THE UNIVERSITY OF TULSA

THE GRADUATE SCHOOL

HISTORY MATCHING PRODUCTION DATA WITH TRUNCATED SVD PARAMETERIZATION

by
Mehrdad Gharib Shirangi

A thesis submitted in partial fulfillment of
the requirements for the degree of Master of Science
in the Discipline of Petroleum Engineering

The Graduate School
The University of Tulsa

2011

THE UNIVERSITY OF TULSA

THE GRADUATE SCHOOL

HISTORY MATCHING PRODUCTION DATA WITH TRUNCATED SVD PARAMETERIZATION

by
Mehrdad Gharib Shirangi

A THESIS

APPROVED FOR THE DISCIPLINE OF

PETROLEUM ENGINEERING

By Thesis Committee

______________________________, Chairperson
Albert C. Reynolds

____
Gaoming Li

____
Peyton Cook

ABSTRACT

Mehrdad Gharib Shirangi (Master of Science in Petroleum Engineering)

History Matching Production Data With Truncated SVD Parameterization

Directed by Albert C. Reynolds

206 pp., Chapter 6: Conclusions

(299 words)

For large scale history matching problems, gradient based algorithms, including the Gauss-Newton and the Levenberg-Marquardt algorithms, require forming the sensitivity matrix which can be computationally very expensive. Parameterization based on the SVD of a dimensionless sensitivity matrix has been discussed by some authors to be a computationally efficient gradient based method of history matching production and, more recently, seismic data. In SVD parameterization algorithms, only a truncated SVD of a dimensionless sensitivity matrix is obtained with the Lanczos method. The SVD parameterization method has been previously applied to generate realizations of horizontal log permeability fields; we apply the method to simulate other rock property fields, i.e., porosity and vertical log permeability fields. This is one step towards the application of the method to large scale history matching problems. In this work, a modified SVD parameterization algorithm is developed which is computationally more efficient than the previous algorithms introduced in this framework. The method is applied to several 2D and 3D synthetic reservoirs for simulation of porosity fields as well as horizontal and vertical log permeability fields.

We also compare the Gauss-Newton and Levenberg-Marquardt algorithms. In this work, we show that the LM algorithm generates a good estimate of the model not only because of damping the change of model parameters, but also because the LM

search direction at early iterations has almost negligible components in the direction of eigenvectors associated with the small eigenvalues of the Hessian matrix. On the other side, at early iterations of the Gauss-Newton algorithm, the search direction vector may have large components in the direction of eigenvectors associated with the small eigenvalues, and thus it adds roughness to the model.

ACKNOWLEDGEMENTS

I would like to express my sincere appreciation to my advisor, Dr. Albert C. Reynolds, Jr., Professor of Petroleum Engineering and Mathematics. I thank Dr. Reynolds for his incredible patience, for his support and encouragement and for all the lessons I learned from him during the past two years. I extend my special thanks to Dr. Gaoming Li, associate director of TUPREP; and to Dr. Peyton Cook, Professor of Mathematical Science of The University of Tulsa, for serving on my thesis committee, and for offering their helpful suggestions and comments.

I would like to extend my thanks to all the other faculty and staff of The University of Tulsa, especially to Dr. Leslie Thompson, Dr. Cem Sarica, Dr. Christian Constanda, Judy Teal, and to Loreta Watkins.

I specially thank my best friends at TU, Alex Emeric and Dr. Reza Tavakoli, TUPREP PhD students, who were always willing to help and discuss various topics in history matching and closed loop reservoir management, for their special support and guidance. I also thank the other TUPREP students who helped me in different situations during these two years: Mei Han, Sy Do, Emilio Coutinho, Emil Nurmammadov, Diego Oliveira, hui zhao and Chaohui Chen.

I would like to thank my friend Andrew Rypkema, for his help in many situations during the period I lived in Tulsa. I also thank my dear uncle, Mohammad Hoseini, for his encouragement, helps and support during my study. This work is dedicated to my mother, Gohar Hoseini and to my grandfather, Khalil Hoseini for their endless love.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Page

CHAPTER 1

INTRODUCTION

1.1 History Matching as an Inverse Problem

In forward problems, the evolutionary state of a system is predicted from some predictive model, given auxiliary conditions and physical properties. Physical properties are referred to as model parameters. One example of a forward problem is to predict pressures and saturations of gridblocks and well rates of a reservoir versus time, given the initial state and the geometry of the reservoir, the rock property fields and fluid properties.

In an inverse problem, given a finite number of observed data which are functions of state variables, one aims to infer information about model parameters. Observed data contain measurement error.

History matching is a discrete inverse problem which is characterized by a finite number of model parameters. The history matching process consists of estimating reservoir properties through matching predicted data to reservoir production history (observed data).

It is well-known that large scale inverse problems are usually ill-posed as there are conceptually an infinite number of models that match the data.

Here, the non-uniqueness property of solution to history matching problems is discussed. A history matching problem does not have a unique solution; there exist infinite number of solutions that match the observed data equally well. Thus the solution to a history matching problem, which is an inverse problem, is not a single model. If one is to pick a single model as solution, there should be a logic to selecting it. One approach to inverse problems is based on Bayesian statistics [28]. In this context a probability distribution is defined for the solution of an inverse problem. The full solution of an inverse problem is the a posteriori probability density function (pdf) for the model variables [28].

1.2 A Priori and A Posteriori Probability Density Functions

In this study, the N_m-dimensional vector of model parameters is denoted by m and it can include horizontal and vertical log permeability and porosity of gridblocks. Permeability fields are assumed to have a log-normal distribution, while porosity fields are assumed to have normal distribution.

In reservoir characterization, probability density functions are usually assumed to have a Gaussian distribution. Gaussian distributions are convenient to deal with in multidimensional space, as they can be fully described by the mean of the random variable and its covariance.

The prior uncertainty in the model parameters is described by a pdf. If the prior pdf has a Gaussian distribution, it can be fully described by its mean and the covariance. To show a Gaussian prior distribution, with m denoting the random vector of model parameters, the following notation is used:

$$m \sim N(m_{\text{prior}}, C_M), \tag{1.1}$$

which means the random vector m has a normal (Gaussian) distribution with mean m_{prior} and covariance matrix C_M. The prior pdf is given by

$$f(m) = a_1 \exp\big[-\frac{1}{2}(m - m_{\text{prior}})^T C_M^{-1}(m - m_{\text{prior}})\big], \tag{1.2}$$

where a_1 is the normalizing constant. The N_d-dimensional column vector of predicted data d is related to the vector of model parameters m by

$$d = g(m). \tag{1.3}$$

If m is the vector of true model parameters, then d is referred to as true data. However, what is available is observed data which is the true data with measurement error added to it. The difference between true data, d_{true}, and the corresponding vector of observed data, d_{obs}, represents measurement error ϵ^d, i.e.,

$$\epsilon^d = d_{\text{true}} - d_{\text{obs}}. \tag{1.4}$$

It is usually reasonable to assume that measurement error has a Gaussian distribution with mean zero and an $N_d \times N_d$ covariance matrix C_D. In other words, observed data is a random vector that has a Gaussian distribution with mean of $d_{\text{true}} = g(m_{\text{true}})$ and covariance matrix C_D. However, in reality m_{true} is unknown; which means that the mean of this pdf, $g(m_{\text{true}})$, is unknown, but a sample of this distribution which is d_{obs} is available. For this reason the Gaussian pdf is expressed with a known sample and unknown mean, $g(m)$:

$$f(d_{\text{obs}}|m) = f(\epsilon^d) = a_2 \exp\big[-\frac{1}{2}(d_{\text{obs}} - g(m))^T C_D^{-1}(d_{\text{obs}} - g(m))\big], \tag{1.5}$$

where a_2 is the normalizing constant. This pdf characterizes the uncertainty in observed data, d_{obs}, given the model parameters, m. As d_{obs} is given, Eq. 1.5 gives the likelihood of m given d_{obs} denoted by $L(m|d_{\text{obs}})$; the value of $L(m|d_{\text{obs}})$ is greater if the predicted data corresponding to the model m is closer to d_{obs}. According to the Bayes Theorem [35, 28], the posterior pdf of the model parameters conditional to the observed data is proportional to the product of the prior pdf and the likelihood function for the model parameters:

$$f(m|d_{\text{obs}}) \propto f(m)L(m|d_{\text{obs}}). \tag{1.6}$$

Using Eqs. 1.2 and 1.5 in Eq. 1.6, the posterior pdf of model parameters conditioned to observed data can be written as

$$f(m|d_{\text{obs}}) = a\exp(-O(m)), \tag{1.7}$$

where a is the normalizing constant and $O(m)$ which is referred to as the total objective function, is

$$\begin{aligned} O(m) = \frac{1}{2}(m - m_{\text{prior}})^T C_M^{-1}(m - m_{\text{prior}}) + \frac{1}{2}(g(m) - d_{\text{obs}})^T C_D^{-1}(g(m) - d_{\text{obs}}) \\ = O_m(m) + O_d(m). \quad (1.8) \end{aligned}$$

The total objective function has two parts, the model mismatch part, O_m, and the data mismatch term, O_d. The model mismatch part, which comes from the prior pdf, provides regularization to avoid unrealistic changes in model parameters.

To sample the posterior pdf and find a highly probable model, $f(m|d_{\text{obs}})$ should be relatively high, which means that the objective function in Eq. 1.8 should be small. Thus to find a probable model, an optimization problem should be solved.

