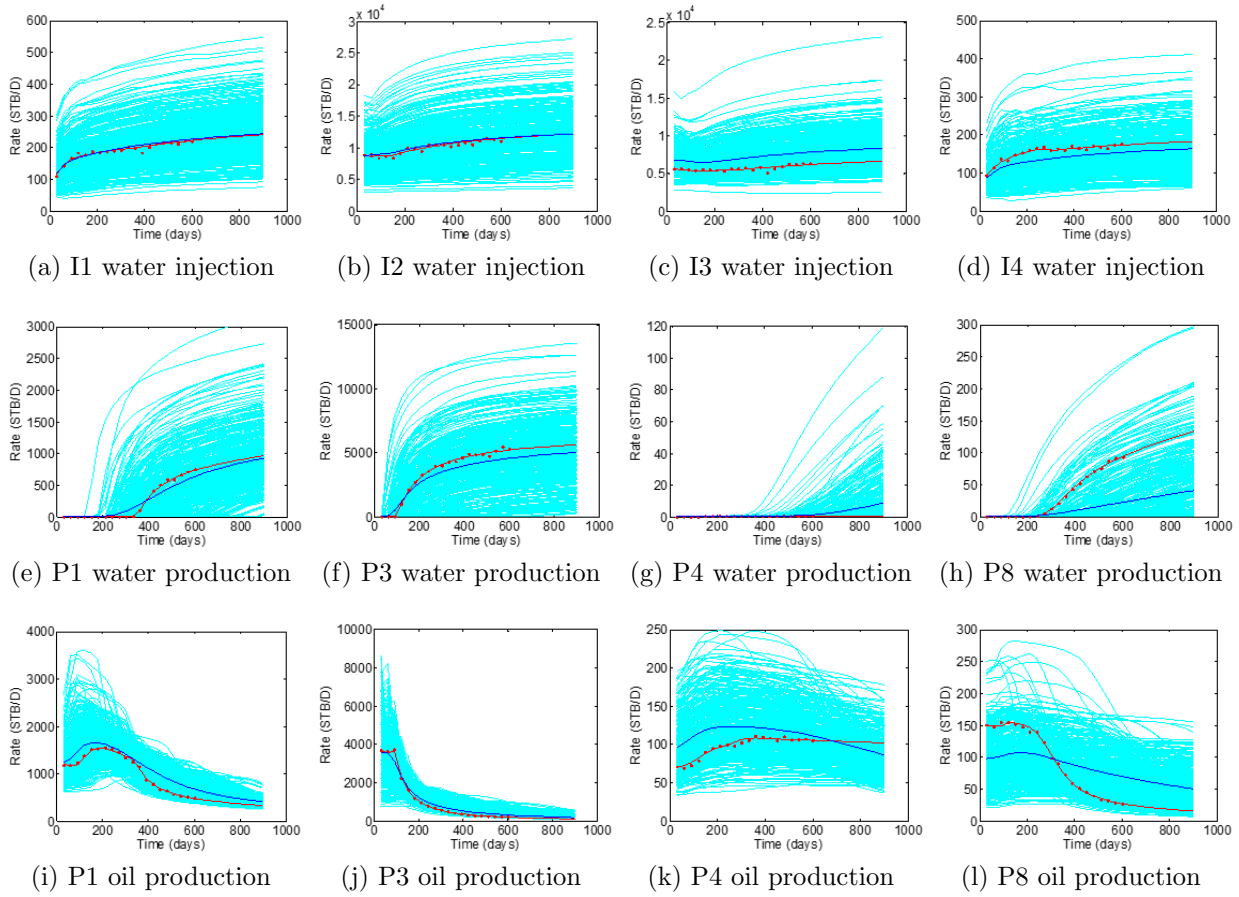
(e) k_{shale}

Figure 5.14: Evolution of model parameters as a function of the ES-MDA step (Case 1: a, λ, s, k_{sand}, k_{levee}, k_{shale}; $N_e = 400$). True (black horizontal straight lines), ensemble mean (red curves), and ensemble mean ± 3 ensemble standard deviation (blue curves).



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

Figure 5.15: True model and three prior realizations (Case 2: a, λ, k_{sand}, k_{levee}, k_{shale}; $N_e = 400$): layer 1 (1st row) and layer 2 (2nd row).



(a) I1 water injection (b) I2 water injection (c) I3 water injection (d) I4 water injection

(e) P1 water production (f) P3 water production (g) P4 water production (h) P8 water production

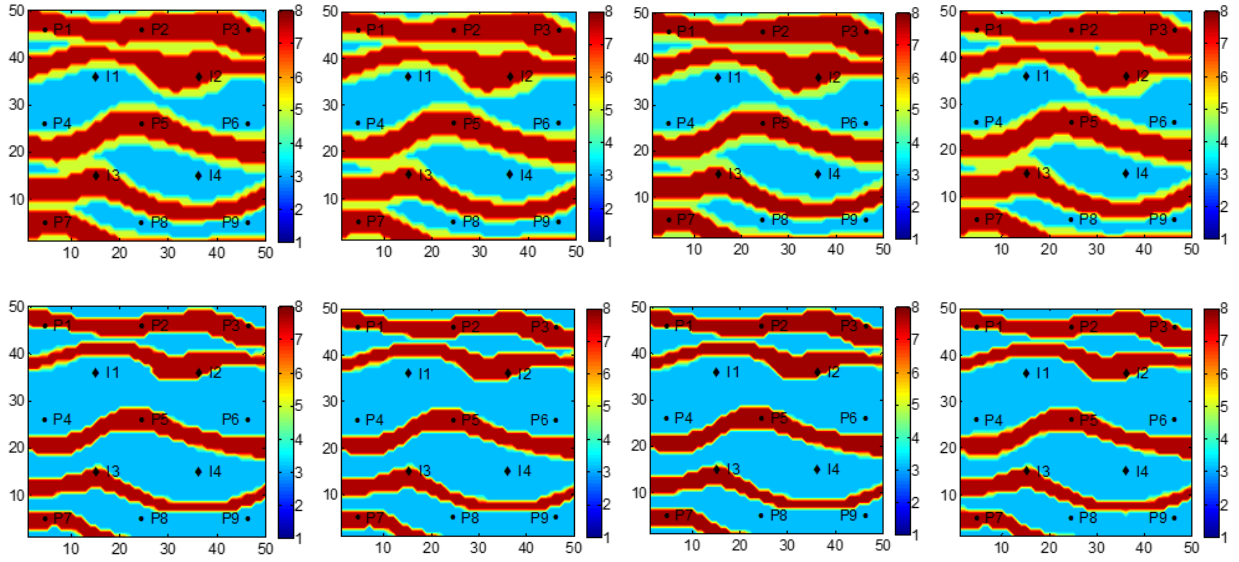
(i) P1 oil production (j) P3 oil production (k) P4 oil production (l) P8 oil production

Figure 5.16: Production data of prior ensemble (Case 2: a, λ, k_{sand}, k_{levee}, k_{shale}; $N_e = 400$). True (red curves), observed data (red dots), simulated data of realizations (light blue curves), and ensemble mean (dark blue curves). History: $0 < t \leq 600$ days, forecast: $t > 600$ days.

In the last case, we retain four parameters including the sinuosity (s) and permeability values of three facies (k_{sand}, k_{levee}, k_{shale}) based on the combination of parameters in Case 1. The vector of model parameters, $\boldsymbol{m}$, is then given by

$$\boldsymbol{m} = \left[s, k_{\text{sand}}, k_{\text{levee}}, k_{\text{shale}}\right]^{\text{T}}. \tag{5.15}$$

Because of different model parameters, the true model and prior realizations (see Fig. 5.21) differ from those in the previous two cases. Based on the discussion of sinuosity in Section 5.1.1, it seems that the true sinuosity is greater than 0.3 since the channel around well I3 is much wider than the one beside well I4. In contrast, the sinuosity of three



(a) True (b) Posterior model #1 (c) Posterior model #2 (d) Posterior model #3

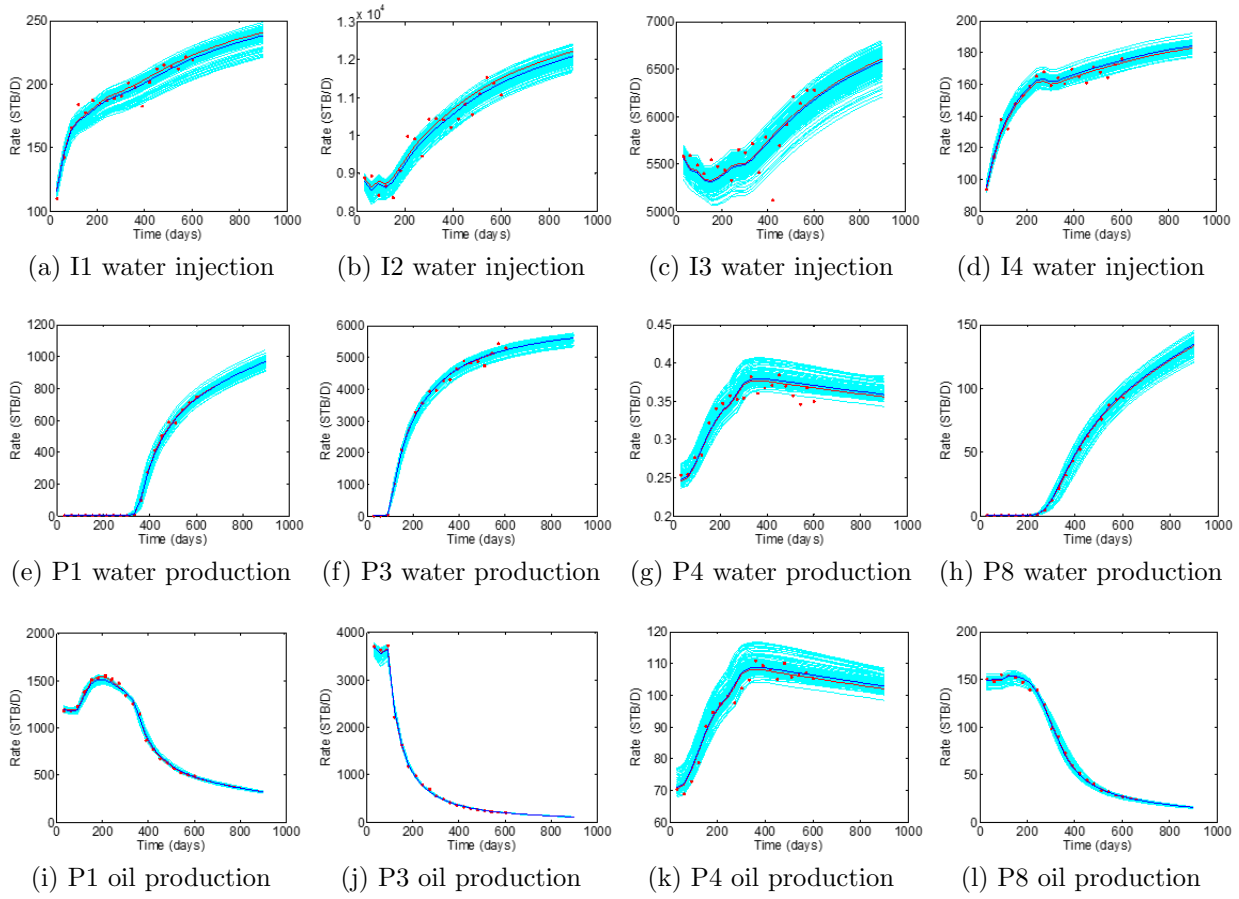
Figure 5.17: True model and three posterior realizations (Case 2: a, λ, k_{sand}, k_{levee}, k_{shale}; $N_e = 400$): layer 1 (1st row) and layer 2 (2nd row).

prior realizations is relatively small because the facies distributes more uniformly along the channels. From Fig. 5.22, the significant uncertainty in the prior production data probably comes from the sinuous geometry of the geological structures and the intersecting patterns of adjacent channels.

