THE UNIVERSITY OF TULSA
THE GRADUATE SCHOOL

DATA ASSIMILATION AND UNCERTAINTY QUANTIFICATION WITH ENSEMBLE METHODS AND MARKOV CHAIN MONTE CARLO

by
Javad Rafiee

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

The Graduate School
The University of Tulsa

2017

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

THE GRADUATE SCHOOL

DATA ASSIMILATION AND UNCERTAINTY QUANTIFICATION WITH ENSEMBLE METHODS AND MARKOV CHAIN MONTE CARLO

by
Javad Rafiee

A DISSERTATION

APPROVED FOR THE DISCIPLINE OF

PETROLEUM ENGINEERING

By Dissertation Committee

Albert C. Reynolds, Chair
Mustafa Onur
Rami Younis
Richard Redner
Brett McKinney

COPYRIGHT STATEMENT

ABSTRACT

Javad Rafiee (Doctor of Philosophy in Petroleum Engineering)

Data Assimilation and Uncertainty Quantification with Ensemble Methods and Markov Chain Monte Carlo

Directed by Albert C. Reynolds

146 pp., Chapter 5: Conclusions

(555 words)

The ensemble smoother with multiple data assimilation (ES-MDA) has proved to be a powerful assisted history-matching method. The only drawback of ES-MDA is that the inflation factors for damping the changes in model parameters have to be determined before starting the history-match. Although various authors have provided suggestions for determining the inflation factors adaptively as the history-match proceeds, these methods often result in a large number of data assimilation steps which can make ES-MDA too computationally inefficient for practical application to large-scale field problems. Here, we provide a theoretical procedure to determine exactly the minimum inflation factor at each data assimilation step that ensures the discrepancy principle is satisfied. Like previous adaptive ES-MDA methods, this theoretical method does not allow one to specify a priori the number of data assimilation steps to be done. Thus, using the exact theoretical procedure as a guide, we provide a practical efficient method for determining the inflation factors which allows one to specify a priori the number of data assimilation steps to be done with ES-MDA which still ensures that the initial inflation factor is chosen so that the discrepancy principle is approximately satisfied. Based on the number of data assimilation steps that the user has specified, the subsequent inflation factors are chosen so they decrease geometrically with data assimilation step in a way such that the sum of the inverse inflation factors is

equal to unity. We refer to our proposed practical method as ES-MDA-GEO. To illustrate the performance of this method, we compare the history matching results of our proposed method with ES-MDA with equal inflation factors as well as other forms of adaptive ES-MDA for three synthetic examples. The results of history matching for three numerical examples show that our new method for choosing inflation factors is computationally efficient and provides a good data match while maintaining a seemingly reasonable degree of uncertainty in the posterior realizations of reservoir model and preserving the features of the assumed geological model.

Ensemble-based methods such as ES, EnKF, or ES-MDA provide a reasonably well approximation of the posterior uncertainty, but they fail when applied to multimodal posteriors arising from nonlinear forward model operators. A full characterization of multimodal posterior distribution is only possible with Markov chain Monte Carlo (MCMC) methods. However, the proposal distribution for MCMC need to be able to generate samples from different modes while it should result in a reasonable acceptance rate to avoid computationally prohibitive long chains. Evaluating the acceptance probability using Metropolis-Hastings criteria requires one forward model run (a reservoir simulation in this work) for each proposed state, and MCMC application is only feasible if the chain start sampling from the target pdf quickly. We introduce a two-level MCMC which is able to sample multimodal posteriors efficiently. In first step, we use distributed Gauss-Newton (DGN) method to generate many modes of the posterior pdf in parallel without the need for an adjoint solution. A Gaussian mixture model (GMM) is then constructed based on the distinct modes that we find in the first step. In the second step, the constructed GMM is used as the proposal distribution for our MCMC algorithm. The application of our method to the test problems shows that this proposed two-level MCMC is much more efficient than the random walk MCMC, however its application to high dimensional problems needs further improvements.

ACKNOWLEDGEMENTS

Firstly, I would like to express my sincere gratitude to my advisor Prof. Albert Reynolds for the continuous support of my Ph.D study and related research, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this dissertation.

Besides my advisor, I would like to thank Drs. Mustafa Onur, Rami Younis, Richard Redner and Brett McKinney for serving as members of my dissertation committee and for their insightful comments and encouragement. 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 at The University of Tulsa.

I gratefully acknowledge financial support from the member companies of the University of Tulsa Petroleum Reservoir Exploitation Projects (TUPREP).

I would like to thank my family: my parents and my brothers and sisters for supporting me spiritually throughout writing this dissertation and my life in general.

This dissertation is dedicated to my wife, Mobina, for her unconditional love and support.

Contents

List of Tables

List of Figures

Chapter 1

INTRODUCTION

1.1 Literature Review

An important aspect of reservoir management is the quantification of the uncertainty in the predictions of the future performance of a field for different development scenarios. The risk associated with decisions for field development depends on the level of uncertainty in the reservoir models and in the associated predicted reservoir performances. History matching (data assimilation) refers to the process of adjusting reservoir model parameters until the predicted data from the simulator is in satisfactory agreement with the inaccurate observed data. To be acceptable, history-matched models should reflect the basic features of the reservoir description developed by geologists. The history-matched models are used for well placement and/or well control optimization in a closed-loop reservoir management framework to find an optimum field development scenario. The history matching problem is an ill-posed inverse problem in the sense that there exist infinite number of solutions which give acceptable data matches due to the high dimension of the parameter space, inaccuracy of the data, the limited number of independent data, and the existence of parameters which have little or no influence on the predicted data. Bayesian statistics provides an elegant framework to formulate the history matching problem. According to Bayes' theorem, the posterior probability density function (pdf) for the model parameters conditional to the observed data can be defined up to a normalizing constant as the product of the prior pdf and the likelihood function. Therefore, the uncertainty quantification problem reduces to sampling this posterior pdf [71]. The set of samples from the posterior pdf (conditional realizations of the model parameters) can be used to quantify the uncertainty in the future performance of the reservoir model and determine the risk associated with field development

decisions.

1.1.1 Markov Chain Monte Carlo

The most common rigorous method available for sampling a target pdf is Markov chain Monte Carlo (MCMC). The goal of MCMC is to approximate a target pdf by generating a Markov chain (the probability of a new state depends only on the last state in the chain) with its stationary distribution equal to the target pdf which we wish to sample. It has been theoretically proved that a well designed MCMC will sample the target posterior pdf correctly as the number of states in the chain approaches infinity [44]. However, it might be necessary to generate millions of states in order for the Markov chain to get to the stationary state where the samples in the chain represent samples from the correct pdf [24, 59]. A complete characterization of the posterior pdf using MCMC for a large scale problem can be computationally infeasible, because a forward model run is required for each proposed state to calculate the likelihood of the proposed state. To improve the performance of MCMC one can improve the computational efficiency by reducing the cost of running the forward model. One idea is to build a fast proxy model to serve as the forward model [10, 46], but these methods introduce modeling error into the uncertainty quantification process. Ma et al. [61] proposed a two-stage sampling method for sampling the permeability field conditional to the production data. In their two-stage method for estimation of permeability field of a fine-grid model, the data mismatch of each new proposal is estimated from the results obtained using a coarse-grid model with upscaled permeability field. If the data mismatch decreases for a proposal, the fine-grid is run for that proposal and the proposed state is accepted or rejected based on Metropolis-Hastings acceptance criteria. Another option to enhance the performance of MCMC method is to improve the proposal distribution such that the chain starts sampling from the target pdf faster. In the ideal case, if the proposal distribution is the same as the target pdf, based on Metropolis-Hastings acceptance criteria any proposed state will be accepted in the chain. Oliver et al. [70] used the Metropolis-Hastings MCMC to sample the posterior log-permeability field conditional to the pressure data in a two-

dimensional single phase problem. In addition to using global and local perturbation methods to propose from the prior Gaussian pdf, they built a Gaussian pdf with the mean of the maximum a posteriori (MAP) estimate and the covariance matrix equal to the inverse of the Hessian matrix evaluated at the MAP estimate. The latter proposal distribution resulted in a higher acceptance rate. Emerick and Reynolds [23] proposed a method combining the EnKF with MCMC which targets improving the MCMC performance by using the EnKF results for the proposal procedure and the calculation of the probability of the new proposed states.

If the relationship between parameters and observations is nonlinear (which is the case in most of the reservoir engineering problems), even with Gaussian prior and measurement errors the posterior pdf may have multiple local modes and sampling a multimodal pdf using MCMC is a difficult task, especially if the modes of the posterior are separated by low probability regions. In this case, the states in the MCMC chain will be sampled around a local mode for a large number of iterations as it is very difficult to make a transition through the low probability region between the modes of the posterior at least using any random walk type of algorithm. Different approaches have been taken to sample such a multimodal posterior. Liang and Wong [57] proposed an evolutionary Monte Carlo (also called population MCMC) based on the conceptual ideas borrowed from simulated annealing and genetic algorithm which can be used to sample a multimodal posterior [63]. Gao et al. [32] used the distributed Gauss-Newton (DGN) method [33] to find multiple minima of the objective function. DGN provides multiple minima as well as the Hessian of the objective function around each minima. After averaging the local minima that are close to each other, a Gaussian mixture model is built to approximate the posterior pdf, where the mean and the covariance of each Gaussian are set to the average local minimum and the inverse of the Hessian at that minimum. Note that Gao et al. [32] did not sample the GMM for uncertainty quantification and assumed that the GMM is the posterior pdf. Li and Reynolds [56] proposed a two-level MCMC method where in the first step, they find many local minima of the objective function using an optimization method starting from different initial guesses.

Then, the set of minima corresponding to a mode are clustered and a Gaussian mixture model (GMM) is constructed to serve as the proposal distribution of the MCMC method. The mean of each Gaussian is set to the local minimum representative of the mean of the cluster and its covariance matrix is defined as inverse of the Hessian evaluated at that minimum. The numerical examples show that their two-level MCMC method requires far less computational cost compared to random-walk MCMC or population MCMC. Note that minimization of the objective function in the first step is usually done with a gradient based method which needs the computation of the sensitivity (or at least the gradient of the objective function) which is only feasible if the adjoint method is available. Unfortunately, the adjoint solution cannot be computed with most commercial simulators.

1.1.2 Randomized Maximum Likelihood

Kitanidis [51] and Oliver et al. [69] independently proposed the randomized maximum likelihood (RML) for generating samples from the posterior pdf defined in a Beysian setting. However, Kitanidis [51] called the algorithm "quasi-linear estimation method". In RML, it is assumed that the distribution of the prior and the measurement error are Gaussian. To generate a sample from the posterior defined in a Bayesian setting, RML minimizes an objective function, which is composed of a model mismatch term and a data mismatch term, using an optimization method, such as Gauss-Newton (GN) or Levenberg-Marquardt (LM), starting from unconditional random realizations of the vector of model parameters drawn from the prior distribution. The observation vector for each minimization is obtained by adding a realization of the measurement error to the observed data. Using two completely different proofs, Oliver [67] and Reynolds et al. [74] showed that for a linear-Gaussian case where the prior model is Gaussian and the forward model is a linear function, RML will sample the posterior pdf correctly. The proof of Reynolds et al. [74] was generalized to the case where the prior mean may be uncertain. Furthermore, it has been shown that RML is able to sample from multiple modes of a multimodal posterior arising from nonlinear toy problems [69, 89, 97]. For the problems where the forward model operator is nonlinear,

although RML samples are from the regions of high probabilities of the target pdf, RML is not guaranteed to generate the correct sampling. In an effort to enhance the sampling performance of RML for nonlinear problems, Stordal and Nævdal [85] presented a generalized version of RML where they multiplied the model mismatch (prior) term of the objective function by a positive weighting factor. Although they provided some theoretical justification for the weighting factor, for real problems, the user need to tune this parameter to obtain a desired sampling performance. As the minimization of the RML objective function at least requires the gradient of the objective function, an ensemble version of RML, which is known as EnRML, was proposed which does not need any gradient computations [14]. Chen and Oliver [14] proposed approximate version of EnRML where the model mismatch term is dropped in the update equation as its calculation requires the inversion of the low rank ensemble representation of the model covariance matrix. Since the number measurements is commonly smaller than the number of model parameters, Stordal and Nævdal [85] proposed a modification of EnRML where the parameter space is mapped to the measurement space to avoid the need for the prior covariance matrix. In this version of EnRML there is no need to drop the model mismatch term, and they showed for toy problems and a synthetic reservoir model, this method with a proper choice of the weighting factor can perform as well as the original EnRML and slightly better than approximate version of EnRML. Oliver [68] suggested an augmented variable RML approach which, if tuned properly, is able to sample multimodal distributions which arise from nonlinear forward models. The methodology is tested for few toy problems however it seems impractical for real applications as it requires the determinant of a Jacobian matrix.

1.1.3 Ensemble Smoother

Among the history matching methods, ensemble-based methods such as ensemble Kalman filter [27, 28] and ensemble smoother [93] have attracted particular attention in the field of petroleum engineering in the last decade. By approximating the first two moments of a target probability density function (pdf), commonly referred to as posterior pdf, with

an ensemble of limited samples of the vector of model parameters, EnKF and ES attempt to characterize the target pdf. The final ensemble of reservoir model parameters after a successful history-match with EnKF or ES provides an approximate quantification of the posterior pdf of reservoir model parameters as well as the uncertainty in production forecasts. EnKF updates the model parameters and state variables (e.g., primary variables of the reservoir simulator) sequentially in time. EnKF assumes that the updated (analyzed) states at a data assimilation step are statistically consistent with the updated states that would be obtained by running the forward model from time zero with the updated realizations of the vector of model parameters. Although EnKF often appears to perform well even in field cases [1, 7, 21, 45], this statistical consistency can only be proved in the linear-Gaussian case [90]. Moreover, in the cases where the structural features [78] or initial depths of fluid contacts [96] are included as reservoir model parameters, this statistical consistency does not hold and the only way to obtain reliable results is by rerunning the forward model from time zero with each updated vector of model parameters. The ensemble smoother (ES), on the other hand, is a pure parameter estimation algorithm which does not update the state variables; the ES assimilates all available data simultaneously and only updates the model parameters. Although ES does not suffer from the inconsistency problem, its final data match is usually not as good as the data match obtained with EnKF. To further improve the performance of ES in terms of the data match, Chen and Oliver [14], Emerick and Reynolds [25, 26] and Luo et al. [60] proposed iterative forms of ES where the iterations are performed similar to those done with iterative optimization algorithms, e.g., Gauss-Newton or Levenberg-Marquardt (LM). A popular iterative form of ES is the ensemble smoother with multiple data assimilation (ES-MDA) [22, 25]. The development of ES-MDA was motivated by the work of Reynolds et al. [75] who showed that the ES update is similar to taking a full-step Gauss-Newton iteration using the same average sensitivity matrix to update each ensemble member. ES-MDA is also a logical generalization of a result presented in Rommelse [77] for a one-parameter, one-datum example. The update equation of ES-MDA is similar to the update equation of ES except that the measurement error covariance is multiplied by an inflation factor at each

update. Emerick and Reynolds [25] showed that ES-MDA will sample the posterior correctly for the linear-Gaussian case as the ensemble size approaches infinity if the sum of the inverse of the inflation factors is equal to one. Although ES-MDA has proved to be a promising method for history matching synthetic and field cases [24, 25, 26], the main drawback of ES-MDA is that the number of assimilation steps and the inflation factors for each data assimilation step have to be specified before starting data assimilation (history-matching). A simple choice of inflation factors is to set all of them equal to N_a where N_a is the number of data assimilation steps and is defined prior to beginning data assimilation. Although this choice of inflation factors often seemingly works well even in field cases, it can cause over-correction of model parameters and lead to unreasonably rough rock property fields [52]. To avoid such issues with ES-MDA, Le et al. [52] proposed two adaptive ES-MDA algorithms in which the inflation factors are selected adaptively during the data-assimilation process. The first method (ES-MDA-RS) determines the inflation factor at each iteration to ensure that the change in the model parameters at each assimilation step is less than a predetermined level (e.g., two prior standard deviations). The second method (ES-MDA-RLM) was inspired by the works of Hanke [41] and Iglesias and Dawson [49] in which the inflation factors were selected based on the discrepancy principle and is conceptually similar to a method proposed earlier by Iglesias [48]. Although the proposed adaptive methods improve the performance of ES-MDA [52], they often require a large number of assimilation steps which may not be feasible for real large-scale field problems. Emerick [20] also proposed an adaptive ES-MDA algorithm where the inflation factor in each data assimilation step is determined as the minimum of the normalized objective function times a heuristical factor (a factor of 0.25 is used in the examples) and a predefined maximum allowable value of each inflation factor (a maximum value of 1000 was used). The iterations end when the summation of inverse inflation factors is equal to one. Chen and Oliver [14] started from the Levenberg-Marquardt (LM) algorithm and derived an iterative ES algorithm which they called LM-ensemble randomized likelihood (LM-EnRML). They avoided the explicit computation of the sensitivity matrix by modifying the approximate Hessian which makes the computations simpler and promotes

stability of the scheme. They used standard suggestions for updating the LM parameter, λ, in LM-EnRML and their iterative process stops based on the magnitude of the data mismatch. As shown in this dissertation, the update equation for the approximate version of EnRML proposed by Chen and Oliver [14], where the model mismatch term of the objective function is dropped is similar to the update equation of ES-MDA. Le et al. [52] compared LM-EnRML with their adaptive ES-MDA schemes for a synthetic three-phase model and found that LM-EnRML results in slightly rougher property maps (overshooting and undershooting) and higher values of the normalized objective function (poorer data match) than those obtained with a Le et al. [52] adaptive ES-MDA algorithm. More importantly, Le et al. [52] found that the performance of LM-EnRML can be highly dependent on the initial choice of λ and for some choices, the algorithm fails to give an acceptable history match. Motivated by the works of Hanke [41] and Iglesias and Dawson [49], Iglesias [48] proposed iterative regularizing ES (IR-ES) as an iterative ensemble smoother. The update equation of the IR-ES is the same as ES-MDA update equation, but the discrepancy principle for iterative regularization is utilized for the choice of the regularizing parameters (or inflation factors) and is also used as the stopping criterion of the algorithm.

1.2 Research Overview

The main objectives of this research are as follows: (i) propose theoretical and efficient practical methods for determining the inflation factors for ES-MDA to improve its performance; (ii) propose a practical methodology to find multiples modes of the posterior pdf when the adjoint solution is not available and use these modes to design an efficient MCMC. This second objective is motivated by the fact that ES-MDA only provides an approximate characterization of uncertainty.

1.3 Dissertation Organization

This dissertation is organized into five chapters, including the Introduction and Conclusions. Chapter 2 presents the theoretical framework for history matching in Bayesian

framework. In this chapter, the formulation of gradient-based history matching, ensemble smoother, and Markov chain Monte Carlo are discussed in detail and connections between between different methods are established. In Chapter 3, a theoretical procedure for choosing the inflation factors for ES-MDA is discussed. Using this theoretical procedure as a guide, a practical efficient method for choosing the inflation factor for ES-MDA is proposed. Chapter 4 presents an efficient two-level MCMC algorithm for sampling multimodal posterior distributions which does not require the availability of the adjoint solution. Chapter 5 presents the conclusions.

Chapter 2

BAYESIAN FRAMEWORK FOR HISTORY MATCHING AND UNCERTAINTY QUANTIFICATION

2.1 Notation and Bayesian Viewpoint

Even though observed data comes from a true model m_{true}, due to the uncertain noise in measurements, the uncertain relationship between data and the fact that, for practical problems, even noise free data and a perfectly known relationship between data cannot resolve the true model, it is convenient to view the inverse problem of estimating a model consistent with observed data in a Bayesian framework [71, 87]. In the Bayesian setting, there generally exist an infinite number of models that are consistent with the data and the prior model and the objectives are to generate some type of estimate of the true model and to quantify the uncertainty in the estimate. Ensemble-based data assimilation methods can be conveniently embedded in this Bayesian framework.

Throughout m denotes an N_m-dimensional random column vector of model parameters which is referred to simply as the model m. The N_d-dimensional column vector d_{obs} denotes a vector of observations and for any given m,

$$d^f = d^f(m) = g(m), \tag{2.1}$$

denotes the N_d-dimensional column vector of predicted data corresponding to d_{obs} where $d^f(m)$ is evaluated using the forward model. Here, we temporarily introduce the standard but superfluous notation $g(m)$ to denote the forward model (for example, a numerical reservoir simulator) that enables the computation of $d^f(m)$ given m. We assume that the probability density function (pdf) for ξ, which denotes the N_d-dimensional column vector of measurement

errors, is $\mathcal{N}(0, C_D)$, i.e., is a multivariate Gaussian distribution with zero mean and $N_d \times N_d$ covariance matrix C_D. The relationship between d_{obs} and the true model can be represented by

$$d_{\text{obs}} = d^f(m_{\text{true}}) + \xi = g(m_{\text{true}}) + \xi, \tag{2.2}$$

where ξ is a specific sample of the random vector of measurement errors generated by sampling $\mathcal{N}(0, C_D)$. A common approach is to assume there is a prior pdf for m so the posterior pdf (pdf conditional to d_{obs}), denoted by $f(m \,|\, d_{\text{obs}})$, is proportional to the prior pdf times the likelihood of the model given d_{obs}. From a Bayesian inverse theory viewpoint, we wish to sample this posterior pdf to characterize the posterior uncertainty in the model and the uncertainty in future predictions generated from the forward model $g(m)$. The linear-Gaussian case refers to the scenario where the prior pdf is Gaussian with mean m_{pr} and covariance matrix C_M and $g(m)$ is linear, e.g., $g(m) = Gm$. In the linear-Gaussian case, the ensemble smoother (ES) and ES-MDA generate a theoretically correct sampling of the posterior pdf as the ensemble size goes to infinity. In the nonlinear case, no such theory exists; nevertheless, ES, ES-MDA and EnKF, have proven to generate useful results for many synthetic and field cases [7, 15, 20, 21, 26, 45, 82], even though one can construct simple synthetic multimodal cases where the ensemble-based methods give a poor representation of the posterior pdf [97]. In the Bayesian viewpoint, a common approach is to assume there is a prior pdf for m so the posterior pdf (pdf conditional to d_{obs}), denoted by $f(m|d_{\text{obs}})$ or $\pi(m)$, is given by

$$\pi(m) \equiv f(m|d_{\text{obs}}) = af(m)\mathcal{L}(m|d_{\text{obs}}), \tag{2.3}$$

where $f(m)$ is the prior pdf, $\mathcal{L}(m|d_{\text{obs}})$ is the likelihood of the model given d_{obs}, and a is a normalizing constant. From a Bayesian inverse theory viewpoint, we wish to sample this posterior pdf to characterize the posterior uncertainty in the model and the uncertainty in future predictions generated from the forward model $g(m)$. Assuming a Gaussian prior pdf with mean m_{pr} and covariance matrix C_M and Gaussian measurement errors, the posterior

pdf can be written as

$$\pi(m) = a \exp\{-O(m)\}, \tag{2.4}$$

where

$$O(m) = \frac{1}{2} (m - m_{\text{pr}})^T C_M^{-1} (m - m_{\text{pr}}) + \frac{1}{2} (g(m) - d_{\text{obs}})^T C_D^{-1} (g(m) - d_{\text{obs}}) . \tag{2.5}$$

In Eq. 2.5, $O(m)$ is called the objective function is composed of two terms: a model mismatch term and a data mismatch term. Instead of maximizing the posterior pdf $f(m|d_{\text{obs}})$, one can minimize the objective function. In other words, the aim of minimizing the objective function is to find models which are close to the prior mean (to preserve the geological features of the prior) and match the observed data (low value of data mismatch term). In typical history matching problems, the number of independent data is lower than the number of parameters which leads to an ill-posed inverse problem. Note that the model mismatch term can be regarded as a regularization term which helps alleviate the ill-posedness of the problem. To characterize the posterior uncertainty in m, we wish to generate multiple samples of $\pi(m) = f(m|d_{\text{obs}})$, i.e., multiple (posterior) realizations of m.

One method to generate approximate conditional realizations of the posterior pdf is called Randomized Maximum Likelihood (RML) which was independently introduced by Kitanidis [51] and Oliver et al. [69]. In the first step of RML, an ensemble of unconditional realizations of the model parameters are generated using the prior distribution, i.e., $m_{\text{uc,j}} \sim \mathcal{N}(m_{\text{pr}}, C_M)$ for $j = 1, 2, \ldots, N_e$. We also generate realizations of the observed data by adding random realizations of the measurement error to d_{obs}, i.e., $d_{\text{uc,j}} \sim \mathcal{N}(d_{\text{obs}}, C_D)$ for $j = 1, 2, \ldots, N_e$. In the second step of RML, the conditional samples of the posterior pdf are obtained by minimizing the following stochastic objective function

$$O_j(m) = \frac{1}{2} (m - m_{\text{uc,j}})^T C_M^{-1} (m - m_{\text{uc,j}}) + \frac{1}{2} (g(m) - d_{\text{uc,j}})^T C_D^{-1} (g(m) - d_{\text{uc,j}}) , \tag{2.6}$$

which results in generating N_e conditional samples of the model parameters. Although RML

can be used to generate approximate samples of the posterior pdf in general, RML samples the posterior pdf correctly only in the case where prior model and measurement errors are Gaussian and the forward model is linear [71]. The strength of RML rests on the fact that it generates samples distributed around the modes of the posterior pdf.

2.2 Maximum Likelihood Estimate

Although the ultimate goal is to characterize the posterior pdf by sampling the pdf, this section is devoted to the viability of optimization methods to minimize the objective function $O(m)$. The minimizer of $O(m)$ is called the maximum a posteriori (MAP) estimate which is the mean of the posterior pdf in the linear Gaussian case. Let's assume that the forward model is linear, i.e., $g(m) = Gm$, where G is a $N_d \times N_m$ matrix mapping the parameter space onto the data space. In order to find the MAP estimate, we have to set the gradient of the objective function with respect to the model parameters equal to zero. The gradient of $O(m)$ is given by

$$\nabla O = C_M^{-1}\left(m - m_{\text{pr}}\right) + G^T C_D^{-1}\left(Gm - d_{\text{obs}}\right), \tag{2.7}$$

and it is straightforward to show that the Hessian matrix is

$$H = \nabla\left[(\nabla O)^T\right] = C_M^{-1} + G^T C_D^{-1} G. \tag{2.8}$$

It can be shown that the Hessian matrix, H, is positive definite; therefore the objective function for the linear case has a unique global minimum which is found by setting the gradient equal to zero. Setting the gradient equal to zero and adding $G(m_{\text{pr}} - m_{\text{pr}})$ to the data mismatch term in Eq. 2.7, we obtain

$$C_M^{-1}\left(m - m_{\text{pr}}\right) + G^T C_D^{-1}\left(Gm - d_{\text{obs}} + G(m_{\text{pr}} - m_{\text{pr}})\right) = 0, \tag{2.9}$$

or equivalently,

$$\left(C_M^{-1} + G^T C_D^{-1} G\right)(m - m_{\text{pr}}) + G^T C_D^{-1}(G m_{\text{pr}} - d_{\text{obs}}) = 0. \tag{2.10}$$

We solve Eq. 2.10 for m and denote the result as m_{MAP} to obtain

$$m_{\text{MAP}} = m_{\text{pr}} + \left(C_M^{-1} + G^T C_D^{-1} G\right)^{-1} G^T C_D^{-1}(d_{\text{obs}} - G m_{\text{pr}}). \tag{2.11}$$

In large scale problems where the number of model parameters, N_m, is much larger than the number of data, N_d, the inversion of $N_m \times N_m$ matrix is prohibitive. The following two matrix inversion lemmas are extensively employed to find equivalent forms of the solution of the inverse problems in order to facilitate an efficient computation of the inverse. The proofs can be found in Oliver et al. [71, Chap. 7].

Lemma 2.1 (First matrix inversion identity)**.** *Assume the $N_m \times N_m$ matrix C_M and the $N_d \times N_d$ matrix C_D are both positive definite. Then for any $N_d \times N_m$ matrix G, the following matrix identity holds:*

$$\left(C_M^{-1} + G^T C_D^{-1} G\right)^{-1} G^T C_D^{-1} = C_M G^T \left(C_D + G C_M G^T\right)^{-1}. \tag{2.12}$$

Lemma 2.2 (Second matrix inversion identity)**.** *Assume the $N_m \times N_m$ matrix C_M and the $N_d \times N_d$ matrix C_D are both positive definite. Then for any $N_d \times N_m$ matrix G, the following matrix identity holds:*

$$\left(C_M^{-1} + G^T C_D^{-1} G\right)^{-1} = C_M - C_M G^T \left(C_D + G C_M G^T\right)^{-1} G C_M. \tag{2.13}$$

Note that according to the second matrix inversion identity, the inversion of an $N_m \times N_m$ matrix given on the left-hand side of Eq. 2.13 can be calculated by inverting an $N_d \times N_d$ matrix which is computationally more efficient to perform compared to the direct inversion of the $N_m \times N_m$ matrix if N_d is much smaller than N_m.

Using the first matrix inversion identity given by Eq. 2.12, one can rewrite the expression for the MAP estimate (Eq. 2.11) as

$$m_{\text{MAP}} = m_{\text{pr}} + C_M G^T \left(G C_M G^T + C_D^{-1} \right)^{-1} \left(d_{\text{obs}} - G m_{\text{pr}} \right). \tag{2.14}$$

For the linear case, it can be shown [71, 87] that the posterior pdf can be written as

$$\pi(m) = \widehat{a} \exp\left(-\frac{1}{2} \left(m - m_{\text{MAP}} \right)^T C_{\text{MAP}}^{-1} \left(m - m_{\text{MAP}} \right) \right), \tag{2.15}$$

where C_{MAP} is called the posterior covariance matrix and it is equal to the inverse of the Hessian matrix, i.e.,

$$\begin{aligned} C_{\text{MAP}} &= H^{-1} \\ &= \left(C_M^{-1} + G^T C_D^{-1} G \right)^{-1} \\ &= C_M - C_M G^T \left(C_D + G C_M G^T \right)^{-1} G C_M, \end{aligned} \tag{2.16}$$

where the last equality is obtained using the second matrix inversion identity (Eq. 2.13).

For the case where the forward operator, $g(m)$, is nonlinear, one can use a gradient-based algorithm to minimize the objective function. However, due to the nonlinearity of the forward model, the posterior pdf may have multiple modes (or equivalently the objective function has multiple local minima) and in such cases, the minimum found using gradient-based algorithms depends on the initial guess. In the next subsection, three gradient-based algorithms which can be used for minimization of the objective function defined in Eq. 2.5 are introduced. The same methods can be used to minimize the objective function given by Eq. 2.6 in an RML setting.

2.2.1 Gauss-Newton

Gauss-Newton method is an iterative algorithm which can be used to minimize the objective function given in Eq. 2.5. Starting from an initial guess, m^0, the search direction

of the Gauss-Newton method to find a minimum of the objective function is

$$\delta m^{i+1} = -H_i^{-1} \nabla O(m^i), \tag{2.17}$$

where $\nabla O(m^i)$ and H_i, respectively, are the gradient of the objective function and the Hessian evaluated at m^i (the vector of model parameters at ith iteration). The gradient vector and Hessian matrix are, respectively, given by

$$\nabla O(m^i) = C_M^{-1} \left(m^i - m_{\text{pr}} \right) + G_i^T C_D^{-1} \left(g(m^i) - d_{\text{obs}} \right), \tag{2.18}$$

and

$$H_i = C_M^{-1} + G_i^T C_D^{-1} G_i, \tag{2.19}$$

where here we have used the Gauss-Newton Hessian. In Eqs. 2.18 and 2.19, G_i, is called the sensitivity matrix evaluated at m^i and is defined as

$$G_i = \left[\nabla_m g(m)^T \right]_{m=m^i}^T. \tag{2.20}$$

The sensitivity matrix, G_i, is a $N_d \times N_m$ matrix which has the partial derivative of the ℓth data with respect to the jth model parameter as the entry in the ℓth row and jth column. The most efficient method for generation of the entire sensitivity matrix is using an adjoint formulation. To compute each row of the sensitivity matrix, one adjoint solution is required; so obtaining the entire sensitivity matrix requires N_d adjoint solutions [54]. Once we find the search direction, δm^{i+1}, then we update the model parameters using

$$m^{i+1} = m^i + \mu_i \delta m^{i+1}, \tag{2.21}$$

to obtain the vector of model parameter at the next iteration. In Eq. 2.21 μ_i is the step size and it can be calculated using a line search method (see Oliver et al. [71, Chap. 8]). Alternatively, we can use a trust region algorithm, which will be discussed later in this

chapter, instead of a line search. One can consider multiple stopping criteria to end Gauss-Newton iterative procedure: the norm of δm^{i+1} is less than a threshold, the change in the value of the objective function in two consecutive iteration is less than a threshold, the value of the objective function is less than a threshold, or the norm of the gradient is less than a threshold. The iterations will continue until one of these stopping criteria is satisfied.

The vector of model parameters in the petroleum engineering inverse problems can include different grid properties (e.g., permeability, porosity at grid blocks) which have different scales. One way to normalize the vector of model parameters is to use the following change of parameters

$$\hat{m}^i = C_M^{-1/2}(m^i - m_{\text{pr}}), \tag{2.22}$$

where $C_M^{-1/2}$ denotes the inverse of the square root of the prior covariance matrix. The calculation of $C_M^{-1/2}$ is computationally expensive for large-scale problems. Dickstein et al. [17] proposed Gauss-Newton and nonlinear conjugate gradient algorithms that avoid computation of this square root matrix but their procedure requires the formation and multiplication by C_M at each iteration. Similarly, we can use the following transformation for the vector of data

$$\hat{d}^i = C_D^{-1/2}\left(g(m^i) - d_{\text{obs}}\right), \tag{2.23}$$

where $C_D^{-1/2}$ is called the inverse square root of the measurement error covariance matrix. Since C_D is commonly assumed diagonal (measurements are independent), computation of $C_D^{-1/2}$ is straightforward. With these changes of variables, the objective function of Eq. 2.5 can be written as

$$O(\hat{m}) = \frac{1}{2}\hat{m}^T\hat{m} + \frac{1}{2}\hat{d}^T\hat{d}. \tag{2.24}$$

Note that for a given m and its corresponding $\hat{m}$, the value of $O(m)$, which is given by Eq. 2.5, is equal to the value of $O(\hat{m})$, which is given by Eq. 2.24. Therefore, in the optimization algorithms, we do not need to distinguish between $O(m)$ and $O(\hat{m})$ when the algorithm requires checking the decrease in the objective function. The gradient of objective function,

$O(\hat{m})$, with respect to $\hat{m}$ evaluated at $\hat{m}^i$ and the Hessian , respectively, are given by

$$\nabla_{\hat{m}} O(\hat{m}^i) = \hat{m}^i + G_{D,i}^T \hat{d}^i, \tag{2.25}$$

and

$$\hat{H}_i = I_{N_m} + G_{D,i}^T G_{D,i}, \tag{2.26}$$

where I_{N_m} is the $N_m \times N_m$ identity matrix and $G_{D,i}$ is called the dimensionless sensitivity matrix at the ith iteration and is defined as

$$G_D = \left[\nabla_{\hat{m}} (\hat{d})^T\right]^T, \tag{2.27}$$

where we temporally drop the subscript i for simplicity. Applying the chain rule we can write Eq. 2.27 as

$$\begin{aligned} G_D &= \left[\nabla_m (\hat{d})^T\right]^T \left[\nabla_{\hat{m}} m^T\right]^T \\ &= \left[\nabla_m \left(C_D^{-1/2}(g(m) - d_{\text{obs}})\right)^T\right]^T \left[\nabla_{\hat{m}} \left(m_{\text{pr}} + C_M^{1/2}\hat{m}\right)^T\right]^T \\ &= \left[G^T C_D^{-1/2}\right]^T C_M^{1/2} \\ &= C_D^{-1/2} G C_M^{1/2}. \end{aligned} \tag{2.28}$$

If we put the subscript i back, the final expression for the dimensionless sensitivity matrix is given by

$$G_{D,i} = C_D^{-1/2} G_i C_M^{1/2}. \tag{2.29}$$

The dimensionless sensitivity matrix given in Eq. 2.29 was first introduced by Zhang et al. [98] for gradient-based history matching. Tavakoli and Reynolds [88, 89] showed that the dimensionless sensitivity given by Eq. 2.29 controls how much the parameters will be changed to match the data during history matching. Moreover, the truncated singular value decomposition (TSVD) of the dimensionless sensitivity has been used to reparameterize the Gauss-Newton or Levenberg-Marquardt algorithms [79, 80, 81, 88, 89]. Similar to Eq. 2.17,

the search direction of the Gauss-Newton method is

$$\delta\hat{m}^{i+1} = -\left(I_{N_m} + G_{D,i}^T G_{D,i}\right)^{-1}\left(\hat{m}^i + G_{D,i}^T \hat{d}^i\right). \tag{2.30}$$

To obtain the updated vector of model parameter, m^{i+1}, we have

$$m^{i+1} = m^i + \mu_i C_M^{1/2}\delta\hat{m}^{i+1}. \tag{2.31}$$

2.2.2 Levenberg-Marquardt

For the Levenberg-Marquardt algorithm, we first describe the method for minimizing the objective function given in Eq. 2.24 based on the normalized vector of model parameters and observed data. Then, we show that we can apply the method to find the minimizer of the objective function given in Eq. 2.5 based on the original vector of model parameters and observations. In the most common Levenberg-Marquardt formulation for minimizing the objective function of Eq. 2.24, the Hessian matrix at the ith iteration, H_i, is replaced by $H_i + \lambda_i I_{N_m}$, to obtain

$$\delta\hat{m}^{i+1} = -\left[(\lambda_i + 1)I_{N_m} + G_{D,i}^T G_{D,i}\right]^{-1}\left(\hat{m}^i + G_{D,i}^T \hat{d}^i\right), \tag{2.32}$$

where λ_i is a positive number and is called Levenberg-Marquardt parameter. The Levenberg-Marquardt parameter controls both the search direction and the step size, so the algorithm does not require a line search anymore. A larger value of λ_i will result in more damping of the update in the model parameters and therefore a smaller step size and a very small value of λ_i brings the algorithm closer to the Gauss-Newton iteration with a full step size ($\mu_i = 1$). Algorithm 2.1 describes the LM method for minimizing the objective function. In Algorithm 2.1, $\zeta > 1$ is used to increase or decrease the LM parameter and the convergence criteria described in Gauss-Newton method can be used to terminate the algorithm.

From Eqs. 2.31 and 2.32, we can see that the update for the original model parameters,

Algorithm 2.1: Pseudo-code for Levenberg-Marquardt method

☐ Given m^0, λ_0, $\lambda_{\max}$, and $\zeta > 1$.

☐ Run the forward model to obtain $g(m^0)$ and G_0.

☐ Calculate the objective function $O(m^0)$.

☐ While(Not Converged)

1. Calculate $\hat{m}^i$ and $\hat{d}^i$ using Eqs. 2.22 and 2.23 respectively.
2. Calculate $\nabla_{\hat{m}} O(\hat{m}^i)$ and $\hat{H}_i$ using Eqs. 2.25 and 2.26 respectively.
3. Calculate $\delta\hat{m}^{i+1} = -(\hat{H}_i + \lambda_i I_{N_m})^{-1} \nabla_{\hat{m}} O(\hat{m}^i)$ (equivalent to Eq. 2.32).
4. Set $m_{\text{tmp}} = m^i + C_M^{1/2} \delta\hat{m}^{i+1}$.
5. Run the forward model to obtain $g(m_{\text{tmp}})$ and G_{tmp} and calculate $O(m_{\text{tmp}})$.
6. If $(O(m_{\text{tmp}}) < O(m^i))$
   - Set $m^{i+1} = m_{\text{tmp}}$, $g(m^{i+1}) = g(m_{\text{tmp}})$, $G_{i+1} = G_{\text{tmp}}$ and $O(m^{i+1}) = O(m_{\text{tmp}})$.
   - Set $\lambda_{i+1} = \lambda_i / \zeta$.
7. Else
   - Reject m_{tmp} and set $\lambda_i = \zeta \lambda_i$.
   - If $\lambda_i > \lambda_{\max}$ stop; otherwise go to step 3.
8. EndIf
9. Set $i = i + 1$.

☐ EndWhile

$\delta m^{i+1} \equiv m^{i+1} - m^i$, can be written as

$$\delta m^{i+1} = C_M^{1/2} \delta\hat{m}^{i+1} = -C_M^{1/2} \left[(\lambda_i + 1) I_{N_m} + G_{D,i}^T G_{D,i}\right]^{-1} \left(\hat{m}^i + G_{D,i}^T \hat{d}^i\right). \tag{2.33}$$

Using the definitions of the $\hat{m}$, $\hat{d}$, and $G_{D,i}$, respectively, given by Eqs. 2.22, 2.23, and 2.29 in Eq. 2.33, we obtain

$$\begin{aligned}
\delta m^{i+1} &= -C_M^{1/2} \left[(\lambda_i + 1) I_{N_m} + C_M^{1/2} G_i^T C_D^{-1} G_i C_M^{1/2}\right]^{-1} \\
&\qquad \left(C_M^{-1/2}(m^i - m_{\text{pr}}) + C_M^{1/2} G_i^T C_D^{-1}(g(m^i) - d_{\text{obs}})\right) \\
&= -C_M^{1/2} \left[C_M^{1/2} \left((\lambda_i + 1) C_M^{-1} + G_i^T C_D^{-1} G_i\right) C_M^{1/2}\right]^{-1} \\
&\qquad \left(C_M^{-1/2}(m^i - m_{\text{pr}}) + C_M^{1/2} G_i^T C_D^{-1}(g(m^i) - d_{\text{obs}})\right).
\end{aligned} \tag{2.34}$$

Factoring out $C_M^{1/2}$ from both sides of the inverse matrix in the squre brackets on the right-hand side of Eq. 2.34, we obtain the following equation for the search direction of the original model parameters using LM algorithm:

$$\delta m^{i+1} = -\left[(\lambda_i + 1)C_M^{-1} + G_i^T C_D^{-1} G_i\right]^{-1} \left(C_M^{-1}(m^i - m_{\text{pr}}) + G_i^T C_D^{-1}(g(m^i) - d_{\text{obs}})\right). \tag{2.35}$$

Algorithm 2.2 describes the non-standard LM method applied to the objective function based on the original vector of model parameters and an advantage of this algorithm is that it avoids switching from m to $\hat{m}$ and vice versa. Comparing Eq. 2.35 with the update equation of Gauss-Newton method (given by Eq. 2.17), we find that the Hessian matrix is modified by adding $\lambda_i C_M^{-1}$ to the Gauss-Newton Hessian given by Eq. 2.19. The non-standard form of LM algorithm given by Eq. 2.35, which was first proposed in Bi [6], has been used in the history matching literature [14, 71, 81, 85]. Note we showed that it can be derived from the standard LM algorithm applied to objective function based on the normalized vector of model parameters, which is a result not previously reported in the literature.

2.2.3 Trust Region

In the trust region Newton or Gauss-Newton, the objective function given by Eq. 2.24 is approximated by the following quadratic model at the ith iteration

$$q_i(s) = g_i^T s + \frac{1}{2} s^T \hat{H}_i s, \tag{2.36}$$

where s is the search direction and is equivalent to $\delta\hat{m}$, g_i^T is the gradient of the objective function evaluated at $\hat{m}^i$, i.e., is equal to $\nabla_{\hat{m}} O(\hat{m}^i)$, which is given by Eq. 2.25, and $\hat{H}_i$ is the Hessian given by Eq. 2.26. Unlike the line search method where the search direction is determined first and then an appropriate distance to move in that direction (step size) is determined, in a trust region, we first estimate the maximum distance from the current iterate, $\hat{m}^i$, that we are allowed to move and then determine a step that minimizes q_i subject to the distance constraint Δ^i which is referred to as the trust region radius (the radius of

Algorithm 2.2: Pseudo-code for modified (non-standard) Levenberg-Marquardt method

☐ Given m^0, λ_0, $\lambda_{\max}$, and $\zeta > 1$.

☐ Run the forward model to obtain $g(m^0)$ and G_0.

☐ Calculate the objective function $O(m^0)$.

☐ While(Not Converged)

1. Calculate $\nabla O(m^i)$ and H_i using Eqs. 2.18 and 2.19 respectively.
2. Calculate $\delta m^{i+1} = -(H_i + \lambda_i C_M^{-1})^{-1} \nabla O(m^i)$ (equivalent to Eq. 2.35).
3. Set $m_{\text{tmp}} = m^i + \delta m^{i+1}$.
4. Run the forward model to obtain $g(m_{\text{tmp}})$ and G_{tmp} and calculate $O(m_{\text{tmp}})$.
5. If $(O(m_{\text{tmp}}) < O(m^i))$
   - Set $m^{i+1} = m_{\text{tmp}}$, $g(m^{i+1}) = g(m_{\text{tmp}})$, $G_{i+1} = G_{\text{tmp}}$ and $O(m^{i+1}) = O(m_{\text{tmp}})$.
   - Set $\lambda_{i+1} = \lambda_i / \zeta$.
6. Else
   - Reject m_{tmp} and $\lambda_i = \zeta \lambda_i$.
   - If $\lambda_i > \lambda_{\max}$ stop; otherwise go to step 2.
7. EndIf
8. Set $i = i + 1$.