1.3 Sampling the a Posteriori PDF

As mentioned before, the full solution of an inverse problem is the a posteriori probability density function for the model variables [28]. The maximum a posteriori estimate (MAP), is the one that maximizes Eq. 1.7 or equivalently minimizes Eq. 1.8. The MAP estimate represents the mean of the posteriori distribution in the Gaussian case and thus is smooth compared to a sample (realization) from the posterior pdf; the MAP does not reflect the full heterogeneity in the model. In order to perform uncertainty analysis on the geological description and performance predictions, one needs to generate a suite of realizations from the a posteriori pdf.

For large scale history matching problems, full characterization of the a posteriori pdf is impractical. Instead, an approximate sample of the a posteriori pdf is obtained with proposed methods, to have an approximate solution of the inverse problem. Sampling the a posteriori pdf is equivalent to constructing a suite of realizations which is referred to as simulation. Constructing a particular estimate of the model is referred to as estimation. In this thesis, emphasis is on estimation and simulation of permeability and porosity fields.

There are two types of sampling methods, those which are known to sample correctly and those that are only approximately correct. Theoretically rigorous methods of sampling such as the Markov chain Monte Carlo (MCMC) method or the rejection algorithm are computationally expensive [23].

Approximate sampling methods include linearization around the MAP (LMAP) and Randomized Maximum Likelihood (RML) method. Linearization around the MAP has been applied to the problem of generating realizations conditional to production data by several authors [6, 26]. The advantage of linearization around the MAP is its low computational cost, which requires minimizing only one objective function. However, in highly non-Gaussian cases the method fails to give an acceptable sample. Liu and Oliver [23] found that for a nonlinear problem, the realizations generated with the LMAP method, did not even approximately honor the data. RML introduced by Oliver et al. [27], is proven to sample correctly in the linear case, i.e, when data are linearly related to the model parameters. Although there is no rigorous theoretical foundation that RML samples correctly when data is not linearly related to the model parameters, several computational experiments have shown that the method performs an adequate sampling in the nonlinear case [23, 31, 14].

The work by Liu and Oliver [23] indicates that RML gives a reasonable characterization of uncertainty compared to MCMC. In this work we mainly use the RML method.

1.3.1 Randomized Maximum Likelihood Method

Randomized Maximum Likelihood(RML), introduced by Oliver et al. [27], is a Monte Carlo method to sample the a posteriori pdf. In this method, Eq. 1.8 is modified such that, m_{prior} is replaced by an unconditional simulation, m_{uc}, from the prior model and d_{obs} is replaced by a sample d_{uc}, from the Gaussian distribution with mean d_{obs} and covariance matrix C_D. A realization from the Gaussian prior pdf is generated from

$$m_{\text{uc}} = m_{\text{prior}} + C_M^{1/2} z_m, \tag{1.9}$$

where z_m is an N_m-dimensional column vector of random normal deviates. The matrix $C_M^{1/2}$ is a square root of C_M, e.g., its Cholesky decomposition. Similarly a realization from $N(d_{\text{obs}}, C_D)$ is generated from

$$d_{\text{uc}} = d_{\text{obs}} + C_D^{1/2} z_d, \tag{1.10}$$

where z_d is an N_d-dimensional column vector of random normal deviates. The matrix $C_D^{1/2}$ is a square root of C_D. Finally the following equation is minimized to generate a sample of the a posteriori pdf.

$$O_j(m) = \frac{1}{2}(m - m_{\text{uc,j}})^T C_M^{-1}(m - m_{\text{uc,j}}) + \frac{1}{2}(g(m) - d_{\text{uc,j}})^T C_D^{-1}(g(m) - d_{\text{uc,j}}). \tag{1.11}$$

In order to generate N_e samples of the a posteriori pdf, N_e simulations of d_{obs} and N_e simulations of m_{prior} are generated and Eq. 1.11 is minimized for $j = 1, \cdots N_e$. Based on the results discussed in Oliver et al. [28], if m_c is the model obtained at convergence, we expect $O_N(m_c)$, the normalized objective function evaluated at m_c, satisfy

$$1 - 5\sqrt{2/N_d} \le O_N(m_c) \le 1 + 5\sqrt{2/N_d}\,. \tag{1.12}$$

When generating a MAP estimate, the normalized objective function is defined by

$$O_N(m) = 2\frac{O(m)}{N_d}, \tag{1.13}$$

and in generating a conditional realization with RML, the normalized objective function is defined by

$$O_{N,j}(m) = \frac{O_j(m)}{N_d}. \tag{1.14}$$

This result will be used to provide an indication of the quality of data match.

1.4 The Sensitivity Matrix

One way to minimize the objective function of Eq. 1.8 is using the gradient based algorithms, in particular the Gauss-Newton method. In the Gauss-Newton method of minimizing the objective function $O(m)$, given by Eq. 1.8, the search direction δm^{l+1} at the lth iteration, is generated from

$$H_l \delta m^{l+1} = -\nabla O^l, \tag{1.15}$$

where ∇O^l is the gradient of $O(m)$ evaluate at m^l, given by

$$\nabla O^l = C_M^{-1}(m^l - m_{\text{prior}}) + G_l^T \{C_D^{-1}(g(m^l) - d_{\text{obs}})\}, \tag{1.16}$$

and H_l is the GN-Hessian matrix, given by

$$H_l = C_M^{-1} + G_l^T C_D^{-1} G_l. \tag{1.17}$$

G is the matrix of total derivatives of the predicted data, g_i, $i = 1, 2, \ldots, N_d$, with respect to the model parameters, which is called the sensitivity matrix and is defined as

$$G = \Big[G_{i,j}\Big] = \Big[\frac{\partial g_i}{\partial m_j}\Big], \tag{1.18}$$

for $i = 1, 2, \ldots, N_d$ and $j = 1, 2, \ldots, N_m$. Thus G is an $N_d \times N_m$ matrix. Here and in the rest of this thesis, G denotes the sensitivity matrix, given by

$$G = \begin{bmatrix} \frac{\partial g_1}{\partial m_1} & \frac{\partial g_1}{\partial m_2} & \cdots & \frac{\partial g_1}{\partial m_{N_m}} \\ \frac{\partial g_2}{\partial m_1} & \frac{\partial g_2}{\partial m_2} & \cdots & \frac{\partial g_2}{\partial m_{N_m}} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial g_{N_d}}{\partial m_1} & \frac{\partial g_{N_d}}{\partial m_2} & \cdots & \frac{\partial g_{N_d}}{\partial m_{N_m}} \end{bmatrix}, \tag{1.19}$$

where the (i, j) entry of the matrix G, denoted as $\frac{\partial g_i}{\partial m_j}$, is the total derivative of the ith predicted data with respect to the jth model parameter.

The product of G times an arbitrary N_d-dimensional vector v, which is a linear combination of the columns of G, can be computed with the “gradient simulator method” [4]. The product of G^T times an arbitrary N_m-dimensional vector u, which is a linear combination of the rows of G, can be computed with the “adjoint method” [22, 43]. These methods are discussed in Oliver et al. [28].

The whole sensitivity matrix, G, can be formed by either N_m applications of the gradient simulator method or N_d adjoint solutions. If both N_m and N_d are very large, forming the matrix G can be computationally very expensive; because of this fact, a direct application of the GN algorithm to large scale problems is not efficient.

For the nonlinear conjugate gradient method [24, 7, 20], and for the quasi-Newton methods, explicit computation of the full sensitivity matrix is not necessary. Both algorithms use the gradient information, and to apply them, only the product of G times a vector and/or the product of G^T times a vector are required. Among the quasi-Newton methods, the limited memory Broyden-Fletcher-Goldfarb-Shanno (LBFGS) method has been successfully applied to history matching problems [43, 13].

1.5 Generating Rock Property Fields

In the examples of this thesis, we generate the synthetic rock property fields by first generating a covariance matrix. Different types of rock property fields, i.e., porosity, horizontal and vertical permeability, are not independent but they are correlated. This is because, at a low permeability region of the reservoir, we usually do not expect high porosity, and vice versa.

The vector of model parameters, m, can be shown with:

$$m = \begin{pmatrix} [\ln(k_h)]^T & [\ln(k_z)]^T & [\phi]^T \end{pmatrix}^T, \tag{1.20}$$

where $\ln(k_h)$ denotes the vector of horizontal log permeability of all gridblocks, $\ln(k_z)$ denotes the vector of vertical log permeability of all gridblocks and ϕ denotes the vector of porosity of all gridblocks.

The covariance matrix for the vector of m can be written as

$$C_M = \begin{pmatrix} C_{\ln k_h} & C_{\ln k_h, \ln k_z} & C_{\ln k_h, \phi} \\ C_{\ln k_z, \ln k_h} & C_{\ln k_z} & C_{\ln k_z, \phi} \\ C_{\phi, \ln k_h} & C_{\phi, \ln k_z} & C_{\phi} \end{pmatrix}, \tag{1.21}$$

where $C_{\ln k_h}$ denotes the covariance matrix for the vector $\ln(k_h)$, and $C_{\ln k_z, \ln k_h}$ denotes the cross covariance between the vectors $\ln(k_h)$ and $\ln(k_z)$. Other vectors are defined the same way. A prior realization of rock property fields is generated with Eq. 1.9 in Subsection 1.3.1.

For details and types of covariance matrices, see Chapter 5 of Oliver et al. [28]. In this thesis, we avoid specific modeling of the cross covariance matrices by using the "screening hypothesis" of Xu et al. [41]. As explained in Reynolds et al. [29], the screening hypothesis implies that

$$C_{\phi, \ln k_h} = C_{\ln k_h, \phi} = \frac{\rho_{\phi, \ln k_h} \sigma_{\phi}}{\sigma_{\ln(k_h)}} C_{\ln k_h}, \tag{1.22}$$

where $\sigma_{\ln(k_h)}$ is the standard deviation of horizontal log permeability, and $\rho_{\phi, \ln k_h}$ is the correlation coefficient between porosity and log permeability at a common location. The cross correlation matrices for other pairs of rock property fields can be defined similar to Eq. 1.22.