Similar to Case 2, we are able to yield very good data matches (see Fig. 5.24) but again the whole ensemble essentially collapses to individual model (see Fig. 5.23) even with the large ensemble of size $N_e = 400$. Regarding the uncertainty quantification, the prior uncertainty has been dramatically reduced (see Fig. 5.25) and the model parameters are calibrated to the true values with almost no variance left (see Fig. 5.26).

5.3.4 Discussion

Based on the previous sensitivity analysis, the failure of proposed workflow in those unsatisfactory cases is probably due to the unstable sensitivities of model parameters which makes it difficult to obtain a good gradient approximation during history matching. To figure out the reason for the unstable sensitivities, it is essential to fully understand the modeling process in the object-based method. Figure 5.27 shows an adaptive channel generated by



(a) I1 water injection (b) I2 water injection (c) I3 water injection (d) I4 water injection

(e) P1 water production (f) P3 water production (g) P4 water production (h) P8 water production

(i) P1 oil production (j) P3 oil production (k) P4 oil production (l) P8 oil production

Figure 5.18: Production data of posterior ensemble (Case 2: a, λ, k_{sand}, k_{levee}, k_{shale}; $N_e = 400$). True (red curves), observed data (red dots), simulated data of realizations (light blue curves), and ensemble mean (dark blue curves). History: $0 < t \leq 600$ days, forecast: $t > 600$ days.

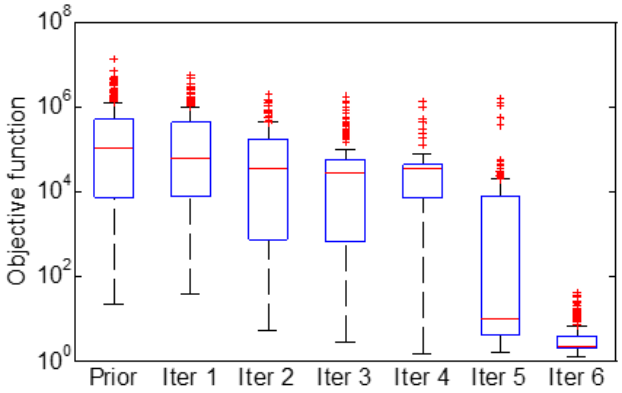
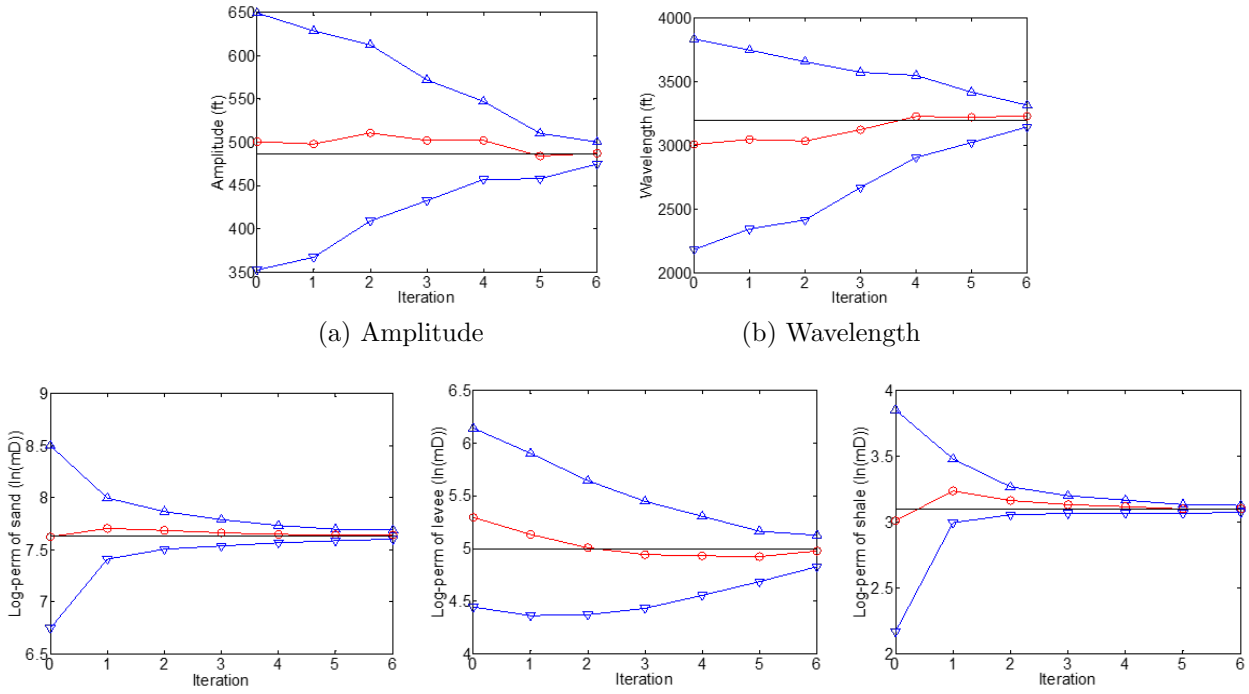


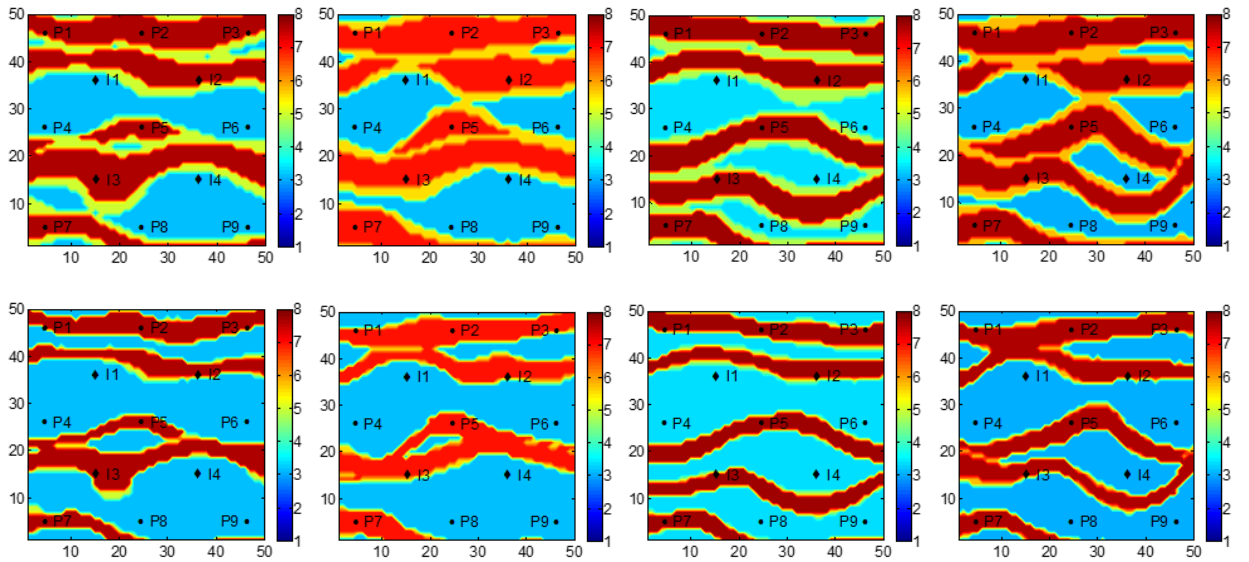
Figure 5.19: Normalized data mismatch (Case 2: a, λ, k_{sand}, k_{levee}, k_{shale}; $N_e = 400$). Median (central red mark), 25th and 75th percentiles (bottom and top edges of the box), extreme data points (whiskers), and outliers ("+" symbol).



(a) Amplitude (b) Wavelength

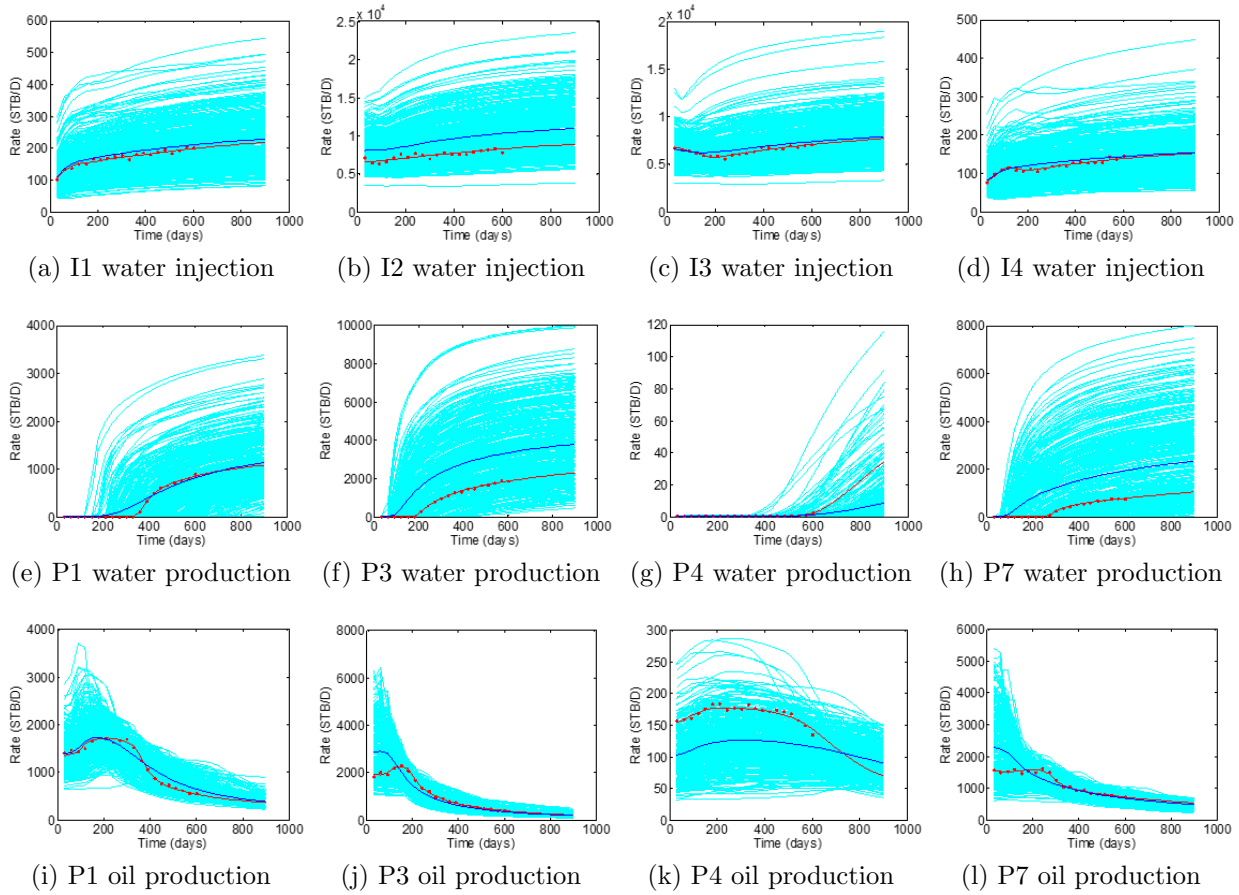
(c) k_{sand} (d) k_{levee} (e) k_{shale}

Figure 5.20: Evolution of model parameters as a function of the ES-MDA step (Case 2: a, λ, k_{sand}, k_{levee}, k_{shale}; $N_e = 400$). True (black horizontal straight lines), ensemble mean (red curves), and ensemble mean ± 3 ensemble standard deviation (blue curves).