☐ EndWhile

a ball-shaped region around the current estimate of the model parameter vector). The step denoted by s^{i+1} is the solution of the following constrained minimization problem:

$$s^{i+1} = \arg\min_s q_i(s) \tag{2.37}$$

$$\text{subject to } \|s\| \leq \Delta^i, \tag{2.38}$$

If the step s^{i+1} provides a sufficient decrease in the objective function, then we set

$$m^{i+1} = m^i + C_M^{1/2} s^{i+1}, \tag{2.39}$$

otherwise we reduce the trust region radius and resolve Eqs. 2.37 and 2.38. If the Hessian matrix is positive definite, then the quadratic approximation of the objective function, $q_i(s)$,

has a unique minimum given by

$$s^{i+1} = -\hat{H}_i^{-1} g_i. \tag{2.40}$$

If the constraint of Eq. 2.38 is satisfied for the s^{i+1} obtained with Eq. 2.40, then we update the model parameters according to Eq. 2.39, otherwise the minimizer of Eq. 2.37 has to satisfy $\|s\| = \Delta^i$, i.e., the solution is at the boundary of the trust region. When $\hat{H}_i$ is real symmetric positive definite, we use the method of Lagrange multipliers [65] to find the solution of Eq. 2.37 subject to the equality constraint of $\|s\| = \Delta^i$. Following Sorensen [84], we define a Lagrange function as

$$\mathfrak{L}(s, \lambda) = g_i^T s + \frac{1}{2} s^T \hat{H}_i s + \frac{1}{2} \lambda (s^T s - (\Delta^i)^2), \tag{2.41}$$

where $\lambda \geq 0$ is called the Lagrange multiplier. Note that $\lambda = 0$ is equivalent to the case where the s^{i+1} obtained with Eq. 2.40 satisfies $\|s\| \leq \Delta^i$. Then, we need to find λ and s minimizing the Lagrange function, $\mathfrak{L}$. The minimizer of Eq. 2.41 needs to satisfy the following two conditions:

$$\frac{\partial \mathfrak{L}(s, \lambda)}{\partial s} = 0, \tag{2.42}$$

and

$$\frac{\partial \mathfrak{L}(s, \lambda)}{\partial \lambda} = 0, \tag{2.43}$$

which, respectively, result in

$$\left(\hat{H}_i + \lambda I_{N_m} \right) s = -g_i, \tag{2.44}$$

and

$$s^T s = (\Delta^i)^2. \tag{2.45}$$

The trust region sub-problem is defined by Eqs. 2.44 and 2.45 which have to be solved simultaneously for λ and s. One way to solve the trust region sub-problem is to use singular value decomposition (SVD) of the positive definite Hessian matrix, $\hat{H}_i$. Since $\hat{H}_i$ is symmetric, we

can write its singular value decomposition (spectral decomposition) as

$$\hat{H}_i = UWU^T, \tag{2.46}$$

where U is the matrix of singular vectors and W is a diagonal matrix with the kth singular value, w_k, as its kth diagonal element. With simple algebraic operations we can show that, if s satisfies Eq. 2.44, then

$$f(\lambda) \equiv s^T s = \sum_{k=1}^{N_m} \left(\frac{u_k^T g_i}{w_k + \lambda} \right)^2, \tag{2.47}$$

where u_k is the kth column of U. The derivative of $f(\lambda)$ is given by

$$f'(\lambda) = -2 \sum_{k=1}^{N_m} \frac{\left(u_k^T g_i\right)^2}{(w_k + \lambda)^3}. \tag{2.48}$$

To obtain the value of λ, we can solve

$$\theta(\lambda) \equiv f(\lambda) - (\Delta^i)^2 = 0, \tag{2.49}$$

iteratively using Newton-Raphson method. Note that for $\lambda = 0$, the solution of Eq. 2.44 reduces to Eq. 2.40 and we have assumed that this solution violates the constraint given by Eq. 2.38, i.e., $\|s^{i+1}\| > \Delta^i$. Therefore, it is clear that $\theta(\lambda = 0) > 0$ and since $f'(\lambda)$ (and consequently θ') is always negative, Eq. 2.49 has a unique solution. Morê and Sorensen [64] showed a fasted convergence of Newton-Raphson method is achieved if it is applied to $1/\sqrt{f(\lambda)} - 1/\Delta^i = 0$ instead of Eq. 2.49. Gould et al. [38] generalized the suggestion of Morê and Sorensen [64] and proposed to solve $[f(\lambda)]^{\beta/2} - (\Delta^i)^\beta = 0$ where β should be selected such that the function behaves linearly as a function of λ. Gao et al. [34] proposed the following power law transformation:

$$\theta(\lambda) \equiv \left(\frac{\sqrt{f(\lambda)}}{\Delta^i} \right)^\beta - 1 = 0. \tag{2.50}$$

To solve $\theta(\lambda) = 0$ using Newton-Raphson method, we need the derivative of $\theta(\lambda)$ with respect to λ which is given by

$$\theta'(\lambda) = \frac{1}{2(\Delta^i)^\beta}\beta\left[f(\lambda)\right]^{\beta/2-1} f'(\lambda). \tag{2.51}$$

The convergence of the Newton-Raphson method is faster if $\theta(\lambda)$ behaves linearly which requires the second derivative of $\theta(\lambda)$ to be equal to zero. Gao et al. [34] proposed a method to estimate the optimum value of β which uses cubic interpolation formulation to estimate the second derivative of $\theta(\lambda)$, however we found that $\beta = -1.5$ results in a fast convergence of the Newton-Raphson method for our problems.

Once we find λ, we solve Eq. 2.44 to obtain s^{i+1}. Using the definitions of g_i and $\hat{H}_i$, respectively given by Eqs. 2.25 and 2.26, Eq. 2.44 can be written as

$$\begin{aligned} s^{i+1} &= -\left[\hat{H}_i + \lambda I_{N_m}\right]^{-1} g_i \\ &= -\left[G_{D,i}^T G_{D,i} + I_{N_m} + \lambda I_{N_m}\right]^{-1}\left(\hat{m}^i + G_{D,i}^T \hat{d}^i\right) \\ &= -\left[(\lambda+1) I_{N_m} + G_{D,i}^T G_{D,i}\right]^{-1}\left(\hat{m}^i + G_{D,i}^T \hat{d}^i\right). \end{aligned} \tag{2.52}$$

Note that s^{i+1} is equivalent to $\delta\hat{m}^{i+1}$ in the LM method of Eq. 2.32. Comparing Eq. 2.52, which is obtained for trust region method, to Eq. 2.32 for the LM algorithm we can see that the Lagrange multiplier in the trust region method is equivalent to the LM parameter (λ_i). However, in the LM algorithm we assume a value λ_i at each iteration, while in the trust region method the Lagrange multiplier is found to minimize $\mathfrak{L}(s, \lambda)$. It should be noted that if s^{i+1} obtained from Eq. 2.40 (which is equivalent to setting λ equal to zero in Eq. 2.44) satisfies the constraint of Eq. 2.38 there is no need to solve the trust region sub-problem. Although the preceding observations are not related to the discussion, to the best of our knowledge no one has reported this exact connection between the Levenberg-Marquardt algorithm and the trust region method.

The trust region solver that we described here includes inversion and spectral decomposition of the Hessian matrix which is $N_m \times N_m$. For the large scale problems it is

impractical to to do so, therefore Gao et al. [34] introduced a trust region solver for the cases where $N_d < N_m$ that uses the second matrix inversion lemma to avoid inverting the Hessian matrix (or solving the linear system of N_m equation given by $\hat{H}_i s = -g_i$ directly). If we use the expression for $\hat{H}_i$ from Eq. 2.26, we can rewrite Eq. 2.44 as

$$\begin{aligned} s &= -\left[\hat{H}_i + \lambda I_{N_m}\right]^{-1} g_i \\ &= -\left[G_{D,i}^T G_{D,i} + (\lambda+1) I_{N_m}\right]^{-1} g_i \\ &= -\left(\frac{1}{\lambda+1} I_{N_m} - \frac{1}{(\lambda+1)^2} G_{D,i}^T \left[I_{N_d} + \frac{1}{\lambda+1} G_{D,i} G_{D,i}^T\right]^{-1} G_{D,i}\right) g_i \\ &= -\frac{1}{\lambda+1} g_i + \frac{1}{(\lambda+1)^2} G_{D,i}^T \left[I_{N_d} + \frac{1}{\lambda+1} G_{D,i} G_{D,i}^T\right]^{-1} G_{D,i} g_i, \end{aligned} \tag{2.53}$$

where the third equality is obtained using the second matrix inversion identity given in Eq. 2.13. We define $\widetilde{y}$ and $z(\lambda)$ are, respectively, as

$$\widetilde{y} = G_{D,i} g_i, \tag{2.54}$$

and

$$z(\lambda) = \left(I_{N_d} + \frac{1}{\lambda+1} G_{D,i} G_{D,i}^T\right)^{-1} \widetilde{y}. \tag{2.55}$$

Note that to find $z(\lambda)$, we only need to invert an $N_d \times N_d$ matrix or solve the following system of N_d equations:

$$\left(I_{N_d} + \frac{1}{\lambda+1} G_{D,i} G_{D,i}^T\right) z(\lambda) = \widetilde{y}. \tag{2.56}$$

From Eq. 2.56 it follows that

$$\frac{G_{D,i} G_{D,i}^T z(\lambda)}{\lambda+1} = \widetilde{y} - z(\lambda). \tag{2.57}$$

Using the definitions of $\widetilde{y}$ and $z(\lambda)$, we can write Eq. 2.53 as

$$s = -\frac{1}{\lambda+1} g_i + \frac{1}{(\lambda+1)^2} G_{D,i}^T z(\lambda). \tag{2.58}$$

Starting from $f(\lambda) \equiv s^T s$ (similar to Eq. 2.47) and using Eq. 2.57, we can write

$$\begin{aligned} f(\lambda) &= \frac{g_i^T g_i}{(\lambda+1)^2} - \frac{2}{(\lambda+1)^3}\widetilde{y}^T z(\lambda) + \frac{1}{(\lambda+1)^4} z(\lambda)^T G_{D,i} G_{D,i}^T z(\lambda) \\ &= \frac{g_i^T g_i}{(\lambda+1)^2} - \frac{1}{(\lambda+1)^3}\left[2\widetilde{y}^T z(\lambda) - z(\lambda)^T \frac{G_{D,i} G_{D,i}^T z(\lambda)}{\lambda+1}\right] \\ &= \frac{g_i^T g_i}{(\lambda+1)^2} - \frac{1}{(\lambda+1)^3}\left[2\widetilde{y}^T z(\lambda) - z(\lambda)^T \left(\widetilde{y} - z(\lambda)\right)\right], \end{aligned} \tag{2.59}$$

or simplifying

$$f(\lambda) = \frac{g_i^T g_i}{(\lambda+1)^2} - \frac{\widetilde{y}^T z(\lambda) + z(\lambda)^T z(\lambda)}{(\lambda+1)^3} \tag{2.60}$$

Recall that λ is the root of $\theta(\lambda)$ which is given by Eq. 2.49 (or Eq. 2.47) where $f(\lambda)$ was defined in Eq. 2.47. Equivalently, we can use $f(\lambda)$ from Eq. 2.60 in Eq. 2.49 (or Eq. 2.50) and solve for λ. In order to use Newton-Raphson method, we need to find the expression for the derivative of $f(\lambda)$ given in Eq. 2.60 with respect to λ. To do so, we first find the derivative of $z(\lambda)$ with respect to λ. If we take the derivative of both sides of Eq. 2.56 with respect to λ we will have

$$\left(I_{N_d} + \frac{1}{1+\lambda} G_{D,i} G_{D,i}^T\right) z'(\lambda) - \frac{1}{(1+\lambda)^2} G_{D,i} G_{D,i}^T z(\lambda), \tag{2.61}$$

then solving for $z'(\lambda)$, we obtain

$$\begin{aligned} z'(\lambda) &= \left(I_{N_d} + \frac{1}{1+\lambda} G_{D,i} G_{D,i}^T\right)^{-1} \frac{G_{D,i} G_{D,i}^T z(\lambda)}{(1+\lambda)^2} \\ &= \frac{1}{1+\lambda}\left(I_{N_d} + \frac{1}{1+\lambda} G_{D,i} G_{D,i}^T\right)^{-1} \left(\widetilde{y} - z(\lambda)\right) \\ &= \frac{z(\lambda) - \widetilde{w}(\lambda)}{\lambda+1}, \end{aligned} \tag{2.62}$$

where we have used given Eqs. 2.55 and 2.57 and $\widetilde{w}(\lambda)$ is given by

$$\widetilde{w}(\lambda) = \left(I_{N_d} + \frac{1}{\lambda+1} G_{D,i} G_{D,i}^T\right)^{-1} z(\lambda), \tag{2.63}$$

or equivalently,

$$\left(I_{N_d} + \frac{1}{\lambda+1} G_{D,i} G_{D,i}^T\right) \widetilde{w}(\lambda) = z(\lambda). \tag{2.64}$$

Differentiation of $f(\lambda)$ given by Eq. 2.60 with respect to λ gives

$$\begin{aligned}
f'(\lambda) &= -\frac{2g_i^T g_i}{(\lambda+1)^3} - \frac{\left[\widetilde{y}^T z'(\lambda) + 2z(\lambda)^T z'(\lambda)\right](\lambda+1)^3 - 3(\lambda+1)^2\left[\widetilde{y}^T z(\lambda) + z(\lambda)^T z(\lambda)\right]}{(\lambda+1)^6} \\
&= -\frac{2g_i^T g_i}{(\lambda+1)^3} - \frac{\widetilde{y}^T\left[(\lambda+1)z'(\lambda) - 3z(\lambda)\right] + z(\lambda)^T\left[2(\lambda+1)z'(\lambda) - 3z(\lambda)\right]}{(\lambda+1)^4} \\
&= -\frac{2g_i^T g_i}{(\lambda+1)^3} - \frac{\widetilde{y}^T\left[(z(\lambda) - \widetilde{w}(\lambda)) - 3z(\lambda)\right] + z(\lambda)^T\left[2\left(z(\lambda) - \widetilde{w}(\lambda)\right) - 3z(\lambda)\right]}{(\lambda+1)^4} \\
&= -\frac{2g_i^T g_i}{(\lambda+1)^3} + \frac{\widetilde{y}^T\left[\widetilde{w}(\lambda) + 2z(\lambda)\right] + z(\lambda)^T\left[2\widetilde{w}(\lambda) + z(\lambda)\right]}{(\lambda+1)^4},
\end{aligned} \tag{2.65}$$

where we have used $(\lambda+1)z'(\lambda) = (z(\lambda) - \widetilde{w}(\lambda))$, which is a direct result of Eq. 2.62, to obtain the third equality. Now that we have an expression for $f'(\lambda)$ we can use the Newton-Raphson method to find λ. Note at each iteration of the Newton-Raphson method, the trust region solver of Gao et al. [34] requires the solution of two $N_d \times N_d$ linear systems (Eqs. 2.56 and 2.64), which is a great advantage for large scale problems where the number of data, N_d, is much smaller than the number of model parameters, N_m.

The trust region radius, Δ^i, determines the region around $\hat{m}^i$ where there is a high level of confidence that the constructed quadratic model provides a good approximation to the objective function which we wish to minimize. A large value of Δ^i is used when our quadratic function approximates the objective function well over a large region around $\hat{m}^i$, and if the approximation is only good over a small region around $\hat{m}^i$, a smaller value of Δ^i should be used. Therefore, it is reasonable to update the trust region radius at each iteration. Before describing the algorithm for updating Δ^i, we define ρ_i as

$$\rho_i \equiv \frac{O(\hat{m}^i) - O(\hat{m}^i + s^{i+1})}{q_i(0) - q_i(s^{i+1})} = \frac{O(\hat{m}^i + s^{i+1}) - O(\hat{m}^i)}{q_i(s^{i+1})}, \tag{2.66}$$

which is the ratio of the actual change of the objective function over the predicted change using the quadratic approximation. Based on the value of ρ_i we update the trust region radius at each iteration following Algorithm 2.3, which is from Gao et al. [35].

Algorithm 2.3: Pseudo-code for trust region method

☐ Given m^0, $\Delta_{\min}$, $\Delta_{\max}$, $\Delta^0 \in (\Delta_{\min}, \Delta_{\max})$, and $0 < \eta_1 < \eta_2 < 1$.

☐ Run the forward model to obtain $g(m^0)$.

☐ For i=0,1,2,...

- Calculate $\hat{m}^i$ and $\hat{d}^i$ using Eqs. 2.22 and 2.23 respectively.
- Solve the constrained minimization problem given in Eqs. 2.37 and 2.38 for s^{i+1}.
- Set $m_{\text{tmp}} = m^i + C_M^{1/2} s^{i+1}$ and obtain $g(m_{\text{tmp}})$.
- Calculate ρ_i from Eq. 2.66, where $O(\hat{m}^i + s^{i+1}) = O(m_{\text{tmp}})$.
- If ($\rho_i > \eta_2$ and $\|s^{i+1}\| > 0.5\Delta^i$)
  - Set $\Delta^{i+1} = \min\{2\Delta^i, \Delta_{\max}\}$
  - Set $m^{i+1} = m_{\text{tmp}}$.
- ElseIf ($\rho_i > \eta_1$ or $\rho_i > \eta_2$)
  - Set $m^{i+1} = m_{\text{tmp}}$.
- Else
  - Set $\Delta^{i+1} = 0.5\Delta^i$
  - Set $m^{i+1} = m^i$.
- EndIf

☐ EndFor

2.3 Ensemble Smoother

As discussed in the previous chapter, the ensemble smoother (ES), which was proposed by van Leeuwen and Evensen [93], is a parameter estimation method that assimilates all the available data once. Following a similar procedure to that of Reynolds et al. [75], we show that the ES update equation is similar to applying a single Gauss-Newton iteration with a full step to minimize the RML objective function of Eq. 2.6 where an average sensitivity matrix approximated from the ensemble is used to update all the ensemble members. If we apply a full step Gauss-Newton iteration (Eq. 2.17 with $\mu_i = 1$) to minimize Eq. 2.6, for the jth unconditional realization (ensemble member), we have

$$m_j^{i+1} = m_j^i - \left[C_M^{-1} + G_{i,j}^T C_D^{-1} G_{i,j}\right]^{-1} \left(C_M^{-1} \left(m_j^i - m_{\text{uc,j}}\right) + G_{i,j}^T C_D^{-1} \left(g(m_j^i) - d_{\text{uc,j}}\right) \right). \quad (2.67)$$

For a single iteration with $i = 0$, if choose $m_j^0 = m_{\text{uc,j}}$, and introduce the notation $m_j^a \equiv m_j^1$, $m_j^f = m_j^0$, so $d_j^f = g(m_j^0) = g(m_j^f)$ (based on Eq. 2.1), then Eq. 2.67 reduces to

$$m_j^a = m_j^f - \left[C_M^{-1} + G_{0,j}^T C_D^{-1} G_{0,j}\right]^{-1} G_{0,j}^T C_D^{-1} \left(d_j^f - d_{\text{uc,j}}\right). \tag{2.68}$$

Using the first matrix inversion identity (Eq. 2.12), we can rewrite Eq. 2.68 as

$$m_j^a = m_j^f + C_M G_{0,j}^T \left[G_{0,j} C_M G_{0,j}^T + C_D\right]^{-1} \left(d_{\text{uc,j}} - d_j^f\right). \tag{2.69}$$

For an ensemble of N_e realizations, we define the $N_m \times N_e$ matrix ΔM, which is referred to here somewhat loosely as the model square root, as

$$\Delta M = \frac{1}{\sqrt{N_e - 1}} \left[m_1^f - \bar{m}^f, ..., m_{N_e}^f - \bar{m}^f\right], \tag{2.70}$$

where

$$\bar{m}^f = \frac{1}{N_e} \sum_{j=1}^{N_e} m_j^f. \tag{2.71}$$

We also define the $N_d \times N_e$ matrix ΔD which we call here the data square root as

$$\Delta D = \frac{1}{\sqrt{N_e - 1}} \left[d_1^f - \bar{d}^f, ..., d_{N_e}^f - \bar{d}^f\right], \tag{2.72}$$

where

$$\bar{d}^f = \frac{1}{N_e} \sum_{j=1}^{N_e} d_j^f. \tag{2.73}$$

If we assume that $\bar{d}^f = g(\bar{m}^f)$, then using a first-order Taylor series expansion around the mean model, $\bar{m}^f$, we can write

$$d_j^f - \bar{d}^f = g(m_j^f) - g(\bar{m}^f) = \bar{G} \left(m_j^f - \bar{m}^f\right), \tag{2.74}$$

where $\bar{G}$ denotes the sensitivity matrix evaluated at $\bar{m}^f$. Using the last equality of Eq. 2.74

and the definitions of ΔM and ΔD given by Eqs. 2.70 and 2.72, respectively, we can write

$$\begin{aligned}\Delta M(\Delta D)^T &= \frac{1}{N_e-1}\sum_{j=1}^{N_e}\left(m_j^f-\bar{m}^f\right)\left(d_j^f-\bar{d}^f\right)^T \\ &= \left[\frac{1}{N_e-1}\sum_{j=1}^{N_e}\left(m_j^f-\bar{m}^f\right)\left(m_j^f-\bar{m}^f\right)^T\right]\bar{G}^T \\ &\approx C_M\bar{G}^T, \end{aligned} \tag{2.75}$$

where the last expression is obtained assuming that the term in square brackets of Eq. 2.75 is the ensemble approximate of the prior covariance matrix, C_M. More specifically, if $m_j^f = m_{\text{uc,j}} \sim \mathcal{N}(m_{\text{pr}}, C_M)$, then the expectation of the term in square brackets of Eq. 2.75 is equal to C_M. With a similar approximation, we can show that

$$\begin{aligned}\Delta D(\Delta D)^T &= \frac{1}{N_e-1}\sum_{j=1}^{N_e}\left(d_j^f-\bar{d}^f\right)\left(d_j^f-\bar{d}^f\right)^T \\ &= \frac{1}{N_e-1}\sum_{j=1}^{N_e}\bar{G}\left(m_j^f-\bar{m}^f\right)\left(m_j^f-\bar{m}^f\right)^T\bar{G}^T \\ &\approx \bar{G}C_M\bar{G}^T. \end{aligned} \tag{2.76}$$

If we replace $G_{0,j}$ by $\bar{G}$ in Eq. 2.69 and use the approximation we obtained in Eqs. 2.75 and 2.76, we obtain

$$m_j^a = m_j^f + \Delta M(\Delta D)^T\left[\Delta D(\Delta D)^T + C_D\right]^{-1}\left(d_{\text{uc,j}} - d_j^f\right), \tag{2.77}$$

which is the update equation for the ensemble smoother. Thus, as was previously done implicitly by Reynolds et al. [75], we have shown that the ensemble smoother is approximately similar to performing one Gauss-Newton iteration with a full step and using the same average sensitivity matrix for updating all realizations. Because one Gauss-Newton iteration generally does not lead to a sufficiently good data match (unless the problem is linear where the Gauss-Newton method converges in one iteration), we cannot expect to obtain a good

data match with a single ES update. Note that in the linear case, the first-order Taylor series approximation that we used in Eq. 2.74 is correct and in this case, ES is equivalent to RML as the ensemble size approaches infinity.

To improve the performance of the ensemble smoother, Emerick and Reynolds [22, 25] proposed to assimilate all the observed data multiple time with an inflated measurement error covariance matrix and called their proposed method the ensemble smoother with multiple data assimilation (ES-MDA). In ES-MDA (and other ensemble-based methods), it is necessary to provide an initial ensemble of N_e models which is denoted by $\{m_j^0\}_{j=1}^{N_e}$ where the superscript 0 refers to an initial model. Generation of this initial ensemble of models requires the existence of a prior geological model, or more theoretically, a prior pdf for m. As ES-MDA involves assimilation of the observed data N_a times, $m_j^{f,i}$ and $m_j^{a,i}$ are used to denote the forecast and analysis of the jth realization at the ith assimilation of the observed data (ith data assimilation step) for $j = 1, 2, \cdots N_e$ and $i = 1, 2, \cdots N_a$ where N_a denotes the total number of data assimilation steps. Similarly, $d_j^{f,i} = d^f(m_j^{f,i})$ denotes the predicted (forecast) data obtained from the forward model evaluated at $m_j^{f,i}$. Starting from $m_j^{a,0} = m_j^0$, before each data assimilation step we set $m_j^{f,i} = m_j^{a,i-1}$; then the update or analysis step at the ith data assimilation step (ith iteration) of ES-MDA is given

$$m_j^{a,i} = m_j^{f,i} + \Delta M^i (\Delta D^i)^T \left[\Delta D^i (\Delta D^i)^T + \alpha_i C_D\right]^{-1} \left(d_{uc,j} - d_j^{f,i}\right) \tag{2.78}$$

for $j = 1, 2, \cdots N_e$ and $i = 1, 2, \cdots N_a$, where α_i is the measurement error inflation factor at the ith iteration and $d_{uc,j}$ is a sample from the normal distribution $\mathcal{N}(d_{\text{obs}}, \alpha_i C_D)$. This addition of noise to d_{obs} to form $d_{uc,j}$ is one of the conditions required in order to show that a correct sampling of a posterior distribution can be obtained in the linear-Gaussian case; see Emerick and Reynolds [22, 25]. Emerick and Reynolds [25] showed that ES-MDA samples correctly in the linear-Gaussian case as the ensemble size approaches infinity if the following condition holds:

$$\sum_{i=1}^{N_a} \frac{1}{\alpha_i} = 1. \tag{2.79}$$

The linear-Gaussian case refers to the scenario where the prior pdf is Gaussian and $g(m)$ is linear. In the linear-Gaussian case, the ensemble smoother (ES), EnKF and ES-MDA generate a theoretically correct sampling of the posterior pdf as the ensemble size goes to infinity. EnKF also needs the forecast equation to be linear (see Thulin et al. [90]). In the nonlinear case, no such theory exists; nevertheless, ES, ES-MDA and EnKF, have proven to generate useful results for many synthetic and field cases [7, 15, 20, 21, 26, 45, 82], even though one can construct simple synthetic multimodal cases where the ensemble methods give a poor representation of the posterior pdf [97].

Note that a simple choice for the inflation factors that will satisfy Eq. 2.79 is $\alpha_i = N_a$ for $i = 1, ..., N_a$, however, this can result in rough property maps of permeability and porosity fields at the end of history matching [52].

2.4 Ensemble Randomized Maximum Likelihood

Chen and Oliver [14] proposed an iterative formula for minimizing the RML objective function using the Levenberg-Marquardt (LM) algorithm which is called LM-EnRML. In LM-EnRML, the sensitivity matrix is approximated using an ensemble of models. Due to the similarity between the update equation of LM-EnRML, this method can be regarded as an iterative ensemble smoother. To derive the LM-EnRML equation, we start by applying the non-standard Levenberg-Marquardt formula of Bi [6] (Eq. 2.35) to minimize the RML objective function given in Eq. 2.6, which results in the following update equation for the jth unconditional realization (ensemble member):

$$m_j^{i+1} = m_j^i - \left[(\lambda_i + 1)C_M^{-1} + G_i^T C_D^{-1} G_i\right]^{-1} \left(C_M^{-1}(m_j^i - m_{\mathrm{uc,j}}) + G_i^T C_D^{-1}(g(m_j^i) - d_{\mathrm{uc,j}})\right), \quad (2.80)$$

provided that m_j^{i+1} decreases the objective function; otherwise λ_i is increased and m_j^{i+1} is recomputed as discussed in Algorithm 2.2. To improve the computational efficiency, Chen and Oliver [14] replaced the covariance matrix in the Hessian term by a positive semi-definite matrix P_i which changes for each iteration. If we replace C_M in the Hessian term of Eq. 2.80

with P_i and set $d_j^i = g(m_j^i)$ (based on Eq. 2.1), we obtain

$$m_j^{i+1} = m_j^i - \left[(\lambda_i + 1)P_i^{-1} + G_i^T C_D^{-1} G_i\right]^{-1} \left(C_M^{-1}(m_j^i - m_{\text{uc,j}})\right) - \left[(\lambda_i + 1)P_i^{-1} + G_i^T C_D^{-1} G_i\right]^{-1} G_i^T C_D^{-1}(d_j^i - d_{\text{uc,j}}). \tag{2.81}$$

Using the first matrix inversion identity given in Eq. 2.12 only for the second matrix inverse in Eq. 2.81, we can write

$$m_j^{i+1} = m_j^i - \left[(\lambda_i + 1)P_i^{-1} + G_i^T C_D^{-1} G_i\right]^{-1} \left(C_M^{-1}(m_j^i - m_{\text{uc,j}})\right) + P_i G_i^T \left[(\lambda_i + 1)C_D + G_i P_i G_i^T\right]^{-1} (d_{\text{uc,j}} - d_j^i), \tag{2.82}$$

To take advantage of the representation of the covariance matrix of model parameters and the sensitivity matrix using standard ensemble-based approximations, Chen and Oliver [14] defined P_i as

$$P_i = \Delta M^i (\Delta M^i)^T, \tag{2.83}$$

where ΔM^i is given by Eq. 2.70 calculated using the ensemble of model at ith iteration. If we define ΔD^i as given by Eq. 2.72, similar to Eqs. 2.75 and 2.76, we can show that

$$P_i G_i^T \approx C_M G_i^T \approx \Delta M^i (\Delta D^i)^T, \tag{2.84}$$

and

$$G_i P_i G_i^T \approx G_i C_M G_i^T \approx \Delta D^i (\Delta D^i)^T. \tag{2.85}$$

Note that ΔM^i is non-square $N_m \times N_e$ and its rank is less than or equal to the minimum of

N_m and N_e. Therefore, P_i in Eq. 2.83 is not insertable and we define its inverse as

$$P_i^{-1} = \left(\Delta M^i (\Delta M^i)^T\right)^+, \tag{2.86}$$

where the superscript "+" is used to denote the pseudo-inverse of the matrix. Substituting Eqs. 2.84, 2.85, and 2.86 into Eq. 2.82, we have

$$\begin{aligned} m_j^{i+1} = m_j^i - \left[(\lambda_i + 1)\left(\Delta M^i (\Delta M^i)^T\right)^+ + G_i^T C_D^{-1} G_i\right]^{-1} \left(C_M^{-1}(m_j^i - m_{\mathrm{uc,j}})\right) \\ + \Delta M^i (\Delta D^i)^T \left[(\lambda_i + 1) C_D + \Delta D^i (\Delta D^i)^T\right]^{-1} (d_{\mathrm{uc,j}} - d_j^i). \end{aligned} \tag{2.87}$$

Factoring out (ΔM^i) and $(\Delta M^i)^T$ from both sides of the first inverse matrix on the right-hand side of Eq. 2.87, and using $G_i \Delta M^i \approx \Delta D^i$ (which can be derived from Eqs. 2.83 and 2.84), Eq. 2.87 becomes

$$\begin{aligned} m_j^{i+1} = m_j^i - \Delta M^i \left[(\lambda_i + 1) I_{N_e} + (\Delta D^i)^T C_D^{-1} \Delta D^i\right]^{-1} (\Delta M^i)^T \left(C_M^{-1}(m_j^i - m_{\mathrm{uc,j}})\right) \\ + \Delta M^i (\Delta D^i)^T \left[(\lambda_i + 1) C_D + \Delta D^i (\Delta D^i)^T\right]^{-1} (d_{\mathrm{uc,j}} - d_j^i). \end{aligned} \tag{2.88}$$

In Eq. 2.88, the covariance matrix of the model parameters, C_M, is approximated using the initial ensemble of model parameters and it does not change with iterations. Inverting the ensemble representation of the prior covariance matrix has its own difficulties, since it is rank deficient. As an approximate method, which avoid inversion of this low rank representation of the covariance matrix, Chen and Oliver [14] proposed to discard the term multiplied by $(m_j^i - m_{\mathrm{uc,j}})$ to obtain

$$m_j^{i+1} = m_j^i + \Delta M^i (\Delta D^i)^T \left[(\lambda_i + 1) C_D + \Delta D^i (\Delta D^i)^T\right]^{-1} (d_{\mathrm{uc,j}} - d_j^i). \tag{2.89}$$

Comparing Eq. 2.89 which is the approximate update equation of LM-EnRML (which ignores the model mismatch term) with the update equation of ES-MDA given by Eq. 2.78 reveals that both equations become essentially identical if $(1 + \lambda_i)$ in Eq. 2.89 is replaced by α_i.

However, the perturbations of the vector of observed data for ES-MDA are obtained from $\mathcal{N}(d_{\text{obs}}, \alpha_i C_D)$ while in LM-EnRML the perturbed vector of observations are samples from $\mathcal{N}(d_{\text{obs}}, C_D)$.

2.5 Markov Chain Monte Carlo

In Markov chain Monte Carlo (MCMC) methods, a target probability distribution function (pdf) is sampled by constructing a Markov chain which has the target pdf as its equilibrium distribution. A Markov chain is a sequence of random vectors denoted by $\{\mathcal{M}^{(0)}, \mathcal{M}^{(1)}, \cdots\}$ such that the probability of $\mathcal{M}^{(n)}$ only depends on $\mathcal{M}^{(n-1)}$, i.e.,

$$P\left(\mathcal{M}^{(n)}|\mathcal{M}^{(n-1)}, \cdots, \mathcal{M}^{(0)}\right) = P\left(\mathcal{M}^{(n)}|\mathcal{M}^{(n-1)}\right) \tag{2.90}$$

Assume that we would like to sample $\pi(m)$ defined by Eq. 2.4 and that, in order to use standard simple expressions, we have a countable number of states denoted by $S \equiv \{m_j\}_{j=1}^{\infty}$ where the random vector, $\mathcal{M}$, can only take any m_j from S and nothing else. We define P_{ij} as the homogeneous transition probability which shows the probability that the Markov process goes from the ith state, m_i, to the jth state, m_j, independent of the trial index. A Markov chain is called stationary if

$$\pi(m_j) = \sum_i \pi(m_i) P_{ij}, \tag{2.91}$$

which means that at some point in the Markov chain, we will start sampling from the target pdf, $\pi(m)$, and all states from that point on will represent samples from the target pdf. A Markov chain is called irreducible if every state in the chain can be reached from every other state. For a stationary chain, if the chain is irreducible and aperiodic (has no periodic states), it is called ergodic. The detailed balance condition introduced by Metropolis et al. [62] is given by

$$\pi(m_i) P_{ij} = \pi(m_j) P_{ji}, \tag{2.92}$$

which is also referred to as the reversibility condition, and is a stronger condition than the stationary condition given in Eq. 2.91 thus if Eq. 2.92 holds, the chain is stationary. It can be easily proved that the stationary condition of Eq. 2.91 holds if the chain is reversible by simply summing Eq. 2.92 over i and using $\sum_i P_{ji} = 1$. Metropolis et al. [62] suggested partitioning the transition probability into a proposal probability and a probability of accepting the proposal, i.e., as

$$P_{ij} = \alpha(m_i, m_j) q(m_i, m_j), \tag{2.93}$$

where $q(m_i, m_j)$ is the probability of proposing a transition from the state m_i to the state m_j and $\alpha(m_i, m_j)$ is the probability of accepting that proposed transition. Hastings [44] proposed using

$$\alpha(m_i, m_j) = \min\left\{1, \frac{\pi(m_j) q(m_j, m_i)}{\pi(m_i) q(m_i, m_j)}\right\}, \tag{2.94}$$

and showed that with this choice of $\alpha(m_i, m_j)$, P_{ij} given by Eq. 2.93 will satisfy the detailed balance condition. Note that since Eq. 2.94 contains the ratio of the values of the target pdf, we only need to know the target pdf up to a normalizing constant to evaluate the probability of accepting a proposed state which is a great advantage in the case where Bayes' theorem is used to define the conditional pdf up to a normalizing constant (see Eq. 2.4). The Metropolis-Hastings MCMC algorithm [44, 62], which is the most popular MCMC method for sampling the posterior pdf given by Eq. 2.4, is presented here. Algorithm 2.4 describes the Metropolis-Hastings MCMC method in details.

Algorithm 2.4: Pseudo-code for Metropolis-Hastings MCMC algorithm

1. Initialize $i = 0$, and $m_i = m_0$, where i is the proposal index.
2. Draw a new model, m_j, from the proposal distribution $q(m_i, m_j)$.
3. Run the forward model (reservoir simulator in this work) to evaluate $\pi(m_j)$ up to its normalizing constant and then evaluate the probability of accepting the new proposal with Eq. 2.94.
4. Sample u from a uniform distribution on $[0, 1]$. If $u \leq \alpha(m_i, m_j)$, accept the proposal and set $m_{i+1} = m_j$. Otherwise, reject the proposal and set $m_{i+1} = m_i$.
5. Set $i = i + 1$ and goto step 2.

The simplest procedure for the proposal mechanism is called global perturbation in which the prior pdf is used for the proposal distribution. In this case, the proposed states are independent of the current state in the chain and the new proposals can be obtained using

$$m_j = m_{\text{pr}} + C_M^{1/2} Z_{N_m}, \tag{2.95}$$

where Z_{N_m} is a random vector generated from $\mathcal{N}(0, I_{N_m})$. Then, the probability of the proposed state can be calculated as

$$q(m_i, m_j) = b \left(-\frac{1}{2} (m_j - m_{\text{pr}})^T C_M^{-1} (m_j - m_{\text{pr}}) \right) = b \left(-\frac{1}{2} Z_{N_m}^T Z_{N_m} \right), \tag{2.96}$$

where b is a normalizing constant which is equal to $\left((2\pi)^{N_m} |C_M| \right)^{-1/2}$, but there is no need to compute it because we only need the ratio of $q(m_j, m_i)$ to $q(m_i, m_j)$ based on Eq. 2.94. The global perturbation algorithm leads to a very low acceptance rate for high-dimensional problems, and most of proposed states will be rejected specially if the observed data are accurate [70]. To improve the acceptance rate, Oliver et al. [70] proposed a local perturbation procedure in which at each iteration of MCMC a random component of Z_{N_m} associated with the last state in the chain is selected and replaced with random number drawn from $\mathcal{N}(0, 1)$. Gelman et al. [36] proposed using the current state of the chain as the mean of the proposal distribution and also replaced the prior covariance with $\sigma^2 C_M$, where σ is a scaling factor less than unity which should be tuned to obtain a chain with an acceptance rate on the order of 25%. Although a small value of σ tends to increase the acceptance rate of the chain, it will result in poorly mixed chain where the states m_{i+k}, $k = 1, 2, .., K$ are strongly correlated for large values of K. Furthermore, a large value of σ will result in a low acceptance rate. An important aspect of this proposal distribution is that it is symmetric, i.e., $q(m_i, m_j) = q(m_j, m_i)$; therefore, the probability of accepting a proposed state will reduce to

$$\alpha(m_i, m_j) = \min \left\{ 1, \frac{\pi(m_j)}{\pi(m_i)} \right\}, \tag{2.97}$$

which is independent of the proposal distribution, q.

It is important to note that if the proposal distribution is the same as the target distribution, any proposed state will be accepted based on Eq. 2.94. Therefore, if we design a proposal distribution which is fairly close the target pdf, we should improve the acceptance rate of the MCMC. In an effort to build a proposal distribution that is close to the posterior pdf, Li and Reynolds [56] proposed to find many local minimums using a gradient-based algorithm and then build a Gaussian mixture model which is used as the proposal distribution in the second step of their two-level MCMC algorithm. Their MCMC algorithm showed promising results for a number of problem they tested, however their first step requires the availability of the adjoint solution to compute the gradient of the objective function. In Chapter 4 we try to eliminate this requirement by using the distributed Gauss-Newton (DGN) which was first introduced in Gao et al. [32]. In DGN we try to find multiple minimizer of the objective function in parallel while the sensitivity matrix is estimated using a simplex gradient approximation.

Chapter 3

GENERATION OF INFLATION FACTORS FOR ES-MDA

Supplemental to comments in Chapter 1, we note that one disadvantage of the iterative ensemble-based data assimilation schemes of IR-ES [48] and ES-MDA-RLM [52] is that the number of iterations required to satisfy their stopping criteria is not known a priori and in some cases the methods require 20 or more iterations to converge. (Later in this chapter, we discuss a case where IR-ES, which is based directly on the discrepancy principle does not converge in 200 iterations.) As each iteration of IR-ES or an adaptive ES-MDA algorithm requires one run of the forward model (numerical reservoir simulator in our examples) for each ensemble member, it is important to minimize the number of iterations to promote computational efficiency. The main contributions of the work presented in this chapter are twofold. First, we derive an analytical equation for the minimum value of the inflation factor that satisfies the discrepancy principle. Unlike IR-ES formulation of Iglesias [48], this equation can be easily solved at each iteration to provide the minimum inflation factor that satisfies the discrepancy principle. Secondly and more importantly, we use this theoretical result to provide a practical way to derive reasonable inflation factors for practical application which allows the user to specify a priori the number of data assimilation steps (iterations) that will be used in ES-MDA. In this practical approach, given the initial ensemble and associated initial data match, the ES-MDA inflation factor at the first data assimilation step is determined using the discrepancy principle. Based on the number of data assimilation steps that the user has specified, the subsequent inflation factors are chosen so they decrease geometrically with iteration number (data assimilation step) in a way such that the sum of the inverse inflation factors is equal to unity. This restriction on the sum ensures that, for the linear-Gaussian case, the resulting ES-MDA algorithm samples the

posterior distribution correctly as the ensemble size goes to infinity. The main advantage of the second method based on geometric factors is that it allows one to specify a priori the number of data assimilation steps that will be used which allows the user to consider the computational resources required.

3.1 Methodology

An analytical formula for calculating the ES-MDA inflation factors based on the discrepancy principle is provided in the next subsection. This theoretical result generates distinct ES-MDA procedures ranging from a strict requirement that the discrepancy principle is satisfied at each iteration to methods for choosing the inflation factors that do not strictly enforce the discrepancy principle at every data assimilation step. Among the latter method is a procedure which approximately enforces the discrepancy principle only at the first iteration but allows the user to specify a priori the number (N_a) of data assimilation steps to be performed, whereas with strict enforcement of the discrepancy principle, there is no control over the number of iterations.