This screening hypothesis eliminates the need to develop a model for the cross-variogram between porosity and permeability.

1.6 Parameterization Algorithms

In real field history matching problems, the number of model parameters to be adjusted is very large; besides that, the number of data can be small and the data may be available only at few locations. The data may have very small or even no sensitivity to some of the model parameters. Using the data to resolve the properties of all parameters individually, is misleading. Instead of finding the individual properties of the gridblocks, one should try to resolve the important geological features of the rock property fields. In geological descriptions based on Gaussian models, there is connectivity between rock properties of neighboring gridblocks.

In history matching problems, the number of model parameters can be very large. Reducing the number of model parameters to be estimated, can significantly reduce the computational costs, and may also introduce a regularization. Many parameterization algorithms are developed to reduce the number of parameters to be estimated.

One of the first parameterization methods applied to history matching problems, is the method of zonation. In this method, in order to reduce the number of model parameters, the reservoir is divided into a small number of zones. In each of the zones the properties are assumed to be uniform. A modeling error is thus introduced through the assumption of uniform properties within each zone and by assigning the boundaries of these zones more or less arbitrarily [16]. Jacquard and Jain [18] and Jahns [19] used the zonation method to estimate to match pressure data in synthetic one dimensional and two dimensional single phase reservoirs.

The work by Gavalas et al. [16] shows that by using a priori statistical information on the unknown parameters, the problem becomes better determined. They showed that the results of Bayesian estimation, where an objective function containing the prior information of the rock properties, is minimized by an optimization algorithm, is more accurate than those of zonation.

Shah et al. [34] proposed a parameterization based on the eigenvectors associated with the largest eigenvalues of the matrix G^TG to reduce the number of parameters. They compared Bayesian estimation, parameterization using sensitivity vectors and parameterization by zonation. They used the trace of the a posteriori covariance matrix for the measure of the uncertainty in the estimate. They concluded that the Bayesian estimation generated a model with the lowest uncertainty.

Gavalas et al. [16] advocated the use of eigenvectors of the prior covariance matrix for parameterization. For the one dimensional problems they considered, they reduced the number of parameters from sixty six to twenty without significantly affecting the accuracy of the estimates of the porosity and permeability fields. Later, Oliver [26] used parameterization based on the Karhunen-Loeve expansion, to condition log permeability fields to well test pressure data, variogram and point measurements of permeability. The Karhunen-Loeve expansion is referred to the decomposition of a random function, in terms of the eigenvectors of the covariance. It is optimal in a sense that it is the most efficient in putting the maximum amount of energy into the fewest modes. In one example by Oliver [26], repararneterization based on the Karhunen-Loeve expansion, reduced the number of parameters to be obtained in the Gauss-Newton matrix problem, from 1089 to 128 without a significant reduction in the quality of the final permeability estimates. Reynolds et al. [30] applied the parameterization based on the spectral decomposition of the prior correlation matrix which is dimensionless. They found that when the overall prior covariance matrix contains information for both porosity and permeability, a straightforward application of the spectral decomposition results in a reparameterization which suppresses much of the porosity information; in case that the covariance matrix is for different types of model parameters, the spectral decomposition should be applied to the prior correlation matrix.

Reynolds et al. [30] used the subspace method to reduce the size of the matrix problem in each of the Gauss-Newton iterations for history matching problems. In the subspace method, the search direction vector is expanded as a linear combination of a few basis vectors. These vectors may be gradients of subobjective functions, e.g., gradient of data mismatch term and gradient of the model misfit part of the objective function. Abacioglu et al. [2] followed the work of Reynolds et al. [30] with a more detailed investigation. The subspace vectors were computed with the adjoint method, and the gradient simulator method was used in calculation of the Hessian matrix. They investigated the effect of the number of the subspace vectors on the efficiency of a subspace method. They also suggested that instead of using a fixed number of subspace vectors, the number of subspace vectors can be gradually increased. They presented a theoretical argument that suggests the eigenvectors of the dimensionless matrix $L^T G^T C_D^{-1} GL$ associated with the largest eigenvalues, form an ideal basis for parameterization. They did not use this approach in example problems, and they also mentioned that the computation of this set of vectors is probably too expensive to be practical. The parameterization they proposed as an ideal basis, is the same as parameterization in terms of the right singular vectors of dimensionless sensitivity matrix, defined by Zhang et al. [44] as

$$G_D = C_D^{-1/2} G C_M^{1/2}, \tag{1.23}$$

noticing that $G_D^T G_D = L^T G^T C_D^{-1} GL$, if one uses a Cholesky decomposition for $C_M^{1/2}$. The Cholesky decomposition of C_M, is written as $C_M = LL^T$, and L can be used for $C_M^{1/2}$. Eigenvectors of $L^T G^T C_D^{-1} GL$ are the same as the right singular vectors of G_D, and the eigenvalues of $L^T G^T C_D^{-1} GL$ are the squares of the singular values of G_D. Later, we discuss the computation of a truncated SVD of G_D using the Lanczos algorithm, without explicitly forming the whole matrix. Abacioglu et al. [2] also found that the number of subspace vectors required to achieve rapid convergence is approximately equal to the number of eigenvalues of $L^T G^T C_D^{-1} GL$ that are greater than 0.1. They noticed that although this matrix changes as the minimum is approached, the number of eigenvalues that are greater than 0.1 was relatively constant in their examples.

Rodriques [33, 32] applied a history matching procedure where the change in the vector of model parameters over an iteration is expressed as a linear combinations of a few right singular vectors of the dimensionless sensitivity matrix. To obtain the singular vectors corresponding to the largest singular values of the dimensionless sensitivity matrix, they used the Lanczos algorithm. At each iteration of the Gauss-Newton or Levenberg-Marquardt algorithm, a truncated singular value decomposition of the dimensionless sensitivity matrix is constructed with the application of the Lanczos method. Lanczos's algorithm requires the product of G times a vector and G^T times a vector; the first product can be computed using the gradient simulator method and the second product from an adjoint solution.

Tavakoli and Reynolds [38] used the right singular vectors of a dimensionless sensitivity matrix, G_D, for parameterizing the vector of the change in model parameters at each iteration of the LM algorithm. They presented a theoretical argument that suggests the principal right singular vectors of the dimensionless sensitivity matrix form an optimal basis of parameterization, as eliminating those corresponding to smaller singular values has a negligible effect on the reduction of uncertainty obtained by conditioning a reservoir model to dynamic data. In their results, they obtained more reasonable permeability maps with SVD parameterization and with LBFGS-SVD, than they obtained with LBFGS.

Dickstein et al. [8] used SVD parameterization in the Gauss-Newton algorithm to condition the log permeability field to production and seismic data. In their synthetic example, in the absence of seismic data, when they used a fixed number of SVD parameters (25) with the GN algorithm, the resulting model was very rough, giving a poor representation of the reservoir, although the data matches were very good. They obtained good representations of the model with an increasing N_{SVD} strategy, i.e., starting with a few SVD parameters and gradually increasing the number of SVD parameters.

1.6.1 Gauss-Newton and Levenberg-Marquardt Algorithms

The results of Dickstein et al. [8] and Tavakoli and Reynolds [37] show that when using a fixed number $(20-25)$ of SVD parameters, with the LM algorithm, one can obtain a good representation of the true model (permeability field), while the GN algorithm may provide a rough and poor estimate of the model. Although, by gradually increasing the number of SVD parameters, both algorithms would provide a good estimate of the true model.

Li et al. [22] applied the GN algorithm and a modified LM algorithm to generate the MAP estimate conditioned to p_{wf} data for a simple 2D synthetic reservoir. In their results, the MAP estimate generated from the Gauss-Newton method is very rough and far from the true model. They concluded that the modified LM algorithm is more reliable for obtaining good estimates of model parameters, because by using a very high value of the LM parameter at early iterations, the changes in model parameters are damped. Although they did not use a parameterization and they formed the whole sensitivity matrix in their implementation, their comparison of the results of GN and LM algorithms is in agreement with the results of Dickstein et al. [8] and Tavakoli and Reynolds [37].

1.7 Research Scope and Thesis Outline

1.7.1 Research Scope

Development of a practical, robust and efficient technique for automatic history matching is a problem of great interest. For large scale history matching problems, $O(m)$ is typically minimized by implementation of the LBFGS algorithm discussed in Zhang and Reynolds [43] and Gao and Reynolds [13]. However in this study, our focus is on applying a modified Levenberg-Marquardt (LM) [21, 25] algorithm described in Oliver et al. [28] for minimization of the objective function. The equation of the modified LM algorithm can be solved with a truncated SVD technique.

Truncated SVD algorithms have been successfully applied to estimate or simulate log permeability fields of synthetic 2D examples [8, 37]. The main contribution of this work is to investigate the applicability of SVD parameterization algorithms for estimation or simulation of rock property fields of 3D reservoirs. This is one step towards applying this gradient based algorithm for large scale history matching problems. In some synthetic examples of this work, realizations of porosity and vertical log permeability as well as horizontal log permeability fields are generated to perform RML. In addition, SVD-EnRML algorithm for generating multiple realizations of rock property fields, which was suggested by Tavakoli and Reynolds [39], is modified to be more efficient and robust.

Tavakoli and Reynolds [39] applied a truncated SVD parameterization method to perform RML, by minimizing the objective functions of all realizations simultaneously.