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

Figure 5.21: True model and three prior realizations (Case 3: s, k_{sand}, k_{levee}, k_{shale}; $N_e = 400$): layer 1 (1st row) and layer 2 (2nd row).



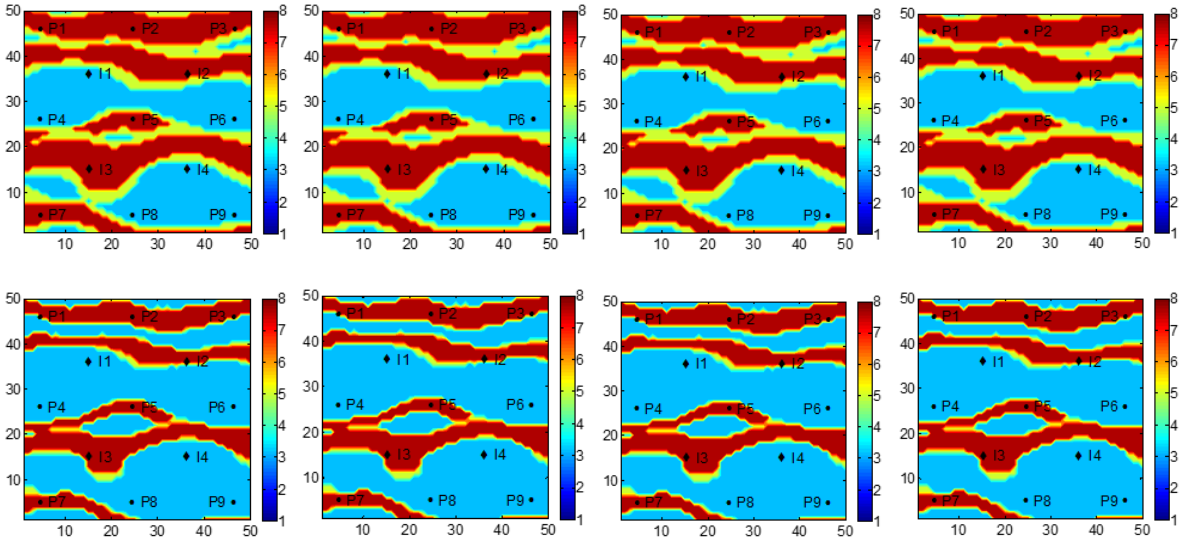
(a) I1 water injection (b) I2 water injection (c) I3 water injection (d) I4 water injection

(e) P1 water production (f) P3 water production (g) P4 water production (h) P7 water production

(i) P1 oil production (j) P3 oil production (k) P4 oil production (l) P7 oil production

Figure 5.22: Production data of prior ensemble (Case 3: s, k_{sand}, k_{levee}, k_{shale}; $N_e = 400$). True (red curves), observed data (red dots), simulated data of realizations (light blue curves), and ensemble mean (dark blue curves). History: $0 < t \leq 600$ days, forecast: $t > 600$ days.

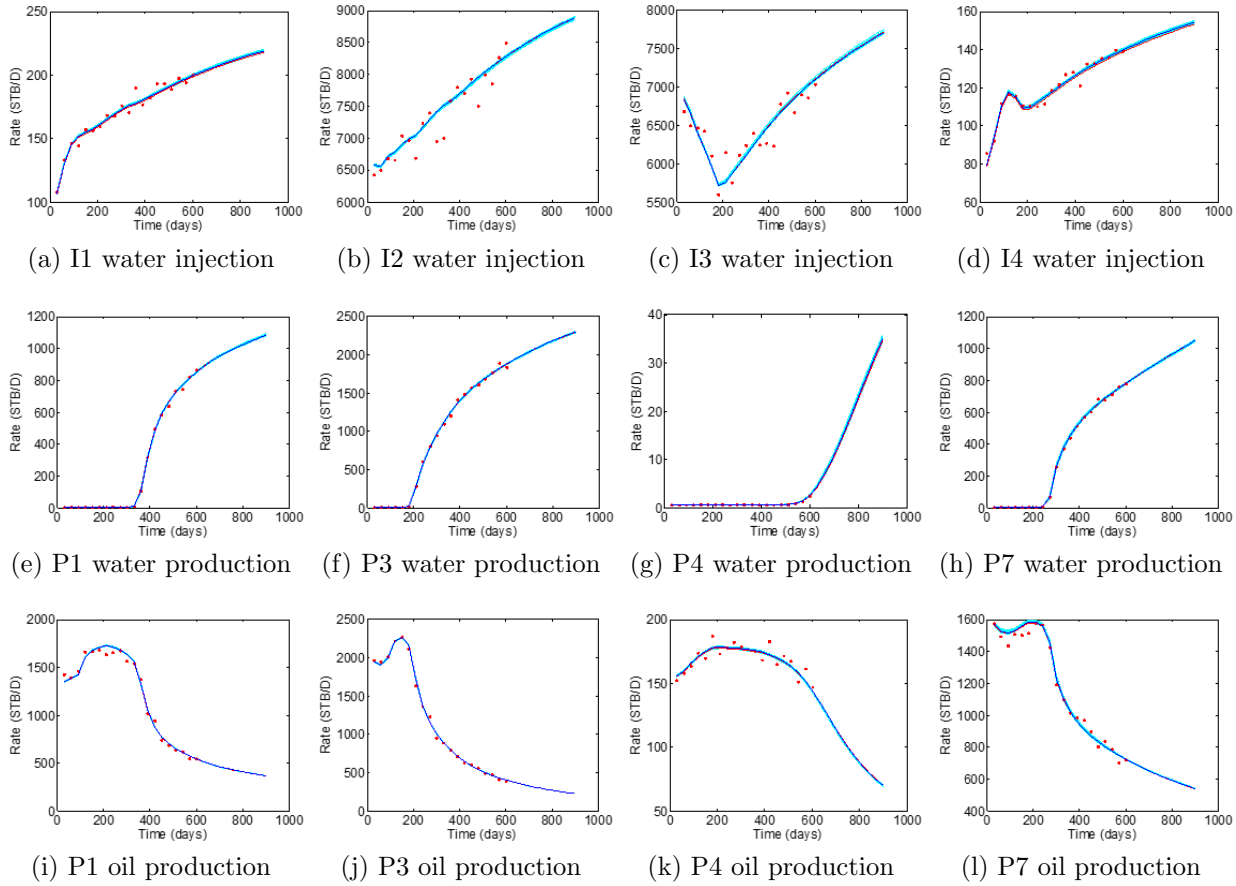
the object-based modeling in Petrel. In one row of the wells (see Fig. 5.27(a)), alternating wells contain channel facies (yellow) or levee facies (light blue) in specific upscaled cells as hard data. In order to honor the hard data information, the first building block containing channel/levee part is placed to fit a given well. Then, a sequence of building blocks are inserted continuously following horizontal and areal search schemes in order to match all the wells. After all well data have been satisfied, the channel or levee facies are added or removed by random such that the average properties (such as amplitude, wavelength, etc.) of large-scale geological structures are consistent with the values we provide to the object-based method in Petrel. During this final stage of modeling, the internal randomness may lead to unexpected facies generations. As a result, the sensitivities of predicated data to the



(a) True (b) Posterior model #1 (c) Posterior model #2 (d) Posterior model #3

Figure 5.23: True model and three posterior realizations (Case 3: s, k_{sand}, k_{levee}, k_{shale}; $N_e = 400$): layer 1 (1st row) and layer 2 (2nd row).

model parameters are not stable and it is hard to approximate the gradient correctly in the data assimilation process.



(a) I1 water injection (b) I2 water injection (c) I3 water injection (d) I4 water injection

(e) P1 water production (f) P3 water production (g) P4 water production (h) P7 water production

(i) P1 oil production (j) P3 oil production (k) P4 oil production (l) P7 oil production

Figure 5.24: Production data of posterior ensemble (Case 3: s, k_{sand}, k_{levee}, k_{shale}; $N_e = 400$). True (red curves), observed data (red dots), simulated data of realizations (light blue curves), and ensemble mean (dark blue curves). History: $0 < t \leq 600$ days, forecast: $t > 600$ days.

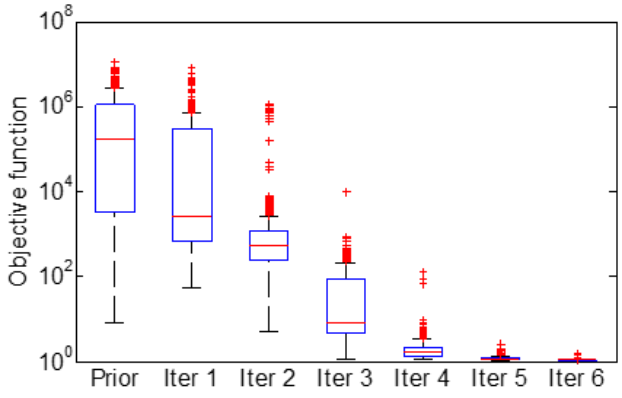
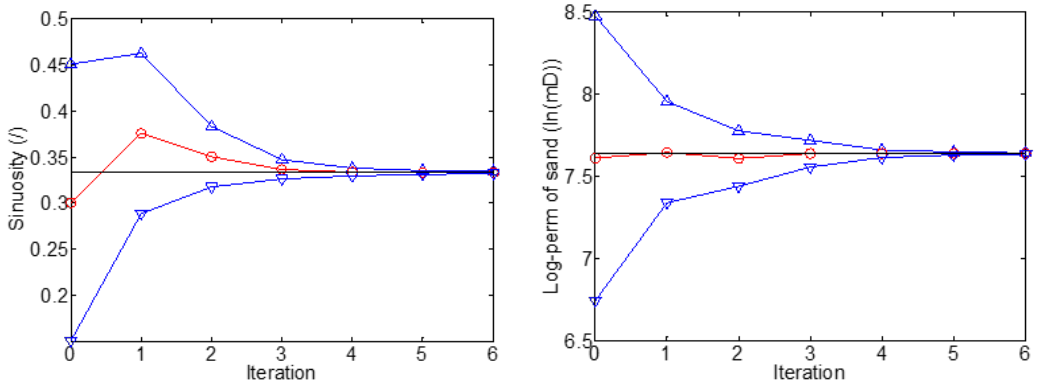
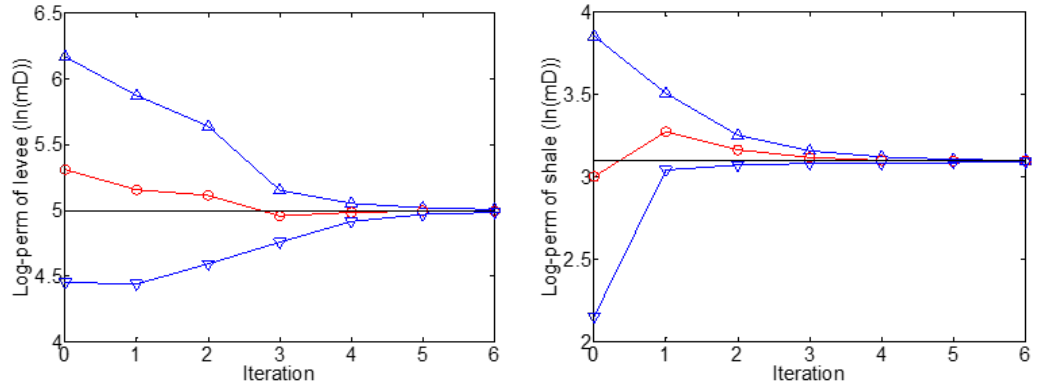


Figure 5.25: Normalized data mismatch (Case 3: s, k_{sand}, k_{levee}, k_{shale}; $N_e = 400$). Median (central red mark), 25th and 75th percentiles (bottom and top edges of the box), extreme data points (whiskers), and outliers ("+" symbol).