3.1.1 Analytical procedure for calculating inflation factors from the discrepancy principle

The truncated singular value decomposition (TSVD) of the standard dimensionless sensitivity matrix has been used to parameterize the gradient-based history matching methods [79, 80, 81, 88, 89]. At any particular model, m, the sensitivity matrix for $d^f(m)$ is denoted by $G(m)$ and is defined as the $N_d \times N_m$ matrix which has $\partial d_i^f(m)/\partial m_j$ as the entry in the ith row and jth column where, here and only here, d_i^f denotes the ith entry of the vector d^f and m_j denotes the jth component of m. If $\bar{m}$ denotes the mean of m, then following Reynolds et al. [75],

$$d^f(m) - d^f(\bar{m}) \approx G(\bar{m})(m - \bar{m}). \tag{3.1}$$

Invoking the approximation $d^f(\bar{m}) = \bar{d}^{f,i}$, Eq. 3.1 indicates that at the ith data assimilation

step (Eq. 2.78),

$$d_j^{f,i} - \bar{d}^{f,i} \approx G(\bar{m}^{f,i})(m_j^{f,i} - \bar{m}^{f,i}), \tag{3.2}$$

for $j = 1, 2, \cdots N_e$. It follows that

$$G_D^i \equiv C_D^{-1/2} \Delta D^i \approx C_D^{-1/2} G(\bar{m}^{f,i}) \Delta M^i, \tag{3.3}$$

where $C_D^{-1/2}$ is the inverse of the square root of C_D. Because ΔM^i is the analogue of the square root of a model covariance matrix, e.g., an analogue of $C_M^{1/2}$, Eq. 3.3 is the direct analogue of the standard dimensionless sensitivity matrix for gradient-based history matching which was defined by Zhang et al. [98] as given in Eq. 2.29. For completeness, we repeat that the standard dimensionless sensitivity matrix in [98] is given by

$$G_{D,i} = C_D^{-1/2} G_i C_M^{1/2}. \tag{3.4}$$

Note that the original dimensionless sensitivity matrix of Eq. 3.4 is $N_d \times N_m$ and its evaluation requires the calculation of the partial derivatives of each data with respect to all of the model parameters whereas the ensemble-based dimensionless sensitivity matrix used here (Eq. 3.3) is $N_d \times N_e$ and does not require the calculation of any derivatives. As shown by Tavakoli and Reynolds [88] and Tavakoli and Reynolds [89], the reduction in uncertainty attained by assimilating data can be directly characterized in terms of the singular values of the dimensionless sensitivity matrix of Eq. 3.4. This motivates rewriting the ES-MDA equation in terms of the dimensionless sensitivity matrix defined in Eq. 3.3. After doing so, the SVD of the dimensionless sensitivity matrix of Eq. 3.3 is used to derive an analytical expression that can be used to directly calculate the minimum inflation factor at each iteration which guarantees that the discrepancy principle is satisfied. By writing the inverse matrix in Eq. 2.78 as

$$\left[\Delta D^i (\Delta D^i)^T + \alpha_i C_D\right]^{-1} = C_D^{-1/2} \left[C_D^{-1/2} \Delta D^i (\Delta D^i)^T C_D^{-1/2} + \alpha_i I_{N_d}\right]^{-1} C_D^{-1/2}, \tag{3.5}$$

the ES-MDA update equation (Eq. 2.78) can be rewritten as

$$m_j^{a,i} = m_j^{f,i} + \Delta M^i (G_D^i)^T \left[G_D^i (G_D^i)^T + \alpha_i I_{N_d} \right]^{-1} C_D^{-1/2} \left(d_{uc,j} - d_j^{f,i} \right), \tag{3.6}$$

for $j = 1, 2, \cdots N_e$. Similar to Eq. 3.6, one can update the mean of m directly as

$$\bar{m}^{a,i} = \bar{m}^{f,i} + \Delta M^i (G_D^i)^T \left[G_D^i (G_D^i)^T + \alpha_i I_{N_d} \right]^{-1} C_D^{-1/2} \left(d_{\text{obs}} - \bar{d}^{f,i} \right), \tag{3.7}$$

where $\bar{d}^{f,i}$ is given by Eq. 2.73. We can show that $\bar{m}^{a,i}$ is the solution of the regularized least squares problem given by

$$x^{a,i} = \arg\min_x \left\{ \frac{1}{2} \left\| G_D^i x - y \right\|^2 + \frac{\alpha_i}{2} \left\| x \right\|^2 \right\}, \tag{3.8}$$

where

$$x = (\Delta M^i)^+ \left(m - \bar{m}^{f,i} \right), \tag{3.9}$$

and

$$y = C_D^{-1/2} \left(d_{\text{obs}} - \bar{d}^{f,i} \right). \tag{3.10}$$

Here, $(\Delta M^i)^+$ is the pseudo-inverse of ΔM^i. To show in the following that $\bar{m}^{a,i}$ is the solution of the regularized least squares problem given by Eq. 3.8, we start by defining the following function:

$$\widetilde{O}(x) = \frac{1}{2} \left\| G_D^i x - y \right\|^2 + \frac{\alpha_i}{2} \left\| x \right\|^2, \tag{3.11}$$

which we wish to minimize. The gradient of $\widetilde{O}$ with respect to x is given by

$$\nabla \widetilde{O} = (G_D^i)^T \left(G_D^i x - y \right) + \alpha_i x. \tag{3.12}$$

If we set the gradient equal to zero and solve for the minimizer, $x^{a,i}$, we obtain

$$\begin{aligned} x^{a,i} &= \left[(G_D^i)^T G_D^i + \alpha_i I_{N_e}\right]^{-1} (G_D^i)^T y \\ &= \frac{1}{\alpha_i} I_{N_e} (G_D^i)^T \left[\frac{1}{\alpha_i} G_D^i (G_D^i)^T + I_{N_d}\right]^{-1} y \\ &= (G_D^i)^T \left[G_D^i (G_D^i)^T + \alpha_i I_{N_d}\right]^{-1} y, \end{aligned} \tag{3.13}$$

where we have used the first matrix inversion lemma given in Eq. 2.12 to obtain the second equality. If we use the definitions of x and y, which are given by Eqs. 3.9 and 3.10 respectively, Eq. 3.13 can be rewritten as

$$(\Delta M^i)^+ \left(\bar{m}^{a,i} - \bar{m}^{f,i}\right) = (G_D^i)^T \left[G_D^i (G_D^i)^T + \alpha_i I_{N_d}\right]^{-1} C_D^{-1/2} \left(d_{\text{obs}} - \bar{d}^{f,i}\right). \tag{3.14}$$

Multiplying both sides of Eq. 3.14 by ΔM^i and moving $\bar{m}^{f,i}$ to the right-hand side of the equality we obtain Eq. 3.7.

Note that Eq. 3.8 is the well-known Tikhonov regularization method with regularization parameter α_i applied to the ill-posed inverse problem:

$$G_D^i x = y. \tag{3.15}$$

The term multiplied by α_i in Eq. 3.8 is called the regularization term and the regularization parameter, α_i, controls the relative importance of the two terms. If α_i is too small, the solution of Eq. 3.7, or, equivalently, Eq. 3.8 includes high frequency components and is generally too oscillatory to be an acceptable solution; moreover, the numerical solution of Eq. 3.7, or, equivalently, Eq. 3.8 can be unstable. On the other hand, if α_i is too large, the high frequency components in the solution are filtered out but the resulting solution may be too smooth to be physically viable. One approach to define and determine the optimum regularization parameter is to use the discrepancy principle [94, p. 104]. It is reasonable to assume that the data mismatch is greater than the noise level associated with the data,

otherwise there is no point in matching the data. This assumption can be written as

$$\|y\| > \eta, \tag{3.16}$$

or, equivalently,

$$\|C_D^{-1/2}\left(d_{\text{obs}} - \bar{d}^{f,i}\right)\| > \eta, \tag{3.17}$$

where η is the noise level defined as

$$\eta^2 \equiv \left\|C_D^{-1/2}\left(d_{\text{obs}} - g(m_{\text{true}})\right)\right\|^2. \tag{3.18}$$

Viewing d_{obs} as a random vector given by $d_{\text{obs}} = g(m_{\text{true}}) + \xi$ (see Eq. 2.2) where ξ is the random noise vector from the distribution $\mathcal{N}(0, C_D)$, η^2 has a χ^2-distribution with mean N_d [4, p. 107]. Therefore, it is reasonable to assume $\eta = \sqrt{N_d}$. Eq. 3.16 or, equivalently, Eq. 3.17 simply states that the signal-to-noise ratio is strictly greater than one. From Eq. 3.15, $(G_D^i x^{a,i} - y)$ is the difference between the data vector y and the regularized solution given by Eq. 3.7. Thus, the discrepancy principle indicates that the minimum regularization parameter, α_i, that should be used in Eq. 3.7, should be selected such that

$$\eta = \|G_D^i x^{a,i} - y\|. \tag{3.19}$$

Eq. 3.19 has a unique solution for the regularization parameter, α_i [39, p. 44]. From Eq. 3.13 we have

$$\begin{aligned} G_D^i x^{a,i} - y &= G_D^i (G_D^i)^T \left[G_D^i (G_D^i)^T + \alpha_i I_{N_d}\right]^{-1} y - y \\ &= G_D^i (G_D^i)^T \left[G_D^i (G_D^i)^T + \alpha_i I_{N_d}\right]^{-1} y \\ &\qquad - \left[G_D^i (G_D^i)^T + \alpha_i I_{N_d}\right]\left[G_D^i (G_D^i)^T + \alpha_i I_{N_d}\right]^{-1} y \\ &= \alpha_i I_{N_d}\left[G_D^i (G_D^i)^T + \alpha_i I_{N_d}\right]^{-1} y. \end{aligned} \tag{3.20}$$

Using Eq. 3.10 in Eq. 3.20, it follows that

$$\eta = \left\| G_D^i x^{a,i} - y \right\| = \alpha_i \left\| \left[G_D^i (G_D^i)^T + \alpha_i I_{N_d} \right]^{-1} C_D^{-1/2} \left(d_{\text{obs}} - \bar{d}^{f,i} \right) \right\|. \tag{3.21}$$

The expression for η given by Eq. 3.21 can be used in Eq. 3.17 to obtain

$$\| C_D^{-1/2} \left(d_{\text{obs}} - \bar{d}^{f,i} \right) \| > \alpha_i \left\| \left[G_D^i (G_D^i)^T + \alpha_i I_{N_d} \right]^{-1} C_D^{-1/2} \left(d_{\text{obs}} - \bar{d}^{f,i} \right) \right\|. \tag{3.22}$$

We can replace the inequality in Eq. 3.22 by an equality by multiplying the left-hand-side by some $\rho \in (0, 1)$. Then we replace Eq. 3.22 by

$$\rho \| C_D^{-1/2} \left(d_{\text{obs}} - \bar{d}^{f,i} \right) \| = \alpha_i \left\| \left[G_D^i (G_D^i)^T + \alpha_i I_{N_d} \right]^{-1} C_D^{-1/2} \left(d_{\text{obs}} - \bar{d}^{f,i} \right) \right\|. \tag{3.23}$$

The term multiplied by ρ is the norm of the dimensionless forecast data mismatch vector (which is obtained using the current estimates of the model) and the right-hand side of Eq. 3.23 is an approximation of the dimensionless data mismatch vector that would be obtained the updated model parameters (dimensionless analysis mismatch vector using a given value of α_i). Evaluating the exact dimensionless data mismatch vector requires running the forward model for the updated model parameters. Eq. 3.23 states that the mismatch of the updated models should be reduced by a factor ($\rho < 1$) times the dimensionless forecast mismatch vector. A higher value of ρ will result in a less decrease in the data mismatch which is achieved by more damping of the updates with a larger value of α_i. However, for regularizing Levenberg-Marquardt, Hanke [41] proposed the following condition for the regularization parameter

$$\rho^2 \left\| C_D^{-1/2} \left(d_{\text{obs}} - \bar{d}^{f,i} \right) \right\|^2 \leq \alpha_i^2 \left\| \left[G_D^i (G_D^i)^T + \alpha_i I_{N_d} \right]^{-1} C_D^{-1/2} \left(d_{\text{obs}} - \bar{d}^{f,i} \right) \right\|^2, \tag{3.24}$$

and showed that any regularization parameter, α_i, satisfying Eq. 3.24 (which is greater than or equal to the regularization parameter obtained from Eq. 3.23) suffices to obtain a stable

solution for the inverse problem. Eq. 3.24 states that instead of finding the exact value of α_i, we can use a larger inflation factor; however, this often will result in more iterations when the algorithm is applied. Iglesias and Dawson [49] adopted the idea of Hanke [41] and Hanke [42] for the selection of the regularization parameter and stopping criteria of Levenberg-Marquardt (LM) as applied to a least-squares problem. As the update equations for regularized LM and ES-MDA have similar forms, Iglesias [48] and Le et al. [52] used the criterion to choose the ES-MDA inflation factor but proposed two somewhat different methods for choosing the inflation factors which are derived by analogy with the choise of the regularization parameter for the regularized LM algorithm [49]. Specifically, Iglesias [48] used Eq. 3.24 for the regularizing parameter, but Le et al. [52] replaced d_{obs} and $\bar{d}^{f,i}$, respectively, with $d_{uc,j}$ and $d_j^{f,i}$ to enforce the condition for each individual realization which results in a much stricter condition and larger inflation factors. Another difference between their methods is the stopping criteria for the algorithms. While ES-MDA-RLM of Le et al. [52] uses Eq. 2.79 as the stopping criterion to ensure that we sample correctly in the linear-Gaussian case, the IR-ES [48] algorithm terminates based on the following criterion according to the discrepancy principle:

$$\left\| C_D^{-1/2} \left(d_{\text{obs}} - \bar{d}^{f,i} \right) \right\| \leq \tau \eta, \tag{3.25}$$

where $\tau > 1$ is a tuning parameter. For the gradient-based regularizing LM algorithm, Hanke [41] showed that $\tau > 1/\rho$ provides a stable approximation of the solution of the inverse problem.

Iglesias [48] proposed estimating an inflation factor that satisfies Eq. 3.24 with an iterative scheme. The iterative scheme starts with $\alpha_i = 1$, and if Eq. 3.24 is not satisfied we double α_i and recheck Eq. 3.24 with the new α_i, otherwise we accept the α_i. The procedure is continued until an α_i that is sufficiently large so that Eq. 3.24 is satisfied is found. By starting IR-ES with $\alpha_1 = 1$, IR-ES will sample correctly in the linear-Gaussian case as the ensemble size goes to infinity if and only if ρ is sufficiently small so that Eq. 3.24 holds when

$i = 1$ (so $\alpha_1 = 1$ is acceptable) and Eq. 3.25 is satisfied after the first iteration. As an alternative to using an iterative procedure to determine α_i, we show below that the right-hand side of Eq. 3.24 can be written in a form that allows its easy evaluation by using the SVD of G_D^i. The resulting expression provides a straightforward method for determining the minimum inflation factor that ensures that Eq. 3.24 is satisfied.

Let $U\Lambda V^T$ represent the SVD of G_D^i, where U is an $N_d \times N_d$ orthogonal matrix with its jth column given by the jth left singular vector, V is an $N_e \times N_e$ orthogonal matrix with its jth column given by the jth right singular vector, and Λ is an $N_d \times N_e$ matrix with the jth singular value, λ_j, as its jth diagonal entry. The singular values are ordered as

$$\lambda_{\max} \equiv \lambda_1 \geq \lambda_2 \geq \cdots \lambda_N \equiv \lambda_{\min} \geq 0 \tag{3.26}$$

where N is the minimum of N_d and N_e. It follows that

$$C \equiv G_D^i (G_D^i)^T + \alpha_i I_{N_d} = \left(U\Lambda V^T\right)\left(U\Lambda V^T\right)^T + \alpha_i I_{N_d}. \tag{3.27}$$

Using $V^T V = I_{N_e}$ and $UU^T = I_{N_d}$, we can write

$$C = U\left(\Lambda\Lambda^T + \alpha_i I_{N_d}\right)U^T, \tag{3.28}$$

with the inverse of C given by

$$C^{-1} = U\Gamma U^T, \tag{3.29}$$

where Γ is an $N_d \times N_d$ diagonal matrix with its jth diagonal entry equal to

$$\gamma_j = \frac{1}{\lambda_j^2 + \alpha_i}. \tag{3.30}$$

Using the definition of the vector y in Eq. 3.10 and defining the matrix C as given in Eq. 3.27, one can write Eq. 3.24 as

$$\rho^2 \leq \alpha_i^2 \frac{\left\|C^{-1}y\right\|^2}{\left\|y\right\|^2}, \tag{3.31}$$

where y is still given by Eq. 3.10. We know that the columns of U form an orthonormal set and therefore the vector y can be expressed as a linear combination of the columns of U, i.e.,

$$y = \sum_{\ell=1}^{N_d} c_\ell u_\ell, \tag{3.32}$$

where c_ℓ's are scalers and u_ℓ is the ℓth column of U. Note c_ℓ is given by

$$c_\ell = u_\ell^T y. \tag{3.33}$$

Using Eq. 3.32 and the orthonormality of $\{u_\ell\}_{\ell=1}^{N_d}$, the norm of y can be written as

$$\|y\|^2 = y^T y = \sum_{i=1}^{N_d} \sum_{j=1}^{N_d} c_i u_i^T u_j c_j = \sum_{j=1}^{N_d} c_j^2. \tag{3.34}$$

Similarly,

$$\left\|C^{-1}y\right\|^2 = \left(C^{-1}y\right)^T \left(C^{-1}y\right) = \left(\sum_{i=1}^{N_d} c_i C^{-1} u_i\right)^T \left(\sum_{j=1}^{N_d} c_j C^{-1} u_j\right). \tag{3.35}$$

As the eigenvalue/eigenvector pairs of the matrix C^{-1} are (γ_j, u_j) where the γ_j's are given by Eq. 3.30, we can replace $C^{-1}u_i$ by $\gamma_i u_i$ in Eq. 3.35 to obtain

$$\left\|C^{-1}y\right\|^2 = \left(\sum_{i=1}^{N_d} c_i \gamma_i u_i\right)^T \left(\sum_{j=1}^{N_d} c_j \gamma_j u_j\right) = \sum_{j=1}^{N_d} \gamma_j^2 c_j^2 \tag{3.36}$$

Using Eqs. 3.34 and 3.36 in Eq. 3.31, we obtain

$$\rho^2 \leq \alpha_i^2 \frac{\sum_{j=1}^{N_d} \gamma_j^2 c_j^2}{\sum_{j=1}^{N_d} c_j^2}. \tag{3.37}$$

Eq. 3.37 is a nonlinear function of α_i which can be solved numerically to obtain the exact value of α_i satisfying the discrepancy principle (Eq. 3.24) for each iteration. The ratio that is multiplied by α_i^2 in Eq. 3.37 represents a weighted average of the squared eigenvalues of

C^{-1}. If y in Eq. 3.32 is equal to a constant times the kth eigenvector, then Eq. 3.37 reduces to

$$\rho^2 \leq \alpha_i^2 \gamma_k^2 = \frac{\alpha_i^2}{(\lambda_k^2 + \alpha_i)^2}, \tag{3.38}$$

which is equivalent to

$$\alpha_i \geq \frac{\rho}{1-\rho} \lambda_k^2. \tag{3.39}$$

More generally, it is easy to show that the largest possible value of α_i that satisfies Eq. 3.37 is obtained when y is aligned with the first eigenvector and the smallest value of α_i that satisfies Eq. 3.37 is acquired when y is in the same direction as the eigenvector corresponding to the smallest singular value. For a given y, the optimum value of α_i is between these two extremes. Therefore, instead of strictly enforcing Eq. 3.24, we propose a modified version of IR-ES, which we call M-IR-ES, where the inflation factor at each data assimilation step is defined by

$$\alpha_i = \frac{\rho}{1-\rho} \overline{\lambda}^2 \tag{3.40}$$

where $\overline{\lambda}$ is the average singular value of the dimensionless sensitivity matrix, G_D^i, and is given by

$$\overline{\lambda} = \frac{1}{N} \sum_{j=1}^{N} \lambda_j, \tag{3.41}$$

where N is the minimum of N_d and N_e. In M-IR-ES, if Eq. 3.40 gives a value of α_i which makes the sum of the inverse of the inflation factors greater than one, we choose α_i so that the inverse of the inflation factors sum to 1. The M-IR-ES is terminated if either the stopping criteria based on discrepancy principle (Eq. 3.25) is satisfied or the sum of the inverse inflation factors becomes equal to unity; see Eq. 2.79. or

3.1.2 ES-MDA with geometric inflation factors

In this section, we introduce a practical and efficient method for generating the inflation factors used in ES-MDA. This method allows the user to specify the number of data assimilation steps (N_a) a priori considering the computational resources available. The mo-

tivation for specifying N_a a priori is that our experience with the other forms of iterative ES, such as IR-ES [48] and ES-MDA-RLM [52], shows that they may require more than 20 iterations, in fact far more, to meet their stopping criteria and using more than 20 iterations is not generally computationally feasible for very large-scale field applications. Our experience to date shows that between 4 to 8 data assimilation steps with an appropriate choice of inflation factors results in a good history-match using ES-MDA and this number is generally computationally feasible. In the practical approach, given the initial ensemble and associated initial data match, the ES-MDA inflation factor at the first data assimilation step is determined using Eq. 3.40 where, in Eq. 3.40, we use $\rho = 0.5$. Thus, the first inflation factor is given by

$$\alpha_1 = \overline{\lambda}^2 = \left(\frac{1}{N} \sum_{j=1}^{N} \lambda_j \right)^2 . \tag{3.42}$$

As the inflation factors, α_i's, are originally introduced in order to damp the ES updates, it is reasonable to require that the inflation factors are a monotonically decreasing function of the data assimilation step index i [25], and the results of [52] on adaptive ES-MDA suggests that the estimated optimal inflation factors often decrease rapidly as i increases. Our implementations of the IR-ES method [48] and ES-MDA-RLM method [52] indicate that the optimal inflation factors usually, but not always, decrease with the ES iterations. This observation inspired us to introduce the inflation factors based on a geometric sequence where the first inflation factor, α_1, is given by Eq. 3.42. We assume that the inflation factors form a geometric sequence, i.e., each inflation factor is a constant multiple of the previous inflation factor so that

$$\alpha_i = \beta^{i-1} \alpha_1, \tag{3.43}$$

where β is called the common ratio of the sequence. To find the common ratio β, we use Eq. 3.43 in Eq. 2.79, which yields

$$\sum_{i=1}^{N_a} \frac{1}{\beta^{i-1}} = \alpha_1. \tag{3.44}$$

The expression for a summation of the geometric series can be used in Eq. 3.44, to obtain

$$\frac{1-(1/\beta)^{N_a-1}}{1-(1/\beta)} = \alpha_1. \tag{3.45}$$

Eq. 3.45 can be solved to find the desired solution in the domain of $(0, 1)$ for β. Once we find β, the subsequent inflation factors are determined based on Eq. 3.43.

In summary, in our practical method for generating geometric inflation factors which we call ES-MDA-GEO, we allow the user to specify the number of data assimilation steps and choose the first inflation factor based on Eq. 3.42. Then we solve Eq. 3.45 for the common ratio of the geometric sequence, β. The rest of the inflation factors are determined using Eq. 3.43. Below in Algorithm 3.1, we summarize in pseudo-code the ES-MDA-GEO algorithm. Before doing so, we should note that, as is relatively standard in ensemble-based methods, we generally use a truncated singular value decomposition (TSVD) in the update equation for an ensemble-based data assimilation method. Letting the TSVD of G_D^i be given by

$$G_D^i = U_p \Lambda_p V_p^T, \tag{3.46}$$

where

$$p \leq N = \min\{N_d, N_e\}, \tag{3.47}$$

the matrix C of Eq. 3.28 is replaced by

$$C = U_p \left(\Lambda_p^2 + \alpha_i I_p\right) U_p^T, \tag{3.48}$$

with the pseudo-inverse of C given by

$$C^+ = U_p \Gamma_p U_p^T, \tag{3.49}$$

where Γ_p is an $p \times p$ diagonal matrix with its jth diagonal entry equal to

$$\gamma_j = \frac{1}{\lambda_j^2 + \alpha_i}. \tag{3.50}$$

It is important to recall that in Eq. 3.42, $N = \min\{N_d, N_e\}$ and that Eq. 3.42 is based on the full SVD of the dimensionless sensitivity matrix, not TSVD, so the choice of the first inflation factor is not affected by truncation. However, after determining the first inflation factor, TSVD can be used to compute the updated solution of Eq. 3.6 as shown immediately below. Finally, using Eqs. 3.46 and 3.49 in Eq. 3.6 and simplifying gives the following ES-MDA update equation for ith data assimilation step:

$$m_j^{a,i} = m_j^{f,i} + \Delta M^i V_p \Lambda_p \Gamma_p U_p^T C_D^{-1/2} (d_{uc,j} - d_j^{f,i}). \tag{3.51}$$

Here, p is chosen as the smallest positive integer that satisfies

$$\sum_{j=1}^{p} \lambda_j \geq \widetilde{a} \sum_{j=1}^{N} \lambda_j. \tag{3.52}$$

In the examples, we have used $\widetilde{a} = 0.99$ but decreasing $\widetilde{a}$ will lead to a smoother solution. It is interesting to note that using the update equation given by Eq. 3.51, the dimensionless data mismatch term, $C_D^{-1/2}(d_{uc,j} - d_j^{f,i})$, is projected onto the subspace spanned by the left singular vectors of G_D^i that correspond to the p largest singular vectors, i.e., the scaled data-mismatch vector, $C_D^{-1/2}(d_{uc,j} - d_j^{f,i})$, is projected onto the subspace span$\{u_i\}_{i=1}^{p}$ which is equivalent to applying principal component analysis.

3.2 Comments on ES-MDA Update

In this section, we investigate the ES-MDA update equation in more detail. We define the normalized data mismatch vector, $\delta\widetilde{d}_j$ by

$$\delta\widetilde{d}_j = C_D^{-1/2} \left(d_{uc,j} - d_j^{f,i} \right). \tag{3.53}$$

Algorithm 3.1: Pseudo-code for ES-MDA with geometric inflation factors

☐ Choose the number of data assimilation steps, N_a.

☐ Generate initial ensemble denoted by $\{m_j^{a,0}\}_{j=1}^{N_e}$.

☐ For $i = 1$ to N_a

- Set $m_j^{f,i} = m_j^{a,i-1}$ for $j = 1, 2, ..., N_e$.
- Run the ensemble from time zero.
- Calculate ΔM^i and ΔD^i using Eqs. 2.70 and 2.72 respectively.
- Calculate G_D^i using Eq. 3.3 ($G_D^i = C_D^{-1/2}\Delta D^i$), compute the TSVD of G_D^i and choose the truncation level based on Eq. 3.52.
- If $(i = 1)$ then
  - Set $\alpha_1 = \max\{\overline{\lambda}^2, N_a\}$, where $\overline{\lambda}$ is given by Eq. 3.41.
  - Solve Eq. 3.45 for β.
- Else
  - Set $\alpha_i = \beta^{i-1}\alpha_1$.
- Endif
- For $j = 1$ to N_e
  - Perturb the observation vector with $d_{uc,j} = d_{\text{obs}} + \sqrt{\alpha_i} C_D^{1/2} Z_j$, where $Z_j \sim \mathcal{N}(0, I_{Nd})$.
  - Update the model parameters based on Eq. 3.51 (or Eq. 3.6).
- Endfor

☐ Endfor

In the case where C_D is diagonal, multiplying the data mismatch vector, $d_{uc,j} - d_j^{f,i}$, by $C_D^{-1/2}$ simply normalizes each component of the data mismatch vector by the standard deviation of the associated measurement error. Similarly, we define

$$\delta\widetilde{m}_j = \left(\Delta M^i\right)^+ \left(m_j^{a,i} - m_j^{f,i}\right), \tag{3.54}$$

where the $N_e \times N_m$ matrix, $(\Delta M^i)^+$, is the pseudo-inverse of ΔM^i. Note that $\delta\widetilde{m}_j$ is an N_e-dimensional column vector. Using Eqs. 3.53 and 3.54, we can rewrite Eq. 3.51 as

$$\delta\widetilde{m}_j = V_p \Lambda_p \Gamma_p U_p^T \delta\widetilde{d}_j, \tag{3.55}$$

where, as noted earlier, $U_p^T \delta\widetilde{d}_j$ is the projection of $\delta\widetilde{d}_j$ onto the $p-$dimensional subspace spanned by the columns of U_p. Thus, using the TSVD effectively replaces an N_d-dimensional data mismatch vector by a p-dimensional data mismatch vector. In most applications, N_e is much smaller than N_d so p will be much smaller than N_d; see Eq. 3.47.

If we truncate too many singular values, i.e., choose p too small, we may not be able to match all data well. On the other hand, if we do not truncate and $N_d >> N_e$, the final set of $\{m_j\}_{j=1}^{N_e}$ may underestimate the uncertainty in m and we may even suffer ensemble collapse. Ensemble collapse refers to the situation where all the models generated become essentially equal. This is because each member of the final ensemble of models must be a linear combination of the initial ensemble of models [53] so there are effectively only N_e degrees of freedom (the coefficients in the linear combination of the initial models) available to assimilate data; when $N_d >> N_e$, assimilating all data has a tendency to cause the set of coefficients for each updated member to be the same and in this case, there is little variability (uncertainty) in the updated (analyzed) ensemble of models.

The columns of the $N_e \times N_e$ matrix V form a basis for R^{N_e} so there exist scalars, c_k^j, $k = 1, 2, \cdots N_e$ such that

$$\delta\widetilde{m}_j = \sum_{\ell=1}^{N_e} c_\ell^j v_\ell, \tag{3.56}$$

where

$$c_\ell^j = v_\ell^T \delta\widetilde{m}_j. \tag{3.57}$$

Multiplying Eq. 3.55 by v_ℓ^T, where v_ℓ is the ℓth right singular vector of G_D^i, we obtain

$$v_\ell^T \delta\widetilde{m}_j = v_\ell^T V_p \Lambda_p \Gamma_p U_p^T \delta\widetilde{d}_j. \tag{3.58}$$

We let e_ℓ be the unit N_e-dimensional column vector with unity as its ℓth component and with all other components equal to zero. Note $v_\ell^T V_p = e_\ell^T$ so using this result in Eq. 3.58

gives

$$v_\ell^T \delta\widetilde{m}_j = e_\ell^T \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_p \end{bmatrix} \begin{bmatrix} \frac{1}{\lambda_1^2+\alpha_i} & & \\ & \ddots & \\ & & \frac{1}{\lambda_p^2+\alpha_i} \end{bmatrix} \delta\widehat{d}_j = \frac{\lambda_\ell}{\lambda_\ell^2+\alpha_i}(\delta\widehat{d}_j)_\ell. \tag{3.59}$$

for $\ell = 1, 2, \cdots p$ where $(\delta\widehat{d}_j)_\ell$ is the ℓth entry of the vector $\delta\widehat{d}_j$ and

$$(\delta\widehat{d}_j)_\ell = U_p^T(\delta\widetilde{d}_j)_\ell. \tag{3.60}$$

Multiplying Eq. 3.55 by v_ℓ^T for $\ell > p$, gives

$$v_\ell^T \delta\widetilde{m}_j = 0. \tag{3.61}$$

Eqs. 3.59 and 3.61 indicate that using TSVD effectively reparameterizes the ES-MDA update of the jth model in terms of the coefficients of the principal p right singular vectors of the dimensionless sensitivity matrix. Moreover, the weight (coefficient) of the ℓth right singular vector in Eq. 3.56 is $\lambda_\ell(\lambda_\ell^2+\alpha_i)^{-1}$ times the projection of the ℓth component of the dimensionless data mismatch terms onto the space spanned by the p principal left singular vectors. Assuming all these projected data mismatch terms are roughly equal in magnitude, the expression for the coefficients suggests that by choosing α_i larger than λ_1^2 we ensure that the weights of the right singular vectors are descending. It is interesting to note that this last statement is consistent with the discussion of Shirangi [80] and Shirangi and Emerick [81] about the choice of the Levenberg-Marquardt (LM) parameter where they parameterize the update in the model parameters using the LM algorithm based on the right singular vectors of the standard dimensionless sensitivity matrix for gradient-based history matching. Since the singular vectors corresponding to the small singular values are of high frequency and often cause rough changes in the model parameter updates, it is reasonable to avoid putting high weights on these singular vectors. However, despite the results of Tavakoli and Reynolds

[88] and Shirangi and Emerick [81], we do not necessarily need to give higher weights to the right singular vectors corresponding to the highest singular values.

3.3 Results and Discussion

In this section, we provide two history-matching examples in order to compare the performance of our methodology for choosing the inflation factors (ES-MDA-GEO) with three iterative forms of ES: ES-MDA with equal inflation factors (ES-MDA-EQL), the adaptive ES-MDA (ES-MDA-RLM) proposed by Le et al. [52], and the iterative regularized ES (IR-ES) proposed by Iglesias [48]. For IR-ES, we also consider its modified version (M-IR-ES) which we introduced earlier. In our implementation of IR-ES, we use the value of $\tau = 1/\rho$ in all cases. To investigate the performance of the methods, we define a number of performance measures which we discuss next. The first metric uses the root-mean-square-error (RMSE) to measure the closeness of the realizations of the posterior to the true model:

$$\text{RMSE} = \frac{1}{N_e}\sum_{j=1}^{N_e}\left(\frac{1}{N_m}\sum_{k=1}^{N_m}(m_{\text{true},k} - m_{j,k})^2\right)^{1/2}, \tag{3.62}$$

where $m_{\text{true},k}$ and $m_{j,k}$, respectively, are the kth model parameter from the true model and the jth realization of the posterior models. The next metric considered is the average standard deviation

$$\overline{\sigma} = \frac{1}{N_m}\sum_{k=1}^{N_m}\sigma_k, \tag{3.63}$$

where σ_k is the standard deviation of the kth model parameter. Finally, we have the average normalized objective function which measures the quality of the data match and is given by

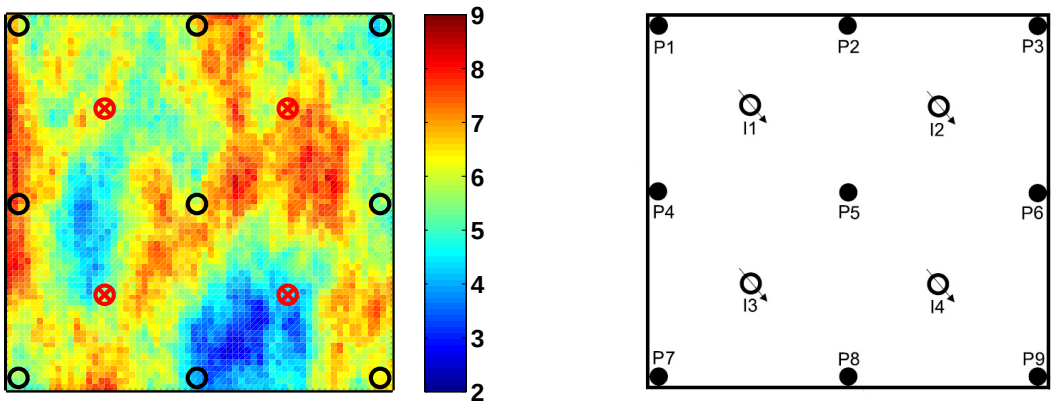
$$O_{Nd} = \frac{1}{N_e N_d}\sum_{j=1}^{N_e}(d_j^f - d_{\text{obs}})^T C_D^{-1}(d_j^f - d_{\text{obs}}). \tag{3.64}$$

The lower the value of the average normalized objective function, the closer the predicted data to the observed data. Based on the results of Gao and Reynolds [30], we expect to be able to obtain values of O_{Nd} fairly close to unity if we generate an appropriate data match.

3.3.1 Case study 1: 2D waterfooding

Example 1 pertains to a synthetic two-dimensional square reservoir with a length equal to 5120 ft in each direction. The reservoir domain is discretized on a $64 \times 64 \times 1$ grid. Each gridblock is $80 \times 80 \times 15$ ft. The true permeability field is generated using a spherical covariance function with the major range of 2560 ft and the minor range of 1280 ft where the major range is oriented in the north direction. The mean and the variance of ln(k) are, respectively, 5.5 and 1.0. The model (vector of reservoir parameters), m, consists of the set of gridblock horizontal log-permeabilities. We assume that permeability is isotropic. Figure 3.1(a) shows the true log-permeability map from which the observed data are generated. The reservoir model contains nine production wells and four water injection wells. Figure 3.1(b) shows the location of the wells. The initial reservoir pressure is 3000 psi. The injection wells operate with constant bottomhole pressure equal to 4500 psi during the whole simulation time while the production wells operate at a constant bottomhole pressure of 2800 psi for the first six months and then the bottomhole pressure constraint is changed to 2500 psi for the rest of the simulation. The observations include the monthly water injection rate of the injection wells and water and oil production rates of the production wells from the first 36 months. The history-match period, which is 36 months, is followed by a 20 month long prediction period. The predicted data from the true model are obtained by running the reservoir simulator model with the true model parameters and the measurements are acquired by adding random Gaussian noise to the data predicted from the true model. The standard deviation of the measurement error is assumed to be equal to 3% of the measured value with the minimum of 2 STB/day. An initial ensemble of 400 realizations is generated with the same covariance function used for generating the truth case. Because of the relatively large ensemble, we do not use any type of covariance localization for this example.

We assimilate the observed data to condition the ensemble of prior models to production data with different methods. Recall that ES-MDA-GEO refers to the proposed new method for choosing the inflation factors based on a geometric sequence where the geometric ratio β is obtained by solving Eq. 3.45 with $\alpha_1 = \bar{\lambda}^2$; see Eq. 3.42. Recall that this value



(a) (b)

Figure 3.1: Schematic of (a) true log-permeability field and (b) the well configuration for the synthetic two-dimensional model.

	Prior	ES-MDA-RLM ρ=0.5	IR-ES ρ=0.5	M-IR-ES ρ=0.5	ES-MDA-EQL N_a =4	ES-MDA-EQL N_a =6	ES-MDA-GEO N_a =4	ES-MDA-GEO N_a =6
RMSE	2.235	0.613	0.902	0.680	1.451	1.093	0.586	0.633
$\overline{\sigma}$	0.995	0.334	0.517	0.363	0.258	0.255	0.380	0.362
O_{Nd}	16121	1.059	8.143	6.899	8.451	1.344	25.246	5.778
Iter	-	21	9	4	4	6	4	6

Table 3.1: Comparison of the performance of different methods for $\rho = 0.5$ and different values of N_a in example 1.

of α_1 is obtained by setting $i = 1$, $\rho = 0.5$ and $\lambda_k = \bar{\lambda}$ in Eq. 3.39 where $\bar{\lambda}$ is the average of the nonzero singular values of G_D^0, the initial dimensionless sensitivity coefficient matrix (Eq. 3.3). Thus, the ES-MDA-GEO algorithm corresponds to using $\rho = 0.5$ in Eq. 3.24; therefore, the results of ES-MDA-GEO should logically be compared with results generated using $\rho = 0.5$ in the ES-MDA-RLM, IR-ES and M-IR-ES algorithms. Such a comparison is presented in Table 3.1. For this example, the first inflation factor determined using Eq. 3.42 is 1049.4 and solving Eq. 3.45 for N_a of 4 and 6, respectively, we obtain β equal to 0.102 and 0.264. Note all IR-ES and M-IR-ES results are generated with $\tau = 1/\rho$. Note that we consider two applications of equal inflation factors in Table 3.1, one uses four data assimilation steps ($N_a = 4$) and one uses six data assimilation steps ($N_a = 6$); in both cases all inflation factors are given by $\alpha_i = N_a$ for $i = 1, 2, \cdots, N_a$.

In Table 3.1, the results from the two ES-MDA cases with equal inflation factors are

inferior to those generated with the other methods because the two applications of ES-MDA with equal inflation factors give the highest values of RMSE, i.e., estimated models which are farthest from the true model, and also give the lowest value of $\bar{\sigma}$ (lowest variation in posterior models) which almost certainly means ES-MDA-EQL results in an underestimation of uncertainty for this example. Plots (a) and (b) of Figure 3.2 also show that using ES-MDA-EQL results in an estimated mean model which is quite rough compared to the mean models estimated with other methods using $\rho = 0.5$; see plots (c) and (d) of Fig. 3.2 and plots (b), (e), and (g) of Fig. 3.3. For completeness, we point out that ES-MDA with equal inflation factors and $N_a = 6$ gives an extremely good data match ($O_{Nd} = 1.344$) but at least based on the discrepancy principle (see IR-ES value of $O_{Nd} = 8.143$), we have overmatched the data in the $N_a = 6$ equal inflation factor case. Excluding the ES-MDA-EQL results of Table 3.1 from further consideration, the second observation from these results is that the best data match obtained with any method remaining under consideration is achieved with ES-MDA-RML but, again according to the discrepancy principle, this method overmatches the data; moreover ES-MDA-RLM is extremely computationally inefficient as it requires 21 iterations to reach convergence. Next, we note that ES-MDA-GEO and M-IR-ES, based on the RMSE values, give estimated models that are closer to the truth than the models obtained from IR-ES. In addition, fewer iterations are required to obtain convergence with ES-MDA-GEO and M-IR-ES than are required with IR-ES. On the other hand, IR-ES gives the largest value of $\bar{\sigma}$ obtained with any method and given the tendency of ensemble-based data assimilation methods to underestimate uncertainty [9, 92], it may be tempting to conjecture that IR-ES gives the most reliable estimate of the posterior uncertainty in the model, but without a correct sampling using a method such as Markov chain Monte Carlo (MCMC), we cannot ascertain whether this conjecture is correct. We should also point out that as we decrease the accuracy of the history-match, we tend to increase the uncertainty in the posterior ensemble of models and this can lead to an overestimation of uncertainty. In Chapter 4 we consider a new for using MCMC for sampling the posterior that is feasible for problems with a small number of parameters, but too computationally expensive which has over 4,000 parameters.

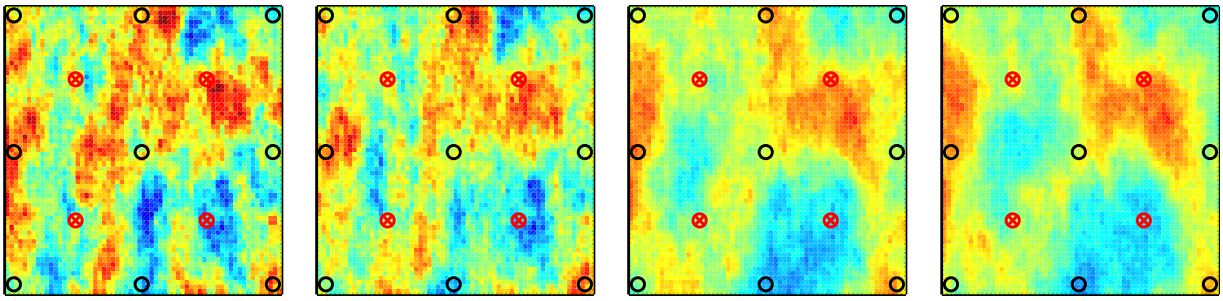
Standard MCMC methods such as random walk are far too computationally expensive for practical applications and have their own sampling issues in practice [23, 59]. Finally, as should be the case, note that the estimates of the mean of the posterior log-permeability field in plots (c) and (d) of Fig. 3.2 and (b),(e) and (g) of Fig. 3.3 are smoother than the true model (Fig. 3.1(a)) but capture some of the dominant large features of the true model. Note that the posterior mean of the log-permeability field generated with IR-ES and M-IR-ES (plots (e) and (g) of Fig. 3.3) do not capture the low permeability region between the wells I4 and P8 (see Fig. 3.1) or the high permeability region connecting P2 and P5 as well as ES-MDA-GEO does. Overall, ES-MDA-GEO requires a fewer number of data assimilation steps and yet resolves the permeability map better compared to other methods under consideration in terms of identifying the high and low permeability regions of the true permeability map.

	Prior	ES-MDA-RLM			IR-ES		M-IR-ES		
		ρ=0.2	ρ=0.5	ρ=0.8	ρ=0.2	ρ=0.5	ρ=0.2	ρ=0.5	ρ=0.8
RMSE	2.235	0.698	0.613	0.601	1.333	0.902	0.721	0.680	0.811
$\overline{\sigma}$	0.995	0.335	0.334	0.320	0.538	0.517	0.378	0.363	0.329
O_{Nd}	16121	1.864	1.059	1.054	28.665	8.143	36.741	6.899	2.689
Iter	-	9	21	45	4	9	3	4	7

Table 3.2: Comparison of the performance of different methods for different values of ρ in example 1.

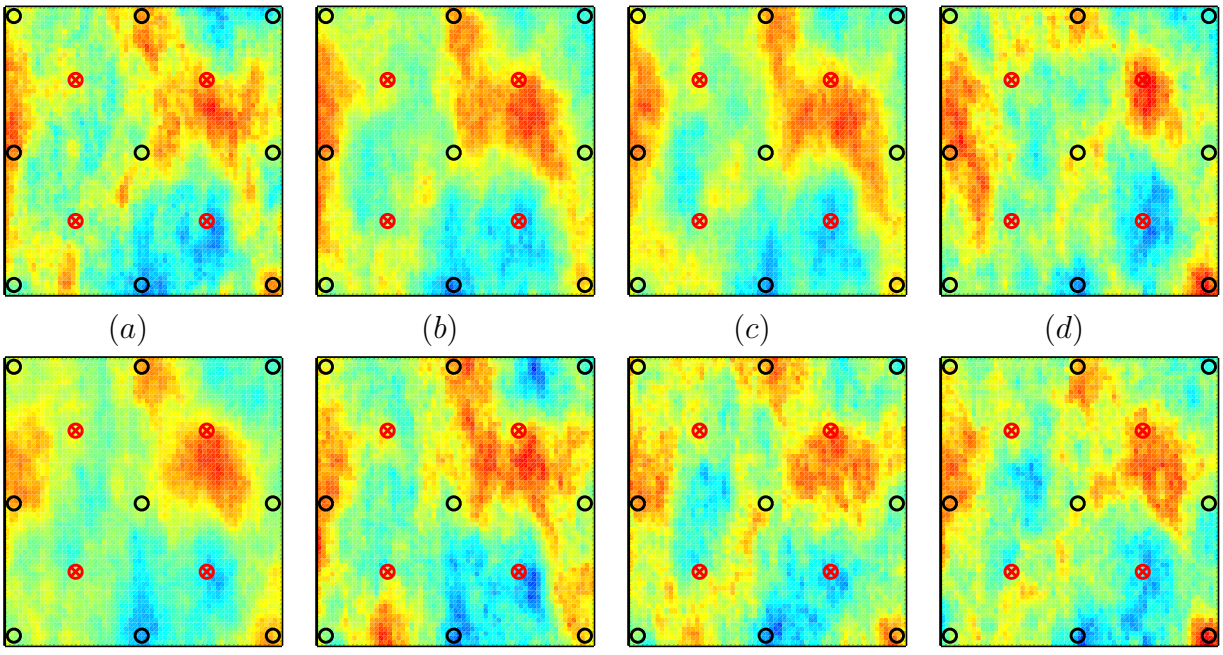
The performance of ES-MDA-RLM, IR-ES and M-IR-ES depends on the value of ρ. For this example, we tried three values of ρ namely, 0.2, 0.5 and 0.8 and the results are summarized in Table 3.2. To include a complete comparison in Table 3.2, the results of $\rho = 0.5$ are repeated from Table 3.2. Note that Iglesias [48] recommended the use of $\rho = 0.8$ and $\tau \simeq 1$, but in this example, IR-ES with $\tau = 1/\rho = 1.25$, which results in faster termination of the method, did not converge after 200 iterations so we obtained no results for this case. The general observation of the results presented in Table 3.2 is that increasing the value of ρ tends to result in models that are closer to the true model (except for M-IR-ES which gives higher value of RMSE with $\rho = 0.8$ than with $\rho = 0.5$). Also for all

methods in Table 3.2, a higher value of ρ gives a smaller value of $\bar{\sigma}$, a smaller value of O_{Nd}, and a higher number of data assimilation steps. Note that the estimates of the mean of the posterior log-permeability field obtained with $\rho = 0.2$ in plots (a), (d), and (f) of Fig. 3.3 are rougher compared to the other plots in Fig. 3.3, which suggest that $\rho = 0.2$ does not provide sufficient damping of the change in model parameters.