They used a modified Levenberg-Marquardt algorithm to minimize the objective functions. Although they gave a brief description on how they use the Levenberg-Marquardt algorithm, they did not discuss an important issue, which is presented here.

At each iteration of SVD-EnRML, a truncated SVD of G_D for a particular realization, m_{base}, is calculated and is used in the LM search direction of all realizations to decrease the objective functions. Although by using this parameterization, the objective functions of all realizations may decrease at early iterations, there is no guarantee that using a truncated SVD of G_D of a particular realization in calculating LM search direction for another realization will decrease the objective function of that realization. To expand this point, let us look at the equation of the LM search direction at iteration l given by [39]

$$\begin{aligned} &L^{-1}\delta m_j^{l+1} = \\ &\quad = -[(1+\gamma_j^l)I_{N_m} + G_{D,l}^T G_{D,l}]^{-1}\{L^{-1}(m_j^l - m_{\text{uc},j}) + G_{D,l}^T C_D^{-\frac{1}{2}}(g(m_j^l) - d_{\text{uc},j})\}, \end{aligned} \tag{1.24}$$

where γ_j^l is the LM parameter of the jth realization at the lth iteration of the algorithm, and m_j^l is the vector of model parameters of the jth realization at the lth iteration of the algorithm. In this equation, the matrix $G_{D,l}$ is supposed to be computed at m_j^l; while in SVD-EnRML algorithm, we approximate $G_{D,l}$ with a truncated SVD of $G_{D,l}$ computed at another model, m_{base}^l. Since this approximation is *ad hoc*, it is possible that in an iteration of SVD-EnRML, applying the LM algorithm without the actual $G_{D,l}$ would not decrease the objective function of a realization; and as discussed later, this often happens, specially after the convergence of the MAP estimate.

In the implementation of the LM algorithm, if the search direction did not decrease the objective function, the LM parameters is multiplied by a factor and the iteration is repeated until a decrease in the objective function is obtained. This procedure is guaranteed to provide a decrease in the objective function, only if the actual $G_{D,l}$ is used in the search direction. However, in using a truncated SVD of $G_{D,l}$ based on another model, there should be an upper bound for the LM parameter.

Tavakoli and Reynolds [39] presented a second algorithm, which they called SVD-EnRML-AG. In this algorithm, they introduced an inner loop that the actual gradient of each realization is computed. They obtained better results with this algorithm in terms of computational costs and final values of objective functions.

Each iteration in the LM algorithm involves a reservoir simulation run. In this work we show that by some simple modifications, we can obtain better results than those obtained by Tavakoli and Reynolds [39] in terms of the computational cost and final values of the normalized objective functions. We present an ad hoc procedure to improve the SVD-EnRML algorithm so that all realizations generated tend to correspond to appropriately low values of the objective functions. We specify a maximum number of iterations in the LM algorithm and an upper bound for the LM parameter.

We also investigate the effect of the number of SVD parameters in truncated SVD algorithms on convergence properties and the final values of normalized objective functions.

As discussed at the end of the previous section, using the GN algorithm in history matching problems may generate a MAP estimate that is rough and far from the true model. On the other side, applying the LM algorithm with a high LM parameter at early iterations, will generate a good estimate of the true reservoir model. In this work, we investigate the details of the search directions for the two algorithms.

The subspace method was introduced into petroleum engineering by Reynolds et al. [30] in 1996. It has been successfully applied to synthetic single phase reservoirs. In this work, the convergence properties of the subspace method is compared with SVD parameterization in some synthetic 2D and 3D examples.

1.7.2 Thesis Outline

There are six chapters in this thesis. In Chapter 2, the formulation of the LM and GN algorithms with SVD parameterization for generating the MAP estimate and RML realizations are presented. Then a new modified SVD-EnRML algorithm is introduced.

In order to solve the issues pointed out in the previous subsection, we present some modifications to SVD-EnRML algorithm. Then the extension of SVD-EnRML to simulate rock property fields of 3D reservoirs are discussed. In the next section of this chapter, the GN and LM algorithms with SVD parameterization for simulation of rock property fields are compared. The details of the search direction and convergence properties of the two algorithms are investigated in an example which is a synthetic 28 by 30 reservoir. This chapter ends with a discussion on the number of required SVD parameters in the SVD-EnRML algorithm.

In Chapter 3, the subspace method and EnKF are first briefly described. Then some 2D examples are presented where the results of algorithms are compared. In the examples of this chapter, both porosity and log permeability fields are assumed to be uncertain. Prior information and fluid properties are assumed to be known. In the first two examples of this chapter, SVD-EnRML algorithms are compared in terms of computational cost and convergence rate with a 21 by 24 synthetic reservoir. We also investigate the effect of the number of singular triplets retained in the approximation of G_D, on the computational cost and convergence rate of the modified SVD-EnRML algorithm. Finally, the results of the subspace method is compared with the results of SVD parameterization. The 3rd example of Chapter 3 is a 29 by 35 synthetic reservoir, where again the results of SVD-EnRML algorithms are compared. The effect of the maximum number of singular triplets retained in the approximation of G_D, on the computational cost is also investigated. For this example, the results of the modified SVD-EnRML are compared with the results of EnKF in terms of quality of predictions and updated realizations.

Chapter 4 includes 3D examples. In the examples of this chapter, all rock property fields, i.e., porosity, horizontal and vertical log permeability fields, are assumed to be uncertain. Prior information and fluid properties are assumed to be known. The first example is a 28 by 30 by 3 synthetic reservoir with rock property fields having correlations in the vertical direction. In this example, we first compare the results from the subspace method and SVD parameterization when generating the MAP estimate. Then the modified SVD-EnRML algorithm is applied for simulation of the rock property fields in two cases. In one case we condition reservoir models only to pressure data, and in the other case, we condition realizations to both pressure and water rate data. In Example 2, a 30 by 30 by 3 reservoir model is presented where the rock property fields of different layers are uncorrelated. The modified SVD-EnRML algorithm is applied to this example and the results are presented. Example 3 of this chapter is a 30 by 15 by 4 reservoir model with the rock property fields of different layers uncorrelated.

In Chapter 5 we use SVD parameterization for estimating a non-Gaussian field for a synthetic 2D example.

Finally in Chapter 6, we present the conclusions.

CHAPTER 2

SIMULATION OF PERMEABILITY AND POROSITY FIELDS WITH SVD-ENRML ALGORITHMS

In this chapter, the SVD parameterization algorithm for generating an estimate of the model introduced by Tavakoli and Reynolds [38] is described first. Then the SVD-EnRML algorithm, for simultaneously generation the MAP estimate and RML realizations, introduced by Tavakoli and Reynolds [39] is explained. After this, our modifications to the original SVD-EnRML algorithm are presented, and the extension of this algorithm to 3D reservoirs is discussed. A modified SVD-EnRML algorithm is also introduced. Then, SVD parameterization with the Gauss-Newton and Levenberg-Marquardt algorithms are compared for an example.

2.1 Levenberg-Marquardt Algorithm with SVD Parameterization

In this section, SVD parameterization algorithm to generate an estimate of the rock property fields, suggested by Tavakoli and Reynolds [39] is briefly described.

In generating the MAP estimate, the search direction in the modified LM algorithm [28] is calculated as

$$\delta m^{l+1} = -[(1+\gamma^l)C_M^{-1} + G_l^T C_D^{-1} G_l]^{-1}\{C_M^{-1}(m^l - m_{\text{prior}}) + G_l^T C_D^{-1}(g(m^l) - d_{\text{obs}})\}, \quad (2.1)$$

where l denotes the iteration index and γ^l denotes the LM parameter; m^l is the vector of model parameters at the lth iteration of the algorithm. m_{prior} is the prior estimate of the model and d_{obs} is the vector of the observed data; G_l is the sensitivity matrix described in Section 1.4; C_M is the prior covariance matrix for the vector of model parameters, m, and C_D is the covariance matrix for the vector of observed data, d_{obs}.

The preceding LM algorithm can be applied successfully if the number of data is sufficiently small so that it is computationally feasible to calculate all entries of the sensitivity matrix with the adjoint method; since this is impractical for large scale history matching problems, it is desired to find a parameterization algorithm which has convergence properties similar or close to those of the LM algorithm but does not require forming the whole sensitivity matrix.

Applying the LM algorithm with SVD parameterization, does not require forming the whole sensitivity matrix. In SVD parameterization, the change of transformed model parameters at each iteration, $\delta\tilde{m}^{l+1}$, is expanded as a linear combination of a few right singular vectors of the dimensionless sensitivity matrix [33]. As in large problems, the computation of all entries of the sensitivity matrix is not feasible, this method can be considered as an alternative to the original LM algorithm.