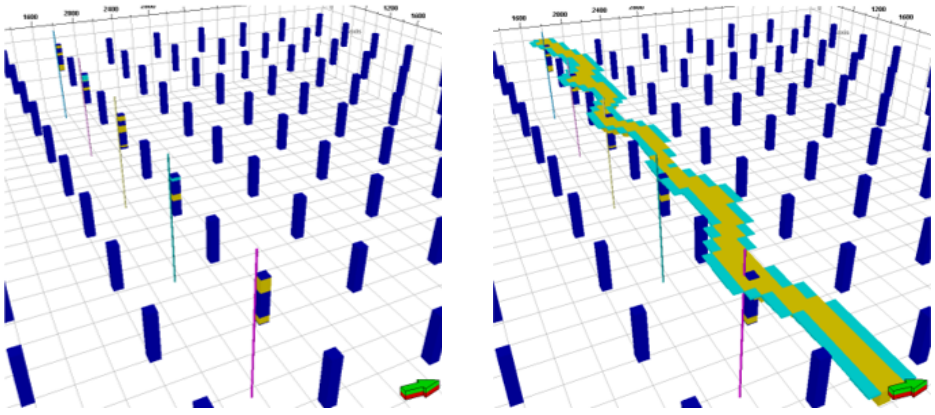


(a) Sinuosity (b) k_{sand}



(c) k_{levee} (d) k_{shale}

Figure 5.26: Evolution of model parameters as a function of the ES-MDA step (Case 3: s, k_{sand}, k_{levee}, k_{shale}; $N_e = 400$). True (black horizontal straight lines), ensemble mean (red curves), and ensemble mean ± 3 ensemble standard deviation (blue curves).



(a) Facies upscaled cells (b) Modeled channel

Figure 5.27: An adaptive channel model honored hard data at wells: sand facies (yellow), levee facies (green) and wells (blue).

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

6.1 Assisted History Matching of Channelized Reservoirs Using ES-MDA with Common Basis DCT

We proposed a new implementation method of the discrete cosine transform (DCT), i.e., common basis DCT, to parameterize discrete facies type indicators. Different from conventional implementations of DCT such as truncated and particular basis DCT, common basis DCT is able to capture the geological similarities among different models by constructing a set of common basis functions that describe the features throughout the whole ensemble. The superior properties of common basis DCT was demonstrated by a reconstruction experiment where the continuity of geological structures is better preserved with common basis DCT.

A history matching workflow for multi-facies channelized reservoirs was presented by combining the common basis DCT parameterization and the ensemble smoother with multiple data assimilation (ES-MDA) algorithm. After every iteration step, an improved post-processing technique is used to map updated continuous variables to discrete facies indicators by considering the prior facies proportions.

The proposed workflow was tested on two synthetic examples. One example is a 2D three-facies reservoir with complex channel distributions and the other example considers a 3D five-layer reservoir with two distinct geological zones. The computational results illustrate that the common basis DCT outperforms the other DCT implementations in the aspects of data match and preservation of geological realism. We also investigated the influence of ensemble size and retained number of DCT basis functions on the performance of history matching workflow. The results show that the ensemble collapse problem can be mitigated

when a large ensemble is used.

6.2 Simultaneous Calibration of the Distribution of Facies and Petrophysical Properties with Non-Negative Matrix Factorization

We introduced the non-negative matrix factorization (NMF) for parameterization of facies fields and history matching of channelized reservoirs. By applying the multiplicative update rules, NMF is able to extract local features of model parameters by imposing the non-negativity constraints on the components in the basis and coefficient matrices. The results of a reconstruction experiment illustrate that the retained number of basis vectors, N_c, has critical impact on the approximation performance of NMF and we developed a heuristic scheme to determine reasonable value of N_c based on requiring that the normalized data mismatch ξ whose value smaller than 0.2.

An integrated history matching workflow combining NMF, ES-MDA and a post-processing technique was presented to calibrate the facies distributions and heterogeneous permeability values within facies simultaneously and was tested on 2D and 3D examples. The computational results show that the key geological structures of true model are mostly resolved in the posterior realizations and we can obtain good data matches as well. Through comparison with common basis DCT and optimization-based principal component analysis (OPCA), NMF presents superior capability of resolving geologic details and preserving the spatial continuity of channelized facies. Since NMF basis tends to describe the local features of the model parameters, a posterior reservoir model can be therefore generated as a positive linear combination of updated NMF basis vectors.

6.3 Parameterization Based on Tensor and Non-Negative Tensor Decomposition for Assisted History Matching of Channelized Reservoirs with Multiple Facies

The recent progress in tensor decomposition (TD) methods allows people to build a multi-dimensional data set and parameterize different dimensions of reservoir models sep-

arately and flexibly. Moreover, its variant, non-negative tensor decomposition (NTD), has gained increasing popularity in various fields but no relevant work has been done to use the NTD for parameterization and history matching. In Chapter 4, we presented the basic concepts, notations and formulations of tensor decomposition. We also discussed different types of tensor and corresponding implementation schemes of TD and NTD. A reconstruction experiment was conducted to compare the performance of various implementations of TD and NTD. An integrated history matching workflow combining TD/NTD, ES-MDA and a post-processing technique was tested on several cases which consider different implementation schemes of TD and NTD, ensemble sizes and sets of wells.

Based on results from the reconstruction experiment and example problems, the following conclusions are drawn:

- The two-mode implementation scheme of TD and NTD gives slightly better model reconstruction quality and data match results than multi-mode schemes, but it takes about 5 times longer to complete two-mode TD and NTD.

- The components in the core tensor and factor matrices of NTD are non-negative and the retained basis functions of two-mode NTD are illustrated to extract and describe latent features hidden in large ensemble, e.g., well locations where facies type observations are available.

- When the retained number of basis functions is identical, the data match results obtained by TD and NTD are better than those obtained by NMF for the same history matching problem.

- The probability maps of channel facies based on posterior realizations provide a good way for uncertainty characterization of facies distributions. In the 100% probability maps, only gridblocks at or very near well gridblocks where the channel facies is observed are occupied by the channel facies when we increase the ensemble size from 200 to 500. This indicates that the probability maps when $N_e = 500$ give better characterization of uncertainty than the $N_e = 200$ case.

- Through visualization and statistical indicators calculated based on the 90% and 80% probability maps, more channel gridblocks in the true model are correctly resolved by NTD than TD. But according to the 0% and 10% probability maps, the proportion of missed channel gridblocks is also higher when implementing NTD.

- In various cases, the performance of TD and NTD are similar in the aspect of data matches and the uncertainty characterization of facies distributions is slightly better with NTD. However, the computing time of NTD can be as 10 times longer as that of TD.

- The ranks of different modes in TD and NTD play an important role in the performance of parameterization. We developed a truncation scheme based on the higher-order SVD (HOSVD) algorithm to determine ranks of different modes in TD and NTD by adjusting a threshold, ζ, in order to balance normalized model mismatch ξ and model reduction ratio η. The usefulness of this truncation scheme was demonstrated by a comprehensive comparison between the TD and NTD with and without truncation.

6.4 Assisted History Matching of Channelized Reservoirs Using the Object-Based Model in Petrel

To completely preserve the spatial continuity of channel facies distribution represented by object-based modeling, we proposed a workflow to adjust by history matching the parameters used to generate a realization of channelized system by object-based modeling in Petrel. The parameters being calibrated include the orientation (θ), amplitude (a), wavelength (λ), sinuosity (s), width (w_{sand}) and thickness (h_{sand}) of sand channels as well as width (w_{levee}) and thickness (h_{levee}) of levee facies.

Although this seems like an obvious approach, the results of a sensitivity analysis show that extremely small changes in some of the parameters (especially the sinuosity s) used in object-based modeling with Petrel lead to an extremely large change in the facies distribution, i.e., the object-based realization is not a continuous or stable function of the

parameters. Due to the internal randomness associated with Petrel modeling process, it is difficult to obtain a reasonable implicit approximation of the gradient in the data assimilation algorithm. Thus, the proposed workflow for history matching of object-based models was not successful.

6.5 Future Work

There are several problems that are not fully resolved by this work and the following topics are suggested for future research:

- Intuitively, one might expect localized basis functions to be superior to global ones for constructing a good approximation of any model. Thus, the properties of localized features retained by NMF and NTD should be further explored.

- We did not really obtain very good history matches for the cases with tensor and non-negative tensor decomposition. More work should be done in order to address this issue including the investigation of truncation level in the SVD used to compute the matrix inverse involved in the ES-MDA update equation.

- Even with the post-processing technique, there still remains small number of gridblocks where the facies type indicator is not discrete and the spatial continuity of geological structures is not guaranteed in the posterior realizations with current workflow. Seismic data should be assimilated as well in order to enhance the preservation of prior geological realism.

- In this work, we applied the parameterization methods to synthetic models. The methodologies and history matching workflow should be extended and tested on more realistic three-dimensional problems.

BIBLIOGRAPHY

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

Afra, S. and E. Gildin, Permeability parametrization using Higher Order Singular Value Decomposition (HOSVD), in *Proceedings of 12th International Conference on Machine Learning and Applications (ICMLA)*, vol. 2, pp. 188–193, IEEE, 2013.

Afra, S. and E. Gildin, Tensor based geology preserving reservoir parameterization with Higher Order Singular Value Decomposition (HOSVD), *Computers & Geosciences*, **94**, 110–120, 2016.

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

Aharon, M., M. Elad, and A. Bruckstein, K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation, *IEEE Transactions on Signal Processing*, **54**(11), 4311, 2006.

Armstrong, M., A. Galli, H. Beucher, G. Loc'h, D. Renard, B. Doligez, R. Eschard, and F. Geffroy, *Plurigaussian simulations in geosciences*, Springer Science & Business Media, 2011.

Bertoncello, A., *Conditioning surface-based models to well and thickness data*, Ph.D. thesis, Stanford University, 2011.

Bhark, E. W., B. Jafarpour, and A. Datta-Gupta, A new adaptively scaled production data

integration approach using the discrete cosine parameterization, in *Proceedings of SPE Improved Oil Recovery Symposium, Tulsa, Oklahoma, 24–28 April*, 2010.

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

Caers, J. and T. Hoffman, The probability perturbation method: A new look at Bayesian inverse modeling, *Mathematical Geology*, **38**(1), 81–100, 2006.

Caers, J. and T. Zhang, *Multiple-point geostatistics: a quantitative vehicle for integrating geologic analogs into multiple reservoir models*, AAPG Special Volumes, 2004.

Canchumuni, S. A., A. A. Emerick, and M. A. Pacheco, Integration of ensemble data assimilation and deep learning for history matching facies models, in *Proceedings of the Offshore Technology Conference, Rio de Janeiro, Brazil, 24-26 October*, 2017.

Canchumuni, S. W., A. A. Emerick, and M. A. C. Pacheco, History matching channelized facies models using ensemble smoother with a deep learning parameterization, in *Proceedings of the 16th European Conference on the Mathematics of Oil Recovery (ECMOR XVI), Barcelona, Spain, 3-6 September*, 2018.