(a) (b) (c) (d)

Figure 3.2: The posterior mean of the log-permeability (a) ES-MDA-EQL $N_a = 4$, (b) ES-MDA-EQL $N_a = 6$, (c) ES-MDA-GEO $N_a = 4$, (d) ES-MDA-GEO $N_a = 6$. The colorbar scale is the same as in Fig. 3.1(a).



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 3.3: The posterior mean of the log-permeability in example 1. (a) ES-MDA-RLM $\rho = 0.2$, (b) ES-MDA-RLM $\rho = 0.5$, (c) ES-MDA-RLM $\rho = 0.8$, (d) IR-ES $\rho = 0.2$, (e) IR-ES $\rho = 0.5$, (f) M-IR-ES $\rho = 0.2$, (g) M-IR-ES $\rho = 0.5$, and (h) M-IR-ES $\rho = 0.8$. The colorbar scale is the same as in Fig. 3.1(a).

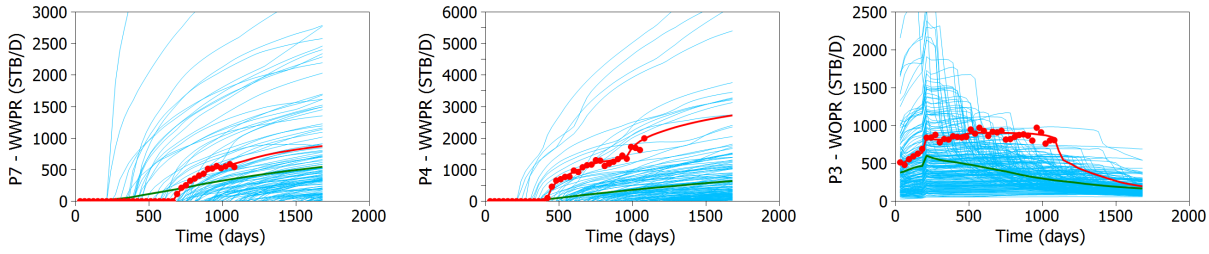


Figure 3.4: Plot of water production rates of wells P7 and P4 and oil production rate of well P3 obtained using the initial ensemble in example 1. The red circles show the history, the red line is true model response, the green line is the mean of the ensemble responses, and the blue lines represent all the ensemble responses.

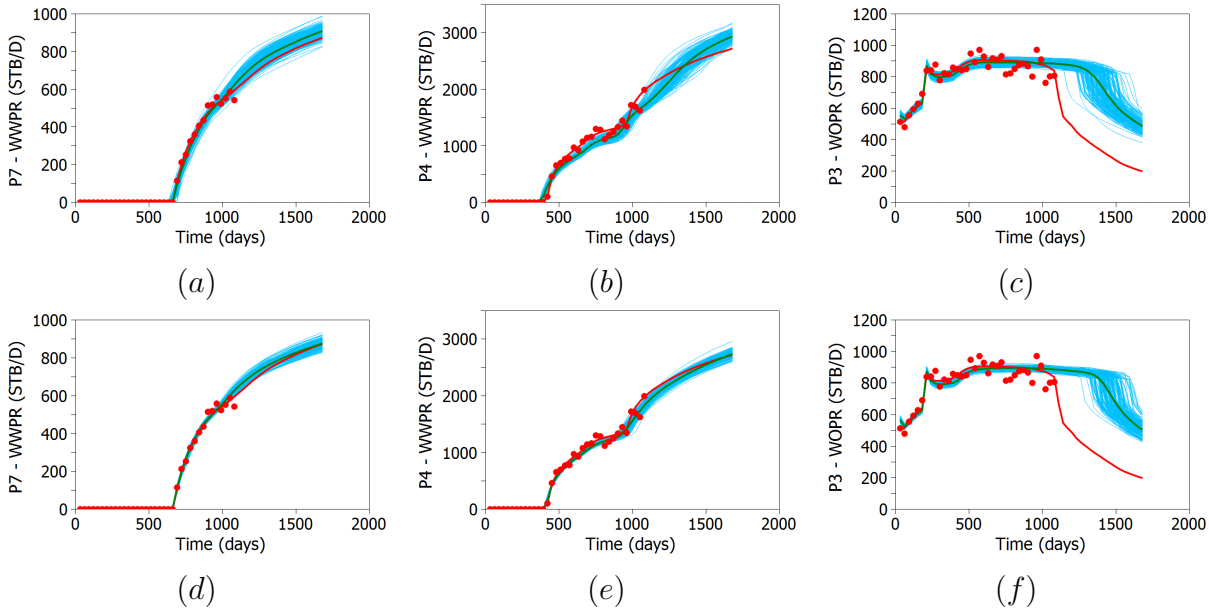


Figure 3.5: Plot of water production rates of wells P7 and P4 and oil production rate of well P3 in example 1. The legend is the same as that of Fig. 3.4. (a)-(c) with ES-MDA-EQL with $N_a = 4$ (d)-(f) with ES-MDA-EQL with $N_a = 6$.

Figure 3.4 shows the initial ensemble predictions for some wells and the data matches and future predictions obtained from different methods for those wells are illustrated in Figures 3.5-3.7. The plots shown are representative of the results from all wells. Note that the data from the first $1,080$ days (36 months) are used for history matching and the predictions for the next 20 months are shown to examine the predictive capabilities of the history-matched models. Despite the aforementioned issues with the result obtained using equal inflation factors for this example, consistent with the results of Table 3.1, the results of

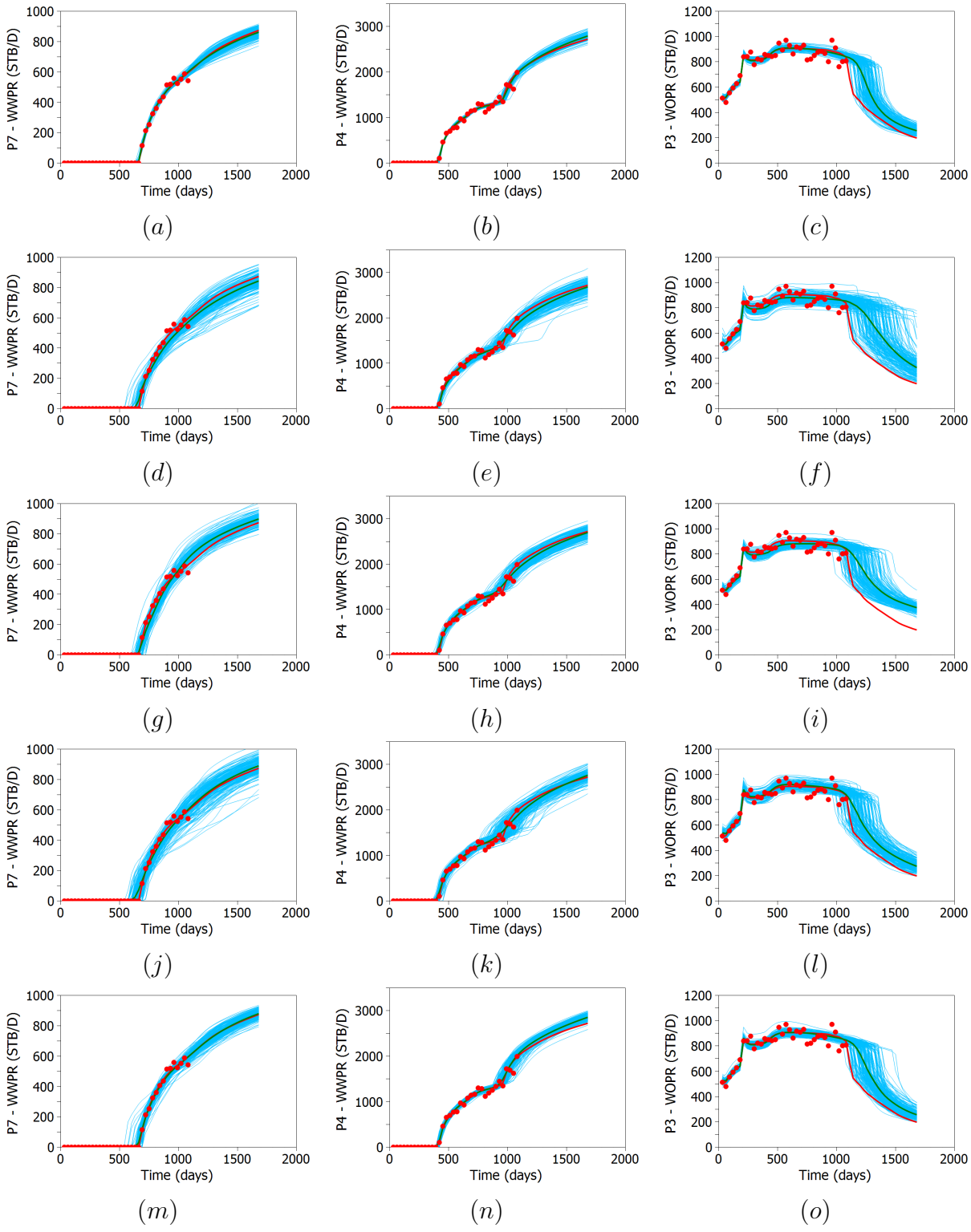


Figure 3.6: Plot of water production rates of wells P7 and P4 and oil production rate of well P3 in example 1. The legend is the same as that of Fig. 3.4. (a)-(c) ES-MDA-RLM with $\rho = 0.5$, (d)-(f) IR-ES with $\rho = 0.5$, (g)-(i) M-IR-ES with $\rho = 0.5$, (j)-(l) ES-MDA-GEO with $N_a = 4$, (m)-(o) ES-MDA-GEO with $N_a = 6$.

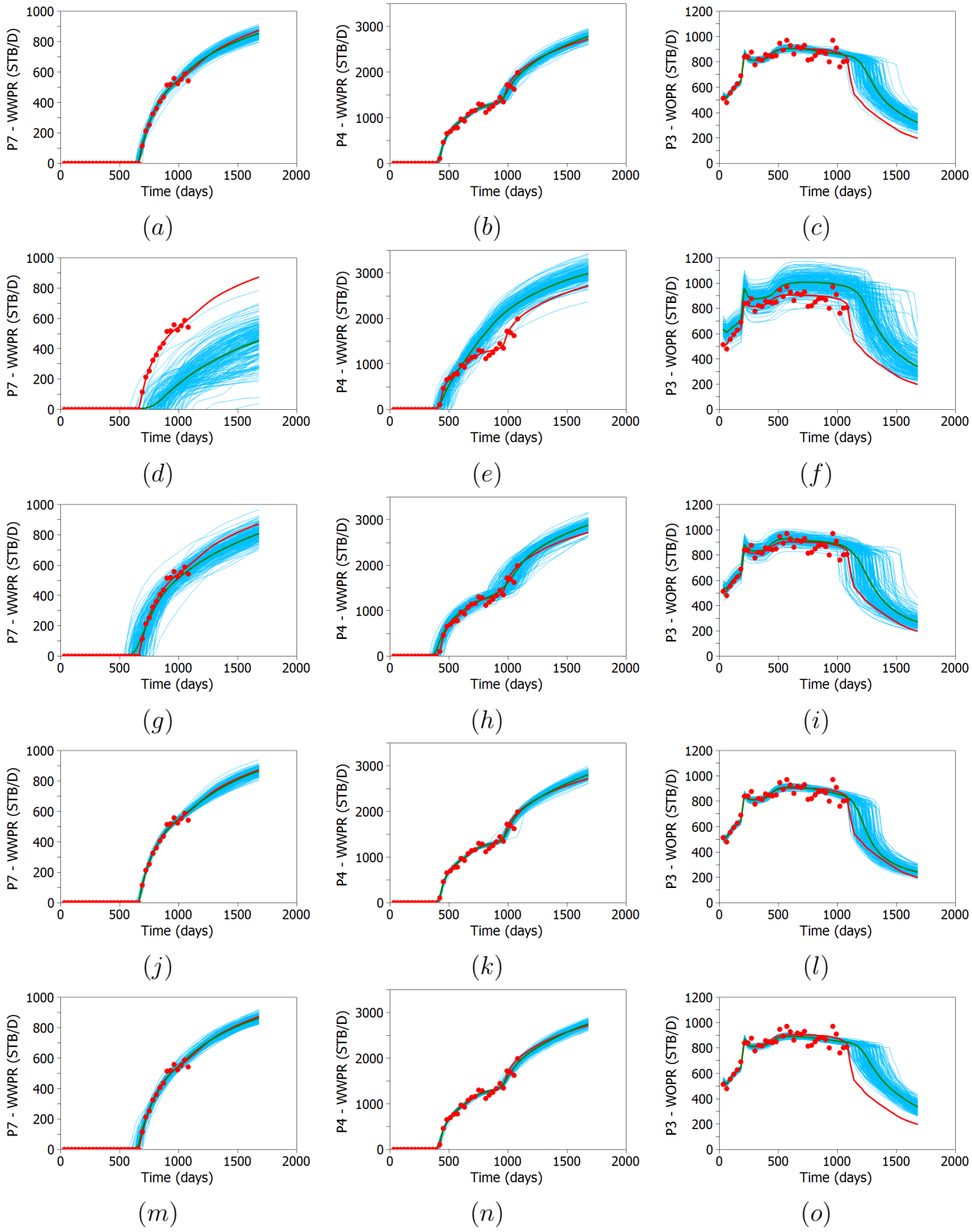


Figure 3.7: Plot of water production rates of wells P7 and P4 and oil production rate of well P3 in example 1. The legend is the same as that of Fig. 3.4. (a)-(c) ES-MDA-RLM with $\rho = 0.2$, (d)-(f) IR-ES with $\rho = 0.2$, (g)-(i) M-IR-ES with $\rho = 0.2$, (j)-(l) ES-MDA-RLM with $\rho = 0.8$, (m)-(o) M-IR-ES with $\rho = 0.8$.

Fig. 3.5 indicate good data matches are obtained with equal inflation factors. Good future rate predictions ($t > 1,080$ days) are obtained in most cases but the predicted oil rate at well P3 (Fig. 3.5 (c) and (f)) is much higher than the true rate. One should note, however, during the period from 200 days to $1,080$ days, the observed rate is approximately constant and there is nothing in the data to suggest that the oil rate will decline immediately after the historical period where data are measured. Thus, predicting the future oil rate at well P3 is quite difficult. Nevertheless, as shown in Fig. 3.6 and Fig. 3.7, all the iterative ensemble-based history matching methods yield a better prediction of well P3's future oil rate compared to ES-MDA with equal inflation factors; however, the predictions of well P3's oil rate shown in Figs. 3.6 and 3.7 are still biased, but the ensemble of predictions obtained with ES-MDA-GEO is the least biased. As mentioned earlier, because the inverse inflation factors always sum to unity (Eq. 2.79) when ES-MDA-GEO is applied, ES-MDA is guaranteed to provide a correct characterization of the posterior pdf in the linear-Gaussian case as the ensemble size goes to infinity whereas, IR-ES and M-IR-ES do not sample the posterior pdf correctly if Eq. 2.79 is not satisfied. In this particular case, the sum of the inverse of the inflation factors for IR-ES and M-IR-ES with $\rho = 0.5$, respectively, are 0.047 and 0.510. Except for the preceding comments on the sum of the inverse inflation factors, there is no intrinsic property of ES-MDA-GEO that makes it give less-biased predictions than are obtained with IR-ES or M-IR-ES for this example; in fact, the objective of ES-MDA-GEO is to find a method that is far more computationally efficient than IR-ES or M-IR-ES that also allows one to specify a priori the number of data assimilation steps but gives reasonable estimates of the true model and posterior uncertainty as well as a good data match without introducing roughness in the posterior models. We can also see that all methods except the IR-ES with $\rho = 0.2$ provide good data matches and future predictions of the water production rate of wells P7 and P4; see plots (d) and (e) of Fig. 3.7.

3.3.2 Case study 2: Brugge model

The Brugge benchmark case is a synthetic reservoir designed by the Dutch Organiza-

tion for Applied Scientific Research (TNO) to test the performance of history matching and production optimization methods in a closed-loop reservoir management framework. The reservoir simulation model consists of 139×48×9 grid cells with 44,550 active cells and the reservoir contains 10 water injection wells and 20 production wells. In the original Brugge data set, TNO provided 104 realizations of porosity (PORO), permeability in x-, y-, and z-directions (PERMX, PERMY, and PERMZ), net-to-gross ratio (NTG) and initial water saturation. A more detailed description of the model can be found in Peters et al. [73]. During the first 10 years of production, each injection well operates under a constant injection rate and each production well produces at a specified total liquid rate. Monthly oil and water production rates at producers and bottomhole pressures at both injection and production wells for the first ten years of the reservoir life were made available for history matching purposes. After history matching, the study participants were asked to estimate the optimal well controls that maximize the net-present value (NPV) of production for the next ten years of the reservoir life where each well has three inflow-control valves (ICV's).

Since the true model is unknown to us, we pick the 44th realization as the true model in order to have a reference for comparison of the performance of the history-matching methods. Gaussian noise is added to the first ten years of production data generated using the true reservoir model in order to generate the observed data. In this procedure, we assume a measurement error with a standard deviation equal to 7.25 psi for the bottomhole pressure data and a measurement error with a standard deviation equal to 30 STB/D for the oil rates and the water rates. Before water breakthrough, we assume that the standard deviation of measurement error for the water rates is 2 STB/D. Here, we consider only the history-matching aspects of the problem, where the history-matching parameters are gridblock porosities (PORO), gridblock log-permeabilities in the x (PERMX), y (PERMY), and z (PERMZ) directions and net-to-gross ratio (NTG) at each gridblock. The remaining 103 realizations of PORO, PERMX, PERMY, PERMZ, and NTG are used for history-matching the 10 years of production data. Because the number of realizations is much smaller than the number of model parameters, we use a distance-based Kalman gain localization

to increase the degrees of freedom, mitigate the effect of spurious correlations and avoid ensemble collapse [99].

The history matching results from four methods, IR-ES with $\rho = 0.5$, M-IR-ES with $\rho = 0.5$, ES-MDA with 6 equal inflation factors and ES-MDA-GEO with 6 data assimilation steps are compared for this example. The results of ES-MDA-RLM with $\rho = 0.5$ are not included as it did not converge for this example. The first inflation factor for ES-MDA-GEO based on Eq. 3.42 is 828.8 for this example and solving Eq. 3.45 with $N_a = 6$, we obtain β equal to 0.278. Iglesias [48] claimed that in IR-ES there is no need for covariance localization, so we use IR-ES both with and without covariance localization. Here, IR-ES refers to the original method without covariance localization while IR-ES-WL refers to the implementation of IR-ES with covariance localization. In this example, RMSE and $\overline{\sigma}$ are calculated separately for each type of model parameter i.e., PORO, PERMX, PERMY, PERMZ, and NTG. For example, RMSE of PORO is calculated using Eq. 3.62 where N_m is replaced by the number of active gridblocks in the model, and $m_{\text{true},k}$ and $m_{j,k}$ are, respectively, the porosity of the kth active cell from the true model and jth realization. Table 3.3 summarizes the results obtained with different methods for the Brugge model.

		Prior	IR-ES-WL ρ=0.5	IR-ES ρ=0.5	M-IR-ES ρ=0.5	ES-MDA-EQL N_a =6	ES-MDA-GEO N_a =6
RMSE	PERMX	1.7469	1.4764	1.3187	1.4748	1.4249	1.4117
	PERMY	1.7534	1.4812	1.3245	1.4799	1.4306	1.4167
	PERMZ	2.0931	1.7800	1.5798	1.7835	1.7104	1.6955
	PORO	0.0318	0.0310	0.0297	0.0308	0.0304	0.0302
	NTG	0.2261	0.2179	0.2054	0.2128	0.2061	0.2083
$\overline{\sigma}$	PERMX	1.0801	0.8791	0.6829	0.8466	0.7436	0.7500
	PERMY	1.0811	0.8798	0.6839	0.8475	0.7445	0.7507
	PERMZ	1.3015	1.0843	0.8479	1.0351	0.9026	0.9146
	PORO	0.0174	0.0166	0.0141	0.0157	0.0140	0.0142
	NTG	0.1059	0.0980	0.0799	0.0895	0.0750	0.0789
O_{Nd}		692.362	5.265	5.251	1.256	1.372	3.693
Iter		-	12	5	4	6	6

Table 3.3: Comparison of the performance of different methods for the Brugge case.

We first consider the results obtained from IR-ES with no localization (IR-ES) in comparison with those obtained from IR-ES with localization (IR-ES-WL). For both methods, we obtain essentially the same value of the average normalized objective function (about 5.3)

which is as expected, because both methods use the same stopping criteria (Eq. 3.25). However, as shown in Table 3.3, compared to the localization case, IR-ES without localization results in lower average posterior standard deviations (smaller uncertainty in the parameters). As expected, this comparison of the two IR-ES methods suggest that localization is actually required in IR-ES because it is expected that in the no localization case, sampling errors due to the limited ensemble size will result in spurious correlations that will cause a reduction in the uncertainty in parameters at gridblocks far from any data measurement whereas, in reality, these parameters should not be changed by data assimilation if the data are insensitive to these gridblock parameters [2, 12, 16, 23, 99]. In short, it appears that IR-ES underestimates to some extent the posterior variance because it causes an incorrect reduction in the variance of parameters at gridblocks far from wells. Although Kalman gain localization mitigates this erroneous reduction in uncertainty, we cannot and do not assert that the standard derivations obtained with IR-ES-WL are correct. Note however that the two ES-MDA with six data assimilations as well as M-IR-ES give standard derivations values between the corresponding ones obtained by IR-ES-WL (which uses Kalman gain localization) and by IR-ES (except for NTG and PORO for ES-MDA-EQL) which again suggests that IR-ES with no localization underestimates the posterior variance in model parameters. Note the results on average standard derivation for the Brugge example are distinctly different than in the first example, where localization was not needed and IR-ES with no localization gave higher values of the average posterior standard deviation than was obtained using ES-MDA with geometric inflation factors. It is important to state, however, that IR-ES-WL requires 12 data assimilation steps to converge whereas only 6 data assimilation steps are used in the two ES-MDA methods. IR-ES with no localization requires only 5 iterations to obtain convergence in this example, but as noted above localization is required to obtain reliable results. It is important to state that M-IR-ES with covariance localization, which selects the inflation factors at each iteration based on Eq. 3.40, converges in 4 data assimilation steps and gives values of RMSE similar to those obtained with the other implementations of ES-MDA while giving average standard deviation values similar

to those obtained with IR-ES-WL and slightly higher than those generated with ES-MDA-GEO. Also note that, according to the results of Table 3.3, our new method for generating inflation factors (ES-MDA-GEO) yields results that are very similar to using six equal inflation factors although using geometric inflation factors does yield very slightly larger values of average standard derivations than are generated with equal inflation factors.

Since we consider only the history-matching aspects of the problems, we use the optimal controls obtained by Chen and Oliver [13] (see also Peters et al. [72]) for the next 20 years to test the predictions of the history-matching problems. Figure 3.8 shows the water production rates for wells P7, P9, and P20 for 30 years which includes 20 years of future predictions obtained using the initial ensemble. Figure 3.9 shows the water production rates for the same wells after history-matching using different methods. These wells are representative of the quality of the history-match and predictions in all wells. We can observe that more uncertainty is associated with the results of IR-ES and IR-ES-WL in the prediction period than is associated with the predictions from the other methods but part of the high variability of predictions in well rates arises from the fact that the data match obtained with IR-ES and IR-ES-WL is worse than the data match obtained with the other methods. As pointed out by Emerick and Reynolds [24], decreasing the accuracy of the data match can lead to an overestimation of the uncertainty in future predictions. The ensemble of predicted water rates at well P7 tends to be biased regardless of the data assimilation method used in part because the observed water rate at well 7 is zero throughout the historical period. However, this bias is greater for the ES-MDA-EQL and ES-MDA-GEO results. Note that IR-ES-WL and IR-ES do improve the predictions of the water rate for well P20 compared to the initial ensemble, although both methods improved the data match in the historical period.

The first realizations of the log-permeability in x-direction and porosity fields are shown in Figures 3.10 and 3.11 respectively. We can see that there the posterior plots are not significantly changed compared to the prior fields and after the history-match the geological structures of the prior model are preserved. This example does not suffer from

the overshooting and undershooting of the properties.

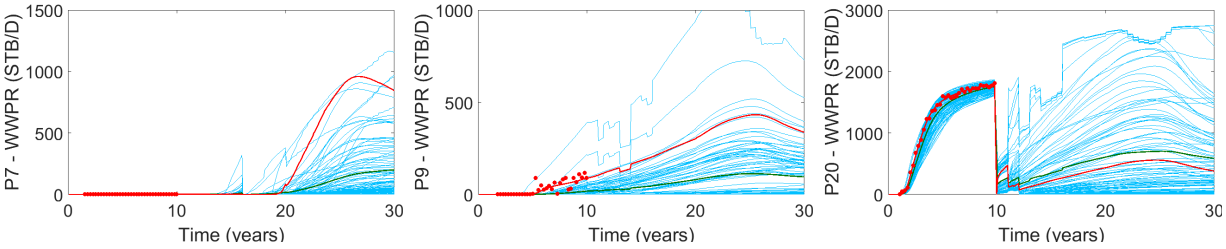


Figure 3.8: The water production rate of wells P7, P9 and P20 obtained using the prior models for Brugge case. The red circles show the history, the red line is true model response, the green line is the mean of the ensemble responses, and the blue lines represent all the ensemble responses.

3.3.3 Case study 3: PUNQ-S3 model

The PUNQ-S3 model contains $19 \times 28 \times 5$ grid blocks with $\Delta x = \Delta y = 590.55$ ft; 1761 blocks are active [3, 29]. The field is bounded to the east and south by a fault, and is supported by a fairly strong aquifer to the north and west. A small gas cap is located in the center of the dome shaped structure. The field initially contains 6 production wells located around the gas-oil contact. None of the wells are perforated in the top two layers. The production consists of an extended well test during the first year, followed by a three year shut-in period, and then 12.5 years of production with two-week shut-in periods annually used to collect buildup pressure data. The extended well testing period is divided into four 3-month periods; during successive periods all wells operate under constant oil rate of 628.9, 1257.9, 628.9, and 314.5 STB/D, respectively. The oil production rate during the last 12.5 years is 943.5 STB/D except for the shut-in periods.

We consider the first 12 years as the history matching period and the remaining 4.5 years as the forecast period. The observed data include the bottomhole pressure (BHP) during both production shut-in periods and gas production rate (GPR) and water production rate (WPR) during the production periods. Gaussian random noise with mean of zero is added to the noise free data obtained by running the reservoir simulator using the true model. The standard deviation for the pressure measurement error is equal to 14.7 psi during shut-in

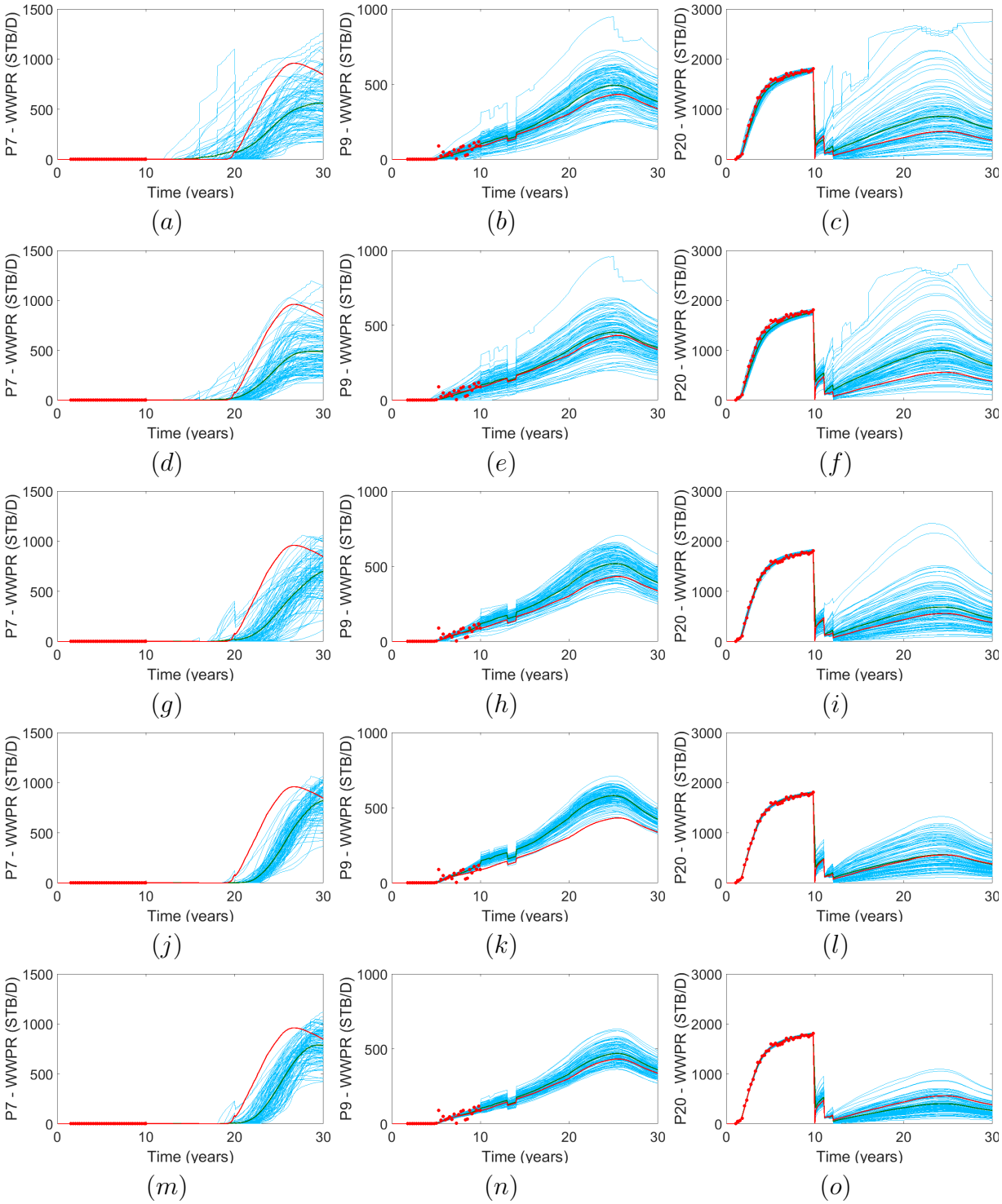


Figure 3.9: The water production rate for history and prediction periods of wells P7, P9, and P20 for Brugge case. The rows ordered from top to bottom are obtained with IR-ES-WL, IR-ES, M-IR-ES, ES-MDA-EQL, and ES-MDA-GEO, respectively. The legend is the same as Fig. 3.8.

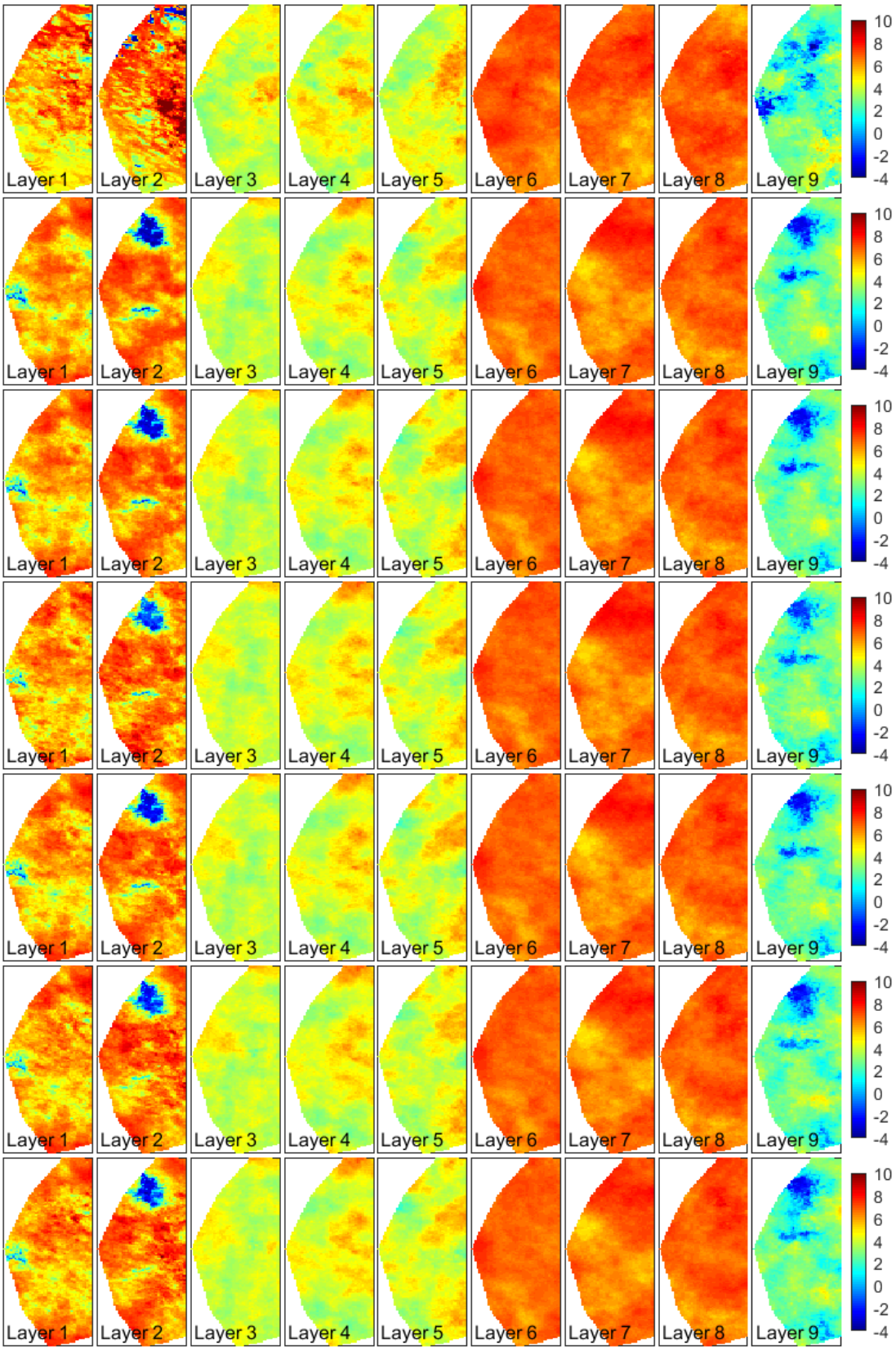


Figure 3.10: Plot of the log-permeability in x-direction. The first row corresponds to the true model, the rest are the realization 1 from prior, IR-ES-WL, IR-ES, M-IR-ES, ES-MDA-EQL, and ES-MDA-GEO respectively.

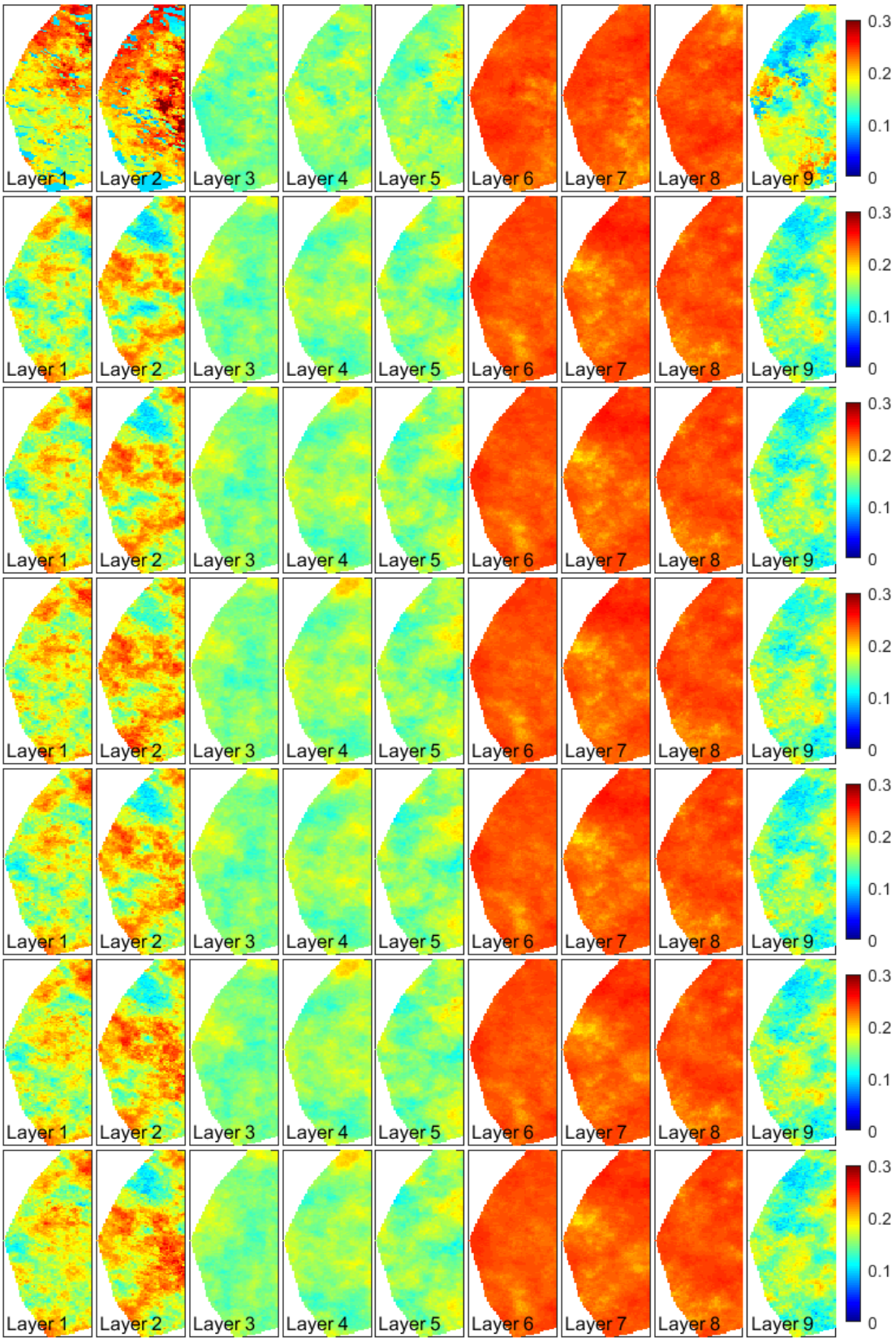
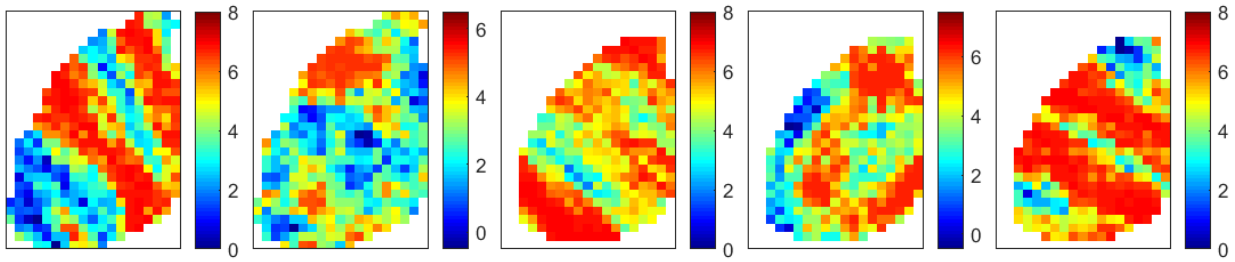


Figure 3.11: Plot of porosity map. The first row corresponds to the true model, the rest are the realization 1 from prior, IR-ES-WL, IR-ES, M-IR-ES, ES-MDA-EQL, and ES-MDA-GEO respectively.

periods, and is equal to 44.1 psi for any pressure measurement obtained during a flowing period. The standard deviations for the measurement errors for water and gas production rates are set equal 3% of the data predicted by the true model.



(a) (b) (c) (d) (e)

Figure 3.12: True horizontal log-permeability field for PUNQ-S3 case. (a) layer 1, (b) layer 2, (c) layer 3, (d) layer 4, and (e) layer 5

The history-matching parameters are horizontal permeability (PERMX), vertical permeability (PERMZ) and porosity (PORO) at each active gridblock. The true horizontal log-permeability for all layers is shown in Figure 3.12. The initial realizations of the model parameters are generated with sequential Gaussian co-simulation based on the geostatistical parameters described in Gao et al. [31] without using any hard data at well locations. An ensemble of size 400 is used to perform history matching. Note in this example, we only consider $\rho = 0.5$ for ES-MDA-RLM, IR-ES and M-IR-ES methods and $N_a = 6$ for ES-MDA-EQL and ES-MDA-GEO methods. We use Kalman gain localization for all methods except for IR-ES method. For ES-MDA-GEO, the first inflation factor based on Eq. 3.42 is 335.8 for this example and solving Eq. 3.45 with $N_a = 6$, we obtain $\beta = 0.339$. Since the model parameters in this example are porosity and horizontal and vertical permeability of the gridblocks, similar to the Brugge model, RMSE and $\overline{\sigma}$ are calculated for each type of model parameters separately, and the results are presented in Table 3.4. The final values of the average normalized objective function obtained with IR-ES and M-IR-ES are fairly similar. As convergence of these two methods are based on the discrepancy principle, the results of Table 3.4 suggest that ES-MDA-RML, and perhaps even ES-MDA-EQL, may have produced an overmatch of the data at least based on the discrepancy principle. The values of the average normalized objective function obtained with IR-ES, M-IR-ES, and ES-MDA-GEO

are close, and as the first two methods have converged based on the discrepancy principle (Eq. 3.25), we can say that ES-MDA-GEO has resulted in an acceptable data match. ES-MDA-RLM results in the lowest values of the RSMEs overall, the method requires 48 data assimilation steps to obtain convergence which makes it far less computationally efficient than the other methods. The RSME and $\overline{\sigma}$ results of ES-MDA-GEO shown in Table 3.4 are very close to those obtained with ES-MDA-RLM but it takes only 6 iterations opposed to the 48 iterations required with ES-MDA-RLM. Note IR-ES has converged in 9 iterations whereas only 6 iterations were used with ES-MDA-GEO and ES-MDA-EQL, and modified M-IR-ES converged in only 4 iterations. In addition, ES-MDA-GEO gives significantly lower RMSE values than are obtained with IR-ES, M-IR-ES, and ES-MDA-EQL which indicates that the models obtained with ES-MDA-GEO are closer to the true model. Despite these issues with IR-ES, this algorithm gives a larger values of $\overline{\sigma}$ compared to all other methods but without knowing the correct posterior we cannot assume that this method gives the best estimate of the posterior uncertainty. Note that ES-MDA-EQL gives the lowest values of $\overline{\sigma}$, and comparing the results of ES-MDA-EQL with those obtained with ES-MDA-GEO, we find that using our method for choosing the inflation factors based on a geometric sequence will alleviate the underestimation of uncertainty without increasing the computational cost of the ES-MDA method.

		Prior	ES-MDA-RLM ρ=0.5	IR-ES ρ=0.5	M-IR-ES ρ=0.5	ES-MDA-EQL N_a =6	ES-MDA-GEO N_a =6
RMSE	PERMX	2.2457	1.9128	2.0547	2.0049	1.9775	1.9251
	PERMZ	2.4992	2.0463	2.2227	2.1103	2.1289	2.0498
	PORO	0.0974	0.0808	0.0877	0.0860	0.0823	0.0801
$\overline{\sigma}$	PERMX	1.4218	0.7022	1.0043	0.8398	0.6728	0.8282
	PERMZ	1.6238	0.8198	1.1609	0.9737	0.7882	0.9635
	PORO	0.0599	0.0300	0.0431	0.0361	0.0290	0.0356
O_{Nd}		6537.15	1.40	8.93	12.16	3.49	11.05
Iter		-	48	9	4	6	6

Table 3.4: Comparison of the performance of different methods for the PUNQ-S3 model.

Figure 3.13 shows the predictions of the initial ensemble for some of the wells. Figures 3.14 and 3.15 shows the predictions of the history-matched models using different methods. Note that in all of these figures the vertical lines correspond to the shut-in and opening

periods of the wells were the rates drop to zero as we shut-in the wells and starts producing after opening the well. As it can be observed in Fig. 4.12, the bottomhole pressure of well PRO-1 drops in most of the posterior realizations in the prediction period which results in producing significant amount of gas while it is not the case for the true model. However, ES-MDA-GEO gives the best predictions of the bottomhole pressure and gas production rate of well PRO-1 and the mean of the ensemble are very close to the predictions of the true model for this well. According to the results of Fig. 4.13, none of the history-matching methods are able to predict the water breakthrough of the well PRO-12. During the historical period well PRO-12 has zero water rate and therefore it is very difficult to predict the water breakthrough for this well.