In order to define the dimensionless sensitivity matrix, a square root of C_M is needed. One may use spectral decomposition or Cholesky decomposition to obtain an expression for $C_M^{1/2}$. With Cholesky decomposition of C_M, $C_M = LL^T$, where L is a lower triangular matrix, it follows that $C_M^{-1} = L^{-T}L^{-1}$, where L^{-T} denotes the inverse of L^T. C_D could be factored in the same way, but if C_D is diagonal, one can simply use $C_D^{-1} = C_D^{-\frac{1}{2}}C_D^{-\frac{1}{2}}$. Let $G_{D,l}$ denote the dimensionless sensitivity matrix corresponding to G_l, i.e., $G_{D,l} = C_D^{-\frac{1}{2}}G_lL$. Left multiplying Eq. 2.1 by L^{-1} and using the decompositions of C_M and C_D, the following equation is obtained:

$$\begin{aligned}
&L^{-1}\delta m^{l+1} = \\
&-L^{-1}[(1+\gamma^l)L^{-T}L^{-1}+L^{-T}L^TG_l^TC_D^{-1}G_l]^{-1}\{L^{-T}L^{-1}(m^l-m_{\text{prior}})+G_l^TC_D^{-1}(g(m^l)-d_{obs})\} \\
&\qquad = -[(1+\gamma^l)I_{N_m}+G_{D,l}^TG_{D,l}]^{-1}\{L^{-1}(m^l-m_{\text{prior}})+G_{D,l}^TC_D^{-\frac{1}{2}}(g(m^l)-d_{obs})\}, \quad (2.2)
\end{aligned}$$

where I_{N_m} denotes $N_m \times N_m$ identity matrix. Defining the transformed model, $\tilde{m}^l$, by

$$\tilde{m}^l = L^{-1}(m^l - m_{\text{prior}}), \tag{2.3}$$

the vector of the change in model parameters at iteration l in the transformed domain can be written as

$$\delta\tilde{m}^{l+1} \equiv \tilde{m}^{l+1} - \tilde{m}^l = L^{-1}(m^{l+1} - m^l) = L^{-1}\delta m^{l+1}. \tag{2.4}$$

Using Eq. 2.4, Eq. 2.2 becomes

$$\delta\tilde{m}^{l+1} = -[(1+\gamma^l)I_{N_m} + G_{D,l}^T G_{D,l}]^{-1}\{\tilde{m}^l + G_{D,l}^T C_D^{-\frac{1}{2}}(g(m^l) - d_{obs})\}, \tag{2.5}$$

or

$$[(1+\gamma^l)I_{N_m} + G_{D,l}^T G_{D,l}]\delta\tilde{m}^{l+1} = -\{\tilde{m}^l + G_{D,l}^T C_D^{-\frac{1}{2}}(g(m^l) - d_{obs})\}. \tag{2.6}$$

The properties of singular value decomposition (SVD) needed for this discussion can be found in Golub and van Loan [17]. Let u_i and v_i, respectively denote the left and right singular vectors corresponding to the singular value λ_i determined from a SVD of the dimensionless sensitivity matrix, G_D. For the normal case where the number of data N_d is less than the number of model parameters N_m, it is easy to show that

$$[(1+\gamma^l)I_{N_m} + G_{D,l}^T G_{D,l}]v_i = \begin{cases} (1+\gamma^l+\lambda_i^2)\,v_i & \text{if} \quad 1 \le i \le N_d \\ (1+\gamma^l)v_i & \text{if} \quad N_d < i \le N_m. \end{cases} \tag{2.7}$$

Thus, the eigenvalue-eigenvector pairs of the $N_m \times N_m$ matrix $[(1+\gamma^l)I_{N_m} + G_{D,l}^T G_{D,l}]$ are $(1+\gamma^l+\lambda_i^2, v_i)$ for $i = 1, 2 \cdots N_d$ and $(1+\gamma^l, v_i)$ for $i = N_d + 1, \cdots N_m$. For a detailed proof of Eq. 2.7, see the appendix of Tavakoli [36]. Note that the Hessian in the transformed space, which appears in the left-hand-side of Eq. 2.6, is real symmetric positive definite. Moreover, the eigenvectors of this matrix are the same as the right singular vectors of $G_{D,l}$. As discussed in the spectral analysis of the LM algorithm in Oliver et al. [28], $\delta\tilde{m}$ obtained by directly solving Eq. 2.6, is a linear combination of the eigenvectors of Hessian. Thus one can write $\delta\tilde{m}^{l+1}$ in the form:

$$\delta\tilde{m}^{l+1} = \sum_{j=1}^{N_m} \alpha_j v_j = V\alpha. \tag{2.8}$$

Note that by using Eq. 2.8, we are just expressing $\delta\tilde{m}$ in an orthogonal coordinate system with axis vectors composed of the right singular vectors of $G_{D,l}$ (which are the same as the eigenvectors of the Hessian), instead of the Cartesian coordinate system. Since the smaller singular values tend to amplify noisy, oscillating components of the solution, using only the largest singular values is a well-known technique to get regularized solutions [32]. Moreover, Tavakoli and Reynolds [38] provided theoretical support that the right singular vectors corresponding to the largest singular values may provide an optimal parameterization of the change in the model over an iteration with a gradient based method. With this motivation, we only compute the p largest singular values of $G_{D,l}$. With the singular vectors corresponding to the p largest singular values, a truncated SVD approximation of $G_{D,l}$ is written as

$$G_{D,l} = U_p \Lambda_p V_p^T = \sum_{i=1}^{p} \lambda_i u_i v_i^T, \tag{2.9}$$

where it is assumed that $p \leq N_d$. Keeping only the components of $\delta\tilde{m}$ along the right singular vectors associated with the p largest singular values, Eq. 2.8 is replaced with

$$\delta\tilde{m}^{l+1} = \sum_{j=1}^{p} \alpha_j v_j = V_p\alpha. \tag{2.10}$$

Using the truncated SVD of $G_{D,l}$, Eq. 2.6 can be approximated by

$$[(1+\gamma^l)I_{N_m} + V_p\Lambda_p^2 V_p^T]\delta\tilde{m}^{l+1} = -\{\tilde{m}^l + G_{D,l}^T C_D^{-\frac{1}{2}}(g(m^l) - d_{obs})\}. \tag{2.11}$$

Substituting Eq. 2.10 into the left hand side of Eq. 2.11, it follows

$$[(1+\gamma^l)I_{N_m} + V_p\Lambda_p^2 V_p^T]V_p\alpha = -\{\tilde{m}^l + G_{D,l}^T C_D^{-\frac{1}{2}}(g(m^l) - d_{obs})\}. \tag{2.12}$$

Using Eq. 2.7 and using the properties of SVD, Eq. 2.12 is equivalent to

$$\sum_{j=1}^{p} \alpha_j(1+\gamma^l+\lambda_j^2)v_j = -\{\tilde{m}^l + V_p\Lambda_p^T U_p^T C_D^{-\frac{1}{2}}(g(m^l) - d_{\text{obs}})\}, \tag{2.13}$$

where $p \leq N_d$. Left multiplying Eq. 2.13 by v_k^T and using the orthogonality of singular vectors, it follows that

$$\alpha_k = \frac{-v_k^T\tilde{m}^l - \lambda_k u_k^T C_D^{-\frac{1}{2}}(g(m^l) - d_{obs})}{1+\gamma^l+\lambda_k^2}, \tag{2.14}$$

for $k = 1$ to p. Using Eq. 2.14 in Eq. 2.10 yields

$$\delta\tilde{m}^{l+1} = \sum_{j=1}^{p}\Big[\frac{-v_j^T\tilde{m}^l - \lambda_j u_j^T C_D^{-\frac{1}{2}}(g(m^l) - d_{obs})}{1+\gamma^l+\lambda_j^2}\Big]v_j. \tag{2.15}$$

So from Eq. 2.4,

$$\delta m^{l+1} = L\delta\tilde{m}^{l+1} = L\sum_{j=1}^{p}\Big[\frac{-v_j^T\tilde{m}^l - \lambda_j u_j^T C_D^{-\frac{1}{2}}(g(m^l) - d_{obs})}{1+\gamma^l+\lambda_j^2}\Big]v_j. \tag{2.16}$$

Once δm^{l+1} is computed, the new vector of model parameters can be computed from

$$m^{l+1} = m^l + \delta m^{l+1}. \tag{2.17}$$

Unlike the GN algorithm where a line search is performed after finding the search direction, in the LM algorithm, the parameter γ^l controls both the search direction and the step size. The LM parameter, γ^l, varies from iteration to iteration. A simple procedure can be used to obtain the value of γ^l for each iteration. If $O(m^{l+1}) < O(m^l)$, m^{l+1} and $\tilde{m}^{l+1}$ are accepted as the new model parameters in the original and transformed space, respectively and γ^l is decreased by a decay factor of 10 ($\gamma_{l+1} = \gamma^l/\ 10$) for the next iteration. Otherwise, γ^l is increased by a growth factor of 10 ($\gamma_{l+1} = 10\gamma^l$) and the iteration is redone with the increased LM parameter.

Oliver et al. [28] suggested that the initial value of γ should be between $\sqrt{O(m^0)/N_d}$ and $O(m^0)/N_d$, where $O(m^0)$ is the initial value of the objective function. Abacioglu et al. [2] stated that if the value of γ is too small at an early iteration, the model acquires roughness which is difficult to remove at late iterations. A very large γ (e.g. 10^{10}) will also result in a very small initial rate of reduction in the objective function. They stated that in their examples it was necessary to start with γ between 10^6 and 10^7 to achieve a small value of the objective function for the 3-D reservoir case.

Based on our experiments with several examples of applying SVD parameterization algorithms, an initial value of 10^7 for γ^0 with a growth and decay factor of 10 works well.

At each iteration of Eq. 2.15, the truncated SVD of $G_{D,l}$ is updated, which is computationally expensive. As mentioned earlier, the Lanczos algorithm is used to compute a truncated SVD of the dimensionless sensitivity matrix. The Lanczos algorithm requires the products of G times a vector and G^T times a vector; the first product can be computed with the gradient simulator method and the second product from the adjoint method. The computational cost of the Lanczos algorithm increases linearly with the level of truncation, p. To obtain p singular triplets, the Lanczos algorithm requires approximately $p + n$ iterations, where each iteration involves one adjoint solution and one application of the gradient simulator method. The value of n depends on the magnitude of assigned relative error (ϵ_{sv}) for convergence of each of the singular values. The details on convergence criteria for the Lanczos algorithm can be found in Chapter 2 of Tavakoli [36]. Each application of the Lanczos algorithm requires roughly the equivalent of $(p + n)/2$ reservoir simulation runs. In all examples of this thesis, $\epsilon_{sv} = 10^{-5}$ is used. With this choice, the value of n varied from 3 to 8.