Carroll, J. D. and J.-J. Chang, Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition, *Psychometrika*, **35**(3), 283–319, 1970.

Chan, S. and A. H. Elsheikh, Parametrization and generation of geological models with generative adversarial networks, *arXiv preprint arXiv:1708.01810*, 2017.

Chan, S. and A. H. Elsheikh, Parametric generation of conditional geological realizations using generative neural networks, *arXiv preprint arXiv:1807.05207*, 2018.

Chang, H. and D. Zhang, History matching of facies distribution with varying mean lengths or different principle correlation orientations, *Journal of Petroleum Science and Engineering*, **124**, 275–292, 2014.

Chang, H., D. Zhang, and Z. Lu, History matching of facies distributions with the EnKF and level set parameterization, *Journal of Computational Physics*, **229**, 8011–8030, 2010.

Chen, C., G. Gao, J. Honorio, P. Gelderblom, E. Jimenez, and T. Jaakkola, Integration of principal-component-analysis and streamline information for the history matching of channelized reservoirs, in *Proceedings of the SPE Annual Technical Conference and Exhibition, Amsterdam, The Netherlands, 27–29 October*, 2014.

Chen, C., G. Gao, B. A. Ramirez, J. C. Vink, and A. M. Girardi, Assisted history matching of channelized models by use of pluri-principal-component analysis, *SPE Journal*, **21**(5), 2016.

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

Cichocki, A., R. Zdunek, A. H. Phan, and S.-i. Amari, *Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation*, John Wiley & Sons, 2009.

De Lathauwer, L., B. De Moor, and J. Vandewalle, A multilinear singular value decomposition, *SIAM journal on Matrix Analysis and Applications*, **21**(4), 1253–1278, 2000a.

De Lathauwer, L., B. De Moor, and J. Vandewalle, On the best Rank-1 and Rank-(R_1, R_2, ..., R_N) approximation of higher-order tensors, *SIAM journal on Matrix Analysis and Applications*, **21**(4), 1324–1342, 2000b.

Dempster, A. P., N. M. Laird, and D. B. Rubin, Maximum likelihood from incomplete data

via the EM algorithm, *Journal of The Royal Statistical Society: Series B*, **39**(1), 1–38, 1977.

Deutsch, C. V. and A. G. Journel, Geostatistical software library and users guide, *Oxford University Press*, 1998.

Deutsch, C. V. and L. Wang, Hierarchical object-based stochastic modeling of fluvial reservoirs, *Mathematical Geology*, **28**(7), 857–880, 1996.

Elsheikh, A. H., V. Demyanov, R. Tavakoli, M. A. Christie, and M. F. Wheeler, Calibration of channelized subsurface flow models using nested sampling and soft probabilities, *Advances in Water Resources*, **75**, 14–30, 2015.

Elsheikh, A. H., M. F. Wheeler, and I. Hoteit, Nested sampling algorithm for subsurface flow model selection, uncertainty quantification, and nonlinear calibration, *Water Resources Research*, **49**(12), 8383–8399, 2013.

Emerick, A. A., Investigation on principal component analysis parameterizations for history matching channelized facies models with ensemble-based data assimilation, *Mathematical Geosciences*, **49**(1), 85–120, 2017.

Emerick, A. A. and A. C. Reynolds, History matching time-lapse seismic data using the ensemble Kalman filter with multiple data assimilations, *Computational Geosciences*, **16**(3), 639–659, 2012.

Emerick, A. A. and A. C. Reynolds, Ensemble smoother with multiple data assimilations, *Computers & Geosciences*, **55**, 3–15, 2013a.

Emerick, A. A. and A. C. Reynolds, Investigation of the sampling performance of ensemble-based methods with a simple reservoir model, *Computational Geosciences*, **17**(2), 325–350, 2013b.

Emery, X., Simulation of geological domains using the plurigaussian model: new developments and computer programs, *Computers & Geosciences*, **33**(9), 1189–1201, 2007.

Evensen, G., Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics, *Journal of Geophysical Research*, **99**(C5), 10,143–10,162, 1994.

Evensen, G., J. Hove, H. C. Meisingset, E. Reiso, K. S. Seim, and O. Espelid, Using the EnKF for assisted history matching of a North Sea reservoir model, in *Proceedings of the SPE Reservoir Simulation Symposium, Houston, Texas, 26–28 February*, 2007.

Falivene, O., P. Arbus, A. Gardiner, G. Pickup, J. A. Muoz, and L. Cabrera, Best practice stochastic facies modeling from a channel-fill turbidite sandstone analog (the Quarry outcrop, Eocene Ainsa basin, northeast Spain), *AAPG bulletin*, **90**(7), 1003–1029, 2006.

Gao, G., M. Zafari, and A. C. Reynolds, Quantifying uncertainty for the PUNQ-S3 problem in a Bayesian setting with RML and EnKF, *SPE Journal*, **11**(4), 506–515, 2006.

Gavalas, G. R., P. C. Shah, and J. H. Seinfeld, Reservoir history matching by Bayesian estimation, *SPE Journal*, **16**(6), 337–350, 1976.

Geman, S. and D. Geman, Stochastic relaxation, Gibbs distributions, and Bayesian restoration of images, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **PAMI-6**(6), 721–741, 1984.

Gentilhomme, T., D. S. Oliver, T. Mannseth, G. Caumon, R. Moyen, and P. Doyen, Ensemble-based multi-scale history-matching using second-generation wavelet transform, *Computational Geosciences*, **19**(5), 999–1025, 2015.

Gómez-Hernández, J. J., A. Sahuquillo, and J. E. Capilla, Stochastic simulation of transmissivity fields conditional to both transmissivity and piezometric data. 1. Theory, *Journal of Hydrology*, (203), 162–174, 1997.

Gonzalez, R. C. and R. E. Woods, *Digital image processing*, Prentice hall Upper Saddle River, 2002.

Goovaerts, P., *Geostatistics for natural resources evaluation*, Oxford University Press, 1997.

Gu, Y. and D. S. Oliver, An iterative ensemble Kalman filter for multiphase fluid flow data assimilation, *SPE Journal*, **12**(4), 438–446, 2007.

Guardiano, F. B. and R. M. Srivastava, Multivariate geostatistics: beyond bivariate moments, in *Geostatistics Troia92*, pp. 133–144, Springer, 1993.

Han, M., Y. Zhao, G. Li, and A. C. Reynolds, Application of EM algorithms for seismic facices classification, *Computational Geosciences*, **15**(3), 421–429, 2011.

Harshman, R. A., Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multimodal factor analysis, 1970.

Håstad, J., Tensor rank is NP-complete, *Journal of Algorithms*, **11**(4), 644–654, 1990.

Hastings, W. K., Monte Carlo sampling methods using Markov chains and their applications, *Biometrika*, **57**(1), 97–109, 1970.

Hitchcock, F. L., The expression of a tensor or a polyadic as a sum of products, *Journal of Mathematics and Physics*, **6**(1-4), 164–189, 1927.

Hoyer, P. O., Non-negative matrix factorization with sparseness constraints, *Journal of Machine Learning Research*, **5**(Nov), 1457–1469, 2004.

Hu, L. and T. Chugunova, Multiple-point geostatistics for modeling subsurface heterogeneity: A comprehensive review, *Water Resources Research*, **44**(11), 2008.

Hu, L. Y., Gradual deformation and iterative calibration of Gaussian-related stochastic models, *Mathematical Geology*, **32**(1), 87–108, 2000.

Iglesias, M. A., Iterative regularization for ensemble-based data assimilation in reservoir models, *Computational Geosciences*, **19**, 177–212, 2015.

Jafapour, B. and M. Khodabakhsi, A probability conditioning method (PCM) for nonlinear flow data integration into multipoint statistical facies simulation, *Mathematical Geology*, **43**(2), 133–164, 2011.

Jafarpour, B., Wavelet reconstruction of geologic facies from nonlinear dynamic flow measurements, *IEEE Transactions on Geoscience and Remote Sensing*, **49**(5), 1520–1535, 2011.

Jafarpour, B. and D. B. McLaughlin, Efficient permeability parameterization with the discrete cosine transform, in *Proceedings of the SPE Reservoir Simulation Symposium, Houston, Texas, 26-28 February*, SPE 106453, 2007.

Jafarpour, B. and D. B. McLaughlin, Estimating channelized reservoir permeabilities with the ensemble Kalman filter: The importance of ensemble design, *SPE Journal*, **14**(2), 374–388, 2009a.

Jafarpour, B. and D. B. McLaughlin, Reservoir characterization with the discrete cosine transform, *SPE Journal*, **14**(1), 182–201, 2009b.

Jansen, J., D. Brouwer, G. Naevdal, and C. van Kruijsdijk, Closed-loop reservoir management, *First Break*, **23**, 43–48, 2005.

Jansen, J. D., S. D. Douma, D. R. Brouwer, P. M. J. V. den Hof, and A. W. Heemink, Closed-loop reservoir management, in *Proceedings of the SPE Reservoir Simulation Symposium, The Woodlands, Texas, 2–4 February*, 2009.

Journel, A., Combining knowledge from diverse sources: An alternative to traditional data independence hypotheses, *Mathematical geology*, **34**(5), 573–596, 2002.

Journel, A. and T. Zhang, The necessity of a multiple-point prior model, *Mathematical Geology*, **38**(5), 591–610, 2006.

Kerrou, J., P. Renard, H.-J. H. Franssen, and I. Lunati, Issues in characterizing heterogeneity and connectivity in non-multiGaussian media, *Advances in Water Resources*, **31**(1), 147–159, 2008.

Khaninezhad, M. M., B. Jafarpour, and L. Li, Sparse geologic dictionaries for subsurface flow model calibration: Part I. Inversion formulation, *Advances in Water Resources*, **39**, 106–121, 2012a.

Khaninezhad, M. M., B. Jafarpour, and L. Li, Sparse geologic dictionaries for subsurface flow model calibration: Part II. Robustness to uncertainty, *Advances in Water Resources*, **39**, 122–136, 2012b.

Khaninezhad, M.-R. and B. Jafarpour, Sparse geologic dictionaries for field-scale history matching application, in *Proceedings of the SPE Reservoir Simulation Symposium, Houston, Texas, 23–25 February*, 2015.

Kiers, H. A., Towards a standardized notation and terminology in multiway analysis, *Journal of Chemometrics*, **14**(3), 105–122, 2000.

Kim, Y.-D. and S. Choi, Nonnegative tucker decomposition, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, IEEE, 2007.

Kitanidis, P. K., Quasi-linear geostatistical theory for inversing, *Water Resources Research*, **31**(10), 2411–2419, 1995.

Kolda, T. G. and B. W. Bader, Tensor decompositions and applications, *SIAM review*, **51**(3), 455–500, 2009.

Koltermann, C. E. and S. M. Gorelick, Paleoclimatic signature in terrestrial flood deposits, *Science*, **256**(5065), 1775–1782, 1992.

Kossaifi, J., Y. Panagakis, A. Anandkumar, and M. Pantic, Tensorly: Tensor learning in python, *The Journal of Machine Learning Research*, **20**(1), 925–930, 2019.

Kruskal, J. B., Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics, *Linear algebra and its applications*, **18**(2), 95–138, 1977.

Le, D. H., A. A. Emerick, and A. C. Reynolds, An adaptive ensemble smoother with multiple data assimilation for assisted history matching, in *Proccedings of the SPE Reservoir Simulation Symposium, Houston, Texas, 23-25 February*, 2015a.

Le, D. H., R. Younis, and A. C. Reynolds, A history matching procedure for non-Gaussian facies based on ES-MDA, in *Proceedings of the SPE Reservoir Simulation Symposium, Houston, Texas, 23–25 February*, 2015b.