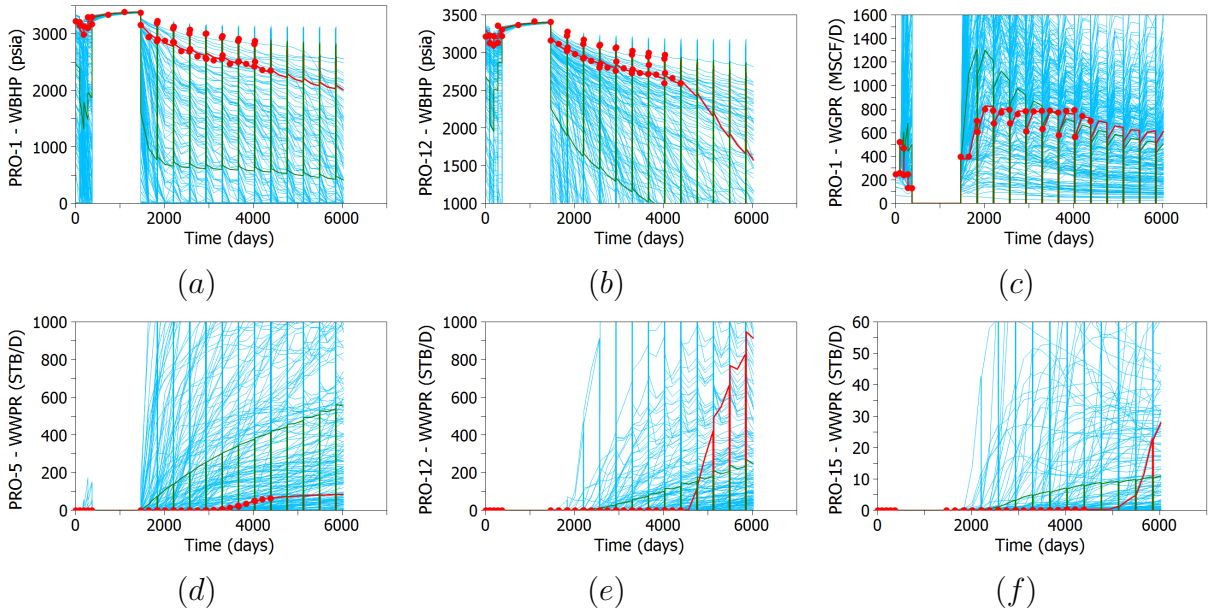


Figure 3.13: The data matches and predictions obtained with the initial ensemble for PUNQ-S3 model. The red circles show the history, the red line is true model response, the green line is the mean of the ensemble responses, and the blue lines represent all the ensemble responses.

Figures 3.16 and 3.17 show horizontal log-permeability for all layers obtained with different methods for the 10th and 20th realizations respectively. This figures show that IR-ES seems to be overshooting the horizontal log-permeability of layer 3 and 4 for both of these realizations. Also, ES-MDA-RLM and ES-MDA-EQL show slight overshooting for layers 3 and 4 in the 20th realization of horizontal log-permeability.

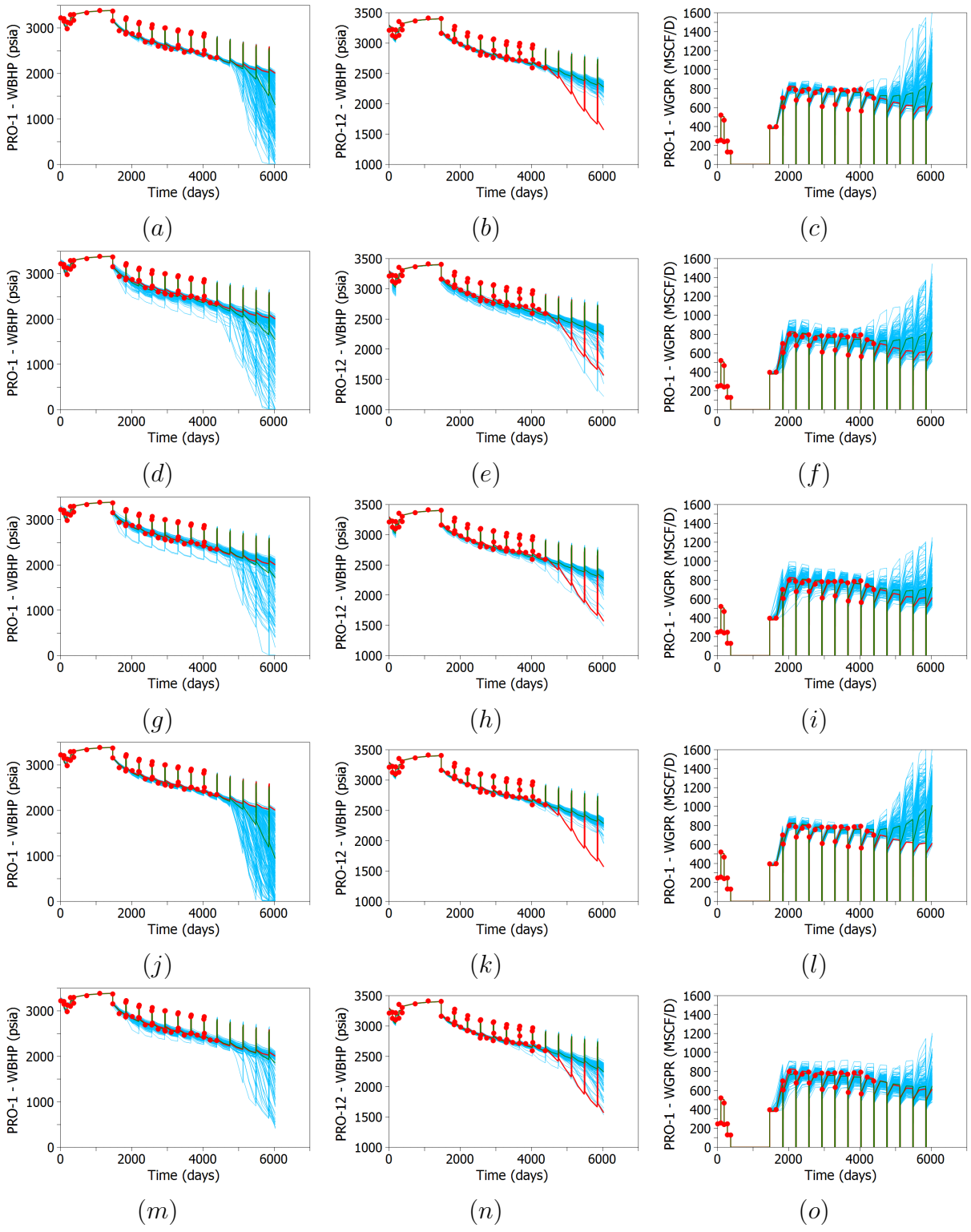


Figure 3.14: Plot of bottomhole pressures of wells PRO-1 and PRO-12 and gas production rate of well PRO-1 for PUNQ-S3 model. (a)-(c) ES-MDA-RLM, (d)-(f) IR-ES, (g)-(i) M-IR-ES, (j)-(l) ES-MDA-EQL and (m)-(o) ES-MDA-GEO. The legend is the same as in Fig. 3.13.

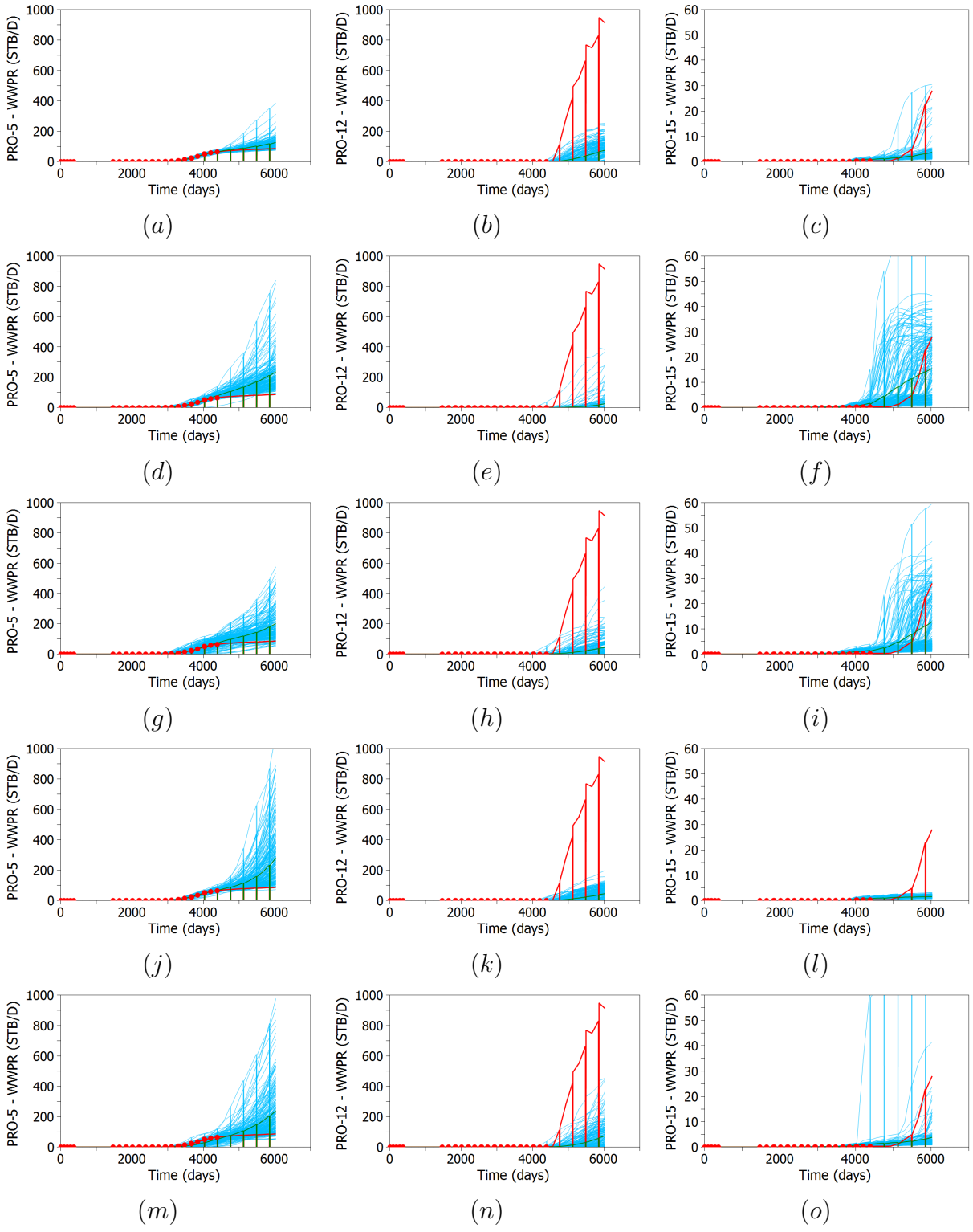


Figure 3.15: Plot of water production rates of wells PRO-5, PRO-12 and PRO-15 for PUNQ-S3 model. (a)-(c) ES-MDA-RLM, (d)-(f) IR-ES, (g)-(i) M-IR-ES, (j)-(l) ES-MDA-EQL and (m)-(o) ES-MDA-GEO. The legend is the same as in Fig. 3.13.

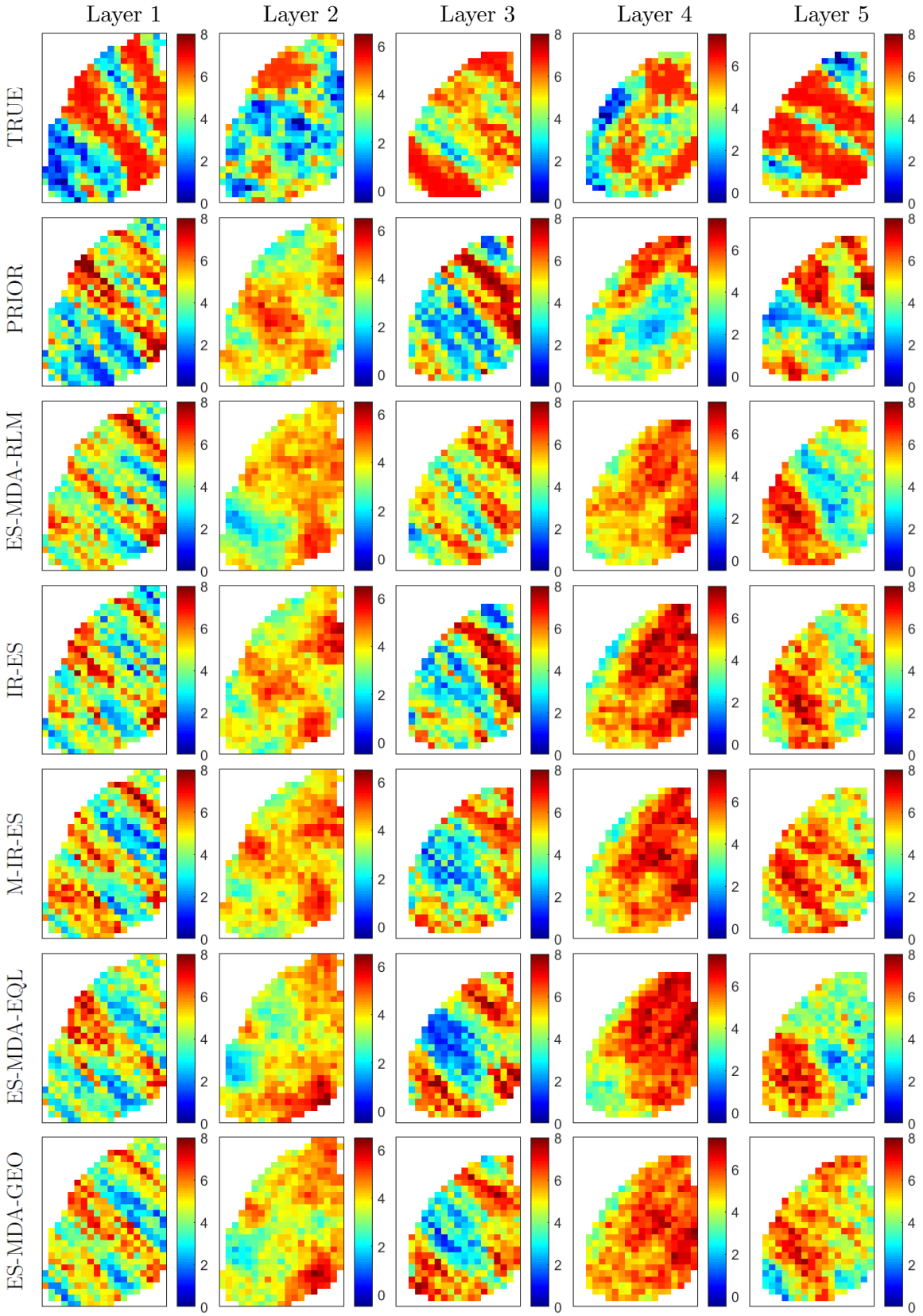


Figure 3.16: Horizontal log-permeability of the 10th realization compared to the true for PUNQ-S3 model

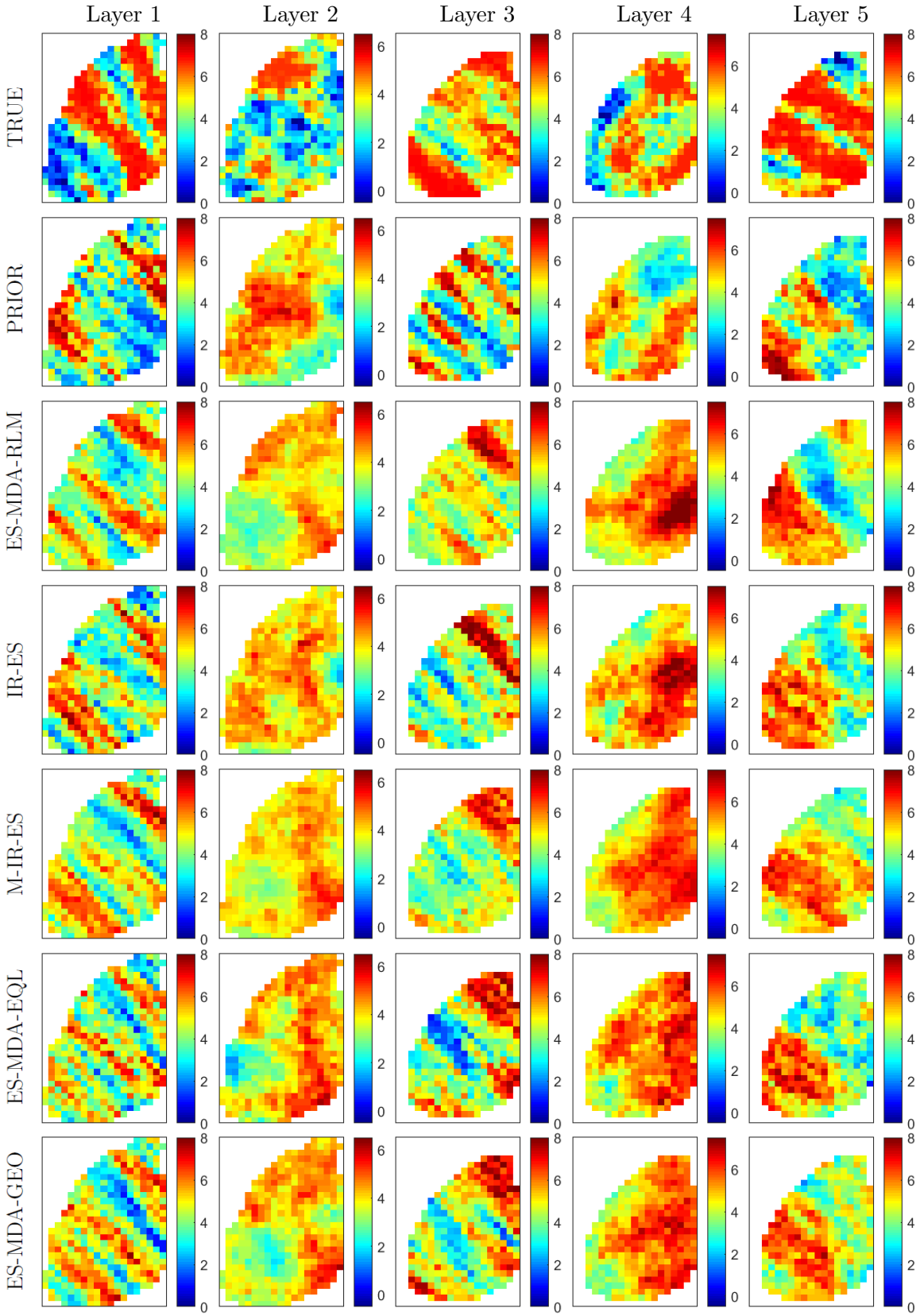


Figure 3.17: Horizontal log-permeability of the 20th realization compared to the true for PUNQ-S3 model.

Chapter 4

AN EFFICIENT MCMC FOR UNCERTAINTY QUANTIFICATION

Our objective in this chapter is to provide a methodology for finding a better characterization of the posterior uncertainty than is obtained with ES, EnKF, or ES-MDA. To do so, we combine the distributed Gauss-Newton (DGN) method of Gao et al. [32] with Markov Chain Monte Carlo. Designing an efficient MCMC method in order to characterize a target pdf is a difficult task especially if the target pdf is multimodal where a simple random walk MCMC can remain trapped near one mode for an extremely large number of iterations before moving to a nearby mode. If the modes of the target pdf are separated with a wide zero or extremely low probability region, it is almost impossible to move from one mode to another unless the proposal distribution is able to propose states far from the current state which generally will result in an unacceptably low acceptance rate. Therefore it is convenient to use a proposal distribution that is close to the target pdf. Li and Reynolds [56] proposed a two-level MCMC algorithm which has shown promising results for sampling a posterior pdf with multiple modes separated by low probability regions. In their method they first find many local minimums of the objective function using a gradient-based algorithm and then build a Gaussian mixture model (GMM) around the modes they find. This GMM serves as the proposal distribution in the second step of their two-level MCMC algorithm. An important advantage of this method is that it is easy to jump from mode to another which leads to a well-mixed chain that does not requires a large number of iterations to characterize the target pdf. The main drawback of their method is the need for an adjoint solution to compute the gradients in the first step where a gradient-based optimization method is used to find multiple local minima starting from different initial guesses. Since the adjoint solution is not typically available in commercial reservoir simulators, we would

like to find multiple modes of the posterior pdf in the first step of the two-level MCMC with a method which does not require an adjoint formulation. To accomplish this purpose, we need a method which is able to find multiple local minima of a multimodal posterior pdf and also provides the covariance matrices for each local minimum. Ensemble-based methods such as EnKF, ES, or their variations, including ES-MDA, are the most popular methods for assisted history-matching which do not need the adjoint formulation and are easy to implement. However, these methods only use the first and second moments of the ensemble to approximate the posterior which makes it suboptimal for highly non-Gaussian pdfs e.g., multimodal pdfs arising from nonlinear forward model operators. Furthermore, following Reynolds et al. [75] we showed in Chapter 2 that the ES update equation is similar to taking a full-step Gauss-Newton iteration using the same average sensitivity matrix to update all ensemble members. Therefore, a single application of ES, EnKF, or ES-MDA to approximate a multimodal pdf will not generally result in finding multiple modes of the pdf. There have been efforts to extend the ensemble based method to handle non-Gaussian problems [5, 18, 19, 83, 86]. Bengtsson et al. [5] introduced a local-local ensemble filter which combines the EnKF with a GMM to handle non-Gaussian problems. The ensemble members are partitioned into clusters and the localized sample covariance obtained for each cluster is used to update the members in each cluster. They found that this method does not provide smooth states, therefore they introduced a hybrid filter where the standard EnKF updates are used to ensure that the updated states from the local-local ensemble filter are spatially smooth. Smith [83] extended the EnKF by performing an expectation-maximization step to estimate the GMM parameters for the ensemble of forecast states. The Kalman gain for each component of the GMM is calculated and each ensemble member is updated with all the Kalman gains resulting in an increased number of state vectors ($N_e \times N_c$, where N_c is the number of GMM components). At the end, the analysis states are remapped to obtain N_e equally weighted analysis states. This step is done by drawing random vectors from a single Gaussian which approximates the GMM the best. A similar method was proposed by Dovera and Della Rossa [18] where they assumed a multimodal prior model approximated

by a GMM and modified EnKF to sample from a GMM posterior. Elsheikh et al. [19] developed a stochastic ensemble method with an update equation similar to EnKF where the gradient is estimated stochastically using an ensemble of directional derivatives. To explore different modes of the posterior, they augmented their method with k-mean clustering algorithm. Gao et al. [32] proposed an assisted history matching algorithm to find multiple local MAP estimates in parallel. They called the method distributed Gauss-Newton (DGN). Although DGN is not an ensemble based method, the sensitivity matrix for Gauss-Newton is found using a simplex gradient based on a sub-ensemble of models. Since the method does require availability of the adjoint solution and is able to find multiple modes concurrently, it seems like a good candidate for the first step of the two-stage MCMC algorithm. We will describe DGN of Gao et al. [32] in details in the next section and discuss its advantages and disadvantages.

4.1 Distributed Gauss-Newton (DGN)

DGN starts with N_e unconditional realizations of m generated from the prior pdf as the base ensemble or the N_e base-cases. At each iteration of DGN, the forward model is run for the N_e (current) models to obtain the predicted data. Similar to the ensemble-based methods, this step can be parallelized to improve the computational performance as it includes running the forward model for N_e independent models. Then, for each member of the base ensemble, the the sensitivity matrix is estimated using a simplex gradient approximation based on the closest models in the training set to that member of the base ensemble which is a great advantage in the sense that the method does not require the availability of the adjoint solution and the information for one base-case can be used for estimation of the sensitivity matrix for other base-cases. The training set includes the models in the current and the previous iterations and it will be updated dynamically during iterations. The initial training set includes the N_e base-cases and it can be larger than the number of base-cases to improve the approximation of the sensitivity matrix for large scale problems. For each base-case a search direction is obtained using a trust region method which gives a new model.

If the new model improves the objective function, the base-case will be updated and trust region radius will be updated. If the new model does not improve the objective function, the base-case will not be updated and the trust region radius will be reduced. In any case, the new model obtained at the end of each trust region iteration and its simulation results are added to the training set provided that its distance to any point in the training set is greater than a minimum distance, $d_{\min}$.

Chen et al. [11] utilized the DGN algorithm in an RML framework and proposed a procedure to find the global minimizer of the objective function. Gao et al. [33] proposed to use DGN to find multiple distinct MAP estimates and the approximate the posterior pdf with a GMM where each MAP estimate is used as the mean of one Gaussian of a GMM with the corresponding covariance matrix of the Gaussian equal to the inverse of the Gauss-Newton Hessian. Gao et al. [33] provided a method to estimate the weights of the GMM and claimed that samples from this GMM can be used for uncertainty quantification without the need for expensive MCMC methods which we will discuss in details later in this chapter. However, we show that this last claim is generally invalid.

4.1.1 Approximation of the sensitivity matrix

As discussed in Chapter 2, the sensitivity matrix is an $N_d \times N_m$ matrix with the the entry in the ith row and the jth column given by the partial derivative of the ith data with respect to the jth model parameter. Although the most efficient and accurate way to compute the sensitivity matrix is with an adjoint formulation, the sensitivity matrix can be estimated using a simplex gradient approach. To calculate the sensitivity matrix for the m_j model, we find the Ns closest models and the corresponding vectors of forecast or predicted data in the training set to each base ensemble member, m_j, for $j = 1, 2, \cdots, N_e$. The closest models are chosen based on the Euclidean distance between the models in the training set and the jth base-case. Next we define two matrices $\Delta M^{(j)}$ and $\Delta D^{(j)}$, respectively, given by

$$\Delta M^{(j)} = \left[m_1^{(j)} - m_j, ..., m_N^{(j)} - m_j \right], \tag{4.1}$$

and

$$\Delta D^{(j)} = \left[d_1^{f,(j)} - d_j^f, ..., d_N^{f,(j)} - d_j^f \right], \tag{4.2}$$

where $d_\ell^{f,(j)} = d^f(m_\ell^{(j)})$ for $\ell = 1, 2, \cdots, N$ and $d_j^f = d^f(m_j)$. Note that $\Delta M^{(j)}$ is $N_m \times N$ and $\Delta D^{(j)}$ is $N_d \times N$. Below, we suppress the superscript "(j)" and simply use ΔM and ΔD in place of $\Delta M^{(j)}$ and $\Delta D^{(j)}$ respectively. The sensitivity matrix evaluated at m_j is approximated by solving

$$\Delta M^T G^T = \Delta D^T. \tag{4.3}$$

In the original DGN of Gao et al. [32], it is proposed to choose N equal to N_m which makes ΔM square. This choice requires that the number of models in the initial training set is much larger than N_m, and the methodology is computationally too expensive for case where the number of model parameters, N_m, is larger than few hundred. Also, the N (which is equal to N_m) points from the training set should be selected such that ΔM is full-rank and thus invertible. However, finding N_m linearly independent models which result in a full-rank matrix has its own implementation difficulties. Here, when N_m is large, we use a value of N that is smaller than N_m and use the pseudo-inverse of ΔM to estimate the sensitivity matrix. Assume that the singular value decomposition of ΔM is given by

$$\Delta M = U \Lambda V^T, \tag{4.4}$$

where U is $N_m \times N_m$ matrix of the left singular vectors, V is $N \times N$ matrix of right singular vectors and Λ is diagonal matrix of the singular values of ΔM. Then, the pseudo-inverse of ΔM which is denoted by ΔM^+ is defined as

$$\Delta M^+ = V_p \Lambda_p^+ U_p^T. \tag{4.5}$$

In Eq. 4.5, p is the number of singular values retained based on the truncation level, V_p and U_p are used to denote the first p columns of V and U and the matrix Λ_p^+ is diagonal and is

defined as

$$\Lambda_p^+ = \text{diag}(\lambda_1^{-1}, \lambda_2^{-1}, \cdots, \lambda_p^{-1}), \tag{4.6}$$

with decreasing eigenvalues of ΔM, i.e., $\lambda_i \geq \lambda_{i+1}$. Note that similar to the discussion of Chapter 3 about the choice of the truncation level, p, one can use

$$\sum_{j=1}^{p} \lambda_j \geq \widetilde{a} \sum_{j=1}^{N} \lambda_j, \tag{4.7}$$

to determine p. Alternatively, we can choose p such that

$$\frac{\lambda_p}{\lambda_1} < \hat{a}, \tag{4.8}$$

is satisfied, where $\hat{a}$ is a number less than one. A larger value of $\hat{a}$ will result in a smaller p which implies more truncation. An advantage of choosing the truncation level based on Eq. 4.8 over the use of Eq. 4.7 is that it does not require the computation of all singular values and singular vectors. Therefore, a truncated singular value decomposition (TSVD) based on Lanczos algorithm [37, 66, 95] can be used to calculate the TSVD and the associated approximation of the sensitivity matrix. Using the expression in Eq. 4.5 for the pseudo-inverse of the ΔM in Eq. 4.3, the approximate sensitivity matrix is obtained as

$$G = \Delta D(U_p \Lambda_p^+ V_p^T). \tag{4.9}$$

Note that in the original DGN the matrix ΔM is assumed square and insertable; therefore there is no need for the calculation of its pseudo-inverse. However, finding an invertible matrix from the training set is not an straight-forward task especially if the number of parameters is large. Using the sensitivity matrix calculated for the base-case m_j, the gradient of the objective function and the Hessian matrix can be calculated based on

$$g = \hat{m}_j + G_D^T \hat{d}^j, \tag{4.10}$$

and

$$\hat{H} = I_{N_m} + G_D^T G_D, \tag{4.11}$$

where similar to Eqs. 2.22, 2.23 and 2.29, we have

$$\hat{m}_j = C_M^{-1/2}(m_j - m_{\text{pr}}), \tag{4.12}$$

$$\hat{d}_j = C_D^{-1/2}\left(d_j^f - d_{\text{obs}}\right), \tag{4.13}$$

and

$$G_D = C_D^{-1/2} G C_M^{1/2}. \tag{4.14}$$

Once we have the gradient vector and the Hessian matrix, an update for the jth base-case can be obtained using the trust region procedure discussed in Chapter 2. A detailed description of the DGN algorithm is given in Algorithm 4.1. Note that the algorithm converges if a specified number of models are converged. For the toy problems we require the convergence of all models while for the large problems we require the convergence of 95 percent of models.

4.1.2 Approximation of GMM weights

Recall that assuming a Gaussian prior pdf with mean m_{pr} and covariance matrix C_M and Gaussian measurement errors, the posterior pdf can be written as

$$\pi(m) = a \exp\{-O(m)\}, \tag{4.15}$$

where

$$O(m) = \frac{1}{2}\left(m - m_{\text{pr}}\right)^T C_M^{-1}\left(m - m_{\text{pr}}\right) + \frac{1}{2}\left(g(m) - d_{\text{obs}}\right)^T C_D^{-1}\left(g(m) - d_{\text{obs}}\right). \tag{4.16}$$

Suppose that using the DGN algorithm we have found N_c distinct minimizers of the objective

Algorithm 4.1: Pseudo-code for distributed Gauss-Newton (DGN) method

☐ Set N_e, N, $d_{\min}$, $i = 0$, Δ_j^0 for $j = 1, 2, .., N_e$, and $S = \{1, 2, .., N_e\}$.

☐ Generate initial ensemble denoted by $\{m_j^0\}_{j=1}^{N_e}$.

☐ Run the forward model for the ensemble to obtain $\{d_j^0\}_{j=1}^{N_e}$.

☐ Initialize the training set.

☐ While (not converged)

- For $j \in S$.
  - Find the closest points in the training set to m_j^i, to calculate Δm and Δd.
  - Estimate the sensitivity matrix, G, from Eq. 4.9.
  - Calculate $\hat{m}_j^i = C_M^{-1/2}(m_j^i - m_{\text{pr}})$, and $\hat{d}_j^i = C_D^{-1/2}\left(g(m_j^i) - d_{\text{obs}}\right)$.
  - Calculate $g_i = \hat{m}_j^i + G_D^T \hat{d}_j^i$ and $\hat{H}_i = I_{N_m} + G_D^T G_D$, where $G_D = C_D^{-1/2} G C_M^{1/2}$.
  - Solve $s_j^{i+1} = \arg\min_s q_i(s)$ subject to $\|s\| \leq \Delta_j^i$, where $q_i(s) = g_i^T s + \frac{1}{2} s^T \hat{H}_i s$.
  - Set $\delta m_j^{i+1} = C_M^{1/2} s_j^{i+1}$.
- EndFor
- For $j \in S$.
  - Update the model parameters $m_j^{i+1} = m_j^i + \delta m_j^{i+1}$.
  - Run the forward model for m_j^{i+1}.
  - Add m_j^{i+1} to the training set, if its distance to all members of the training set is greater than $d_{\min}$.
  - Based on the improvement in the objective function, accept or reject m_j^{i+1} and update the trust-region radius, Δ_j^{i+1} (see Algorithm 2.3).
  - If model j is converged, delete j from set S.
- EndFor
- Set $i = i + 1$.

☐ EndWhile

function given by Eq. 4.16. Then we can write Taylor's series approximation of the objective function in the neighbourhood of the ℓth mode, m_ℓ^*, as

$$O(m) = O(m_\ell^*) + (m - m_\ell^*)\nabla O(m_\ell^*) + \frac{1}{2}(m - m_\ell^*)^T C_\ell^{-1}(m - m_\ell^*) \tag{4.17}$$

where C_ℓ is the covariance matrix at the ℓth mode which is equal to the inverse of the Hessian matrix at the ℓth mode. Assuming that ∇O is equal to zero at the modes, we can rewrite

Eq. 4.17 as

$$\exp(-O(m)) = \exp\left(-O(m_\ell^*)\right) \exp\left(-\frac{1}{2}(m - m_\ell^*)^T C_\ell^{-1}(m - m_\ell^*)\right) \tag{4.18}$$

We let

$$\mathcal{G}_\ell(m) = \exp\left(-\frac{1}{2}(m - m_\ell^*)^T C_\ell^{-1}(m - m_\ell^*)\right), \quad \text{for } \ell = 1, ..., N_c \tag{4.19}$$

denote the ℓth component of the GMM which has a mean of m_ℓ^* and covariance matrix, C_ℓ, equal to the inverse of the Gauss-Newton Hessian. The posterior pdf can be approximated using the GMM as

$$a \exp(-O(m)) = \sum_{\ell=1}^{N_c} w_\ell c_\ell \mathcal{G}_\ell(m) \tag{4.20}$$

where a is an unknown normalizing constant, w_ℓ is the weight of the ℓth Gaussian component of the Gaussian mixture model which are also unknown, and c_ℓ is the normalizing constant for the ℓth Gaussian given by

$$c_\ell = \frac{1}{\sqrt{(2\pi)^{N_m}|C_\ell|}}. \tag{4.21}$$

If we assume that for any $\widetilde{m}$ that is a sample of the ℓth Gaussian, $\mathcal{G}_\ell(m)$, the values of the other Gaussians evaluated at $\widetilde{m}$ are negligible compared to $\mathcal{G}_\ell(\widetilde{m})$, i.e., $G_i(\widetilde{m}) << G_\ell(\widetilde{m})$ for $i \neq \ell$, then using Eqs. 4.18, 4.19 and 4.20, we can write

$$\begin{aligned} w_\ell c_\ell \exp\left(-\frac{1}{2}(m - m_\ell^*)^T C_\ell^{-1}(m - m_\ell^*)\right) = \\ a \exp\left(-O(m_\ell^*)\right) \exp\left(-\frac{1}{2}(m - m_\ell^*)^T C_\ell^{-1}(m - m_\ell^*)\right), \end{aligned} \tag{4.22}$$

which can be simplified to

$$w_\ell c_\ell = a \exp\left(-O(m_\ell^*)\right), \quad \text{for } \ell = 1, ..., N_c. \tag{4.23}$$

Note that we have $N_c + 1$ unknowns including N_c values of w_ℓ and the normalizing constant

a. However, we have the following constraint as the last equation:

$$\sum_{\ell=1}^{N_c} w_\ell = 1. \tag{4.24}$$

Eq. 4.24 is a necessary condition for the right-hand side of Eq. 4.20 to be pdf. To solve for a, we can find the expression for w_ℓ from Eq. 4.23 and use that expression in Eq. 4.24, to obtain

$$a = \frac{1}{\sum_{\ell=1}^{N_c} \left[\exp\left(-O(m_\ell^*)\right)/c_\ell\right]}. \tag{4.25}$$

Once, we a is found, Eq. 4.23, can be used to find w_ℓ's which are given by

$$w_\ell = a \exp\left(-O(m_\ell^*)\right)/c_\ell, \quad \text{for } \ \ell = 1, ..., N_c. \tag{4.26}$$

We refer to this procedure for determining the weights as method 1. Gao et al. [33] proposed to use the information from the models in the training set close to the MAP estimates (modes of the posterior pdf) to improve the estimation of GMM weights. To achieve that, suppose that we have N_{con} well-conditioned models close to the N_c minimizers where the well-conditioned models are defined as the models, m_j, that satisfy two conditions: $(O(m_j) - O(m_\ell^*))/O(m_\ell^*) < \epsilon_1$ and $(m_j - m_\ell^*)^T C_\ell^{-1}(m_j - m_\ell^*) < \epsilon_2$, where $\epsilon_1 > 0$ and $\epsilon_2 > 0$ are both user specified parameters. Note that Gao et al. [33] did not provide any guidelines for choosing ϵ_1 and ϵ_2. They proposed to define a N_{con}-dimensional vector p^{con} where its jth element is

$$p_j^{con} = \exp\left(-O(m_j) + O_{\min}\right), \tag{4.27}$$

where m_j is the jth well-conditioned model and $O_{\min}$ is minimum objective function value for all N_{con} points. By adding $O_{\min}$ to $-O(m_j)$, the maximum value of the elements of p^{con} will be equal to one. Assuming that $O_{\min} = O(m_k)$, then the kth element of p^{con} is equal to one, i.e., $p_k^{con} = 1$. Also they defined a N_c-dimensional vector w_G with w_i as its ith element

and introduced an $N_{con} \times N_c$ matrix A^{con} whose elements are given by

$$a^{con}_{(\ell,j)} = c_\ell \exp\left(-\frac{1}{2}(m_j - m^*_\ell)^T C^{-1}_\ell (m_j - m^*_\ell)\right), \tag{4.28}$$

for $j = 1, 2, \cdots, N_{con}$ and $\ell = 1, 2, \cdots, N_c$. A distance function is then defined by Gao et al. [33] to measure the distance between the GMM approximation and the actual posterior pdf as

$$d(w_G) = [A^{con} w_G - p^{con}]^T [A^{con} w_G - p^{con}]. \tag{4.29}$$

Note $A^{con} w_G$ is a vector of size N_{con} and its jth element is the probability of m_j based on the GMM. The weights of the GMM are chosen as the minimizer of $d(w_G)$ which is given by

$$w_G = \left((A^{con})^T A^{con}\right)^{-1} \left((A^{con})^T p^{con}\right). \tag{4.30}$$

Since there is no guarantee that $(A^{con})^T A^{con}$ is nonsingular, we can use singular value decomposition of A^{con} to find the pseudo-inverse of $(A^{con})^T A^{con}$. Note that since w_G is the solution of an unconstrained minimization problem, some of the weights (components of w_G) may be negative and if a component of w_G of Eq. 4.30 is negative, we replace it by zero. Finally, we need to normalize the remaining positive weights so that their summation is equal to one. We refer to this procedure for approximating the weights of the GMM as method 2. Note that in both methods, if the value of the objective function for different modes differ significantly one local minimum may get a weight equal to one; see field example in Gao et al. [33] where 890 local MAPs are obtained at the end of DGN but one of them got a weight of 0.9999. It is important to emphasis that Gao et al. [33] proposed building a Gaussian mixture model using the modes obtained at the end of DGN approach and assumed that the posterior pdf is well approximated by this GMM. Thus they assumed that random samples drawn from the GMM can be used for uncertainty quantification., i.e., they assume that sampling the GMM constructed is equivalent to sampling the target pdf, $\pi(m)$ given in Eq. 4.15.

4.1.3 DGN for toy problems

In this section we test the capabilities of DGN and approximation of the weights as described above for the uncertainty quantification proposed in [33] for two toy problems. In both toy problems, since we only have one model parameter, the sensitivity at a given model is calculated based on a simple finite difference using the closest point in the training set to that model.

Toy problem 1: Gao et al. [33] designed a scalar toy problem to test their uncertainty quantification method using DGN algorithm. The relationship between the model parameter and the observations is nonlinear. For a given model, m, the predicted data are given

$$d_k(m) = k + 10(m - 1.5)\left[m - (2 + 0.04k)\right]\left\{m - [2.5 + 0.015(3 - k)]\right\}, \tag{4.31}$$

for $k = 1, 2, 3$, where $d_k(m)$ denotes the kth element of $d(m)$. To obtain the observations, the data predicted with $m_{\text{true}} = 1.5$ are perturbed by adding random Gaussian noise with mean of zero and standard deviation of 0.05. The prior distribution for m is also assumed Gaussian with a mean of 2.1 and a variance of 0.2. In this case, the objective function would be

$$O(m) = \frac{1}{2}\left[\frac{(m - 2.1)^2}{0.2} + (d(m) - d_{\text{obs}})^T C_D^{-1}(d(m) - d_{\text{obs}})\right), \tag{4.32}$$

where C_D is a diagonal matrix and is equal to the identity matrix multiplied by 0.05^2. The posterior pdf we wish to sample is still $\pi(m) = f(m|d_{\text{obs}}) = a\exp[-O(m)]$. Due to the nonlinearity of the forward model, the objective function of Eq. 4.32 has three local minima which means that the posterior pdf for this toy problem has three local maximums, i.e., three modes. The DGN algorithm is run to minimize the objective function given by Eq. 4.32 starting from 20 unconditional realizations drawn randomly from the prior distribution. Figure 4.1(a) shows the number of base-cases still running at each iteration of DGN. The results of Fig. 4.1(a) indicates that the DGN algorithm converges in 8 iterations and that three of the minimization processes converge in three iterations. From Fig. 4.1(b), we observe

that 17 base-cases have converged to 2.08, 2 base-cases have converged to 2.51 and only one base-case has converged to 1.50 which is the value of the true model parameter. Note that since the prior mean of 2.1 is close to the middle mode ($m = 2.08$), the majority of the initial base-case which are generated using the prior distribution converged to the middle mode. However, if we generated the initial base-cases uniformly the converged models would be distributed more evenly (as it is the case in [33]).

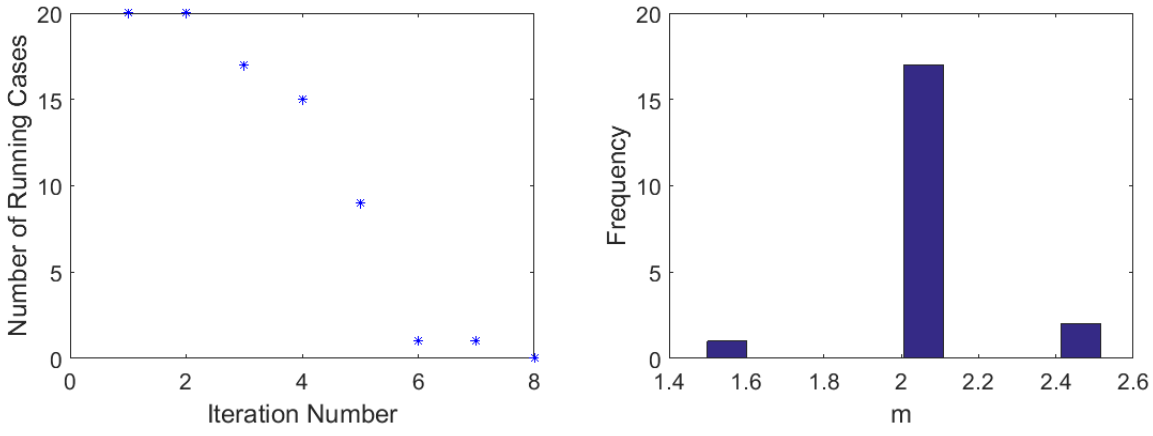
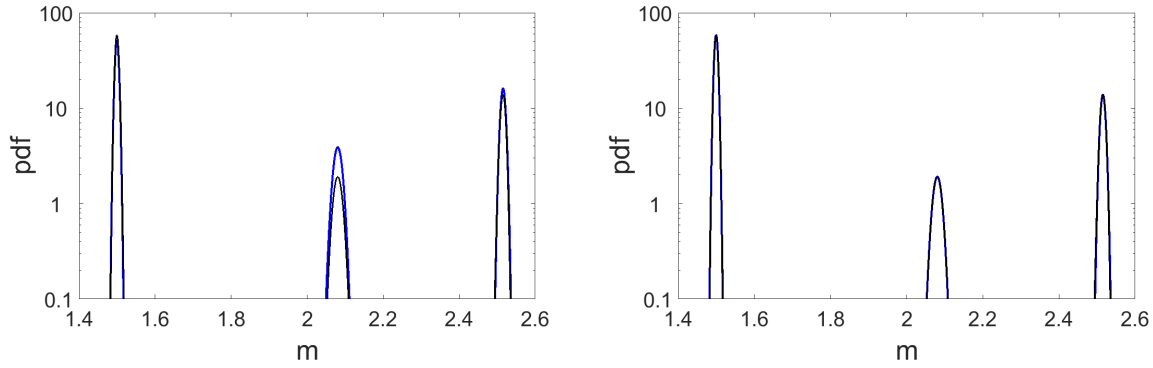


Figure 4.1: The results of DGN algorithm for toy problem 1 (a) the number of running base-cases (b) the histogram of the converged models.