In SVD parameterization, the number of retained SVD parameters is gradually increased as iteration proceeds. A parameter called singular cutoff is used to determine the truncation level, p, at each iteration of the algorithm. The singular cutoff is denoted by "sv-cut" and is an input parameter for the Lanczos method. The Lanczos algorithm is terminated when the ratio of the smallest singular value retained to the largest singular value is less than or equal to sv-cut, i.e., $\lambda_p/\lambda_1 \leq$ sv-cut.

In SVD parameterization algorithm, two parameters are defined for sv-cut, which are μ_1 and $\mu_{\min}$. At the first iteration of the algorithm, $l = 1$, one sets sv-cut $= \mu_1$ and sv-cut is divided by 2 at each subsequent iteration until it reaches the final sv-cut of $\mu_{\min}$ where this value is used in all iterations until the convergence.

Although sv-cut is used to change the number of SVD parameters from one iteration to another, we noticed that for a small value of sv-cut (at late iterations), the number of retained singular values can be very sensitive to a small change of sv-cut, i.e., with a small change of sv-cut, the number of singular values significantly change. In addition, the distribution of the singular values of G_D for each example can be different. Thus, we modified the algorithm, so that the maximum allowable number of SVD parameters ($N_{\text{SVD,max}}$) to be calculated by the Lanczos method is also specified as input, i.e., the Lanczos method is terminated at either $p = N_{\text{SVD,max}}$ or the smallest value of p such that $\lambda_p/\lambda_1 \leq$ sv-cut, whichever is reached first. We usually tend to set a small $\mu_{\min}$, e.g., 0.0002, so that at late iterations of the algorithm, the Lanczos method would compute the specified number of singular triplets, $N_{\text{SVD,max}}$. However, $\mu_{\min}$ should not be unreasonably very small or high.

In examples of this work we used the following parameters for sv-cut, unless stated otherwise.

$$\mu_1 = 0.5,$$

$$\mu_{\min} = 0.0002. \tag{2.18}$$

Throughout, the objective function for generating the MAP estimate, $O(m)$ or

$O_0(m)$, refers to the following equation:

$$O(m) = \frac{1}{2}(m - m_{\text{prior}})^T C_M^{-1}(m - m_{\text{prior}}) + \frac{1}{2}(g(m) - d_{\text{obs}})^T C_D^{-1}(g(m) - d_{\text{obs}}), \tag{2.19}$$

and the objective function for generating an RML realization, $O_j(m)$, refers to

$$O_j(m) = \frac{1}{2}(m - m_{\text{uc,j}})^T C_M^{-1}(m - m_{\text{uc,j}}) + \frac{1}{2}(g(m) - d_{\text{uc,j}})^T C_D^{-1}(g(m) - d_{\text{uc,j}}). \tag{2.20}$$

2.1.1 Steps of Levenberg-Marquardt Algorithm with SVD Parameterization

Here the specific steps of the LM algorithm with SVD parameterizaton for generating an estimate of the model is presented.

Throughout l is the iteration index of the Levenberg-Marquardt algorithm, where a truncated SVD of $G_{D,l}$ is computed using the Lanczos method.

1. Set $l = 0$ and assign the initial guess of $m^0 = m_{\text{prior}}$ for the MAP estimate. Assign the initial value of the LM parameter, γ^0. Set the initial sv-cut, μ_1. Also, set the final value of the singular cutoff, $\mu_{\min}$, and the maximum number of SVD parameters ($N_{\text{SVD,max}}$) to be calculated by the Lanczos method.

2. Run the simulator with initial guess of the model parameters for the MAP estimate, and Compute $O(m^l)$ from Eq. 2.19.

3. Use the Lanczos method with the inputs sv-cut and $N_{\text{SVD,max}}$, to compute a truncated SVD of the dimensionless sensitivity matrix associated with the MAP estimate at iteration l, m^l.

4. (a) Compute the LM search direction with γ^l using Eq. 2.16, then use

$$m_{\text{temp}}^{l+1} = m^l + L\delta\tilde{m}^{l+1} \tag{2.21}$$

to calculate a proposed new update of the MAP estimate. Run the simulator with m_{temp}^{l+1} and calculate the objective function $O(m_{\text{temp}}^{l+1})$ given with Eq. 2.19.

(b) If $O(m_{\text{temp}}^{l+1}) < O(m^l)$, then set

$$\gamma^{l+1} = \gamma^l/10,$$

$$m^{l+1} = m_{\text{temp}}^{l+1}. \tag{2.22}$$

Else, if $O(m_{\text{temp}}^{l+1}) \geq O(m^l)$, then set $\gamma^l = \max(10\gamma^l, 10)$ and go to step (a).

5. Check for convergence using the criteria discussed later. If the algorithm has not converged, increase iteration index l by 1, replace sv-cut by $\max\left\{\frac{\text{sv-cut}}{2}, \mu_{\min}\right\}$ and go to step 3; otherwise, if the convergence criteria are satisfied, go to step 6.

6. End.

Note that the reason we set $\gamma^l = \max(10\gamma^l, 10)$ in step 4(b), is that a small value of the LM parameter (say 0.1) usually does not make an effective change in the search direction, so if $10\gamma^l < 10$, we set it to 10.

In large scale history matching problems, large changes in the objective function at an early iteration may result in convergence to a model with a high value of the objective function. However, based on our experience, and based on the results of Example 1 of Chapter 3, with a large starting LM parameter (e.g. 10^6) in generating an estimate of the model with the algorithm just presented, there is no need to control the change in the objective function at early iterations.

2.2 SVD-EnRML Algorithms

SVD-EnRML algorithms, proposed by Tavakoli and Reynolds [39], aim to simultaneously minimize the objective function of the MAP estimate and N_e RML realizations to sample the posterior pdf. The main idea of this algorithm is that in the LM search direction in the transformed space, at each iteration l, $G_{D,l}$ of all realizations are approximated by a truncated SVD of $G_{D,l}$ of a particular realization. In addition, the right singular vectors of this truncated SVD are used to parameterize the vector of the change in model parameters of all realizations at that iteration of the algorithm. Tavakoli and Reynolds [39] introduced two SVD-based algorithms to perform RML, which they called SVD-EnRML-SMM and SVD-EnRML-AG. The SVD-EnRML-SMM is the basic SVD-EnRML algorithm, and SMM refers to sensitivity matrix multiplication. Since in SVD-EnRML-SMM, the dimensionless sensitivity matrix in both the Hessian and gradient is approximated with a truncated SVD of $G_{D,l}$ of a particular realization, the gradient is computed by multiplying the transpose of the truncated SVD matrix and a vector, hence the name SMM. The second algorithm is called SVD-EnRML-AG, where AG refers to adjoint gradient. In the added inner loop of this algorithm, the gradient of each objective function is computed using the adjoint method, hence the name AG.

At each iteration l of SVD-EnRML, a truncated SVD of dimensionless sensitivity matrix, $G_{D,l}$, for a particular realization is computed by using the Lanczos method. In SVD-EnRML, similar to the SVD parameterization algorithm, the number of retained SVD parameters is gradually increased as iteration proceeds. The singular cutoff, sv-cut, is used to determine the number of retained singular triplets at each iteration of the algorithm.

In addition to μ_1 and $\mu_{\min}$, we define another parameter, μ_2, for sv-cut in SVD-EnRML. At the first iteration of the SVD-EnRML algorithms, one sets sv-cut $= \mu_1$ and this number is divided by 2 at each subsequent iteration until it reaches the final sv-cut of $\mu_{\min}$, where this value is used in all iterations until the convergence of the MAP estimate. After the convergence of the MAP estimate, at each iteration of SVD-EnRML, a truncated SVD of the dimensionless sensitivity matrix corresponding to the realization with the maximum objective function is calculated. At this point sv-cut is set to μ_2 and again the same procedure is followed to determine sv-cut at each iteration. In examples of this work for SVD-EnRML, we used the following parameters for sv-cut, unless stated otherwise.

$$\mu_1 = 0.5,$$

$$\mu_2 = 0.05,$$

$$\mu_{\min} = 0.0002. \tag{2.23}$$

Similar to the SVD parameterization algorithm for generating an estimate of the model, in SVD-EnRML, the maximum allowable number of SVD parameters ($N_{\text{SVD,max}}$) to be calculated by the Lanczos method is also specified as input, i.e., the Lanczos algorithm is terminated at either $p = N_{\text{SVD,max}}$ or the smallest value of p such that $\lambda_p/\lambda_1 \leq$ sv-cut, whichever is reached first.

2.2.1 SVD-EnRML-SMM Algorithm

A detailed step-by-step description of the original algorithm is provided in Tavakoli and Reynolds [39]. What is given for step by step description of the algorithm in Subsection 2.2.4, is with our modifications.