Lee, D. D. and H. S. Seung, Learning the parts of objects by non-negative matrix factorization, *Nature*, **401**(6755), 788–791, 1999.

Lee, D. D. and H. S. Seung, Algorithms for non-negative matrix factorization, in *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pp. 556–562, 2001.

Lee, S.-Y., S. F. Carle, and G. E. Fogg, Geologic heterogeneity and a comparison of two geostatistical models: Sequential Gaussian and transition probability-based geostatistical simulation, *Advances in Water Resources*, **30**(9), 1914–1932, 2007.

Levin, J., Three-mode factor analysis, *Psychological Bulletin*, **64**(6), 442, 1965.

Li, G. and A. C. Reynolds, Iterative ensemble Kalman filters for data assimilation, *SPE Journal*, **14**(3), 496–505, 2009.

Linde, N., P. Renard, T. Mukerji, and J. Caers, Geological realism in hydrogeological and geophysical inverse modeling: A review, *Advances in Water Resources*, **86**, 86–101, 2015.

Liu, N. and D. S. Oliver, Automatic history matching of geologic facies, *SPE Journal*, **9**(4), 429–436, 2004.

Liu, N. and D. S. Oliver, Critical evaluation of the ensemble Kalman filter on history matching of geologic facies, *SPE Reservoir Evaluation & Engineering*, **8**(4), 470–477, 2005a.

Liu, N. and D. S. Oliver, Ensemble Kalman filter for automatic history matching of geologic facies, *Journal of Petroleum Science and Engineering*, **47**(3–4), 147–161, 2005b.

Liu, Y., A. Harding, W. Abriel, and S. Strebelle, Multiple-point simulation integrating wells, three-dimensional seismic data, and geology, *AAPG bulletin*, **88**(7), 905–921, 2004.

Luo, X., A. S. Stordal, R. J. Lorentzen, and G. Naevdal, Iterative ensemble smoother as an approximate solution to a regularized minimum-average-cost problem: Theory and applications, *SPE Journal*, **20**(5), 962–982, 2015.

Mariethoz, G. and J. Caers, *Multiple-point geostatistics: stochastic modeling with training images*, John Wiley & Sons, 2014.

Metropolis, N., A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, Equations of state calculations by fast computing machines, *Journal of Chemical Physics*, **21**, 1087–1092, 1953.

Moreno, D. and S. I. Aanonsen, Stochastic facies modelling using the level set method, in *Extended abstracts book of EAGE Conference on Petroleum Geostatistics 2007, Cascais, Portugal, September, 10-14*, 2007.

Nicholas, A., P. Ashworth, G. Sambrook Smith, and S. Sandbach, Numerical simulation of bar and island morphodynamics in anabranching megarivers, *Journal of Geophysical Research: Earth Surface*, **118**(4), 2019–2044, 2013.

Nordahl, K. and P. S. Ringrose, Identifying the representative elementary volume for permeability in heterolithic deposits using numerical rock models, *Mathematical geosciences*, **40**(7), 753–771, 2008.

Oliver, D. S., N. He, and A. C. Reynolds, Conditioning permeability fields to pressure data, in *Proccedings of the European Conference for the Mathematics of Oil Recovery*, 1996.

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

Osher, S. and J. A. Sethian, Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulation, *Journal of Computational Physics*, **79**(1), 12–49, 1988.

Paola, C., Quantitative models of sedimentary basin filling, *Sedimentology*, **47**, 121–178, 2000.

Ping, J. and D. Zhang, History matching of channelized reservoirs with vector-based level-set parameterization, *SPE Journal*, **19**(3), 514–529, 2014.

Rafiee, J. and A. C. Reynolds, Theoretical and efficient practical procedures for the generation of inflation factors for ES-MDA, *Inverse Problems*, **33**(11), 115,003, 2017.

Ramanathan, R., A. Guin, R. W. Ritzi, D. F. Dominic, V. L. Freedman, T. D. Scheibe, and I. A. Lunt, Simulating the heterogeneity in braided channel belt deposits: 1. a geometric-based methodology and code, *Water Resources Research*, **46**(4), 2010.

RamaRao, B. S., A. M. LaVenue, G. de Marsily, and M. G. Marietta, Pilot point methodology for automated calibration of an ensemble of conditionally simulated transmissivity fields, 1. Theory and computational experiments, *Water Resources Research*, **31**(3), 475–493, 1995.

Rao, K. R. and P. Yip, *Discrete cosine transform: algorithms, advantages, applications*, Academic press, 1990.

Refsgaard, J. C., E. Auken, C. A. Bamberg, B. S. Christensen, T. Clausen, E. Dalgaard, F. Effersø, V. Ernstsen, F. Gertz, A. L. Hansen, et al., Nitrate reduction in geologically heterogeneous catchments – A framework for assessing the scale of predictive capability of hydrological models, *Science of the Total Environment*, **468**, 1278–1288, 2014.

Remy, N., A. Boucher, and J. Wu, *Applied geostatistics with SGeMS: a user's guide*, Cambridge University Press, 2009.

Reynolds, A. C., M. Zafari, and G. Li, Iterative forms of the ensemble Kalman filter, in *Proceedings of 10th European Conference on the Mathematics of Oil Recovery, Amsterdam, 4–7 September*, 2006.

Sahni, I. and R. N. Horne, Multiresolution wavelet analysis for improved reservoir description, *SPE Reservoir Evaluation & Engineering*, **8**(1), 53–69, 2005.

Sarma, P. and W. H. Chen, Generalization of the ensemble Kalman filter using kernel for non-Gaussian random fields, in *Proccedings of the SPE Reservoir Simulation Symposium, The Woodlands, Texas, 2-4 February*, SPE 119177, 2009.

Sarma, P. and W. H. Chen, Preventing ensemble collapse and honoring multipoint geostatistics with the subspace EnKF/EnS and kernel PCA parameterization, in *Proceedings of the SPE Reservoir Simulation Symposium*, 2013.

Sarma, P., L. J. Durlofsky, and K. Aziz, Kernel principal component analysis for efficient differentiable parameterization of multipoint geostatistics, *Mathematical Geosciences*, **40**, 3–32, 2008.

Sarma, P., L. J. Durlofsky, K. Aziz, and W. Chen, A new approach to automatic history matching using kernel PCA, in *Proceedings of the SPE Reservoir Simulation Symposium*, SPE 106176, 2007.

Sebacher, B. and R. Hanea, Channelized reservoir estimation using a low dimensional parameterization based on High Order Singular Value Decomposition, in *Proceedings of the ECMOR XVI-16th European Conference on the Mathematics of Oil Recovery, Barcelona, Spain, 3–6 September*, 2018.

Sebacher, B., R. Hanea, and A. S. Stordal, An adaptive pluri-Gaussian simulation model for geological uncertainty quantification, *Journal of Petroleum Science and Engineering*, **158**, 494–508, 2017.

Sebacher, B., A. Stordal, and R. Hanea, Complex geology estimation using the iterative adaptive Gaussian mixture filter, *Computational Geosciences*, **20**(1), 133–148, 2016.

Shashua, A. and T. Hazan, Non-negative tensor factorization with applications to statistics and computer vision, in *Proceedings of the 22nd international conference on Machine learning*, pp. 792–799, ACM, 2005.

Sousa, E. and A. C. Reynolds, Combining principal component analysis, a machine learning proxy and Markov chain Monte Carlo for history matching and uncertainty quantification for large scale reservoirs, TUPREP research report, The University of Tulsa, 2019.

Straubhaar, J., A. Walgenwitz, and P. Renard, Parallel multiple-point statistics algorithm based on list and tree structures, *Mathematical Geosciences*, **45**(2), 131–147, 2013.

Strebelle, S., Conditional simulation of complex geological structures using multiple-point statistics, *Mathematical Geology*, **34**(1), 1–22, 2002.

Strebelle, S. and C. Cavelius, Solving speed and memory issues in multiple-point statistics simulation program SNESIM, *Mathematical Geosciences*, **46**(2), 171–186, 2014.

Tahmasebi, P., M. Sahimi, G. Mariethoz, and A. Hezarkhani, Accelerating geostatistical simulations using graphics processing units (GPU), *Computers & Geosciences*, **46**, 51–59, 2012.

Tarantola, A., *Inverse problem theory and methods for model parameter estimation*, SIAM, Philadelphia, USA, 2005.

Tavakoli, R. and A. C. Reynolds, Monte Carlo simulation of permeability fields and reservoir performance predictions with SVD parameterization in RML compared with EnKF, *Computational Geosciences*, **15**(1), 99–116, 2011.

Tavakoli, R., S. Srinivasan, A. H. ElSheikh, and M. F. Wheeler, Efficient integration of production and seismic data into reservoir models exhibiting complex connectivity using an

iterative ensemble smoother, in *Proceedings of the SPE Reservoir Simulation Symposium, Houston, Texas, 23–25 February*, 2015.

Tavakoli, R., S. Srinivasan, and M. F. Wheeler, Rapid updating of stochastic models by use of an ensemble-filter approach, *SPE Journal*, **19**(3), 500–513, 2014.

Thulin, K., G. Li, S. I. Aanonsen, and A. C. Reynolds, Estimation of initial fluid contacts by assimilation of production data with EnKF, in *Proceedings of the SPE Annual Technical Conference and Exhibition, Anaheim, California, 11–14 November*, SPE 109975, 2007.

Thulin, K. and G. Nævdal, Ensemble Kalman filter for field estimation – investigations on the effect of the ensemble size, in *Proceedings of the 10th European Conference on the Mathematics of Oil Recovery*, A009, 2006.

Tucker, L. R., Implications of factor analysis of three-way matrices for measurement of change, *Problems in measuring change*, **15**, 122–137, 1963.

Tucker, L. R., The extension of factor analysis to three-dimensional matrices, *Contributions to mathematical psychology*, **110119**, 1964.

Tucker, L. R., Some mathematical notes on three-mode factor analysis, *Psychometrika*, **31**(3), 279–311, 1966.

van Leeuwen, P. J. and G. Evensen, Data assimilation and inverse methods in terms of a probabilistic formulation, *Monthly Weather Review*, **124**, 2898–2913, 1996.

Vervliet, N., O. Debals, and L. De Lathauwer, Tensorlab 3.0-Numerical optimization strategies for large-scale constrained and coupled matrix/tensor factorization, in *Proceedings of 2016 50th Asilomar Conference on Signals, Systems and Computers*, pp. 1733–1738, IEEE, 2016a.

Vervliet, N., O. Debals, L. Sorber, M. Van Barel, and L. De Lathauwer, Tensorlab 3.0, *available online, URL: www.tensorlab.net*, 2016b.

Vesselinov, V. V., B. S. Alexandrov, and D. O'Malley, Nonnegative tensor factorization for contaminant source identification, *Journal of contaminant hydrology*, **220**, 66–97, 2019.

Vo, H. X. and L. J. Durlofsky, A new differentiable parameterization based on principal component analysis for the low-dimensional representation of complex geological models, *Mathematical Geosciences*, **46**(7), 775–813, 2014.

Vo, H. X. and L. J. Durlofsky, Data assimilation and uncertainty assessment for complex geological models using a new PCA-based parameterization, *Computational Geosciences*, **19**(4), 747–767, 2015.

Vo, H. X. and L. J. Durlofsky, Regularized kernel PCA for the efficient parameterization of complex geological models, *Journal of Computational Physics*, **322**, 859–881, 2016.

Wang, Y., G. Li, and A. C. Reynolds, Estimation of depths of fluid contacts by history matching using iterative ensemble-Kalman smoothers, *SPE Journal*, **15**(2), 509–525, 2010.