From the modes generated using DGN algorithm, we can build the Gaussian mixture model that approximates the target pdf. Note that each Gaussian component is centered at a mode with a covariance matrix which is equal to the inverse of the Hessian at that mode. Figure 4.2 shows the comparison of the GMM distribution with the target distribution for two sets of weights. In Fig. 4.2(a) the weights are obtained using method 1 where the normalizing constant, a, is calculated using Eq. 4.25 and then the weights are determined using Eq. 4.23. In this case, the calculated weights for the modes from left to right are 0.624, 0.112, and 0.264 and it is clear that there exist some discrepancy between the target pdf and the GMM distribution using these weights. In Fig. 4.2(b) we used method 2 to determine the weights where the weights are obtained based on Eq. 4.30 which gives the minimizer of Eq. 4.29. This procedure results in weights of 0.718, 0.055, and 0.227 for the modes from left to right. Similar to the results presented in [33] the GMM distribution with this set of

weights and the target pdf are overlapping. It is important to point out that the modes in this toy problem are well separated and the target pdf appears to have three Gaussian peaks and therefore the weights can be tuned such that the target pdf is approximated with the GMM perfectly.



(a) (b)

Figure 4.2: Gaussian mixture model compared to the true posterior pdf for toy problem 1 (a) weights obtained using method 1 (b) weights obtained using method 2. The black curve is the true pdf and the blue curve represents the GMM.

Toy problem 2: The second example is also a nonlinear problem with one model parameter and one datum. This example is similar to the toy problem designed in [97]. The forward model, $g(m)$, is defined as

$$d = g(m) = 1 - 4.5(m - 2\pi/3)^2, \tag{4.33}$$

where m is the model parameter which is a scaler with the true value of $m_{\text{true}} = 1.8836$. A Gaussian prior distribution with the mean equal to 2.3 and the standard deviation of 0.2 is considered. The data from the true model is perturbed by adding random Gaussian noise with a mean of zero and standard deviation of 0.1 to obtain the observed data, d_{obs}. Because the prior distribution and the measurement errors are Gaussian, the posterior pdf is given by Eq. 4.15, where $O(m)$ is defined as

$$O(m) = \frac{1}{2}\left[\left(\frac{m - 2.3}{0.2}\right)^2 + \left(\frac{d_{\text{obs}} - g(m)}{0.1}\right)^2\right]. \tag{4.34}$$

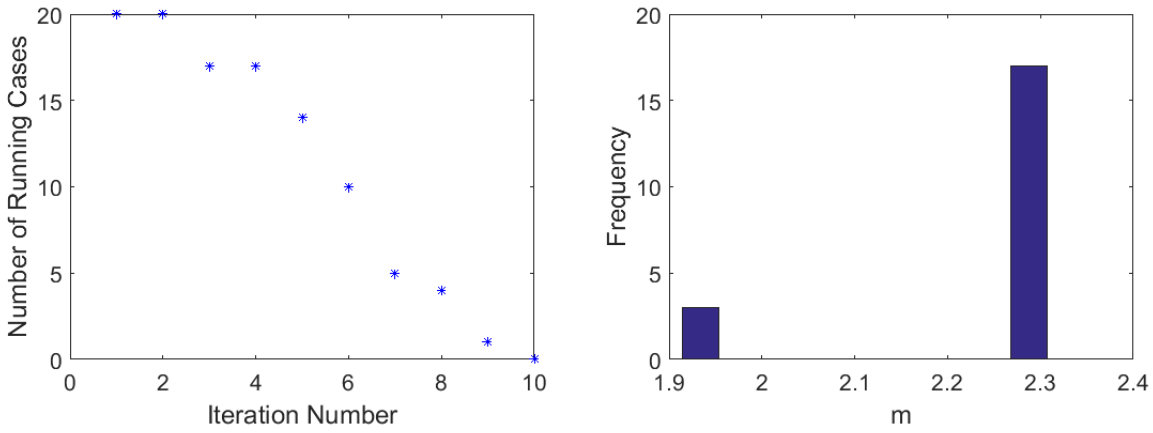


Figure 4.3: The results of DGN algorithm for toy problem 2 (a) the number of running base-cases (b) the histogram of the converged models.

The DGN algorithm is run to minimize the objective function given by Eq. 4.34 with 20 initial base-cases drawn from the prior distribution. Figure 4.3(a) shows the number of base-cases running at each iteration of DGN and as we can see the algorithm converged in 10 iterations. From Fig. 4.1(b) we observe that two distinct modes are found at the end of the DGN algorithm which are located at 1.9140 and 2.3077. Note that the true value of the model parameter is 1.8836, however it turns out that the value of the objective function at $m = 1.9140$ is less than the value of the objective function at $m_{\text{true}} = 1.8836$ because of contribution of the model mismatch term in the objective function. Fig. 4.3(b) also shows that 17 base-cases have converged to 2.3077 and the remaining three base-cases have converged to 1.9140 which, similar to the first toy problem, is due to the fact that the prior mean is close to the mode at 2.3077 and we only generated 20 initial base-cases.

We build the Gaussian mixture model with two components centered around the modes we found using DGN. The weights are determined using both method 1 and method 2 and the resulting GMM distributions are plotted in Figure 4.4. The weights obtained using method 1 for the modes located at 1.9140 and 2.3077 are respectively 0.151 and 0.849. Method 2 results in weights equal to 0.132 and 0.868 which are only slightly different from those obtained with method 1. From Fig. 4.4 we observe that although we have found the modes of the posterior accurately the Gaussian mixture model distributions do not

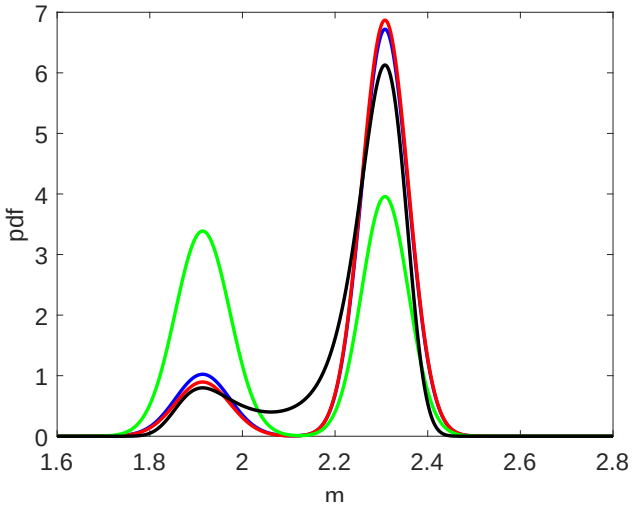


Figure 4.4: Gaussian mixture model with different weights compared to the true posterior pdf for toy problem 2. The black curve is the true posterior pdf, the blue curve is GMM with weights obtained using method 1, the red curve is the GMM with weights obtained using method 2, and the green curve is the GMM with equal weights.

approximate the target pdf well especially in the area between the two modes. Note that the weights are obtained based on the assumption that the modes of the posterior pdf are well separated and thus they only give a good estimate of the posterior pdf if the modes of the pdf are not connected. This example shows that in order to obtain the correct characterization of the posterior pdf, we generally cannot assume that the GMM gives a good representation of the posterior pdf. However, following the idea of Li and Reynolds [56] we can use the GMM constructed with DGN as the proposal distribution in a Markov chain Monte Carlo (MCMC) algorithm to obtain a more efficient MCMC implementation.

4.2 An Efficient Two-Level MCMC Algorithm

A nonlinear relationship between the model parameters and the observations can lead to a multimodal posterior distribution. Uncertainty quantification of the posterior distribution for such problems using a Metropolis-Hastings MCMC method requires a carefully designed proposal distribution. An essential key to improve the computational efficiency of MCMC method based on the Metropolis-Hastings acceptance criteria is to have a proposal distribution which is close to the target distribution. Following the idea of Li and Reynolds [56], we would like to build a Gaussian mixture model to serve as the proposal distribution

for MCMC. Therefore, the first step of our efficient two-level MCMC algorithm is to use DGN in order to find multiple modes of the posterior distribution. An advantage of our two-level MCMC compared to the work of Li and Reynolds [56] is that in our first step we do not require the analytical computation of the gradient of the objective function. At the end of the first step, we build a GMM where the components of GMM are Gaussians centered at the modes of the posterior pdf that we find in the first step. The second step of the two-level MCMC method is to use this GMM as the proposal distribution for MCMC. This step is identical to the second step of the two-level MCMC proposed in Li and Reynolds [56]. For large scale problems it is possible to find a large number modes and many of these modes are close to each other [71], therefore it is reasonable to reduce the number of modes before building the GMM. In order to reduce the number of modes for large scale problems, we first keep only the modes which give a value of the normalized objective function (objective function divided by the number of data) less than a threshold and then cluster the modes that satisfy this condition into a user defined number of clusters using the k-medoids clustering algorithm [50]. The final modes are m_ℓ^*, for $\ell = 1, 2, \cdots, N_c$ where N_c is the number of modes after clustering. Letting C_ℓ denote the inverse Hessian evaluated at m_ℓ^*, the GMM distribution is given by

$$p(m) = \sum_{\ell=1}^{N_c} w_\ell \mathcal{N}\left(m_\ell^*, C_\ell\right), \tag{4.35}$$

where $\mathcal{N}\left(m_\ell^*, C_\ell\right)$ denotes a Gaussian distribution with mean of m_ℓ^* and the covariance of C_ℓ and w_ℓ is called the weight of ℓth Gaussian component. The weights w_ℓ can be determined by method 1 or 2 discussed earlier in DGN section or assigned by any other procedure but we must have $\sum_{\ell=1}^{N_c} w_\ell = 1$ in order for $p(m)$ to be a pdf. The pdf $p(m)$ is used as the proposal distribution. The probability of proposing a transition from the state m_i to m_j, which is required in Metropolis-Hastings method, is then defined by

$$q(m_i, m_j) = p(m_j) = \sum_{\ell=1}^{N_c} w_\ell c_\ell \exp\left(-\frac{1}{2}(m_j - m_\ell^*)^T C_\ell^{-1}(m_j - m_\ell^*)\right). \tag{4.36}$$

From Eq. 4.36, it is clear that the probability of proposing a new state is independent of

the current state in the chain, therefore our two-level MCMC algorithm can be classified as an independent Metropolis sampler [91]. This feature of the proposal distribution is a great advantage for sampling multimodal target distributions and tends to avoid being trapped in one mode for a very large number of states. Independent samplers promote good mixing of the chain, i.e., it is easy to move between the modes of the posterior or more generally move around the sample space.

4.2.1 Covariance matrices

As mentioned earlier, the covariance matrix at ℓth mode, C_ℓ, is set equal to the inverse of the Gauss-Newton Hessian matrix which the is given by

$$H_\ell = C_M^{-1} + G_\ell^T C_D^{-1} G_\ell, \tag{4.37}$$

where G_ℓ is calculated based on Eq. 4.9 using N points from the training set which are closet to m_ℓ^*. However, since we use an approximate estimate of G_ℓ to find the Hessian matrix, using H_ℓ^{-1} as the covariance matrix in the proposal distribution results in a large number of rejections as most of the proposed states have high values of the objective function. Therefore, following the suggestion of Li and Reynolds [56] we assume that the covariance matrix at the ℓth mode is given by

$$C_\ell = \beta H_\ell^{-1}, \tag{4.38}$$

where $0 < \beta \leq 1$ is a scaler which is used to rescale the covariance matrices to make the two-level MCMC more efficient. Note that rescaling the Gauss-Newton covariance matrix is not required for toy problems but it is crucial to do so as the dimension of the problem increases. In our examples, β is chosen such that if we generate 200 realizations from the GMM, at least 20 of those give a normalized objective function less than 5.

4.2.2 Covariance adaptation

Following the discussion of Tierney [91] on independent Metropolis chains, we can

rewrite the Metropolis-Hastings acceptance probability of the candidate state m_j as

$$\alpha(m_i, m_j) = \min\left\{1, \frac{w(m_j)}{w(m_i)}\right\}, \tag{4.39}$$

where $w(m_j) = \pi(m_j)/p(m_j)$ is called the importance weight function in importance sampling process [44]. Note that $p(m_j) = q(m_i, m_j)$ is the probability of proposing m_j given that we are at state m_i. With this notion, a close relationship between the independent Metropolis chains and the importance sampling process can be observed [91]. According to Eq. 4.39 we can see that candidate states with low importance weights are rarely accepted. On the contrary, candidate states associated with high importance weights are usually accepted and will remain in the chain for several steps to increase their weight in the chain; e.g., for calculating the mean of the chain a state which is repeated for a large number has more influence on the mean than states which are repeated a few times in the chain. However, it is possible that the chain get stuck at some points for a long number of iterations, if their importance weights are very high. In the context of importance sampling, it is recommended that we choose the proposal distribution such that it produces a weight function that is approximately constant. It is obvious that if the weight function is constant, all samples will be accepted based on the Metropolis-Hastings acceptance criteria. As it is not possible to find the proposal distribution that results in almost constant weight function, we can alternatively improve the proposal distribution as we proceed in the chain using a covariance adaptation scheme. Haario et al. [40] proposed to use the samples in the chain to improve the estimate of the mean and the covariance of the proposal distribution. Assuming that the initial proposal distribution is a Gaussian with mean of $\bar{m}$ and covariance of C (i.e., $\mathcal{N}(\bar{m}, C)$), the adaptive Metropolis algorithm of Haario et al. [40] can be summarized as

1. Initialize $i = 0$, and $m_i = m_0$, where i is the proposal index.

2. Draw a new model, m_j, from the proposal distribution $\mathcal{N}(\bar{m}, C)$.

3. Run the forward model for m_j and evaluate α based on Eq. 4.39.

4. Generate a random number, u, from the uniform distribution in $[0, 1]$.

5. If $u \leq \alpha$, set $m_{i+1} = m_j$, otherwise set $m_{i+1} = m_i$.

6. Update the mean and the covariance

$$\bar{m} = \bar{m} + \gamma_{i+1}(m_{i+1} - \bar{m}) \tag{4.40}$$

$$C = C + \gamma_{i+1}\left[(m_{i+1} - \bar{m})(m_{i+1} - \bar{m})^T - C\right] \tag{4.41}$$

7. Set $i = i + 1$ and goto step 2.

In Eqs. 4.40 and 4.41 γ_{i+1} is called the learning rate or the gain factor. The sequence of γ_i should satisfy

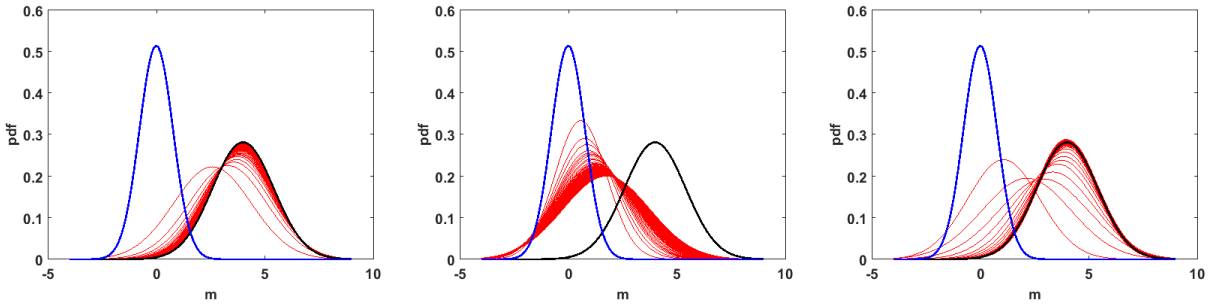
$$\sum_{i=1}^{\infty} \gamma_i = \infty, \tag{4.42}$$

and

$$\sum_{i=1}^{\infty} \gamma_i^{1+\delta} < \infty, \tag{4.43}$$

for some $\delta \in (0, 1]$ in order for the chain to converge or at least, no proof of convergence of the chain have been established without assuming that Eqs. 4.42 and 4.43and hold. The first condition ensures that any point of the target pdf can be reached eventually and the second condition ensures that noise introduced due to new samples is contained and does not interfere with the convergence of the chain [58]. Haario et al. [40] proposed to choose $\gamma_i = O(1/i)$ where i is the iteration index. However, Liang et al. [58, p. 306] pointed out that "the diminishing adaptation condition may not necessarily hold for adaptive independent Metropolis-Hastings algorithms," even though, adaptive independent Metropolis-Hastings samplers have been successfully applied for uncertainty quantification even though their adaptation is not diminishing [47]. In order to check the effect of the learning rate on adaptation of an independent MH algorithm, we designed a simple problem where the aim is to sample a Gaussian target pdf with mean of 4 and variance of 2 i.e., $\pi(m) = \mathcal{N}(4, 2)$. We assume that the proposal distribution has a mean of zero and variance of 0.6, i.e.,

$p(m) = \mathcal{N}(0, 0.6)$. Note that both mean and variance of the proposal distribution are smaller than those of the target pdf. We use the adaptive MCMC algorithm discussed above to sample the target pdf while the mean and the variance of the proposal distribution are updated with two options for the learning rate, γ_i. In the first two cases, we use the suggestion of Haario et al. [40] and set $\gamma_i = 1/i$ and $\gamma_i = 0.1/i$. In the third case, we use a constant value of 0.001 for γ_i. Figure 4.5 shows the change in the proposal distribution (initially blue curve) as we proceed in the chain. From Fig. 4.5(a) it is clear that the proposal moves toward the target pdf as we adapt both its mean and the variance and once it reaches to the target pdf it stays there. However, from Fig. 4.5(b) it is clear that if we use $\gamma_i = 0.1/i$, the adaptation scheme fails to converge to the target pdf. Fig. 4.5(c) uses a constant value for the learning rate which violates the diminishing condition of Eq. 4.43 but it is able to update the proposal distribution to approximately the target pdf and stay there.



(a) (b) (c)

Figure 4.5: Effect of adaptation learning rate for simple Gaussian target and proposal distributions (a) $\gamma_i = 1/i$ (b) $\gamma_i = 0.1/i$ (c)$\gamma_i = 0.001$. The black curve is the target pdf, the blue curve is the initial proposal pdf, and the red curves are the proposal pdfs plotted every 500 iterations.

The two-level MCMC proposed in this work requires covariance adaptation in order to improve its performance. However, unlike Haario et al. [40], at the ith iteration of the chain, we only update the covariance matrix of the ℓth Gaussian components of GMM using

$$C_\ell = (1 - \gamma_{i+1})C_\ell + \gamma_{i+1}\left[(m_{i+1} - \bar{m}_\ell)(m_{i+1} - \bar{m})^T\right], \tag{4.44}$$

which is equivalent to Eq. 4.41. In Eq. 4.44, m_{i+1} belongs to the ℓth Gaussian component and

we assume that the learning rate, γ_i, is constant and equal to μ/N_m^2 following the suggestion of Hansen [43]. The parameter μ is tuned such that we obtain an acceptance rate between 15 to 50 percent for large scale problems [76].

4.2.3 Evaluating the acceptance probability

In high dimensional problems where the number of model parameters is large, evaluating the Metropolis-Hastings acceptance probability is not always straight forward as both the numerator and denominator may be very close close to zero. Recall that the probability of accepting a proposed state, m_j, in the chain is given by

$$\alpha(m_i, m_j) = \min\left\{1, \frac{\pi(m_j)q(m_j, m_i)}{\pi(m_i)q(m_i, m_j)}\right\}, \tag{2.94}$$

where the current state in the chain is m_i. Using the expressions for $\pi(.)$ and $q(.,.)$, we can write

$$\begin{aligned}
&\frac{\pi(m_j)q(m_j, m_i)}{\pi(m_i)q(m_i, m_j)} \\
&\quad = \frac{\exp\left(-O(m_j)\right)\sum_{\ell=1}^{N_c}\frac{w_\ell}{\sqrt{(2\pi)^{N_m}|C_\ell|}}\exp\left(-\frac{1}{2}(m_i - m_\ell^*)^T C_\ell^{-1}(m_i - m_\ell^*)\right)}{\exp\left(-O(m_i)\right)\sum_{\ell=1}^{N_c}\frac{w_\ell}{\sqrt{(2\pi)^{N_m}|C_\ell|}}\exp\left(-\frac{1}{2}(m_j - m_\ell^*)^T C_\ell^{-1}(m_j - m_\ell^*)\right)}.
\end{aligned} \tag{4.45}$$

Note that the determinant of C_ℓ, which is denoted by $|C_\ell|$, for high dimensional problems can be zero when evaluated numerically i.e., all the eigenvalues of C_ℓ are nonzero but since most of them are less than one, the determinant which is the product of all the eigenvalues is numerically zero. If the determinant of the covariance matrix for one component of the GMM is zero or extremely close to zero, it causes problems for evaluating Eq. 4.45. To avoid determinants that are very close to zero, we multiply both the denominator and the

numerator of the fraction on the right-hand side of Eq. 4.45 by $\kappa^{N_m/2}$ to obtain

$$\begin{aligned}
&\frac{\pi(m_j)q(m_j,m_i)}{\pi(m_i)q(m_i,m_j)} \\
&\quad= \frac{\exp\left(-O(m_j)\right)\sum_{\ell=1}^{N_c}\frac{w_\ell}{\sqrt{(2\pi\kappa)^{N_m}|C_\ell|}}\exp\left(-\frac{1}{2}(m_i-m_\ell^*)^T C_\ell^{-1}(m_i-m_\ell^*)\right)}{\exp\left(-O(m_i)\right)\sum_{\ell=1}^{N_c}\frac{w_\ell}{\sqrt{(2\pi\kappa)^{N_m}|C_\ell|}}\exp\left(-\frac{1}{2}(m_j-m_\ell^*)^T C_\ell^{-1}(m_j-m_\ell^*)\right)} \\
&\quad= \frac{\exp\left(-O(m_j)\right)\sum_{\ell=1}^{N_c}\frac{w_\ell}{\sqrt{|2\pi\kappa C_\ell|}}\exp\left(-\frac{1}{2}(m_i-m_\ell^*)^T C_\ell^{-1}(m_i-m_\ell^*)\right)}{\exp\left(-O(m_i)\right)\sum_{\ell=1}^{N_c}\frac{w_\ell}{\sqrt{|2\pi\kappa C_\ell|}}\exp\left(-\frac{1}{2}(m_j-m_\ell^*)^T C_\ell^{-1}(m_j-m_\ell^*)\right)}.
\end{aligned} \tag{4.46}$$

Therefore instead of calculating the determinant of C_ℓ, we calculate the determinant of $2\pi\kappa C_\ell$. The value of $\kappa \geq 1$ is selected using an iterative process starting from $\kappa = 1$ and we increment the value of κ by 0.1 until the determinant of $2\pi\kappa C_\ell$ is greater than zero for $\ell = 1, 2, \cdots, N_c$. We can rewrite Eq. 4.46 as

$$\begin{aligned}
&\frac{\pi(m_j)q(m_j,m_i)}{\pi(m_i)q(m_i,m_j)} \\
&\quad= \frac{\exp\left(-O(m_j)\right)\sum_{\ell=1}^{N_c}\exp\left(\ln\left(\frac{w_\ell}{\sqrt{|2\pi\kappa C_\ell|}}\right)-\frac{1}{2}(m_i-m_\ell^*)^T C_\ell^{-1}(m_i-m_\ell^*)\right)}{\exp\left(-O(m_i)\right)\sum_{\ell=1}^{N_c}\exp\left(\ln\left(\frac{w_\ell}{\sqrt{|2\pi\kappa C_\ell|}}\right)-\frac{1}{2}(m_j-m_\ell^*)^T C_\ell^{-1}(m_j-m_\ell^*)\right)} \\
&\quad= \frac{\exp\left(-O(m_j)\right)\sum_{\ell=1}^{N_c}\exp\left(A_\ell\right)}{\exp\left(-O(m_i)\right)\sum_{\ell=1}^{N_c}\exp\left(B_\ell\right)},
\end{aligned} \tag{4.47}$$

where A_ℓ and B_ℓ are, respectively, define as

$$A_\ell = \ln\left(\frac{w_\ell}{\sqrt{|2\pi\kappa C_\ell|}}\right) - \frac{1}{2}(m_i-m_\ell^*)^T C_\ell^{-1}(m_i-m_\ell^*), \tag{4.48}$$

and

$$B_\ell = \ln\left(\frac{w_\ell}{\sqrt{|2\pi\kappa C_\ell|}}\right) - \frac{1}{2}(m_j-m_\ell^*)^T C_\ell^{-1}(m_j-m_\ell^*). \tag{4.49}$$

If we denote the maximum values of A_ℓ and B_ℓ over ℓ, respectively, with $A_{\max}$ and $B_{\max}$,

Eq. 4.47 can be written as

$$\frac{\pi(m_j)q(m_j,m_i)}{\pi(m_i)q(m_i,m_j)} = \frac{\exp\left(-O(m_j)\right)\exp\left(A_{\max}\right)\sum_{\ell=1}^{N_c}\exp\left(A_\ell - A_{\max}\right)}{\exp\left(-O(m_i)\right)\exp\left(B_{\max}\right)\sum_{\ell=1}^{N_c}\exp\left(B_\ell - B_{\max}\right)}$$
$$= \exp\left(\left[O(m_i) - O(m_j)\right] + \left[A_{\max} - B_{\max}\right]\right)\frac{\sum_{\ell=1}^{N_c}\exp\left(A_\ell - A_{\max}\right)}{\sum_{\ell=1}^{N_c}\exp\left(B_\ell - B_{\max}\right)} \quad (4.50)$$

4.3 Results and Discussion

In this section, our proposed two-level MCMC method is applied to two problems. The first case is the toy problem 2 which was discussed earlier in this chapter. We showed that finding the GMM will not provide the correct characterization of the posterior pdf for this problem and we need to perform sampling in order to capture the correct posterior distribution. The second example is a 1-dimensional reservoir model which has been designed by Emerick and Reynolds [24] to investigate the sampling performance of ensemble-based methods. Li and Reynolds [56] used this example in order to compare the performance of their two-level MCMC algorithm with other MCMC methods.

4.3.1 Revisiting toy problem 2

Earlier in this chapter, we discussed a procedure to find a GMM where the Gaussians in the GMM have the modes of the posterior pdf as their means and these modes are found by applying DGN to find the minimum of the objective function associated with the posterior pdf. We showed that DGN is able to find both modes of the posterior pdf for this scalar toy problem, but the GMM does not characterize the posterior pdf exactly especially in the low probability region between the modes which contributes significantly to the probability density despite its lower probability compared to the modes. Using the GMM distribution that we obtained in the first step, we run 5 parallel Markov chains starting from different initial states drawn randomly from the proposal distribution. The maximum number of iterations for all the five chains is set equal to 50 thousand. In this toy problem we set $\beta = 1$ in Eq. 4.38 to obtain the covariance for each mode. Figure 4.6 shows the change in the GMM proposal distribution for one of the chains as we adapt the covariances of the GMM

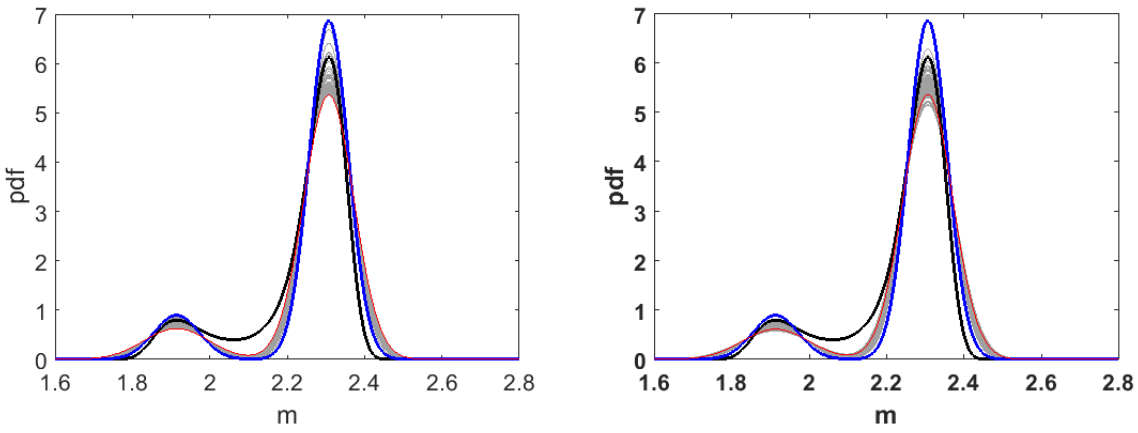
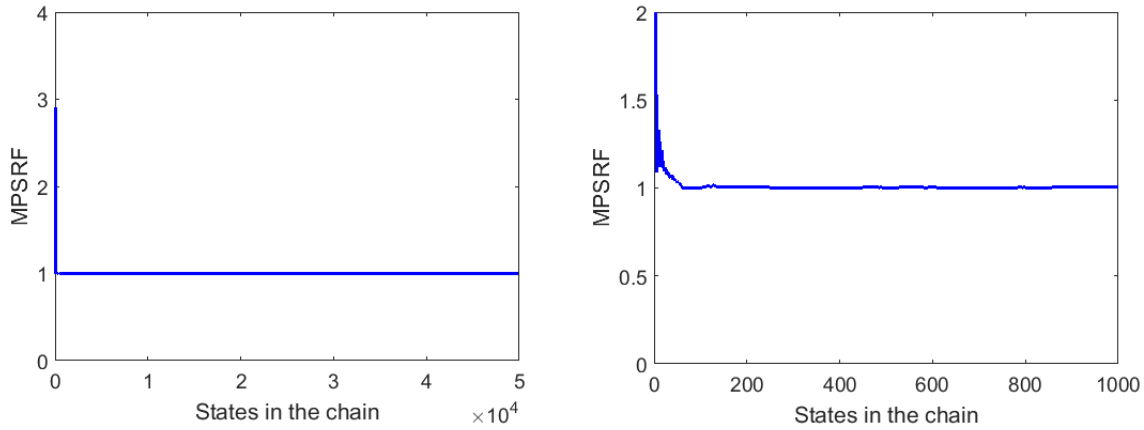


Figure 4.6: Change in the proposal distribution with adaptation for toy problem 2 with (a) $\gamma_i = 10^{-4}$ and (b) $\gamma_i = 10^{-3}$. The black curve is the true posterior pdf, the blue curve is the initial GMM, the red curve is the final GMM, and the grey curves are adapted GMM every 1000 iteration.

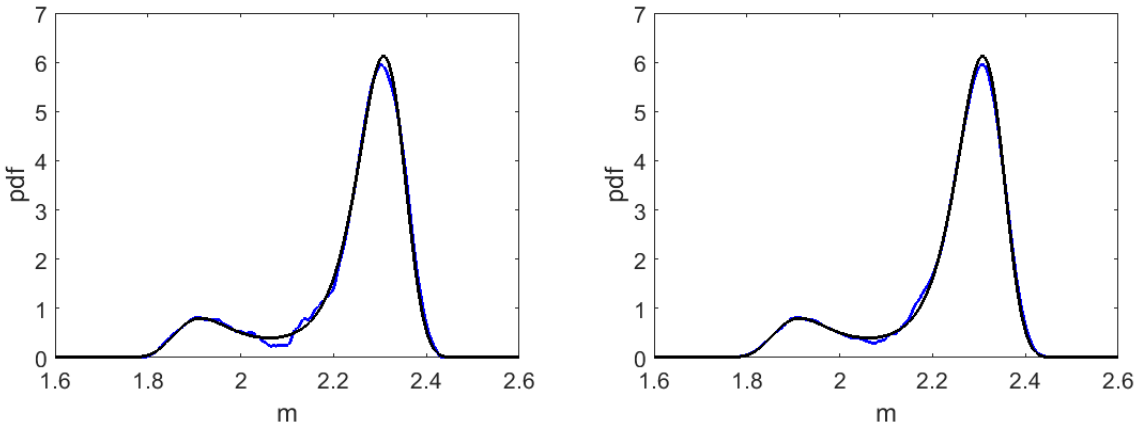
with γ_i equal to 10^{-4} and 10^{-3}. We observe for both values of γ_i the probability of the region between the two modes which was close to zero initially is increased with adaptation which means that the initial GMM proposal distribution becomes closer to the target pdf with adaptation which should enable each chain to converge more quickly to the target pdf. Since both values of γ result in very similar final GMMs, in the remainder of this section we only discuss the results obtained with $\gamma_i = 10^{-4}$. It is also important to note that even with covariance adaptation the GMM is not exactly the same as the target pdf and this is due to the fact that we cannot find a GMM with two modes that is able to match the target pdf. However, the average acceptance rate based on all five chains is 80 percent where the acceptance rate for a chain is defined as the number of accepted states divided by the total number of iterations multiplied by 100. In order to monitor the convergence of the chains, we use the multivariate potential scale reduction factor (MPSRF) [8, 56] which should approach a value of 1.0 when the chains converge. Figure 4.7 shows the plot of MPSRF with the iterations in the chains and we observe that the chains converge quickly with the value of MPSRF very close to one from states 500 onward. Therefore, the states from 501 from all the chains are combined to approximate the posterior pdf. The approximation of the posterior pdf is plotted in Figure 4.8 compared to the true posterior pdf. In Fig. 4.8(a)

we used only 1500 samples from each to approximate the posterior pdf and in Fig. 4.8(b) we used 5000 samples from each chain. Overall, we are able to resolve the posterior pdf with a good approximation. The slight discrepancy between the sample distribution and the posterior pdf is due to the small sample size and it can be observed that with increasing the sample size the approximation of the posterior improves specially in the region between the two modes. Although it is not shown here, if we use all the states after 500, i.e., states from 500 to 50,0000, the sample distribution will be indistinguishable from the target pdf.



(a) (b)

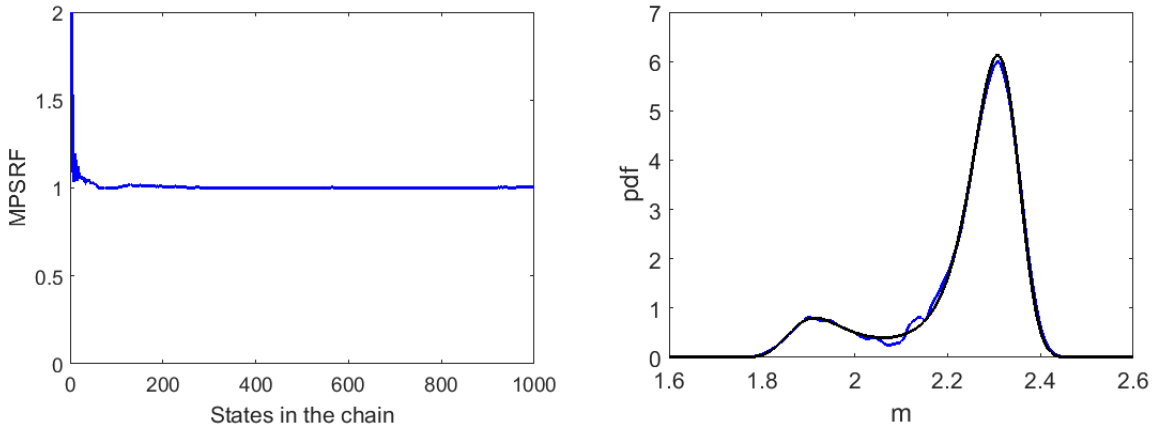
Figure 4.7: Convergence of the chains for toy problem 2. (a) full range of iterations (b) zoom in for the first 1000 iterations.



(a) (b)

Figure 4.8: Comparison of the posterior pdf with distribution obtained using samples from all chains starting from iteration 500 (a) 1500 samples from each chain (b) 5000 samples from each chain. The black curve is the posterior pdf and the blue curve is sample distribution.

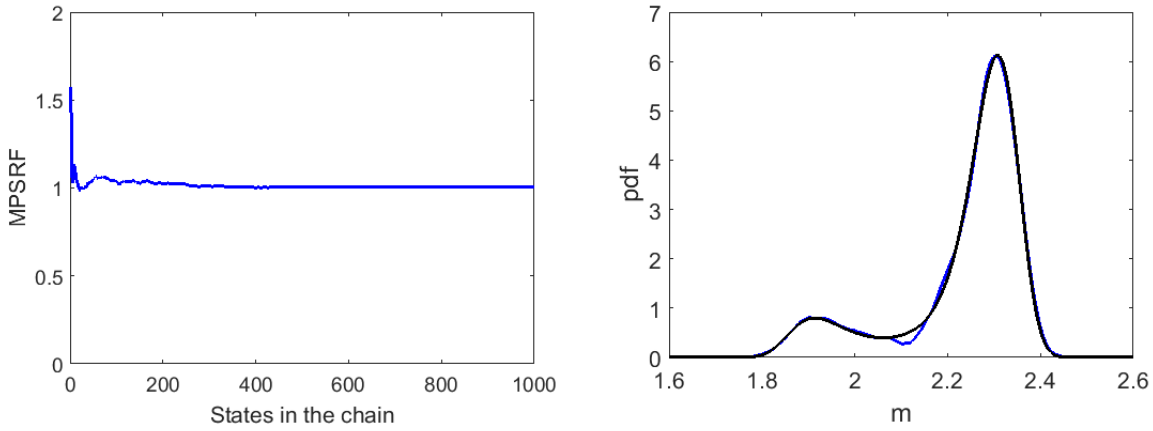
To investigate the effect of covariance adaptation on the performance of the two-level MCMC method, we repeated the sampling step for this toy problem without adapting the covariances of the GMM. Figure 4.9 shows the results of the two-level MCMC without covariance adaptation. The plot of MPSRF, which is shown in Fig. 4.9(a), indicates that the chains converge quickly and the value of MPSRF is very close to 1 from iteration 500 onward. The sample distribution obtained with 5000 samples from each chain starting from 501 state in the chains is plotted in Fig. 4.9(b) and we can see that it approximate the posterior pdf slightly worse compared to the case with covariance adaptation (see Fig. 4.8(b)). The average acceptance rate between chains for the case without adaptation is 79 percent which is very close to the average acceptance rate with covariance adaptation. Although the covariance adaptation does not provide much improvement in this toy problem in terms of acceptance rate, covariance adaptation is important to obtain a reasonable acceptance rate for a higher dimension problems.



(a) (b)

Figure 4.9: MCMC performance for toy problem 2 without covariance adaptation. (a) Convergence of the chains (b) The target pdf (black curve) compared to sample distribution (blue curve) using 5000 samples from each chain.

Finally, we investigate the effect of weights of the GMM components on the sampling performance of the MCMC. In this case, we assume that the weights of the GMM components are equal (see the green curve in Fig. 4.4) and the results of the MCMC are plotted in Figure 4.10. Although the average acceptance rate for the chains is 55 percent in this



(*a*) (*b*)

Figure 4.10: MCMC performance for toy problem 2 with equal weights for the GMM components. (*a*) Convergence of the chains (*b*) The target pdf (black curve) compared to sample distribution (blue curve) using 5000 samples from each chain.

case, the convergence of the chains is similar to the case where the weights of the GMM components are obtained based on the value of the objective function and the approximation of the posterior pdf with sample distribution is slightly different but is almost as well as in Fig. 4.8(b).

4.3.2 1D model

The second test problem is the one designed by Emerick and Reynolds [24] for testing the performance of different history matching methods which is a one-dimensional reservoir model descritized into 31 gridblocks, see Figure 4.11. Each gridblock is 50 ft in all directions. The porosity for all gridblocks is equal to 0.25 and oil and water viscosities, respectively, are 2 and 1 cp. The initial reservoir pressure is 3500 psi. The model includes a water injection well and a producer. The injection well perforated in the first gridblock which injects water at constant bottomhole pressure of 4000 psi. The producer is perforated in the 31st gridblock and produces under constant bottomhole pressure of 3000 psi. As shown in Fig. 4.11, a monitoring well is located at the center of the reservoir model. The gridblock log-permeabilities are considered as the model parameters. The prior covariance is generated using an exponential covariance function with the practical range corresponding to 10 grid-

blocks. The prior mean and variance of $\ln(k)$, respectively, are 5.0 and 1.0. The observed data includes the monthly pressure measurements at the monitoring well for the first 360 days which means that we have 12 pressure measurements as the observed data. The true model is the same one used by Emerick and Reynolds [24] and was generated randomly from the prior model. The data predicted from the true model is perturbed by adding random Gaussian noise with mean of zero and standard deviation of one psi to obtain the vector of observations, d_{obs}. The vector of observations, d_{obs}, is the same one used by Emerick and Reynolds [24]. As a reference to compare our two-level MCMC results we use the results

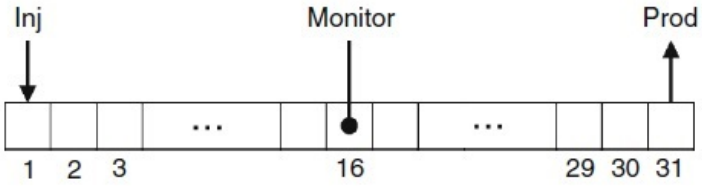


Figure 4.11: The gridblocks and well locations in 1-D waterflooding example.

of random walk chain which was generated by Li [55] to sample the posterior pdf given by Eq. 4.15. The length of the chain is 23 million where the initial state is a random sample from the prior model. The proposal distribution is a Gaussian centered around the current state in the chain with the covariance of $\sigma^2 C_M$. The scaling factor, σ, of 0.005 was used by Li [55] for this example which results in 22.9 percent acceptance rate. Figure 4.12 shows the marginal posterior distributions of permeability and water production rate obtained with the long random walk MCMC. Note that these marginal distributions are generated using every 25th state in the last 500 thousand states in the chain.

As the first step of our two-level MCMC method, we apply the DGN method according to Algorithm 4.1 starting from 200 realizations drawn randomly from the prior distribution, i.e., $N_e = 200$. These 200 realizations represent the base realizations or the base-cases and the initial training set. We use $d_{\min} = 0.001$ and the new models are added to the training set if the distance between the model and all models in the training set is greater than or equal to $d_{\min}$. In Eqs. 4.1 and 4.2, we use $N = 20$ which means that $\Delta M^{(j)}$ and $\Delta D^{(j)}$

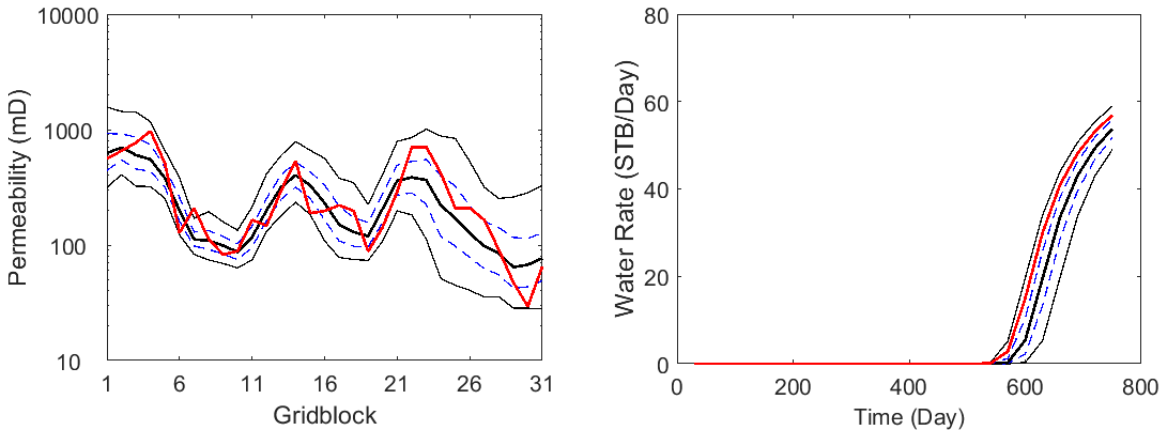


Figure 4.12: Marginal distribution of (a) permeability and (b) water production rate obtained using a long random walk MCMC (plots are regenerated from Li [55]). In both figures, red curve represents the true, the solid black curves are P5 and P95, the blue dashed lines curves are P25 and P75, and the thick solid black curve is the median (P50).

are constructed using the 20 closest models to m_j from the training set (which is less than the number of parameters, $N_m = 31$). The sensitivity matrix for each model is calculated using Eq. 4.9 where p is selected based on Eq. 4.8 with $\hat{a} = 0.001$. The initial value of the trust region radius for all base-cases is set to 1 with a maximum value of 2 and minimum value of 0.01 and if the trust region radius for a given base-case is reduced to a value less than this minimum value we flag that base-case as converged. A base-case also is flagged as converged if at some iteration its the normalized objective objective function (objective function divided by the number of data) is less than 3.0. The DGN algorithm converges if 95 percent of base-cases are converged and this convergence criterion is satisfied after 61 iterations. Figure 4.13 shows the number of running base-cases during the iterations of the DGN. The total number of forward model runs that the DGN algorithm required is 5068. At the end of DGN algorithm, 60 models give a normalized objective function value less than 5 and we use these models the modes to build the GMM. Before discussing the MCMC results, we should mention that the calculation of the weights based on the objective function as in Eqs. 4.26 results in assigning 93 percent of the total weights to the four minimizers which give normalized objective function values very close to 2. Because the sensitivity matrix is approximated the gradient of the objective function is not accurate and the algorithm cannot

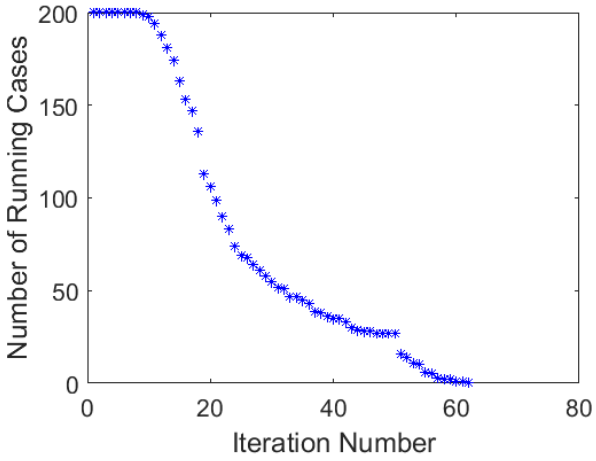


Figure 4.13: Plot of the number of running base-cases for DGN algorithm applied to 1D problem.

decrease the objective function to the same level for all modes.