Throughout, m_0^l denotes the approximation of the MAP estimate at the lth iteration of the LM algorithm when minimizing the objective function of Eq. 2.19, and m_j^l, $j = 1, 2, \cdots N_e$ denotes the estimate of the jth RML realization at iteration l in minimizing $O_j(m)$ given by Eq. 2.20. Let (λ_i, u_i, v_i), $i = 1, 2, \cdots p$ denote the ith singular triplet from the truncated SVD of the dimensionless sensitivity matrix, $G_{D,l}$, corresponding to a particular realization, m_{base}, at iteration l. The truncated SVD of $G_{D,l}$ is computed based on the MAP estimate; and after the convergence of the MAP estimate, a truncated SVD of $G_{D,l}$ for the realization with the maximum objective function is computed. In applying the LM algorithm, the search direction to generate the MAP estimate, $\delta\tilde{m}_0^{\ell+1}$, is obtained from Eq. 2.15, whereas the search direction to generate the jth RML realization is calculated by the obvious analogue of Eq. 2.15,

$$\delta\tilde{m}_j^{l+1} = \sum_{i=1}^{p} \left[\frac{-v_i^T \tilde{m}_j^l - \lambda_i u_i^T C_D^{-\frac{1}{2}}(g(m_j^l) - d_{\text{uc, j}})}{1 + \gamma_j^l + \lambda_i^2} \right] v_i, \tag{2.24}$$

for $j = 1, 2, \cdots N_e$. Once $\delta \tilde{m}_j^{l+1}$ is obtained, δm_j^{l+1} is calculated from:

$$\delta m_j^{l+1} = L \delta \tilde{m}_j^{l+1}. \tag{2.25}$$

The new candidate for updated model is calculated using $m_j^{l+1} = m_j^l + \delta m_j^{l+1}$, which is accepted only if it decreases the objection function, $O_j(m)$.

After decreasing the objective functions of all realizations, a truncated SVD of $G_{D,l}$ is computed again, and the loop is repeated until convergence.

Note that although the algorithm was derived directly from a gradient-based LM algorithm, neither sensitivities nor gradients of the objective functions are explicitly computed.

2.2.2 SVD-EnRML-AG Algorithm

A second algorithm, introduced by Tavakoli and Reynolds [39], is called SVD-EnRML-AG. This algorithm, is a modification to SVD-EnRML-SMM. In SVD-EnRML-AG an inner iteration is added to the basic procedure described for SVD-EnRML-SMM. In the inner loop, for each realization, a damped objective function instead of the actual objective function is minimized. In the inner loop iterations, the SVD parameterization is kept fixed and the gradient of each objective function is calculated with the adjoint method. In applying the LM algorithm in the inner iteration, the search direction $\delta \tilde{m}_j^{k+1}$ for each realization is calculated using the following equation:

$$\delta \tilde{m}_j^{k+1} = \sum_{i=1}^{p} \left[\frac{-v_i^T \tilde{m}_j^k - v_i^T L^T G_{k,j}^T \tilde{C}_{D,j}^{-1} (g(m_j^k) - d_{\text{uc, j}})}{1 + \gamma_j^l + \lambda_i^2} \right] v_i, \tag{2.26}$$

where k is the iteration index of the inner loop. The matrix vector product $G_{k,j}^T \tilde{C}_{D,j}^{-1} (g(m_j^k) - d_{\text{uc, j}})$ is computed using the adjoint method. For each realization, the inner iteration is ended when the change in the objective function is small.

The damped objective function for generating a realization, refers to

$$O_{\mathrm{damp},j}(m) = \frac{1}{2}(m - m_{\mathrm{uc,j}})^T C_M^{-1}(m - m_{\mathrm{uc,j}}) + \frac{1}{2}(g(m) - d_{\mathrm{uc,j}})^T \tilde{C}_{D,j}^{-1}(g(m) - d_{\mathrm{uc,j}}), \quad (2.27)$$

where $\tilde{C}_{D,j}$ is obtained by multiplying the nth diagonal entry of C_D by ψ_n^2 where ψ_n is calculated by

$$\psi_n = \max\{1, |\frac{g_n(m_j^l) - [d_{\mathrm{uc},j}]_n}{3\sigma_{d,n}}|\}. \quad (2.28)$$

In the preceding equation, $\sigma_{d,n}$ denotes the standard deviation of the nth measurement error, m_j^l is the last updated model obtained at the outer iteration, $g_n(m_j^l)$ is the nth component of the predicted data vector evaluated at m_j^l and $[d_{\mathrm{uc},j}]_n$ denotes the nth entry of the perturbed data vector $d_{\mathrm{uc},j}$. In the inner loop at iteration l, damping factors and thus $\tilde{C}_{D,j}$ are fixed.

2.2.3 Levenberg-Marquardt Algorithm in SVD-EnRML

In SVD-EnRML-SMM, when $G_{D,l}$ in the LM search direction for minimizing the objective function $O_j(m)$ of an RML realization, is approximated by a truncated SVD of $G_{D,l}$ for another realization, m_{base}, there is no guarantee that the objective function O_j decreases, as the $G_{D,l}$ in the LM search direction is supposed to be computed for m_j^l. In practice, what happens is that in some cases, when the truncated SVD of $G_{D,l}$ of a particular realization (m_{base}) is used to try to decrease the objective function of realization j ($m_j \neq m_{\mathrm{base}}$), no matter how much the LM parameter is increased, the objective function of realization j does not decrease. Note that in the original LM search direction given by Eq. 2.1, a large LM parameter is similar to taking a small step in the steepest descent direction; however in SVD-EnRML-SMM, since we are approximating the gradient using a truncated SVD of $G_{D,l}$ computed at another model, the search direction is not necessarily downhill. A search direction is downhill, if its inner product with the gradient of objective function is negative.

Based on the discussion in this subsection, we limit the number of iterations in the LM algorithm for decreasing the objective function $O_j(m)$, at the lth iteration of SVD-EnRML. We also assign an upper bound for the LM parameter. With these simple modifications, many unnecessary simulation runs are avoided.

At iteration l of SVD-EnRML, a truncated SVD of $G_{D,l}$ is computed based on a particular realization, m^l_{base}. For realization m^l_{base}, the SVD parameterization should decrease the objective function, unless the algorithm has converged for this realization. However, for realization j ($m_j \neq m_{\text{base}}$), we specify a maximum number of iterations, $s_{\max}$ and an upper bound for the LM parameter, $\gamma_{\max}$.

Throughout, l refers to the lth updating of SVD parameters by the Lanczos method; $\gamma_{\max}$ is an upper bound for the LM parameter which is specified as input. s denotes the iteration index in an application of the LM algorithm at iteration l to decrease an objective function. s_{max} refers to the maximum allowable iterations in an application of the LM algorithm for a realization j ($m_j \neq m_{base}$) which is specified as input.

Throughout, $\gamma_{j,s}$ (with a subscript s) is referred to the LM parameter which is used in the LM search direction for the jth realization at a fixed l. γ^l_j (with a superscript l) is the initial value of $\gamma_{j,s}$, i.e., $\gamma_{j,s=0} = \gamma^l_j$.

2.2.4 Steps of SVD-EnRML-SMM and SVD-EnRML-AG

The specific steps for the implementation of the SVD-EnRML-AG algorithm are given first. Then it will be shown how to convert the steps to the SVD-EnRML-SMM algorithm. Note that the steps given here are with our modifications discussed in the previous subsection.

1. Set $l = 0$ and assign the initial guess of $m^l_j = m_{\text{uc},j}$ for $j = 1, 2, \cdots N_e$ for RML realizations and an initial guess of $m^l_0 = m_{\text{prior}}$ for the MAP estimate. Assign the initial value of the LM parameter, γ^0, and set $\gamma^l_j = \gamma^0$ for $j = 0, 1, \cdots N_e$. Set the initial sv-cut and sv-cut after the convergence of the MAP estimate, μ_1 and μ_2, respectively. Set the final value of the singular cutoff, $\mu_{\min}$ and the maximum

number of SVD parameters ($N_{\text{SVD,max}}$) to be calculated by the Lanczos method; also set $\gamma_{\max}$ and $s_{\max}$.

2. Calculate the initial values of the objective functions by running the simulator to the final time with initial guesses of the model parameters for the MAP estimate and all realizations. Compute $O_j(m_j^l)$ with Eq. 2.20 for $j = 1, 2, \cdots N_e$ and compute $O(m_0^l)$ from Eq. 2.19.

3. If $O_{N,\text{MAP}}(m_0^l) < 1 + 5\sqrt{2/N_d}$ or if Eqs. 2.37 and 2.38 are both satisfied, i.e., the MAP estimate has converged, use the Lanczos method to compute a truncated SVD of the dimensionless sensitivity matrix pertaining to m_{base}^l where

$$O_{j_{\text{base}}}(m_{\text{base}}^l) = \max_{1 \le j \le N_e} O_j(m_j^l). \tag{2.29}$$

   In this case as the truncated SVD of $G_{D,l}$ is computed for m_{base}, we want to have a large step for the change of the model in decreasing the maximum objective function, $O_{j_{\text{base}}}$; in addition, γ^l for the base realization may have a large value that would result in a small decrease of the maximum objective function, $O_{j_{\text{base}}}$. Thus, set γ_{base}^l (γ^l pertaining to m_{base}^l) to a small number, e.g., 0.1.

   Otherwise, use the Lanczos method to compute a truncated SVD of the dimensionless sensitivity matrix for the MAP estimate at iteration l, m_0^l.

4. For $j = 0, 1, 2, \cdots N_e$

   (a) Set $s = 0$, where s is the iteration index in the LM algorithm for decreasing the objective function of the jth realization, at a fixed l.

   (b) Set the initial value for the LM parameter,

$$\gamma_{j,s} = \gamma_j^l. \tag{2.30}$$

   (c) Compute the LM search direction, $\delta\tilde{m}_j^{l+1}$, using $\gamma_{j,s}$ in Eq. 2.24 for an RML realization, or in Eq. 2.15 for the MAP estimate, then use

$$m^{l+1}_{\text{temp},j} = m^l_j + L\delta\tilde{m}^{l+1}_j, \tag{2.31}$$

to calculate a proposed new update of the jth realization. Run the simulator and calculate the objective function $O_j(m^{l+1}_{\text{temp},j})$.