Xu, Y. and W. Yin, A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion, *SIAM Journal on imaging sciences*, **6**(3), 1758–1789, 2013.

Zhang, F., A. C. Reynolds, and D. S. Oliver, Evaluation of the reduction in uncertainty obtained by conditioning a 3D stochastic channel to multiwell pressure data, *Mathematical Geology*, **34**(6), 713–740, 2002.

Zhao, Y. and F. Forouzanfar, A simultaneous Bayesian estimation of channelized facies and reservoir properties under prior uncertainty, in *Proceedings of the SPE Europec featured at 79th EAGE Conference and Exhibition, Paris, France, 12–15 June*, 2017.

Zhao, Y., F. Forouzanfar, and A. C. Reynolds, Assisted history matching for multi-facies channelized reservoir using ES-MDA with common basis DCT, in *Proceedings of the EC-MOR XV-15th European Conference on the Mathematics of Oil Recovery, Amsterdam, Netherlands, 29 August – 1 September*, 2016.

Zhao, Y., F. Forouzanfar, and A. C. Reynolds, History matching of multi-facies channelized reservoirs using ES-MDA with common basis DCT, *Computational Geosciences*, **21**(5-6), 1343–1364, 2017.

Zhao, Y., A. C. Reynolds, and G. Li, Generating facies maps by assimilating production data and seismic data with the ensemble Kalman filter, in *Proccedings of the SPE Improved Oil Recovery Symposium, Tulsa, Oklahoma, 20–23 April*, SPE 113990, 2008.

APPENDIX A

DATA ASSIMILATION ALGORITHM

A.1 ES-MDA-GEO

For a vector of model parameters, $\boldsymbol{y}$, with dimension of N_y, the ES-MDA-GEO update equation at the ℓth data assimilation step (iteration) is

$$\boldsymbol{y}_j^{\ell+1} = \boldsymbol{y}_j^{\ell} + \Delta\boldsymbol{Y}^{\ell}\boldsymbol{V}_p\boldsymbol{\Lambda}_p\boldsymbol{\Gamma}_p\boldsymbol{U}_p^{\mathrm{T}}\boldsymbol{C}_{\mathrm{D}}^{-1/2}(\boldsymbol{d}_{\mathrm{uc},j}^{\ell} - \boldsymbol{d}_j^{\ell}), \quad \text{for } j = 1, \cdots, N_e, \tag{A.1}$$

where N_e is the ensemble size; $\boldsymbol{d}_j^{\ell}$ is the N_d-dimensional vector of forecast data corresponding to $\boldsymbol{y}_j^{\ell}$; $\boldsymbol{d}_{\mathrm{uc},j}^{\ell}$ is the observed data perturbed by random noise sampled from the distribution $\mathcal{N}(0, \alpha_\ell \boldsymbol{C}_{\mathrm{D}})$; and $\boldsymbol{C}_{\mathrm{D}}$ is a $N_d \times N_d$ covariance matrix of the observed data measurement errors. For the ensemble of N_e realizations, $\Delta\boldsymbol{Y}^{\ell}$ is the $N_y \times N_e$ matrix defined as

$$\Delta\boldsymbol{Y}^{\ell} = \frac{1}{\sqrt{N_e - 1}}\left[\boldsymbol{y}_1^{\ell} - \bar{\boldsymbol{y}}^{\ell}, \cdots, \boldsymbol{y}_{N_e}^{\ell} - \bar{\boldsymbol{y}}^{\ell}\right], \tag{A.2}$$

where

$$\bar{\boldsymbol{y}}^{\ell} = \frac{1}{N_e}\sum_{j=1}^{N_e}\boldsymbol{y}_j^{\ell}. \tag{A.3}$$

Similarly, we can also define the $N_d \times N_e$ matrix $\Delta\boldsymbol{D}^{\ell}$ as

$$\Delta\boldsymbol{D}^{\ell} = \frac{1}{\sqrt{N_e - 1}}\left[\boldsymbol{d}_1^{\ell} - \bar{\boldsymbol{d}}^{\ell}, \cdots, \boldsymbol{d}_{N_e}^{\ell} - \bar{\boldsymbol{d}}^{\ell}\right], \tag{A.4}$$

where

$$\bar{\boldsymbol{d}}^{\ell} = \frac{1}{N_e}\sum_{j=1}^{N_e}\boldsymbol{d}_j^{\ell}. \tag{A.5}$$

Then, a dimensionless sensitivity matrix, $\boldsymbol{G}_D^\ell$, is given by

$$\boldsymbol{G}_D^\ell = \boldsymbol{C}_{\mathrm{D}}^{-1/2} \Delta \boldsymbol{D}^\ell, \tag{A.6}$$

where $\boldsymbol{C}_{\mathrm{D}}^{-1/2}$ is the inverse of the square root of $\boldsymbol{C}_{\mathrm{D}}$. Generally, a truncated singular value decomposition (TSVD) is used in the update equation for an ensemble-based data assimilation method as follows

$$\boldsymbol{G}_D^\ell = \boldsymbol{U}_p \boldsymbol{\Lambda}_p \boldsymbol{V}_p^{\mathrm{T}}, \tag{A.7}$$

where p denotes the truncation level and is chosen as the smallest positive integer that satisfies

$$\sum_{j=1}^{p} \lambda_j \geq \zeta \sum_{j=1}^{N} \lambda_j, \tag{A.8}$$

where $\zeta = 0.99$ in this work; λ_j is the jth singular value after the SVD of $\boldsymbol{G}_D^\ell$ and the singular values are ordered as

$$\lambda_{\max} = \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N = \lambda_{\min} \geq 0, \tag{A.9}$$

where $N = \min\{N_d, N_e\}$. And $\boldsymbol{\Gamma}_p$ in Eq. A.1 is an $p \times p$ diagonal matrix with its jth diagonal entry equals to

$$\gamma_j = \frac{1}{\lambda_j^2 + \alpha_\ell}. \tag{A.10}$$

The complete derivation procedure and calculation details of the geometric inflation factors, α_ℓ, can be referred to the original paper by Rafiee and Reynolds (2017).

APPENDIX B

TENSOR AND NON-NEGATIVE TENSOR DECOMPOSITION

B.1 Vector Outer Product

For an Nth-order tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, we call it a rank-one tensor if it can be written as the outer product of N vectors, i.e.,

$$\mathcal{A} = \boldsymbol{u}^{(1)} \otimes \boldsymbol{u}^{(2)} \otimes \cdots \otimes \boldsymbol{u}^{(N)}, \tag{B.11}$$

where vectors $\boldsymbol{u}^{(1)} \in \mathbb{R}^{I_1}, \cdots, \boldsymbol{u}^{(n)} \in \mathbb{R}^{I_n}, \cdots, \boldsymbol{u}^{(N)} \in \mathbb{R}^{I_N}$ and the symbol "$\otimes$" represents the vector outer product. The outer product of the first two vectors, $\boldsymbol{u}^{(1)} \in \mathbb{R}^{I_1}$ and $\boldsymbol{u}^{(2)} \in \mathbb{R}^{I_2}$, gives a second-order tensor (matrix), $\mathcal{B} \in \mathbb{R}^{I_1 \times I_2}$, which is given by

$$\mathcal{B} = \boldsymbol{u}^{(1)} \otimes \boldsymbol{u}^{(2)} = \boldsymbol{u}^{(1)} \big(\boldsymbol{u}^{(2)}\big)^{\mathrm{T}}. \tag{B.12}$$

Then, the outer product of the first three vectors, $\boldsymbol{u}^{(1)} \in \mathbb{R}^{I_1}$, $\boldsymbol{u}^{(2)} \in \mathbb{R}^{I_2}$ and $\boldsymbol{u}^{(3)} \in \mathbb{R}^{I_3}$, leads to a three-order tensor, $\mathcal{C} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ where the i_3th "slice" can be obtained by multiplying the second-order tensor, $\mathcal{B}$, with the i_3th element in vector $\boldsymbol{u}^{(3)}$ as follows

$$\mathcal{C}_{::i_3} = u^{(3)}_{i_3} \mathcal{B}, \quad \text{for } i_3 = 1, \cdots, I_3. \tag{B.13}$$

In a similar way, we can obtain the rank-one tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ through the outer product of N vectors. Therefore, each element of the tensor $\mathcal{A}$ is the product of the

corresponding vector elements as

$$a_{i_1 i_2 \cdots i_n \cdots i_N} = u^{(1)}_{i_1} u^{(2)}_{i_2} \cdots u^{(n)}_{i_n} \cdots u^{(N)}_{i_N}, \quad \text{for } i_n \text{ such that } 1 \leq i_n \leq I_n. \tag{B.14}$$

B.2 Tensor Multiplication

Here we consider only the tensor n-mode product, i.e., we provide the equation for multiplying a tensor by a matrix (or a vector) in mode n. The n-mode product of a tensor $\mathcal{B} \in \mathbb{R}^{R_1 \times R_2 \times \cdots \times R_n \times \cdots \times R_N}$ with a matrix $\boldsymbol{U}^{(n)} \in \mathbb{R}^{I_n \times R_n}$ is defined by

$$\left[\mathcal{B} \times_n \boldsymbol{U}^{(n)}\right]_{r_1 \cdots r_{n-1} i_n r_{n+1} \cdots r_N} = \sum_{r_n=1}^{R_n} b_{r_1 \cdots r_{n-1} r_n r_{n+1} \cdots r_N} u_{i_n r_n}, \tag{B.15}$$

leading to a tensor $\mathcal{B} \times_n \boldsymbol{U}^{(n)} \in \mathbb{R}^{R_1 \times R_2 \times \cdots \times I_n \times \cdots \times R_N}$. With the n-mode product, the singular value decomposition of a matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, i.e., $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{V}^{\mathrm{T}}$, can be written as $\boldsymbol{A} = \boldsymbol{\Lambda} \times_1 \boldsymbol{U} \times_2 \boldsymbol{V}$ in the tensor framework with the following property:

$$(\boldsymbol{\Lambda} \times_1 \boldsymbol{U}) \times_2 \boldsymbol{V} = (\boldsymbol{\Lambda} \times_2 \boldsymbol{V}) \times_1 \boldsymbol{U}, \tag{B.16}$$

where $\boldsymbol{U} \in \mathbb{R}^{m \times m}$, $\boldsymbol{\Lambda} \in \mathbb{R}^{m \times n}$ and $\boldsymbol{V} \in \mathbb{R}^{n \times n}$.

B.3 Higher-Order Orthogonal Iteration (HOOI)

In this work, the tensor decomposition (TD) is implemented using the Tucker method (Tucker, 1966) with a Matlab package called "Tensorlab 3.0" (Vervliet et al., 2016a,b) where the HOSVD algorithm (De Lathauwer et al., 2000a) is available.

De Lathauwer et al. (2000b) proposed the higher-order orthogonal iteration (HOOI) algorithm in order to improve the calculations of factor matrices but HOOI is more computationally expensive than HOSVD (Cichocki et al., 2009). In this appendix, the implementation of HOOI algorithm is described in details. Moreover, since there is no HOSVD algorithm for non-negative tensor decomposition (NTD) in Tensorlab 3.0, we employ a Python package called "TensorLy" (Kossaifi et al., 2019) to implement NTD using the HOOI algorithm in

this work.