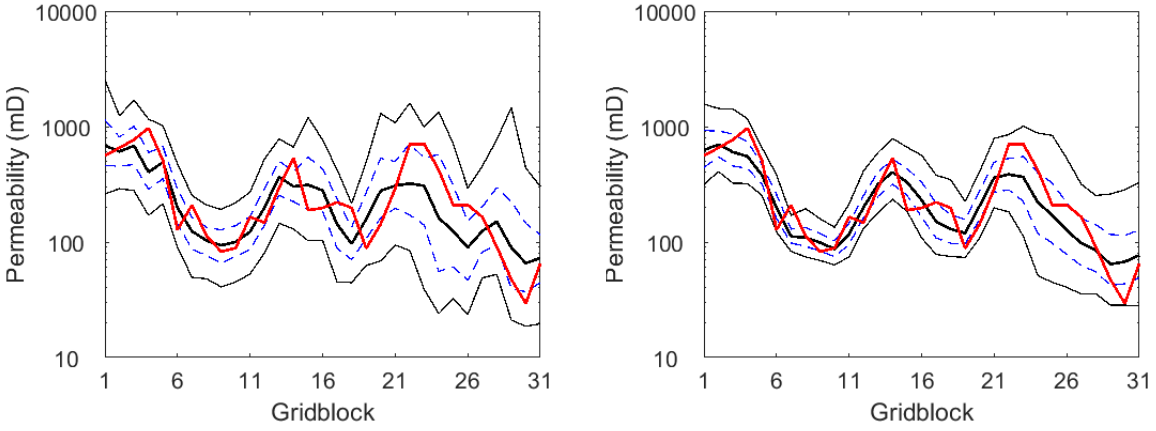
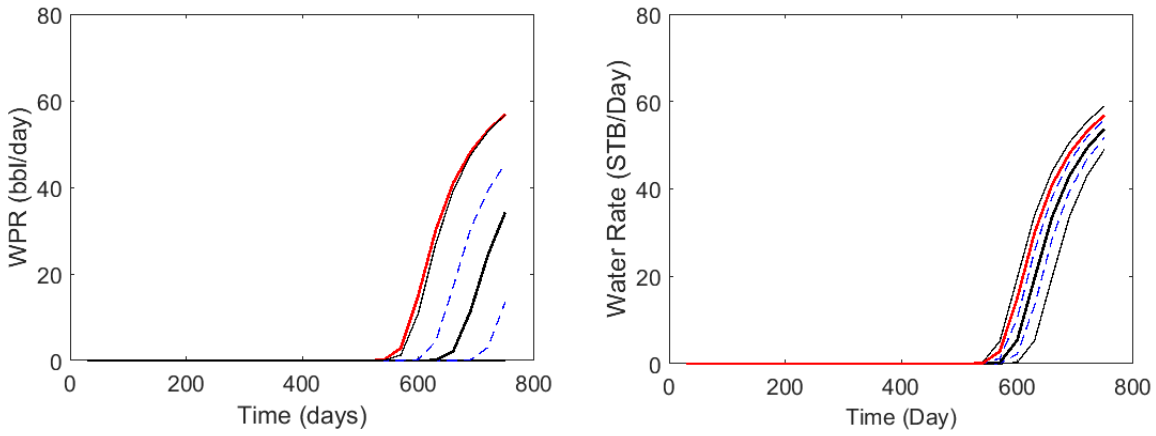


Figure 4.14: Marginal distribution of permeability obtained using (a) 1000 realizations drawn randomly from GMM as proposed in [33] (b) random walk MCMC. The legend is the same as Fig. 4.12.

In the field case presented in [33], a weight of 0.9999 is assigned to one minimizer and they used only that minimizer and its corresponding covariance matrix to build a Gaussian distribution and generated 1000 samples using this Gaussian to quantify uncertainty. Therefore, we built a GMM using all the 60 modes where the weights are calculated using Eq. 4.26 and generated 1000 random realizations from the GMM. We run the forward model for these realizations to obtain the predicted data. Figures 4.14 and 4.15 show, respectively,



(a) (b)

Figure 4.15: Marginal distribution of water production rate obtained using (*a*) 1000 realizations drawn randomly from GMM as proposed in [33] (*b*) random walk MCMC. The legend is the same as Fig. 4.12.

the marginal distribution of permeability and water production rate obtained using the 1000 realizations generated directly from the GMM compared to the random walk MCMC results. From Fig. 4.14(a) we observe that the marginal distribution of permeability covers the true permeability, but the distribution is wider compared to the marginal distribution of permeability obtained from the long random walk MCMC given in Fig. 4.14(b). The marginal distribution of the water production rate plotted in Fig. 4.15(a) is not a good approximation of the water production rate generated using the long random walk MCMC (see Fig. 4.15(b)) and the water production rate from the true model is outside the P95 curve, Fig. 4.15(a). Although generating a set of realizations from the GMM is computationally less expensive than sampling using MCMC, this example confirms that generating a set of realizations from the GMM does not provide an acceptable quantification of uncertainty.

In order to build the GMM proposal distribution for the second stage of the MCMC method, we cluster the 60 distinct modes into 25 clusters using the k-medoids algorithm and use the 25 medoids to build the GMM. Although we used k-medoids clustering to reduce the number of modes, but we can use all the 60 modes to build the GMM. It is showed in Li [55] that the difference between the MCMC results with all modes and reduced number of modes is negligible. As discussed earlier, due to approximation of the sensitivity

matrix we cannot find the modes exactly for high dimensional problems. Therefore weighting the GMM components based on the objective function value does not result in reasonable weights (most modes will have weight very close to zero). We showed that for toy problem 2, a GMM with equal weights is less similar to the target pdf (see Fig. 4.4), but when we performed sampling using equal weights we were able to characterize the target pdf (see Fig. 4.10(b)). Thus we set all the weights equal to build the GMM. The covariance matrices are multiplied by 0.1 ($\beta = 0.1$ in Eq. 4.38). We use $\gamma_i = 4/N_m^2$ which is approximately equal to 0.004 to adapt the covariances based on Eq. 4.44. Starting from different initial states, which are drawn randomly from the proposal distribution, we started 5 parallel chains with the maximum number of iterations equal to 100 thousand. The average acceptance rate of

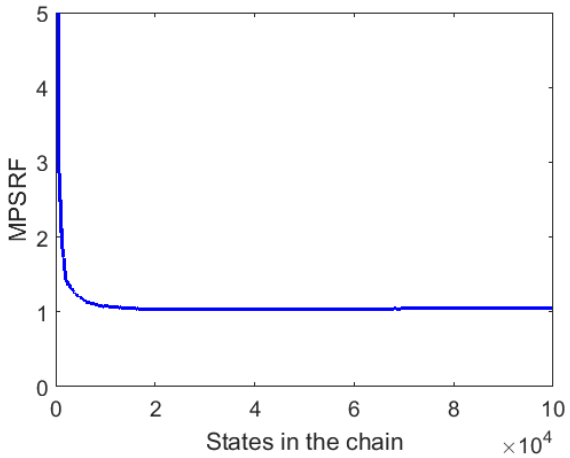


Figure 4.16: Plot of MPSRF versus the states in the chain calculated using 5 parallel chains for the 1D waterflooding problem.

all chains is about 21 percent. Figure 4.16 shows that the MPSRF value stabilizes from the state 20 thousand, however there is only slight change in the MPSRF value from state 20 thousand onward. Therefore, we plotted the marginal distribution of permeability and water production rate using 3000 samples from each chain starting from state 20 thousand in Figures 4.17 and 4.18 respectively. The marginal distribution of permeability covers the true permeability field in Fig. 4.17(a) and similar to the result obtained from the random walk (Fig. 4.17(b)) there exist less variation in the permeability values of gridblocks to the left of the monitoring well where the pressure measurement are acquired. In Fig. 4.18(a) we

observe that the true water production rate curve is almost the same as the P75 curve as it is in Fig. 4.18(b) too, however the range between P95 and P5 curves is wider in Fig. 4.18(a) compared to in Fig. 4.18(b).

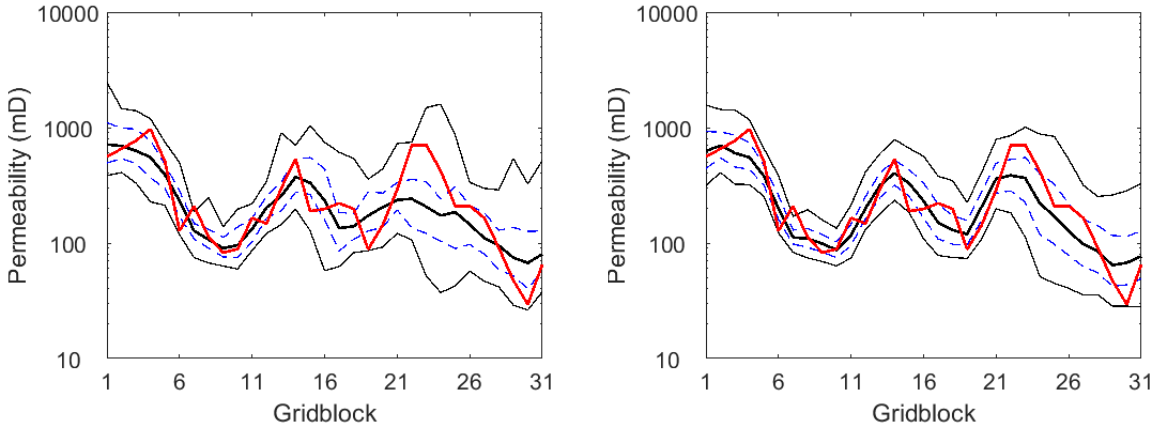


Figure 4.17: Marginal distribution of permeability obtained using (a) 3000 states from each chain of our two-level MCMC starting from state 20 thousand (b) random walk MCMC. The legend is the same as Fig. 4.12.

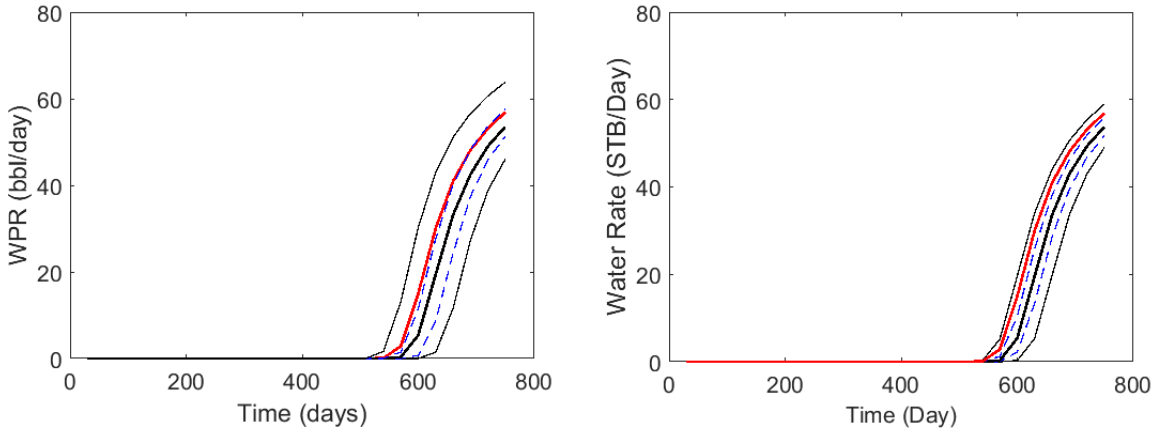


Figure 4.18: Marginal distribution of water production rate obtained using (a) 3000 states from each chain of our two-level MCMC starting from state 20 thousand (b) random walk MCMC. The legend is the same as Fig. 4.12.

Starting from the state 40 thousand in the chains, we use 3000 from each chain to construct the marginal distributions of permeability and water production rate and the result is given in Figure 4.19. Comparing Fig. 4.19(a) and Fig. 4.17(a) we can see that only the

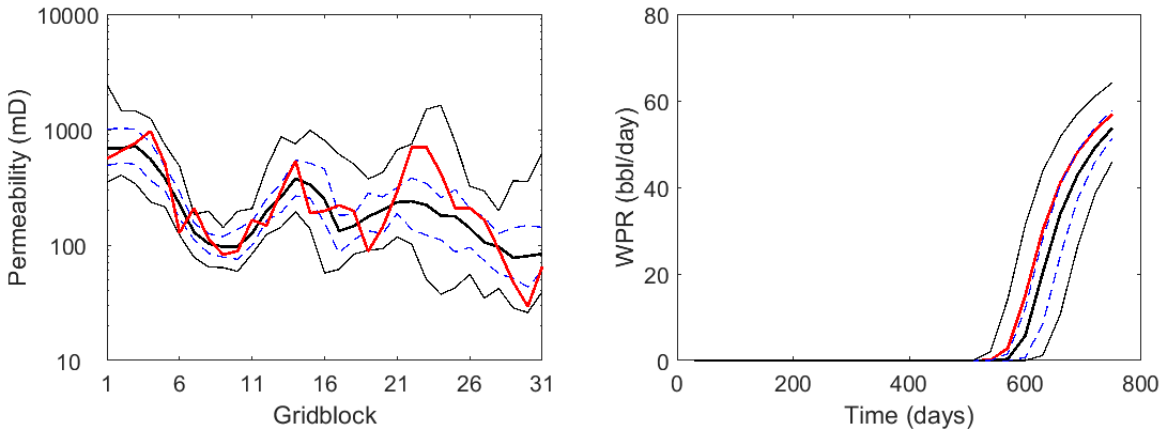


Figure 4.19: Marginal distribution of (a) permeability and (b) water production rate obtained using two-level MCMC. 3000 states from each chain starting from state 40 thousand are used. The legend is the same as Fig. 4.12.

P5 and P95 curves are slightly different where both curves are smoother in Figure 4.19(a). However, there is no visual difference between the water production rate curves in Fig. 4.19(b) and Fig. 4.18(a).

Li and Reynolds [56] used their two-level MCMC in order to sample the posterior pdf for this problem. In their method, they started from 200 initial guesses generated randomly from the prior distribution of the model parameters and minimized the objective function using a gradient-based algorithm. The minimizers with normalized objective function are clustered into 25 clusters using the k-medoids algorithm and they used the models selected as the medoids to build a GMM. The weights of the GMM are assigned proportional to the number of models in each cluster. This GMM is then used as the proposal distribution in the second stage of their two-level MCMC. The results obtained with their two-level MCMC algorithm for this problem are presented in Figures 4.20 and 4.21. Comparing the marginal distribution of permeability in Fig. 4.20(a) with Fig. 4.19(a), we observe that the uncertainty in the permeability of the first few gridblocks obtained by Li [55] is lower than the results of our proposed two-level MCMC. Considering the results of random walk MCMC given in Fig. 4.20(b) as the reference, it seems that our results give a slightly better estimation of uncertainty for the first few gridblocks. The P5 and P95 curves of the marginal pdf for the

water production rate obtained using the two-level MCMC of Li and Reynolds [56] are more similar to the P5 and P95 curves generated using the random walk MCMC than the P5 and P95 curves generated from our two-level MCMC. However, the P75 and true curves in Fig. 4.21(a) are clearly distinct while the true water production rate is very close to P75 curve in Fig. 4.21(b). It is important to note that Li [55] uses modes with normalized objective function values less than 1.5 which are more accurate modes and result in a narrower bounds of uncertainty.

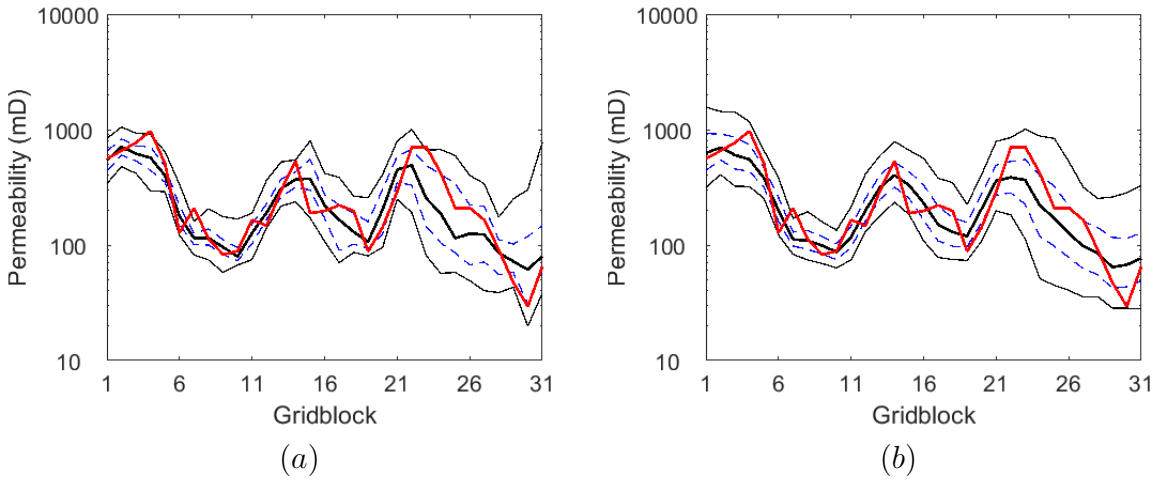


Figure 4.20: Marginal distribution of permeability obtained using (a) the states between 15 thousand to 25 thousand from 5 chains from the two-level MCMC of Li [55] (b) random walk MCMC. The legend is the same as Fig. 4.12.

Finally, we use the ES-MDA algorithms discussed in Chapter 3 to check their performances for this example. Similar to Emerick and Reynolds [24], we use ten independent initial ensembles, each of which contains 100 realizations which are generated using the prior distribution. Note that an ensemble of size 100 (more than three times the number of model parameters) is sufficiently large for ensemble-based methods so that we do not require any covariance localization. The history matching results from four methods, IR-ES with $\rho = 0.5$, M-IR-ES with $\rho = 0.5$, ES-MDA with 10 equal inflation factors and ES-MDA-GEO with 10 data assimilation steps are compared for this problem. The results of ES-MDA-RLM with $\rho = 0.5$ for this example are not included as it did not converge for any of the ten ensembles with less than 100 data assimilation steps. Also, IR-ES with $\rho = 0.5$ did not converge after

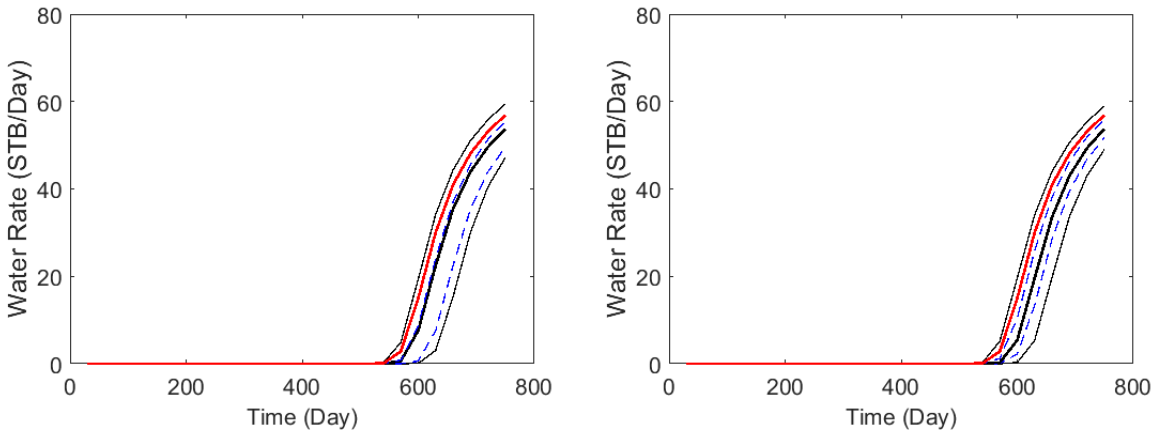


Figure 4.21: Marginal distribution of water production rate obtained using (a) the states between 15 thousand to 25 thousand from 5 chains from the two-level MCMC of Li [55] (b) random walk MCMC. The legend is the same as Fig. 4.12.

250 iterations for the first ensemble and the results of this ensemble is not included here and the average number of iterations that IR-ES takes for all the remaining nine ensembles is 18 iterations. Note that the average iterations required for M-IR-ES is 12 iterations. Figure 4.22 shows the box plot of the normalized objective function for all methods as well as the two level MCMC algorithms. Among the ensemble-based methods, IR-ES results in the lowest value for the median of the normalized objective function. Note that IR-ES iterates until the data mismatch satisfies Eq. 3.25 and thus requires a low data mismatch. Note that we used ES-MDA-GEO with 10 iterations which is almost half of the average number of iterations required by IR-ES to converge. Figures 4.23 and 4.24, respectively, show the marginal distribution of permeability and water production rate obtained with different methods including the random walk MCMC and our two-level MCMC. According to the results of Fig. 4.23(c), ES-MDA with equal inflation factors results in a poor approximation of the marginal distribution of permeability compared to other methods. This last statement is consistent with the marginal distribution of the water production rate given in Fig. 4.24(c), which pertains to ES-MDA with equal inflation factors, where the water production rate curve from the true model is outside the P95 curve while it should be close to the P75 curve based on the MCMC results. According to the MCMC results given in

plots (e) and (f) of Fig. 4.23, the true permeability curve should be almost enclosed within the P5 and P95 curves, which is not the case for any of ES-MDA plots (see plot (a)-(d) of Fig. 4.23). However, the marginal distribution of permeability obtained with ES-MDA

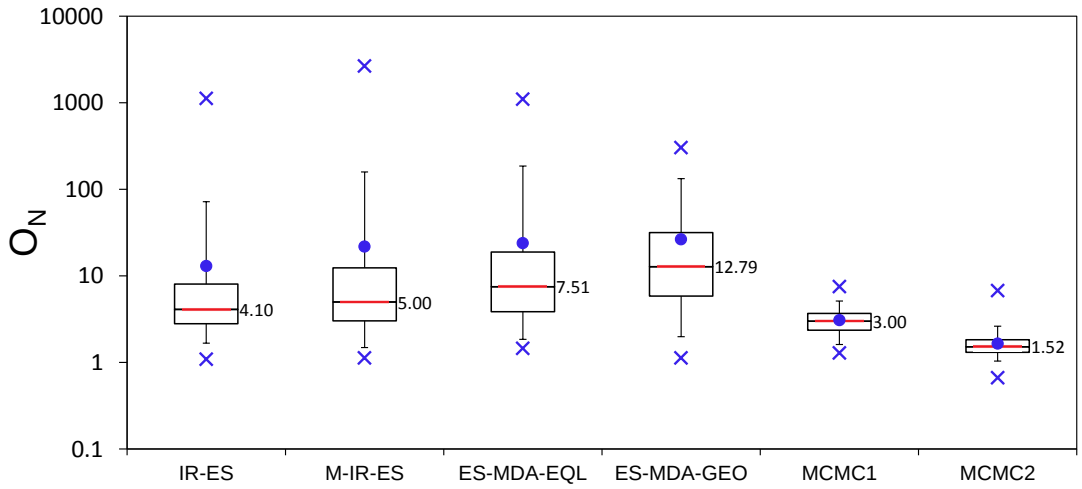


Figure 4.22: Box plot of the normalized objective function. The red line within each box corresponds to the median, and the bottom and top of each box correspond to P25 and P75. The horizontal black dash lines correspond to P2 and P98. The blue circle corresponds to the mean and blue crosses are the minimum and maximum values. The number next to the boxes correspond to the median of O_N. MCMC1 refers to our two-level MCMC algorithm and MCMC2 refers to the two-level MCMC of Li and Reynolds [56].

with geometric inflation factors (Fig. 4.23(d)) includes the true permeability curve within its P5 and P95 curves whereas this is not the case with the marginal distributions generated from IR-ES, M-IR-ES, and ES-MDA-EQL. From Fig. 4.24 it can be observed that the water production curves of the true model is very close to the P75 curve obtained with both IR-ES and ES-MDA-GEO which is consistent with the results obtained with random walk MCMC and our two level MCMC. Note that the marginal distribution of the water production rate obtained with IR-ES and ES-MDA is very similar to the result obtained with random walk MCMC despite of the differences in their marginal permeability distributions. Although the ES-MDA does not provide the exact characterization of the posterior pdf for this example, it provides a reasonably good approximation of uncertainty and is far more computationally efficient than the MCMC algorithm.

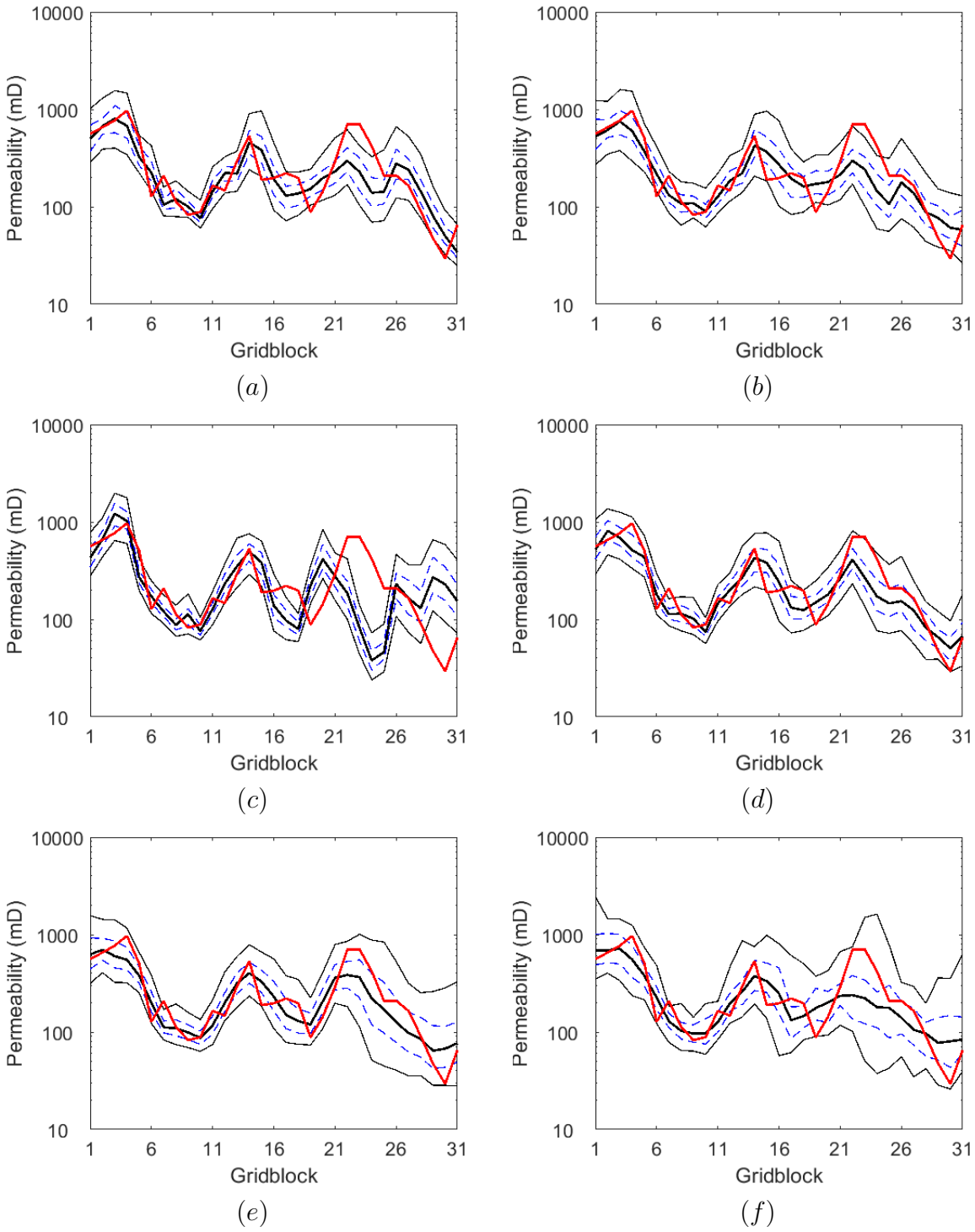


Figure 4.23: Marginal distribution of permeability obtained using (a) IR-ES with $\rho = 0.5$ (b) M-IR-ES with $\rho = 0.5$ (c) ES-MDA-EQL with $N_a = 10$ (d) ES-MDA-GEO with $N_a = 10$ (e) random walk MCMC (f) our two-level MCMC. The legend is the same as Fig. 4.12.

4.3.3 2D model

This example pertains to a synthetic two-dimensional square reservoir which is de-

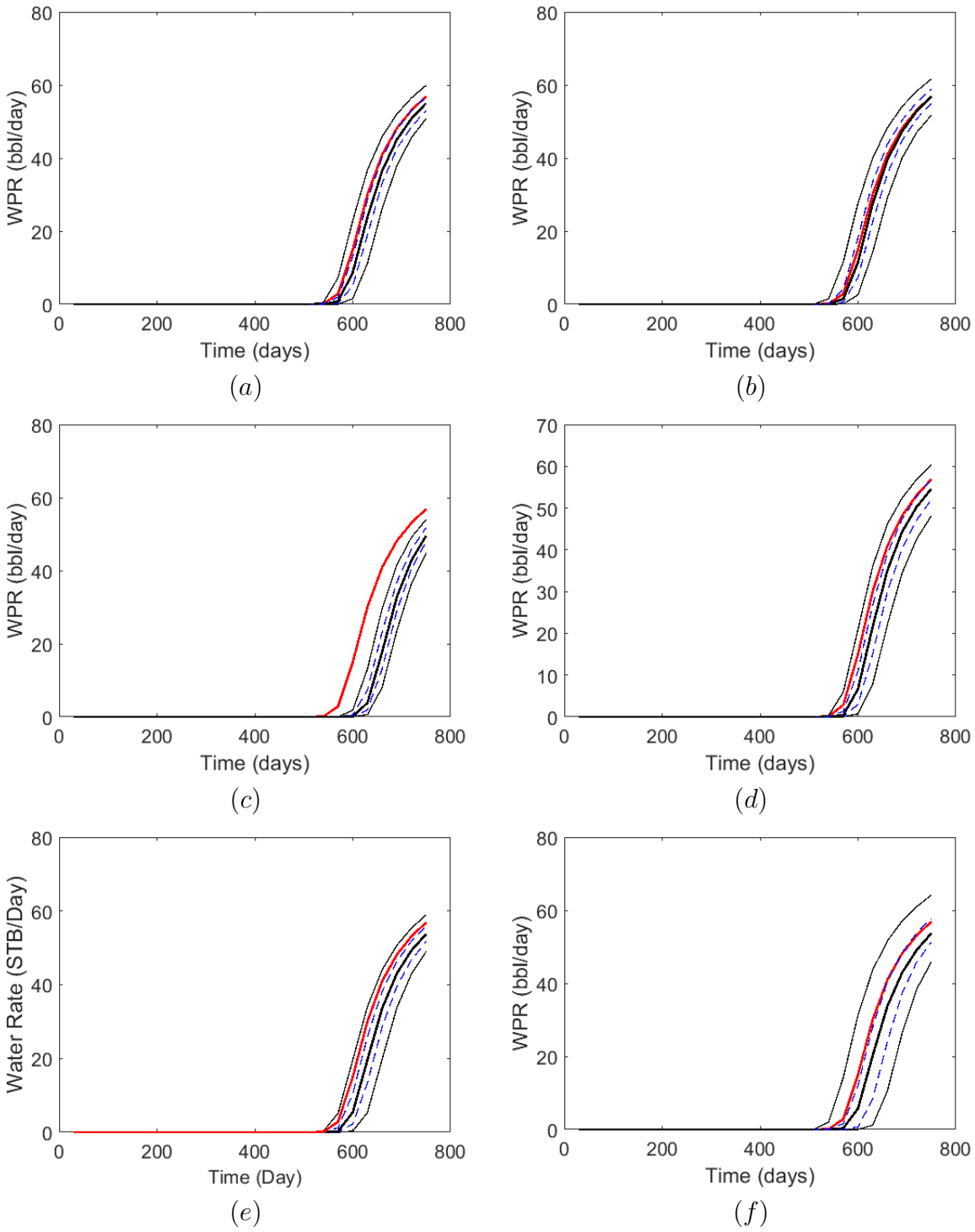


Figure 4.24: Marginal distribution of water production rate obtained using (a) IR-ES with $\rho = 0.5$ (b) M-IR-ES with $\rho = 0.5$ (c) ES-MDA-EQL with $N_a = 10$ (d) ES-MDA-GEO with $N_a = 10$ (e) random walk MCMC (f) our two-level MCMC. The legend is the same as Fig. 4.12.

signed in Li [55] to test their two-level MCMC algorithm. The reservoir domain is discretized on a $44 \times 44 \times 1$ grid. Each gridblock is $100 \times 100 \times 15$ ft. The true permeability field is generated using an exponential covariance function with the major range of $2,500$ ft and the minor range of $1,100$ ft where the major range is oriented in 45° from the positive y-direction. The mean and the variance of ln(k) are, respectively, 5 and 1.0. The model (vector of reservoir parameters), m, consists of the set of gridblock horizontal log-permeabilities. We assume that permeability is isotropic. Figure 4.25 shows the true log-permeability map (true model). The reservoir model contains nine production wells and four water injection wells. The initial reservoir pressure is $3,000$ psi. The injection wells operate with constant bottomhole pressure equal to $4,500$ psi during the whole simulation time while the production wells operate at a constant bottomhole pressure of $2,800$ psi for the first six months and then the bottomhole pressure constraint is changed to $2,500$ psi for the rest of the simulation. The observations include the monthly water injection rate of the injection wells and water and oil production rates of the production wells from the first 36 months. The history-match period, which is 36 months, is followed by a 20 month long prediction period. The predicted data from the true model are obtained by running the reservoir simulator model with the true model parameters and the measurements are acquired by adding random Gaussian noise to the data predicted from the true model. The standard deviation of the measurement error is assumed to be equal to 5% of the measured value with the minimum of 2 STB/day.

As the first step of our two-level MCMC method, we apply the DGN method according to Algorithm 4.1 starting from 250 realizations drawn randomly from the prior distribution, i.e., $N_e = 250$. These 250 realizations represent the base realizations or the base-cases. The initial training set includes 400 realizations which includes the 250 base-cases. We use $d_{\min} = 0.001$ and the new models are added to the training set if the distance between the model and all models in the training set is greater than or equal to $d_{\min}$. In Eqs. 4.1 and 4.2, we use $N = 100$ which means that $\Delta M^{(j)}$ and $\Delta D^{(j)}$ are constructed using the 100 closest models to m_j from the training set (which is less than the number of parameters, $N_m = 1936$). The sensitivity matrix for each model is calculated using Eq. 4.9 where p is

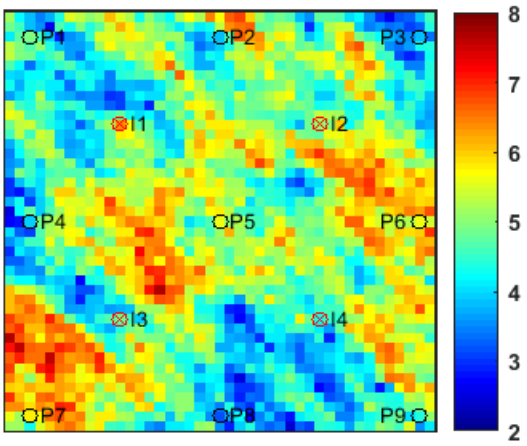


Figure 4.25: True log-permeability map for the synthetic 2D model. The black circles are the production wells and the red circles with cross are the injection wells.

selected based on Eq. 4.8 with $\hat{a} = 0.001$. The initial value of the trust region radius for all base-cases is set to 1 with a maximum value of 2 and minimum value of 0.1 and if the trust region radius for a given base-case is reduced to a value less than this minimum value we flag that base-case as converged. A base-case also is flagged as converged if at some iteration its the normalized objective objective function (objective function divided by the number of data) is less than 3.0. The DGN algorithm converges if 95 percent of base-cases are converged and this convergence criterion is satisfied after 25 iterations. Figure 4.26 shows the number of running base-cases during the iterations of the DGN. The total number of forward model runs that the DGN algorithm required is 4722. At the end of DGN algorithm, 52 models

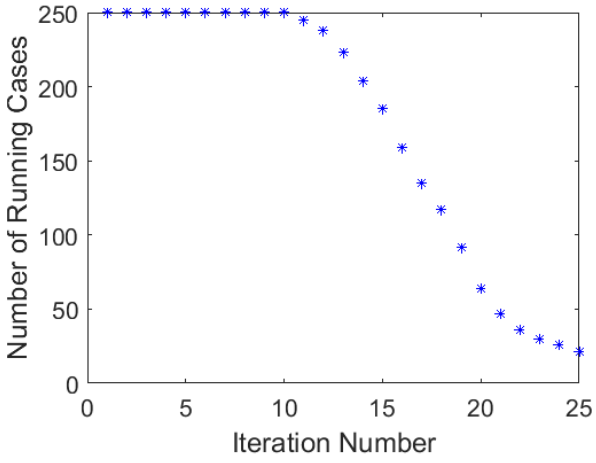


Figure 4.26: Plot of the number of running base-cases for DGN algorithm applied to 2D problem.

give a normalized objective function value less than 3.5. In order to build the GMM proposal distribution for the second stage of the MCMC method, we cluster the 52 distinct modes into 25 clusters using the k-medoids algorithm and use the 25 medoids to build the GMM. Since our modes are not exact weighting the GMM components based on the objective function value does not results in reasonable weights and similar t the previous example we set all the weights equal. The covariance matrices are multiplied by 0.08 ($\beta = 0.08$ in Eq. 4.38) and we use $\gamma_i = 500/N_m^2$ which is approximately equal to 0.0001 to adapt the covariances based on Eq. 4.44. Starting from different initial states, which are drawn randomly from the proposal distribution, we started 5 parallel chains with the maximum number of iterations equal to 50 thousand. The average acceptance rate of all the five chains is about 23 percent for this problem. According to Figure 4.27, which shows the value of MPSRF versus the iteration number in the chains, we observe that the five chains converge after about 30 thousand iterations as the value of MPSRF stabilizes to a value close to one. Therefore, we use the states between 30 to 40 thousands to approximate the posterior distribution.

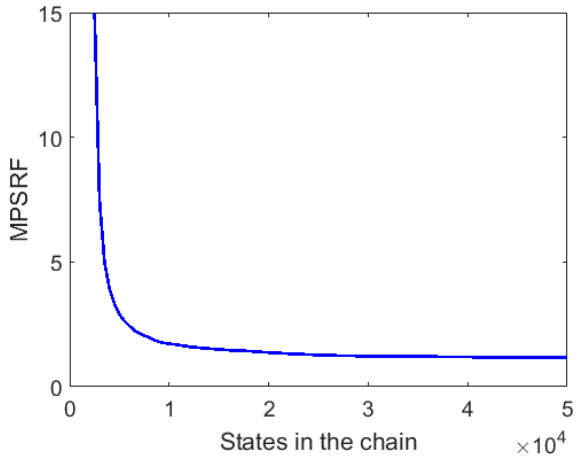
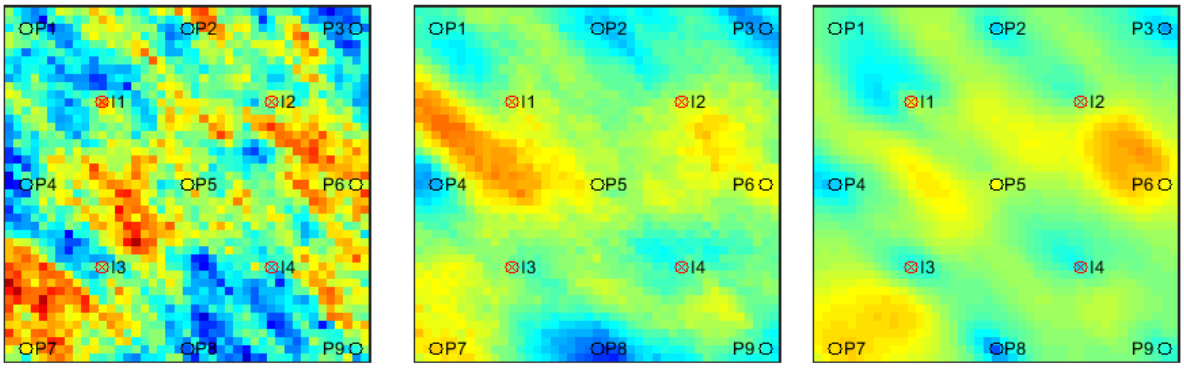


Figure 4.27: Plot of MPSRF versus the states in the chain calculated using 5 parallel chains for the 2D problem.

We compare the results of our two-level MCMC algorithm for this 2D problem with the results obtained by the two-level MCMC of Li [55] which uses gradient-based minimization to find minimizers of the objective function. For this example, Li [55] started with 250 random realizations of the prior as the initial guesses for their gradient-based minimization



(a) (b) (c)

Figure 4.28: The posterior mean of the log-permeability compared to true model. (a) True model (b) our two-level MCMC and (c) two-level MCMC of Li and Reynolds [56]. The colorbar scale is the same as in Fig. 4.25

algorithm. Then, the minimizers with normalized objective function less than 1.5 are clustered into 35 modes using k-medoid clustering algorithm. The GMM is then constructed using these 35 modes and is used as the proposal distribution for their two-level MCMC algorithm, which shows convergence after 30 thousand iterations. Figure. 4.28 shows the mean of the log-permeability obtained with both MCMC algorithms compared to the true model. Fig. 4.28 indicates that both two-level MCMC algorithms are able to capture the mean features (structures) of the true model, but the structures (high or low permeability regions) captured from our two-level MCMC (Fig. 4.28(b)) are larger than those obtained with the two-level MCMC of Li [55] (Fig. 4.28(c)). The main reason for this last observation is that Li [55] uses analytical gradient of the objective function which is more accurate than our simplex gradient approximation. This approximation of the gradient causes changing the model parameters in a larger region that is required with analytical gradient.

Figures 4.29-4.31 compares the data match and prediction of well rates for some wells using both MCMC algorithms. It is clear that the data match obtained with our two-level MCMC is inferior compared to those obtained with the two-level MCMC of Li [55] which is an expected result due to the accuracy of the modes used to build the GMM. In our algorithm, we used modes with normalized objective function less than 3.5 to construct our GMM while Li [55] used the modes with normalized objective function less than 1.5. Note

that since we use an approximate gradient we cannot decrease the objective function as low as it can be reduced with a gradient-based method. As a result, the predictions obtained with our method also have wider bands compared to those Li [55] obtained.

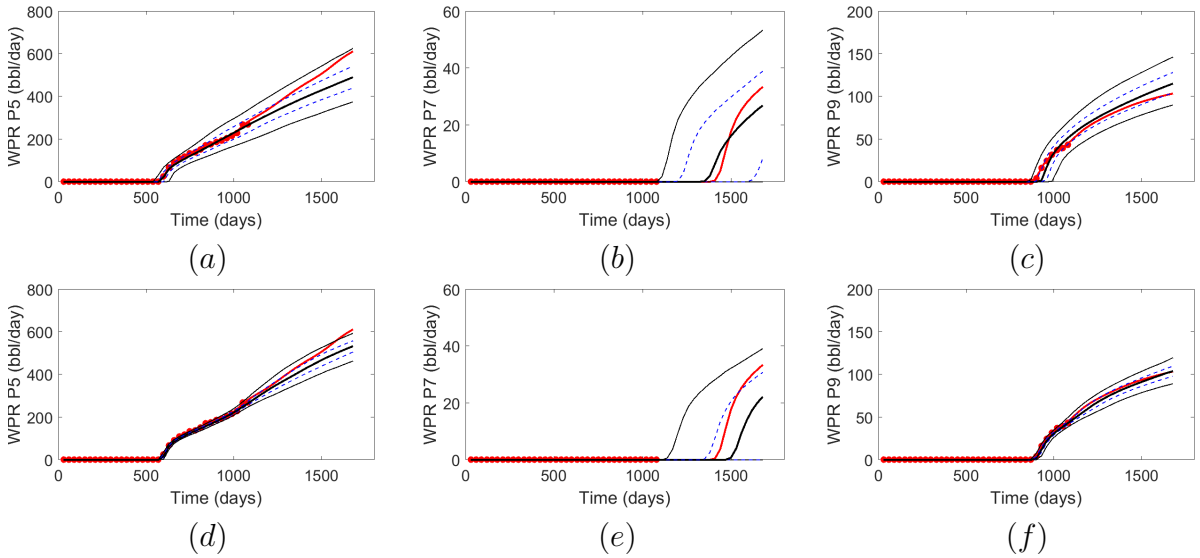


Figure 4.29: Plot of water production rate of wells P5, P7 and P9 for 2D model. (a)-(c) our two-level MCMC and (d)-(f) two-level MCMC of Li and Reynolds [56]. The red curve represents the true, the red circles are the observed data, the solid black curves are P5 and P95, the blue dashed lines curves are P25 and P75, and the thick solid black curve is the median (P50).

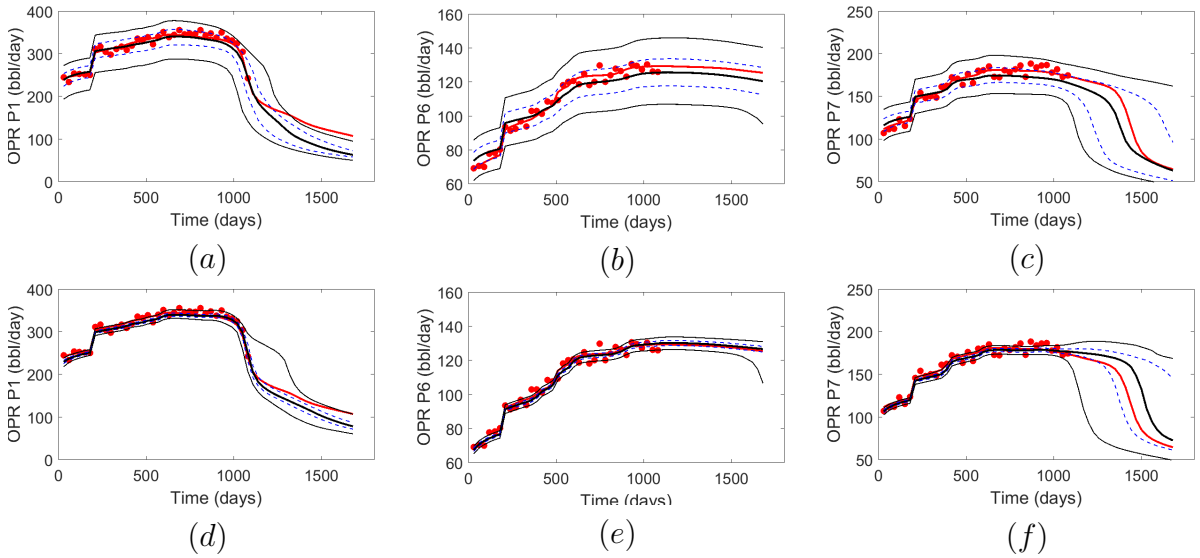


Figure 4.30: Plot of oil production rate of wells P1, P6 and P7 for 2D model. (a)-(c) our two-level MCMC and (d)-(f) two-level MCMC of Li and Reynolds [56]. The legend are the same as in Fig. 4.29.

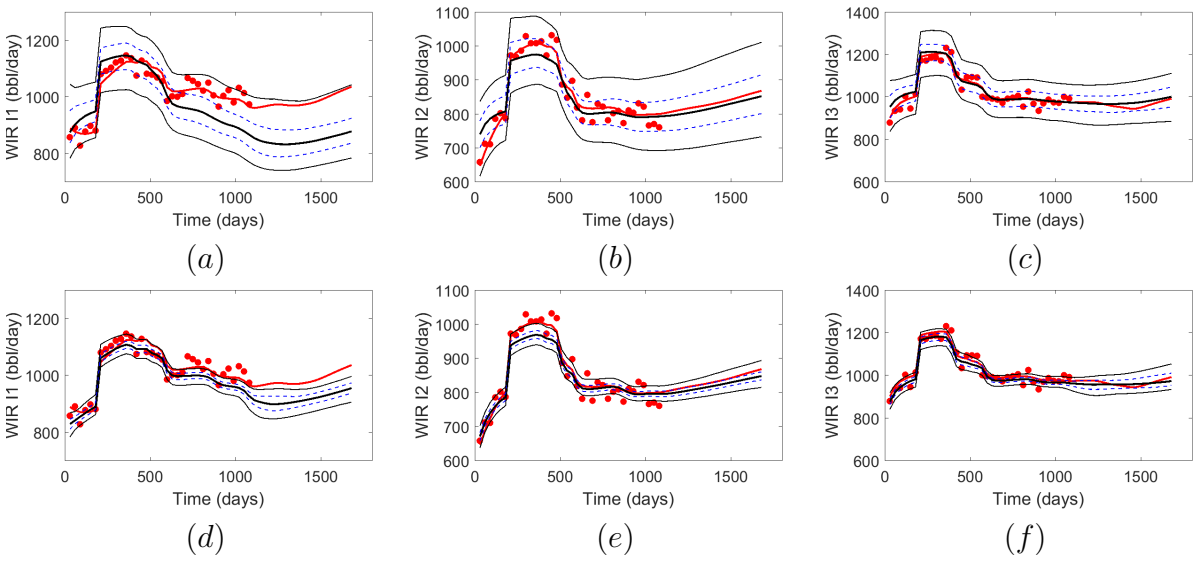


Figure 4.31: Plot of water injection rate of wells I1, I2 and I4 for 2D model. (a)-(c) our two-level MCMC and (d)-(f) two-level MCMC of Li and Reynolds [56]. The legend are the same as in Fig. 4.29.

Finally, we compare the performance of different ES-MDA methods (discussed in Chapter 3) for this problem. An initial ensemble of size 400 is generated using the prior model and are used to perform history matching. Note in this example, we consider $\rho = 0.5$ for ES-MDA-RLM, IR-ES and M-IR-ES methods and $N_a = 6$ for ES-MDA-EQL and ES-MDA-GEO methods. For ES-MDA-GEO, the first inflation factor based on Eq. 3.42 is 1058.4 for this example and solving Eq. 3.45 with $N_a = 6$, we obtain $\beta = 0.264$. Table 4.1 summarizes the results obtained with different methods for this example. In Table 4.1, RMSE is define as

$$\text{RMSE} = \frac{1}{\sqrt{N_m}} \|m_{\text{true}} - \overline{m}\|, \tag{4.51}$$

which compares the mean of the posterior model parameters, $\overline{m}$, with the true model, m_{true}. Also, $\overline{O}_N$ is the average normalized objective function which is different from O_{Nd} introduced in Chapter 3 which only includes the data mismatch term. First, we compare the results obtained with the two MCMC methods. We refer to our two-level MCMC as MCMC1 and we use MCMC2 to refer to the two-level MCMC of Li [55]. We can see that the values of RSME, $\overline{\sigma}$, and $\overline{O}_N$ are higher for MCMC1 compared to those obtained with MCMC2. A higher RSME indicated that the mean model farther from the true model, the higher value of

$\overline{\sigma}$ shows that the variation in the models is higher, and higher value of $\overline{O}_N$ is an indication of inferior data match. These results are consistent with the results of Figs. 4.28-4.31. For the ensemble-based method, we observe that ES-MDA-RLM results in the closest value of RSME to that obtained with MCMC2. However, ES-MDA-RLM converges in 19 iterations which is more than three times the iterations we used for ES-MDA-GEO and its final value $\overline{O}_N$ is lower than that of the MCMC2 method which is an indication of overmatching the data. Ignoring the results of ES-MDA-RLM, ES-MDA-GEO results in the closest value of RSME to that obtained with MCMC, while ES-MDA-EQL and M-IR-ES give the highest values of RSME which is consistent with rougher mean log-permeability maps shown in Figure 4.32. IR-ES gives the highest value of $\overline{\sigma}$ which is close to that given by MCMC2. ES-MDA-GEO

	Prior	ES-MDA-RLM	IR-ES	M-IR-ES	ES-MDA-EQL	ES-MDA-GEO	MCMC1	MCMC2
RMSE	0.936	0.791	0.833	0.921	0.882	0.770	0.885	0.793
$\overline{\sigma}$	0.993	0.450	0.701	0.445	0.375	0.491	0.872	0.734
$\overline{O}_N$	8367	1.19	4.45	3.03	1.42	2.26	3.76	2.25
Iter	-	19	8	4	6	6	-	-

Table 4.1: Comparison of the performance of different methods for 2D problem. MCMC1 refers to our two-level MCMC and MCMC2 refers to the two-level MCMC of Li [55].

results in the second highest value $\overline{\sigma}$ among the ensemble-based methods. Regarding the average normalized objective function, $\overline{O}_N$, ES-MDA-GEO gives the closet value of $\overline{O}_N$ to that obtained with MCMC2 while IR-ES results in the highest value of $\overline{O}_N$. Figures 4.33-4.35 compares the data match and predictions for some wells obtained with ensemble-based methods and MCMC2. Based on the results of these figures, we observe that IR-ES is not able to match the data well compared to other method (see plot (i) of Fig. 4.33, plots (g)-(i) of Fig. 4.33, and plots (g)-(h) of Fig. 4.33), and the uncertainty in the predictions is higher than those obtained with MCMC2. Overall, the uncertainty in the predictions of ES-MDA-RLM, M-IR-ES, and ES-MDA-EQL is lower than the results of MCMC2 and ES-MDA-GEO gives the best quantification of the uncertainty in the predictions compared to other methods. The results of this example confirms that ES-MDA provides a reasonably good approximation of uncertainty while it requires far less computational effort than the MCMC algorithm.

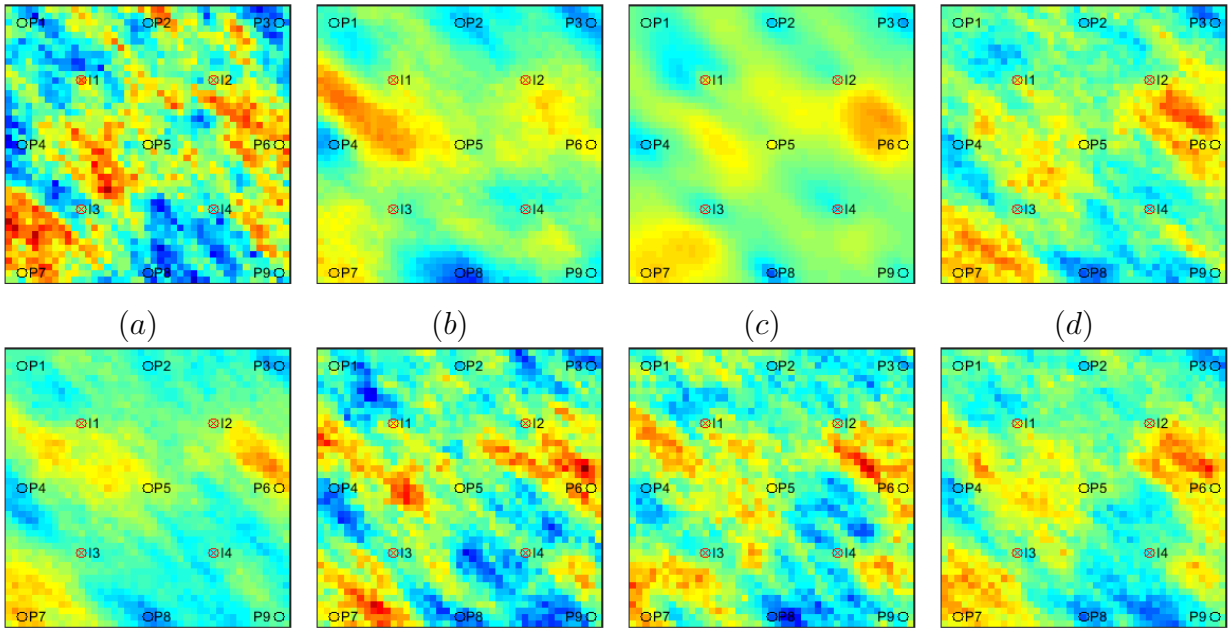


Figure 4.32: The posterior mean of the log-permeability compared to true for 2D model. (a) True, (b) our two-level MCMC, (c) two-level MCMC of Li and Reynolds [56], (d) ES-MDA-RLM $\rho = 0.5$, (e) IR-ES $\rho = 0.5$, (f) M-IR-ES $\rho = 0.5$, (g) ES-MDA-EQL $N_a = 6$, and (h) ES-MDA-GEO $N_a = 6$. The colorbar is the same as in Fig. 4.25.

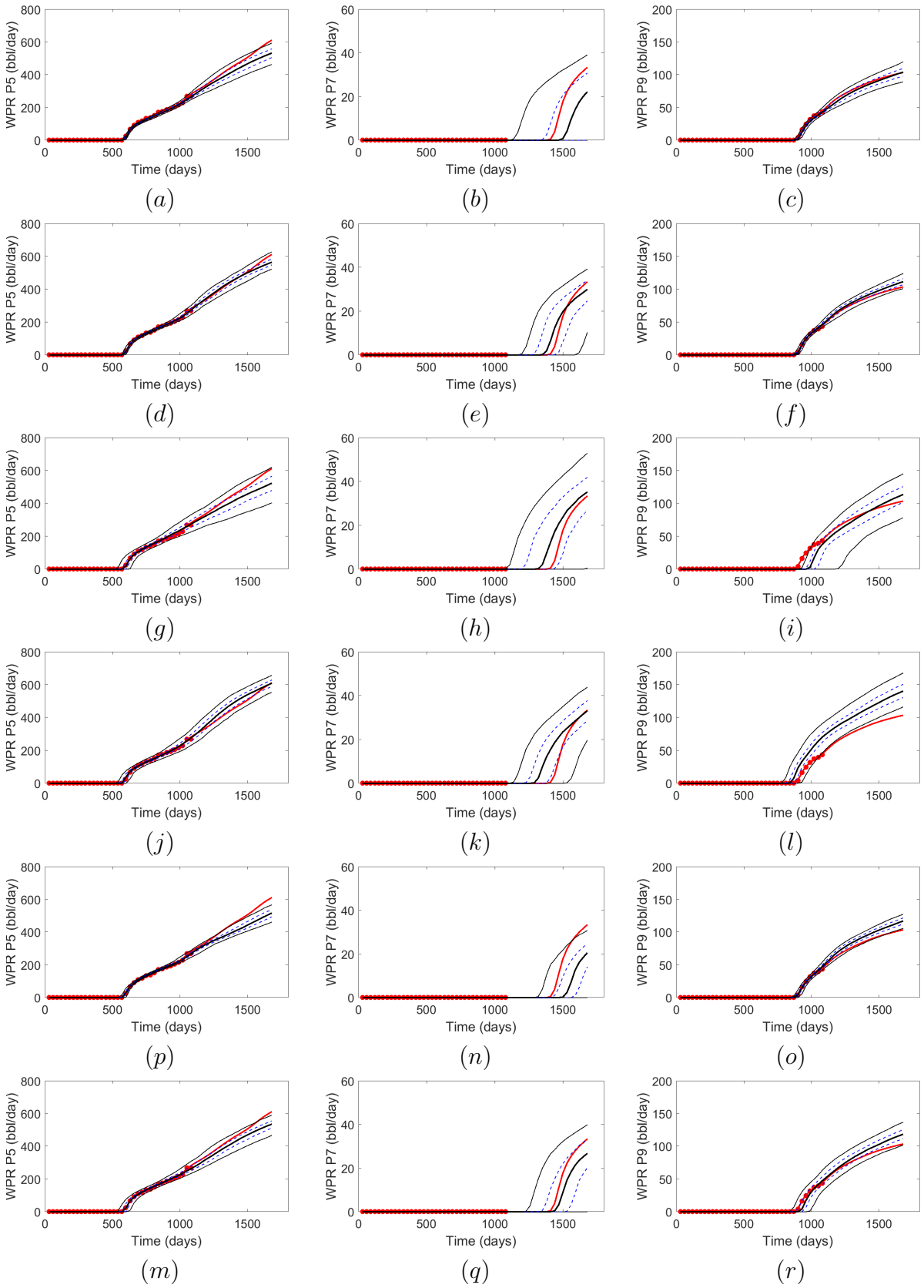


Figure 4.33: Plot of water production rate of wells P5, P7 and P9 for 2D model. (a)-(c) two-level MCMC of Li and Reynolds [56], (d)-(f) ES-MDA-RLM, (g)-(i) IR-ES, (j)-(l) M-IR-ES, (m)-(o) ES-MDA-EQL, and (p)-(r) ES-MDA-GEO. The legend is the same as in Fig. 4.29.

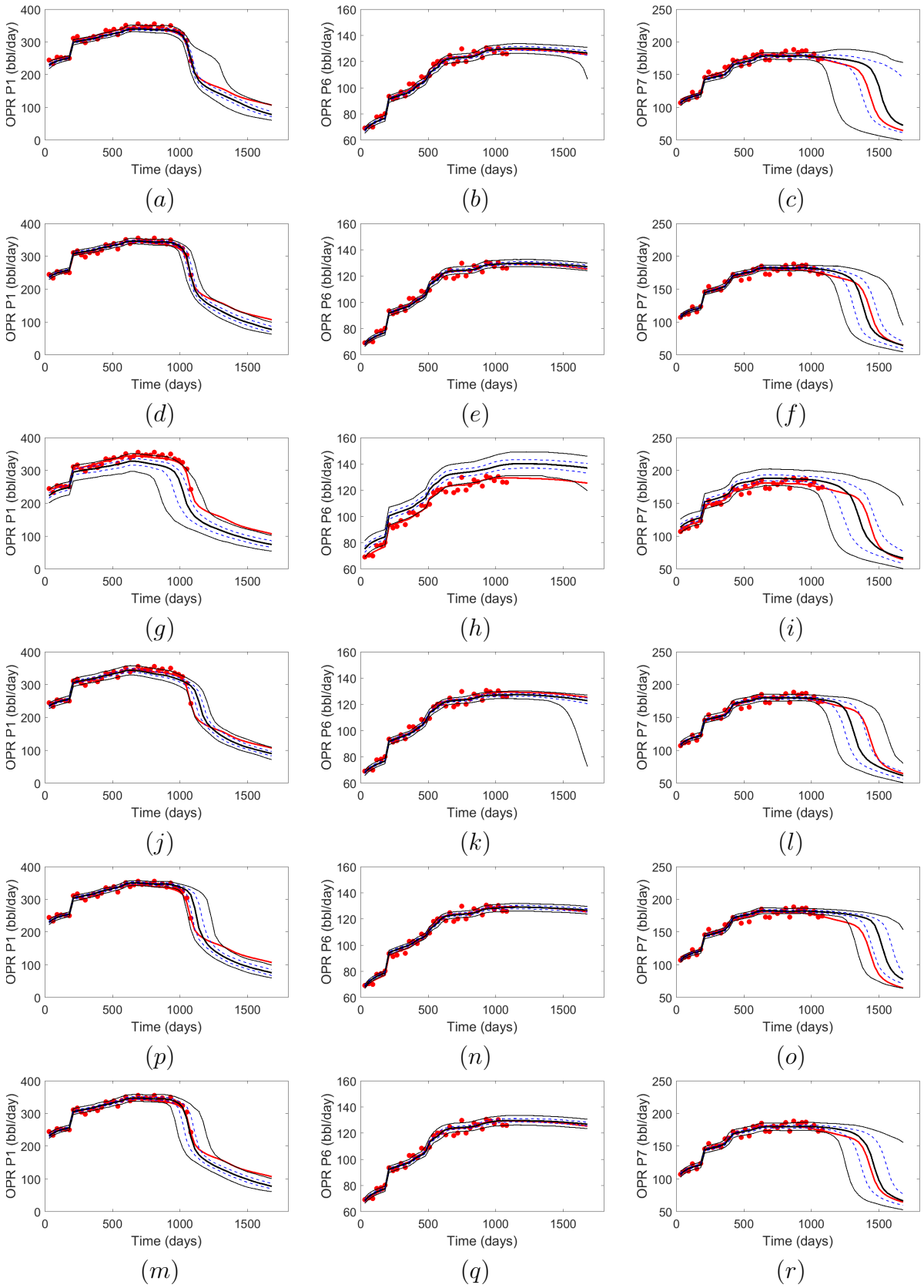


Figure 4.34: Plot of oil production rate of wells P1, P6 and P7 for 2D model. (a)-(c) two-level MCMC of Li and Reynolds [56], (d)-(f) ES-MDA-RLM, (g)-(i) IR-ES, (j)-(l) M-IR-ES, (m)-(o) ES-MDA-EQL, and (p)-(r) ES-MDA-GEO. The legend is the same as in Fig. 4.29.

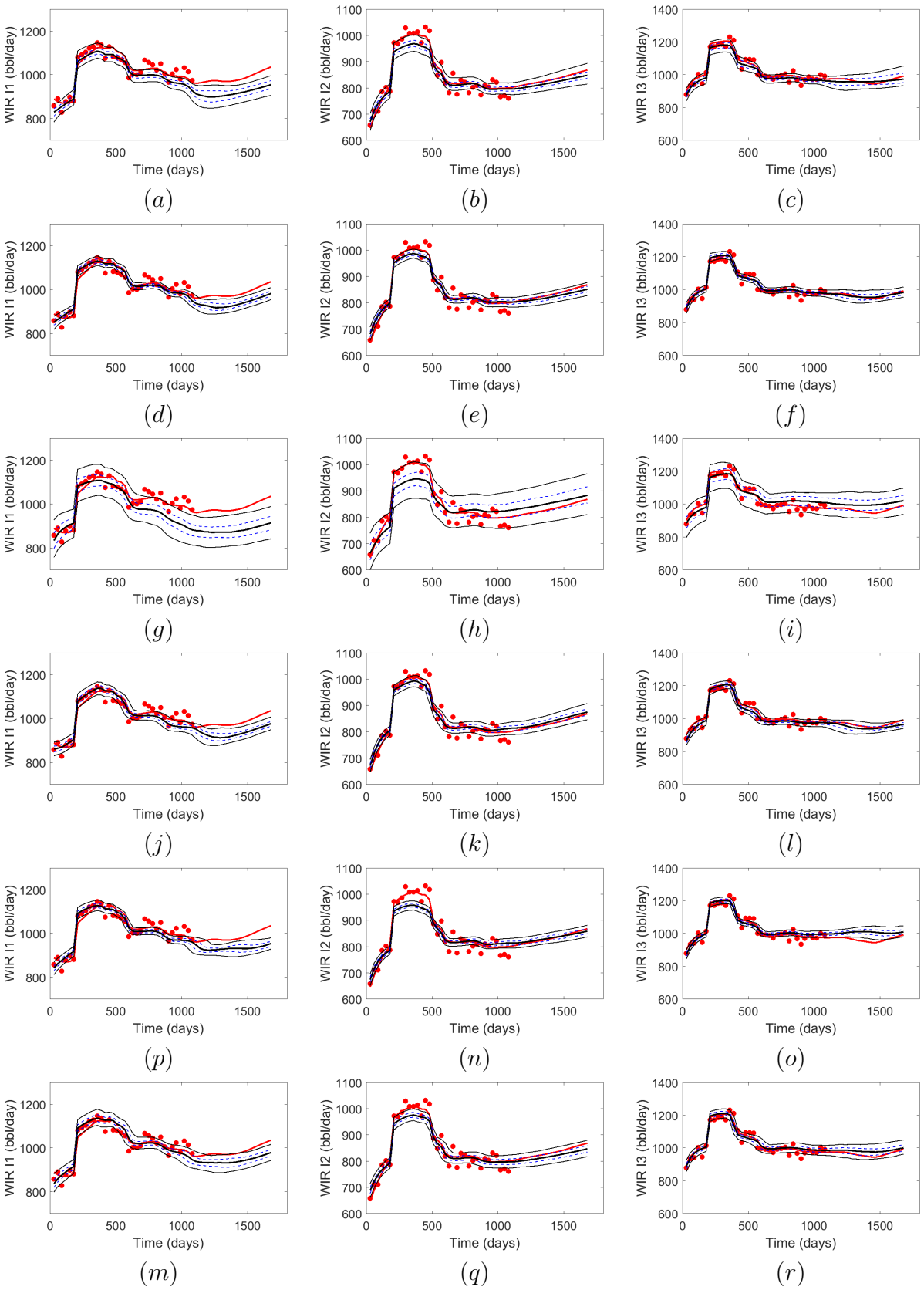


Figure 4.35: Plot of water injection rate of wells I1, I2 and I3 for 2D model. (a)-(c) two-level MCMC of Li and Reynolds [56], (d)-(f) ES-MDA-RLM, (g)-(i) IR-ES, (j)-(l) M-IR-ES, (m)-(o) ES-MDA-EQL, and (p)-(r) ES-MDA-GEO. The legend is the same as in Fig. 4.29.

Chapter 5

CONCLUSIONS

5.1 Generation of Inflation Factors for ES-MDA

The ES-MDA-GEO algorithm developed in this work is an efficient data assimilation method that allows the user to specify a priori the number of data assimilation steps that will be done based on computational resources while at the same time providing sufficient damping of changes in the realizations of the reservoir model at each iteration to control overshooting and undershooting that can lead to overly rough estimates of reservoir models. ES-MDA-GEO is more robust than using the original ES-MDA algorithm with equal inflation factors since ES-MDA-GEO generally generates better estimates of the true model preserves more variability (uncertainty) in the posterior ensemble of realizations of the reservoir model obtained by assimilating observed data. However, there exist examples where the performance of ES-MDA-GEO and ES-MDA with equal inflation factors generate very similar results, e.g., the Brugge example.

We presented an analytical expression that enables the exact calculation of the minimum inflation factor that satisfies the inequality derived from the discrepancy principle that is the basis of IR-ES. However, in order to enhance computational efficiency, we use a smaller than theoretically optimum value of this first inflation factor to generate the inflation factor at the first data assimilation step of ES-MDA-GEO. Overall, the resulting ES-MDA-GEO algorithm produces model estimates and estimates of posterior uncertainty that are somewhat similar to those generated with IR-ES, which strictly enforces the discrepancy principle, but ES-MDA is not only more computationally efficient but also allows the user to control the total computational effort by specifying the number of data assimilations to be done a priori. Based on the approximation of the inflation factors that satisfy the discrepancy principle,

we also developed a modified IR-ES algorithm that gives fairly similar estimates of the true model and the posterior uncertainty as IR-ES but is more computationally efficient than IR-ES.

We no longer recommend use of ES-MDA-RML due to the fact that this method tends to be very computationally inefficient than the other methods considered and may tend to underestimate the posterior variances.

With a choice of $\rho = 0.8$, which has been recommended previously, the iterative regularizing ensemble smoother method (IR-ES) may require such an extremely high number of iterations to obtain convergence that the method becomes computationally infeasible for practical problems. For one example provided, IR-ES failed to converge in 200 iterations. In this regard, the modified IR-ES algorithm we developed here is more robust.

A previous claim to the contrary, results presented in the Brugge case make it clear that at least in some cases, covariance or Kalman gain localization will generally be required with using IR-ES.

5.2 An Efficient MCMC for Uncertainty Quantification

We presented a computationally efficient two-level MCMC algorithm to characterize the posterior pdf which generates samples of the vector of model parameters conditional to the vectors of observed data. The main contribution of this work is that in the first step of the proposed two-level MCMC we do not require the availability of the adjoint solution to find multiple modes of the posterior pdf. Instead, we use the DGN method which approximates the sensitivity matrix using a simplex gradient approach and finds multiple modes of the posterior pdf in parallel. A Gaussian mixture model is then built which is used as the proposal distribution for the second step of our two-level MCMC scheme. Since the GMM expected to approximate the posterior pdf locally, it improves the efficiency of the MCMC scheme for sampling multimodal posteriors.

The computational results confirm that this proposed two-level MCMC method provides a correct sampling of the posterior pdf for the reservoir model parameters which is

comparable to the two-level MCMC algorithm which uses adjoint solution to find the modes while it requires far less computational effort than is required by random walk MCMC methods. For the 2D problem which has a relatively large number of parameters, the DGN is not able to decrease the objective function as low as it is possible with gradient-based method and therefore the two-level MCMC results show higher uncertainty in the predictions and model parameters than it is obtained with the two-level MCMC based on analytical gradient.

While the use of the constructed GMM as the proposal distribution promotes the performance of the MCMC scheme, we showed that generated samples from this GMM (without a rejection mechanism), which is claimed by some researchers, can not be used to characterize the posterior pdf in general.

Bibliography

[1] S. I. Aanonsen, 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.

[2] J. L. Anderson. Exploring the need for localization in ensemble data assimilation using a hierarchical ensemble filter. *Physica D: Nonlinear Phenomena*, 230(1):99–111, 2007.

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

[4] R. Barlow. *Statistics, A Guide to the Use of Statistical Methods in the Physical Sciences*. John Wiley & Sons, Chichester, England, 1989.

[5] T. Bengtsson, C. Snyder, and D. Nychka. Toward a nonlinear ensemble filter for high-dimensional systems. *Journal of Geophysical Research: Atmospheres*, 108(D24), 2003.

[6] Z. Bi. *Conditional 3D Stochastic Channels to Well-Test Pressure Data*. Ph.D. dissertation, The University of Tulsa, Tulsa, OK, 1999.

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

[8] S. P. Brooks and A. Gelman. General methods for monitoring convergence of iterative simulations. *Journal of computational and graphical statistics*, 7(4):434–455, 1998.

[9] G. Burgers, P. van Leeuwen, and G. Evensen. Analysis scheme in the ensemble Kalman filter. *Monthly Weather Review*, 126(6):1719–1724, 1998.

[10] B. Chen, J. He, X. Wen, W. Chen, and A. C. Reynolds. Pilot design analysis using proxies and Markov chain Monte Carlo method. In *ECMOR XV-15th European Conference on the Mathematics of Oil Recovery*, 2016.

[11] C. Chen, G. Gao, R. Li, R. Cao, T. Chen, J. C. Vink, P. Gelderblom, et al. Integration of distributed gauss-newton with randomized maximum likelihood method for uncertainty quantification of reservoir performance. In *SPE Reservoir Simulation Conference*. Society of Petroleum Engineers, 2017.

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

[13] Y. Chen and D. S. Oliver. Ensemble-based closed-loop optimization applied to Brugge field. *SPE Reservoir Evaluation & Engineering*, 13(1):56–71, 2010.

[14] Y. Chen 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.

[15] Y. Chen and D. S. Oliver. History matching of the Norne full-field model with an iterative ensemble smoother. *SPE Reservoir Evaluation & Engineering*, 17(02):244–256, 2014.

[16] D. Devegowda, E. Arroyo-Negrete, A. Datta-Gupta, and S. G. Douma. Efficient and robust reservoir model updating using ensemble Kalman filter with sensitivity-based covariance localization. In *Proceedings of the SPE Reservoir Simulation Symposium, Houston, Texas, 26–28 February (SPE 106144)*, 2007.

[17] F. Dickstein, P. Goldfeld, G. T. Pfeiffer, and R. V. Pinto. Truncated conjugate gradient and improved LBFGS and TSVD for history matching. *Computational Geosciences*, pages 1–19, 2017.

[18] L. Dovera and E. Della Rossa. Multimodal ensemble Kalman filtering using Gaussian mixture models. *Computational Geosciences*, 15(2):307–323, 2011.

[19] A. H. Elsheikh, M. F. Wheeler, and I. Hoteit. Clustered iterative stochastic ensemble method for multi-modal calibration of subsurface flow models. *Journal of Hydrology*, 491:40–55, 2013.

[20] A. A. Emerick. Analysis of the performance of ensemble-based assimilation of production and seismic data. *Journal of Petroleum Science and Engineering*, 139:219–239, 2016.

[21] A. A. Emerick and A. C. Reynolds. History matching a field case using the ensemble Kalman filter with covariance localization. In *Proccedings of the SPE Reservoir Simulation Symposium, The Woodlands, Texas, USA, 21–23 February (SPE 141216)*, 2011.

[22] A. A. Emerick 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.

[23] A. A. Emerick and A. C. Reynolds. Combining the ensemble Kalman filter with Markov chain Monte Carlo for improved history matching and uncertainty characterization. *SPE Journal*, 17(3):639–659, 2012.

[24] A. A. Emerick and A. C. Reynolds. Investigation of the sampling performance of ensemble-based methods with a simple reservoir model. *Computational Geosciences*, 17(2):325–350, 2013.

[25] A. A. Emerick and A. C. Reynolds. Ensemble smoother with multiple data assimilations. *Computers & Geosciences*, 55:3–15, 2013.

[26] A. A. Emerick and A. C. Reynolds. History-matching production and seismic data in a real field case using the ensemble smoother with multiple data assimilation. In

Proccedings of the SPE Reservoir Simulation Symposium, The Woodlands, Texas, USA, 18-20 February (SPE 163645), 2013.

[27] G. Evensen. Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *Journal of Geophysical Research*, 99 (C5):10143–10162, 1994.

[28] G. Evensen. *Data Assimilation: The Ensemble Kalman Filter*. Springer, Berlin, 2007.

[29] F. J. T. Floris, M. D. Bush, M. Cuypers, F. Roggero, and A.-R. Syversveen. Methods for quantifying the uncertainty of production forecasts: A comparative study. *Petroleum Geoscience*, 7(S):87–96, 2001.

[30] G. Gao and A. C. Reynolds. An improved implementation of the LBFGS algorithm for automatic history matching. *SPE Journal*, 11(1):5–17, 2006.

[31] G. Gao, 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.

[32] G. Gao, J. Vink, C. Chen, Y. El Khamra, and M. Tarrahi. Distributed Gauss-Newton method for history matching problems with multiple best matches. In *ECMOR XV-15th European Conference on the Mathematics of Oil Recovery*, 2016.

[33] G. Gao, J. C. Vink, C. Chen, M. Tarrahi, and Y. El Khamra. Uncertainty quantification for history matching problems with multiple best matches using a distributed Gauss-Newton method. In *SPE Annual Technical Conference and Exhibition*. Society of Petroleum Engineers, 2016.

[34] G. Gao, H. Jiang, P. V. Hagen, J. C. Vink, T. Wells, et al. A Gauss–Newton trust region solver for large scale history matching problems. In *SPE Reservoir Simulation Conference*. Society of Petroleum Engineers, 2017.

[35] G. Gao, J. C. Vink, C. Chen, Y. El Khamra, and M. Tarrahi. Distributed Gauss–Newton optimization method for history matching problems with multiple best matches. *Computational Geosciences*, May 2017.

[36] A. Gelman, G. O. Roberts, and W. R. Gilks. Efficient Metropolis jumping rules. *Bayesian Statistics. Oxford University Press*, 5(5):599–607, 1996.

[37] G. H. Golub and C. F. van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, 2nd edition, 1989.

[38] N. I. Gould, D. P. Robinson, and H. S. Thorne. On solving trust-region and other regularised subproblems in optimization. *Mathematical Programming Computation*, 2 (1):21–57, 2010.

[39] C. W. Groetsch. *The theory of Tikhonov regularization for Fredholm equations of the first kind*. 1984.

[40] H. Haario, E. Saksman, J. Tamminen, et al. An adaptive metropolis algorithm. *Bernoulli*, 7(2):223–242, 2001.

[41] M. Hanke. A regularizing Levenberg-Marquardt scheme, with applications to inverse groundwater filtration problems. *Inverse problems*, 13(1):79–95, 1997.

[42] M. Hanke. The regularizing Levenberg-Marquardt scheme is of optimal order. *J. Integral Equations Applications*, 22(2):259–283, 2010.

[43] N. Hansen. The CMA evolution strategy: A tutorial. *arXiv preprint arXiv:1604.00772*, 2016.

[44] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[45] V. Haugen, G. Nævdal, L.-J. Natvik, G. Evensen, A. M. Berg, and K. M. Flornes. History matching using the ensemble Kalman filter on a North Sea field case. *SPE Journal*, 13(4):382–391, 2008.

[46] J. He, J. Xie, P. Sarma, X.-H. Wen, W. H. Chen, and J. Kamath. Proxy-based work flow for a priori evaluation of data-acquisition programs. *SPE Journal*, 2016.

[47] L. Holden, R. Hauge, and M. Holden. Adaptive independent metropolis-hastings. *The Annals of Applied Probability*, 19(1):395–413, 2009.

[48] M. A. Iglesias. Iterative regularization for ensemble data assimilation in reservoir models. *Computational Geosciences*, 19(1):177–212, 2015.

[49] M. A. Iglesias and C. Dawson. The regularizing Levenberg–Marquardt scheme for history matching of petroleum reservoirs. *Computational geosciences*, 17(6):1033–1053, 2013.

[50] L. Kaufman and P. J. Rousseeuw. *Finding groups in data: an introduction to cluster analysis*, volume 344. John Wiley & Sons, Hoboken, New Jersey, 2009.

[51] P. K. Kitanidis. Quasi-linear geostatistical theory for inversing. *Water Resources Research*, 31(10):2411–2419, 1995.

[52] D. H. Le, A. A. Emerick, and A. C. Reynolds. An adaptive ensemble smoother with multiple data assimilation for assisted history matching. *SPE Journal*, 21(6):2195–2207, 2015.

[53] G. Li and A. C. Reynolds. Iterative ensemble Kalman filters for data assimilation. *SPE Journal*, 14(3):496–505, 2009.

[54] R. Li, A. C. Reynolds, and D. S. Oliver. Sensitivity coefficients for three-phase flow history matching. *J. Canadian Pet. Tech.*, 42(4):70–77, 2003.

[55] X. Li. *Efficient MCMC Characterization of Uncertainty in Reservoir Description and Production Forrecasts*. Ph.D. dissertation, The University of Tulsa, Tulsa, OK, 2017.

[56] X. Li and A. C. Reynolds. Generation of a proposal distribution for efficient MCMC characterization of uncertainty in reservoir description and forecasting. In *SPE Reservoir Simulation Conference*. Society of Petroleum Engineers, 2017.

[57] F. Liang and W. H. Wong. Evolutionary Monte Carlo: Applications to C_p model sampling and change point problem. *Statistica sinica*, 14(3):317–342, 2000.

[58] F. Liang, C. Liu, and R. Carroll. *Advanced Markov chain Monte Carlo methods: learning from past samples*, volume 714. John Wiley & Sons, 2011.

[59] N. Liu and D. S. Oliver. Evaluation of Monte Carlo methods for assessing uncertainty. *SPE Journal*, 8(2):188–195, 2003.

[60] X. Luo, A. S. Stordal, R. J. Lorentzen, and G. Nævdal. Iterative ensemble smoother as an approximate solution to a regularized minimum-average-cost problem: Theory and applications. *SPE Journal*, 20(5):962–982, 2015.

[61] X. Ma, M. Al-Harbi, A. Datta-Gupta, and Y. Efendiev. An efficient two-stage sampling method for uncertainty quantification in history matching geological models. *SPE Journal*, 13(01):77–87, 2008.

[62] N. Metropolis, 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.

[63] L. Mohamed, B. Calderhead, M. Filippone, M. Christie, and M. Girolami. Population MCMC methods for history matching and uncertainty quantification. *Computational Geosciences*, 16(2):423–436, 2012.

[64] J. J. Morê and D. C. Sorensen. Computing a trust region step. *SIAM Journal on Scientific and Statistical Computing*, 4(3):553–572, 1983.

[65] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, New York, 1999.

[66] D. P. OLeary and J. A. Simmons. A bidiagonalization–regularization procedure for large scale discretizations of ill-posed problems. *SIAM Journal on Scientific and Statistical Computing*, 2(4):474–489, 1981.

[67] D. S. Oliver. On conditional simulation to inaccurate data. *Mathematical Geology*, 28 (6):811–817, 1996.

[68] D. S. Oliver. Metropolized randomized maximum likelihood for improved sampling from multimodal distributions. *SIAM/ASA Journal on Uncertainty Quantification*, 5 (1):259–277, 2017.

[69] D. S. Oliver, N. He, and A. C. Reynolds. Conditioning permeability fields to pressure data. In *ECMOR V-5th European Conference on the Mathematics of Oil Recovery*, 1996.

[70] D. S. Oliver, L. B. Cunha, and A. C. Reynolds. Markov chain Monte Carlo methods for conditioning a permeability field to pressure data. *Mathematical Geology*, 29(1):61–91, 1997.

[71] D. S. Oliver, A. C. Reynolds, and N. Liu. *Inverse Theory for Petroleum Reservoir Characterization and History Matching*. Cambridge University Press, Cambridge, UK, 2008.

[72] E. Peters, Y. Chen, O. Leeuwenburgh, and D. S. Oliver. Extended Brugge benchmark case for history matching and water flooding optimization. *Computers & Geosciences*, 50:16–24, 2013.

[73] L. Peters, R. J. Arts, G. K. Brouwer, C. R. Geel, S. Cullick, R. J. Lorentzen, Y. Chen, K. Dunlop, F. C. Vossepoel, R. Xu, P. Sarma, A. H. Alhuthali, and A. C. Reynolds. Results of the Brugge benchmark study for flooding optimisation and history matching. *SPE Reservoir Evaluation & Engineering*, 13(3):391–405, 2010.

[74] A. C. Reynolds, N. He, and D. S. Oliver. Memoir 71, chapter 10: Reducing uncertainty in geostatistical description with well-testing pressure data. 1999.

[75] A. C. Reynolds, M. Zafari, and G. Li. Iterative forms of the ensemble Kalman fil-

ter. In *Proceedings of 10th European Conference on the Mathematics of Oil Recovery, Amsterdam, 4–7 September 2006*, 2006.

[76] G. O. Roberts and J. S. Rosenthal. Optimal scaling for various Metropolis-Hastings algorithms. *Statistical Science*, 16:351–367, 2001.

[77] J. Rommelse. *Data Assimilation in Reservoir Management*. Ph.D. dissertation, Technical University of Delft, Delft, The Netherlands, 2009.

[78] A. Seiler, S. I. Aanonsen, G. Evensen, and J. C. Reivenæs. Structural surface uncertainty modeling and updating using the ensemble Kalman filter. *SPE Journal*, 15(4):1062–1076, 2010.

[79] M. G. Shirangi. History matching production data with truncated SVD parameterization. Masters thesis, The University of Tulsa, Tulsa, OK, 2011.

[80] M. G. Shirangi. History matching production data and uncertainty assessment with an efficient TSVD parameterization algorithm. *Journal of Petroleum Science and Engineering*, 113:54–71, 2014.

[81] M. G. Shirangi and A. A. Emerick. An improved TSVD-based Levenberg–Marquardt algorithm for history matching and comparison with Gauss–Newton. *Journal of Petroleum Science and Engineering*, 143:258–271, 2016.

[82] J. A. Skjervheim, G. Evensen, S. I. Aanonsen, B. O. Ruud, and T. A. Johansen. Incorporating 4D seismic data in reservoir simulation models using ensemble Kalman filter. *SPE Journal*, 12(3):282–292, 2007.

[83] K. W. Smith. Cluster ensemble Kalman filter. *Tellus A*, 59(5):749–757, 2007.

[84] D. C. Sorensen. Newtons method with a model trust region modification. *SIAM Journal on Numerical Analysis*, 19(2):409–426, 1982.

[85] A. S. Stordal and G. Nævdal. A modified randomized maximum likelihood for improved Bayesian history matching. *Computational Geosciences*, pages 1–13, 2017.

[86] A. S. Stordal, H. A. Karlsen, G. Nævdal, H. J. Skaug, and B. Vallès. Bridging the ensemble Kalman filter and particle filters: the adaptive Gaussian mixture filter. *Computational Geosciences*, 15(2):293–305, 2011.

[87] A. Tarantola. *Inverse Problem Theory and Methods for Model Parameter Estimation*. SIAM, 2005.

[88] R. Tavakoli and A. C. Reynolds. History matching with parameterization based on the SVD of a dimensionless sensitivity matrix. *SPE Journal*, 15(12):495–508, 2010.

[89] R. Tavakoli 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.

[90] K. Thulin, 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.

[91] L. Tierney. Markov chains for exploring posterior distributions. *the Annals of Statistics*, 22(4):1701–1728, 1994.

[92] P. J. van Leeuwen. Comment on "Data assimilation using an ensemble Kalman filter technique". *Monthly Weather Review*, 127(6):1374–1377, 1999.

[93] P. J. van Leeuwen and G. Evensen. Data assimilation and inverse methods in terms of a probabilistic formulation. *Monthly Weather Review*, 124:2898–2913, 1996.

[94] C. R. Vogel. *Computational Methods for Inverse Problems*. Society for Industrial & Applied Mathematics (SIAM), Jan. 2002.

[95] C. R. Vogel and J. G. Wade. Iterative SVD-based methods for ill-posed problems. *SIAM Journal on Scientific Computing*, 15(3):736–754, 1994.

[96] Y. Wang, 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.

[97] M. Zafari and A. C. Reynolds. Assessing the uncertainty in reservoir description and performance predictions with the ensemble Kalman filter. *SPE Journal*, 12(3):382–391, 2007.

[98] F. Zhang, A. C. Reynolds, and D. S. Oliver. Evaluation of the reduction in uncertainty obtained by conditioning a 3D stochastic channel to multiwell pressure data. *Mathematical Geology*, 34(6):713–740, 2002.

[99] Y. Zhang and D. S. Oliver. Evaluation and error analysis: Kalman gain regularization versus covariance regularization. *Computational Geosciences*, 15(3):489–508, 2011.