Given an Nth-order tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, an approximate tensor $\hat{\mathcal{A}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, can be obtained by minimizing the least squares objective function as follows

$$O(\hat{\mathcal{A}}) = ||\mathcal{A} - \hat{\mathcal{A}}||_{\mathrm{F}}^2, \tag{B.17}$$

where the subscript "F" denotes the Frobenius norm. With the Tucker method, the approximate tensor, $\hat{\mathcal{A}}$, can be written as

$$\hat{\mathcal{A}} = \mathcal{B} \times_1 \boldsymbol{U}^{(1)} \times_2 \boldsymbol{U}^{(2)} \cdots \times_N \boldsymbol{U}^{(N)}, \tag{B.18}$$

where the core tensor $\mathcal{B} \in \mathbb{R}^{R_1 \times R_2 \times \cdots \times R_N}$, and the factor matrices $\boldsymbol{U}^{(1)} \in \mathbb{R}^{I_1 \times R_1}$, $\boldsymbol{U}^{(2)} \in \mathbb{R}^{I_2 \times R_2}$, $\cdots$, $\boldsymbol{U}^{(N)} \in \mathbb{R}^{I_N \times R_N}$ have orthonormal columns and represent the feature changes in the 1st, 2nd, $\cdots$, Nth dimension of the tensor, respectively. R_1, R_2, $\cdots$, R_N are the 1-mode, 2-mode, $\cdots$, N-mode rank of $\mathcal{A}$, respectively. In other words, R_1, R_2, $\cdots$, R_N are the column rank of factor matrix $\boldsymbol{U}^{(1)}$, $\boldsymbol{U}^{(2)}$, $\cdots$, $\boldsymbol{U}^{(N)}$, respectively.

Based on the derivations in the original work (De Lathauwer et al., 2000b), the factor matrices are determined in the optimization process while the core tensor is obtained by a theorem giving

$$\mathcal{B} = \mathcal{A} \times_1 (\boldsymbol{U}^{(1)})^{\mathrm{T}} \times_2 (\boldsymbol{U}^{(2)})^{\mathrm{T}} \cdots \times_N (\boldsymbol{U}^{(N)})^{\mathrm{T}}. \tag{B.19}$$

In De Lathauwer et al. (2000b), the objective function defined in Eq. B.17 can be rewritten as

$$O(\hat{\mathcal{A}}) = ||\mathcal{A} - \hat{\mathcal{A}}||_{\mathrm{F}}^2 = ||\mathcal{A}||_{\mathrm{F}}^2 - 2 < \mathcal{A}, \hat{\mathcal{A}} > + ||\hat{\mathcal{A}}||_{\mathrm{F}}^2, \tag{B.20}$$

where the inner product, $< \mathcal{A}, \hat{\mathcal{A}} >$, is obtained with Eqs. B.18 and B.19 as follows

$$\begin{aligned} < \mathcal{A}, \hat{\mathcal{A}} > &= < \mathcal{A}, \mathcal{B} \times_1 \boldsymbol{U}^{(1)} \times_2 \boldsymbol{U}^{(2)} \cdots \times_N \boldsymbol{U}^{(N)} > \\ &= < \mathcal{A} \times_1 (\boldsymbol{U}^{(1)})^{\mathrm{T}} \times_2 (\boldsymbol{U}^{(2)})^{\mathrm{T}} \cdots \times_N (\boldsymbol{U}^{(N)})^{\mathrm{T}}, \mathcal{B} > \\ &= ||\mathcal{B}||_{\mathrm{F}}^2. \end{aligned} \tag{B.21}$$

Since the factor matrices $\boldsymbol{U}^{(1)}, \boldsymbol{U}^{(2)}, \cdots, \boldsymbol{U}^{(N)}$ have orthonormal columns and they do not affect the Frobenius norm, we have $||\hat{\mathcal{A}}||_{\mathrm{F}}^2 = ||\mathcal{B}||_{\mathrm{F}}^2$. Substitution of the above expressions in Eq. B.20 yields

$$\begin{aligned} O(\hat{\mathcal{A}}) &= ||\mathcal{A} - \hat{\mathcal{A}}||_{\mathrm{F}}^2 = ||\mathcal{A}||_{\mathrm{F}}^2 - 2 < \mathcal{A}, \hat{\mathcal{A}} > + ||\hat{\mathcal{A}}||_{\mathrm{F}}^2 = ||\mathcal{A}||_{\mathrm{F}}^2 - 2||\mathcal{B}||_{\mathrm{F}}^2 + ||\mathcal{B}||_{\mathrm{F}}^2 \\ &= ||\mathcal{A}||_{\mathrm{F}}^2 - ||\mathcal{B}||_{\mathrm{F}}^2 = ||\mathcal{A}||_{\mathrm{F}}^2 - ||\mathcal{A} \times_1 (\boldsymbol{U}^{(1)})^{\mathrm{T}} \times_2 (\boldsymbol{U}^{(2)})^{\mathrm{T}} \cdots \times_N (\boldsymbol{U}^{(N)})^{\mathrm{T}}||_{\mathrm{F}}^2. \end{aligned} \tag{B.22}$$

Therefore, the original minimization problem is equivalent to the maximization of the function

$$g(\boldsymbol{U}^{(1)}, \boldsymbol{U}^{(2)}, \cdots, \boldsymbol{U}^{(N)}) = ||\tilde{\mathcal{U}}^{(n)} \times_n (\boldsymbol{U}^{(n)})^{\mathrm{T}}||_{\mathrm{F}}^2, \tag{B.23}$$

where

$$\tilde{\mathcal{U}}^{(n)} = \mathcal{A} \times_1 (\boldsymbol{U}^{(1)})^{\mathrm{T}} \cdots \times_{n-1} (\boldsymbol{U}^{(n-1)})^{\mathrm{T}} \times_{n+1} (\boldsymbol{U}^{(n+1)})^{\mathrm{T}} \cdots \times_N (\boldsymbol{U}^{(N)})^{\mathrm{T}}. \tag{B.24}$$

The complete higher-order orthogonal iteration (HOOI) algorithm, for an Nth-order tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, is presented in Algorithm 1.

Algorithm 1: Higher-Order Orthogonal Iteration (HOOI)

Input: $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$.

Output: $\hat{\mathcal{A}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ which is a rank-$(R_1, R_2, \cdots, R_N)$ approximation of $\mathcal{A}$.

Initialization: $\boldsymbol{U}_0^{(n)} \in \mathbb{R}^{I_n \times R_n}$ for $n = 1, \cdots, N$ using HOSVD.

for $\ell = 0, \cdots, \ell_{max}$ **do**

  $\tilde{\mathcal{U}}_{\ell+1}^{(1)} = \mathcal{A} \times_2 (\boldsymbol{U}_\ell^{(2)})^{\mathrm{T}} \times_3 (\boldsymbol{U}_\ell^{(3)})^{\mathrm{T}} \cdots \times_N (\boldsymbol{U}_\ell^{(N)})^{\mathrm{T}}$;

  Maximize over $\boldsymbol{U}_\ell^{(1)} \in \mathbb{R}^{I_1 \times R_1}$ with constraint $(\boldsymbol{U}_\ell^{(1)})^{\mathrm{T}} \boldsymbol{U}_\ell^{(1)} = \mathbf{I}$ to obtain $\boldsymbol{U}_{\max}^{(1)}$:

  $g(\boldsymbol{U}_{\ell+1}^{(1)}) = ||\tilde{\mathcal{U}}_{\ell+1}^{(1)} \times_1 (\boldsymbol{U}_\ell^{(1)})^{\mathrm{T}}||_{\mathrm{F}}^2$;

  $\boldsymbol{U}_{\ell+1}^{(1)} = \boldsymbol{U}_{\max}^{(1)}$.

  for $n = 2, \cdots, N$ **do**

    $\tilde{\mathcal{U}}_{\ell+1}^{(n)} = \mathcal{A} \cdots \times_{n-1} (\boldsymbol{U}_{\ell+1}^{(n-1)})^{\mathrm{T}} \times_{n+1} (\boldsymbol{U}_\ell^{(n+1)})^{\mathrm{T}} \cdots$;

    Maximize over $\boldsymbol{U}_\ell^{(n)} \in \mathbb{R}^{I_n \times R_n}$ with constraint $(\boldsymbol{U}_\ell^{(n)})^{\mathrm{T}} \boldsymbol{U}_\ell^{(n)} = \mathbf{I}$ to obtain $\boldsymbol{U}_{\max}^{(n)}$:

    $g(\boldsymbol{U}_{\ell+1}^{(n)}) = ||\tilde{\mathcal{U}}_{\ell+1}^{(n)} \times_n (\boldsymbol{U}_\ell^{(n)})^{\mathrm{T}}||_{\mathrm{F}}^2$;

    $\boldsymbol{U}_{\ell+1}^{(n)} = \boldsymbol{U}_{\max}^{(n)}$.

    end (for)

  if *Converged* **then**

    Return: $\boldsymbol{U}^{(1)}, \boldsymbol{U}^{(2)}, \cdots, \boldsymbol{U}^{(N)}$; $\mathcal{B} = \tilde{\mathcal{U}}^{(N)} \times_N (\boldsymbol{U}^{(N)})^{\mathrm{T}}$; $\hat{\mathcal{A}}$

    break;

  end (for)

B.4 Kronecker, Khatri–Rao, and Hadamard Products

The Kronecker product of matrices $\mathbf{A} \in \mathbb{R}^{I \times J}$ and $\mathbf{B} \in \mathbb{R}^{K \times L}$ is denoted by $\mathbf{A} \otimes \mathbf{B}$.

The result is a matrix of size $(IK) \times (JL)$ and defined by

$$\begin{aligned} \mathbf{A} \otimes \mathbf{B} &= \begin{bmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & \cdots & a_{1J}\mathbf{B} \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & \cdots & a_{2J}\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{I1}\mathbf{B} & a_{I2}\mathbf{B} & \cdots & a_{IJ}\mathbf{B} \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{a}_1 \otimes \boldsymbol{b}_1 & \boldsymbol{a}_1 \otimes \boldsymbol{b}_2 & \boldsymbol{a}_1 \otimes \boldsymbol{b}_3 & \cdots & \boldsymbol{a}_J \otimes \boldsymbol{b}_{L-1} & \boldsymbol{a}_J \otimes \boldsymbol{b}_L \end{bmatrix}. \end{aligned} \tag{B.25}$$

The Khatri–Rao product is the "matching columnwise" Kronecker product. Given matrices $\mathbf{A} \in \mathbb{R}^{I \times K}$ and $\mathbf{B} \in \mathbb{R}^{J \times K}$, their Khatri-Rao product is denoted by $\mathbf{A} \odot \mathbf{B}$. The result is a matrix of size $(IJ) \times K$ defined by

$$\mathbf{A} \odot \mathbf{B} = \begin{bmatrix} \boldsymbol{a}_1 \otimes \boldsymbol{b}_1 & \boldsymbol{a}_2 \otimes \boldsymbol{b}_2 & \cdots & \boldsymbol{a}_K \otimes \boldsymbol{b}_K \end{bmatrix}. \tag{B.26}$$

The Hadamard product is the elementwise matrix product. Given matrices $\mathbf{A} \in \mathbb{R}^{I \times J}$ and $\mathbf{B} \in \mathbb{R}^{I \times J}$, their Hadamard product is denoted by $\mathbf{A} \circledast \mathbf{B}$. The result is also of size $I \times J$ and defined by

$$\mathbf{A} \circledast \mathbf{B} = \begin{bmatrix} a_{11}b_{11} & a_{12}b_{12} & \cdots & a_{1J}b_{1J} \\ a_{21}b_{21} & a_{22}b_{22} & \cdots & a_{2J}b_{2J} \\ \vdots & \vdots & \ddots & \vdots \\ a_{I1}b_{I1} & a_{I2}b_{I2} & \cdots & a_{IJ}b_{IJ} \end{bmatrix}. \tag{B.27}$$