(d) If $O_j(m^{l+1}_{\text{temp},j}){\geq}O_j(m^l_j)$, then

i. If $s = s_{\max}$ or $\gamma_{j,s} \geq \gamma_{\max}$, and $m_j \neq m_{\text{base}}$, then set

$$\gamma^{l+1}_j = \gamma^l_j,$$

$$m^{l+1}_j = m^l_j, \tag{2.32}$$

and go to step (a) for the next realization, i.e., terminate the LM algorithm for the jth realization at iteration l. Note that the jth realization and its corresponding LM parameter are not updated at this iteration.

ii. Else, if $s < s_{\max}$ and $\gamma_{j,s} < \gamma_{\max}$, or $m_j = m_{\text{base}}$, set

$$\gamma_{j,s+1} = \max(10\gamma_{j,s}, 100),$$

$$s = s + 1. \tag{2.33}$$

and return to step (c).

Else, if $O_j(m^{l+1}_{\text{temp},j}) < O_j(m^l_j)$, set

$$m^{l+1}_j = m^{l+1}_{\text{temp},j},$$

$$\gamma^{l+1}_j = \gamma_{j,s}/10. \tag{2.34}$$

5. Check for convergence using the criteria discussed later. If the algorithm has not converged, increase iteration index l by 1, replace sv-cut by $\max\left\{\frac{\text{sv-cut}}{2}, \mu_{\min}\right\}$ and go to step 6; otherwise, go to step 7, i.e., terminate the algorithm.

6. For $j = 0, 1, 2, \cdots N_e$

   (a) With k denoting the iteration index of the inner loop, set $k = 0$ and set the initial guess of the model parameters for the inner loop as $m_j^k = m_j^l$, where m_j^l is the last updated model obtained at the outer loop.

   (b) Compute the damping factors by Eq. 2.28, then compute the damped objective function $O_{\text{damp},j}(m_j^k)$ by Eq. 2.27. Then, set the initial value of the LM parameter for the inner loop, $\gamma_{j,k} = O_{\text{damp},j}(m_j^k)/N_d$ for $j \neq 0$, and $\gamma_{j,k} = 2O_{\text{damp},j}(m_j^k)/N_d$ for $j = 0$.

   (c) Calculate the product of $G_{k,j}^T$ with the vector $\tilde{C}_{D,j}^{-1}(g(m_j^k) - d_{\text{uc},j})$ using the adjoint method.

   (d) Calculate $\delta\tilde{m}_j^{k+1}$ by Eq. 2.26 using $\gamma_{j,k}$, then use

   $$m_{\text{temp},j}^{k+1} = m_j^k + L\delta\tilde{m}_j^{k+1} \tag{2.35}$$

   to calculate a proposed new update of the jth realization, $m_{\text{temp},j}^{k+1}$.

   (e) Run the simulator with $m_{\text{temp},j}^{k+1}$ and calculate the value of the damped objective function, $O_{\text{damp},j}(m_{\text{temp},j}^{k+1})$.

   (f) If $O_{\text{damp},j}(m_{\text{temp},j}^{k+1}) \geq O_{\text{damp},j}(m_j^k)$, then set $\gamma_{j,k} = 10\gamma_{j,k}$, and return to step (d).

   Else, if $O_{\text{damp}}(m_{\text{temp},j}^{k+1}) < O_{\text{damp},j}(m_j^k)$, then set $m_j^{k+1} = m_{\text{temp},j}^{k+1}$, $\gamma_{j,k+1} = \gamma_{j,k}/10$, and go to the next step.

   (g) Check for convergence of the inner loop. The convergence criterion is

   $$\frac{O_{\text{damp},j}(m_j^k) - O_{\text{damp},j}(m_j^{k+1})}{O_{\text{damp},j}(m_j^k)} < 5 \times 10^{-3}. \tag{2.36}$$

If the condition given by Eq. 2.36 is not satisfied, increase the iteration index, k, by one and return to step (c).

Once convergence of all damped objective functions is obtained, set $m_j^l = m_j^{k+1}$ for each j and go to step 3.

7. End.

To implement SVD-EnRML-SMM, the step 6 of the algorithm described above is deleted and step 5 is replaced by:

Step 5: Check for convergence using the criteria discussed later. If the algorithm has not converged, increase iteration index l by 1, replace sv-cut by $\max\left\{\frac{\text{sv-cut}}{2}, \mu_{\min}\right\}$ and go to step 3. Otherwise terminate the algorithm.

In this modified SVD-EnRML-SMM algorithm, each application of the LM algorithm requires at most $s_{\max}$ simulation runs; since, if the objective function did not decrease after $s_{\max}$ iterations, the LM algorithm is terminated and m_j^l and γ_j^l are kept the same for the next iteration. Before the convergence of the MAP estimate, we set $s_{\max} = 5$, and after the convergence of the MAP estimate we set $s_{\max} = 3$. We also set $\gamma_{\max} = 10^8$, for the upper bound of the LM algorithm.

In the step by step description of the SVD-EnRML-AG algorithm, given in Tavakoli and Reynolds [39], they simply used the same LM parameter in the inner loop as they described for the outer loop. Based on our experience, decreasing the LM parameter in both the inner loop and outer loop is not efficient. We modified the inner loop of the SVD-EnRML-AG, so that the first value of the LM parameter in the inner loop is set to normalized damped objective function, and the LM parameter of the outer loop for the next iteration, is not influenced by the final γ from the inner loop. In the inner loop, since we are minimizing a damped objective function, the data mismatch terms are damped and using a high LM parameter will provide a search direction with very small entries. Thus we set the initial value of the LM parameter to the the normalized damped objective function.

In our implementation of the SVD-EnRML algorithms, when the normalized objective function of a realization, $O_N(m)$, decreases to a value less than $1+5\sqrt{2/N_d}$, that realization is accepted as a conditional realization and is not minimized any more; this is because we want to use the same criterion for all examples and all SVD-EnRML algorithms; however, the convergence criteria of the algorithm are not based on the values of $O_N(m)$.

2.2.5 Convergence Criteria

The convergence criteria for termination of SVD-EnRML algorithms are based on the requirement that both the maximum relative change in the objective function and the maximum relative change in the model parameters of all realizations be small, i.e.,

$$\triangle O(m) = \max_{1\le j\le N_e} \left\{ \frac{|O_j(m_j^{l+1}) - O_j(m_j^l)|}{O_j(m_j^{l+1}) + 10^{-5}} \right\} < \eta_o, \tag{2.37}$$

and

$$\triangle m = \max_{1\le j\le N_e} \left\{ \frac{\|m_j^{l+1} - m_j^l\|}{\|m_j^{l+1}\| + 10^{-5}} \right\} < \eta_m. \tag{2.38}$$

In the examples presented here, the following values are used:

$$\eta_o = 10^{-3}, \tag{2.39}$$

$$\eta_m = 10^{-2}. \tag{2.40}$$

In practice, one may also specify a maximum number of iterations, e.g., terminate the algorithm at a maximum of 50 applications of the Lanczos method.

2.3 SVD-EnRML With Multiple Iteration Method

In SVD-EnRML, a significant cost is for calculation and updating singular triplets with the Lanczos algorithm. Each iteration of the Lanczos algorithm for computing a truncated SVD of $G_{D,l}$, requires one application of the gradient simulator method to obtain $G_l x$ and one adjoint solution to obtain $G_l^T y$. Therefore, it is desirable to avoid extensive updates of the singular triplets during the minimization. Computational cost can be reduced if either the number of iterations of SVD-EnRML or the number of SVD triplets at each iteration are reduced.

The main idea of the SVD-EnRML-SMM algorithm, which is the original SVD-EnRML algorithm, was to approximate the truncated SVD of $G_{D,l}$ of all realizations by a truncated SVD of $G_{D,l}$ for a particular realization. In addition, the right singular vectors of this truncated SVD are used to parameterize the vector of the change in the model parameters of all realizations at that iteration of the LM algorithm. The idea of this multiple iteration method is to use the truncated SVD of $G_{D,l}$ in more than one application of the LM algorithm. We add an inner loop to the basic algorithm, where the truncated SVD of $G_{D,l}$, at a fixed l, is used multiple times to decrease the objective functions. Unlike the SVD-EnRML-AG algorithm, in the inner loop of this multiple iteration method, we use the same equation of the outer loop for computing the search direction and the gradient is not computed with the adjoint method.

At early iterations, where the truncated SVD of $G_{D,l}$ is computed based on the MAP estimate, the rate of decrease of all objective functions is quite fast; thus, as long as the SVD parameters are computed based on the MAP estimate, or $m_{\text{base}} = m_0$, we follow the same steps of the SVD-EnRML-SMM algorithm. After the convergence of the MAP estimate, the algorithm is modified as follows. Like the original SVD-EnRML-SMM algorithm, after computing a truncated SVD of $G_{D,l}$ for the realization with the maximum objective function by using the Lanczos method, the LM algorithm is used to decrease the objective function of each realization, m_j for $j = 1, 2, \cdots N_e$. In the application of the LM algorithm, if the objective function of realization m_j decreases by more than 1%, the realization enters a loop where the same SVD parameters are used to further decrease the objective function, $O_j(m)$. In this loop, the LM parameter is not changed, i.e., the same LM parameter is used in the search direction at every inner iteration. This loop is ended if the objective function does not decrease more than 1%. We call this method SVD-EnRML-MI which refers to SVD-Ensemble-RML-Multiple-

Iterations.

Keeping the LM parameter constant in the inner loop is *ad hoc*, however there is a motivation for it. As the outer iteration, (with iteration index of l) proceeds, the truncated SVD of $G_{D,l}$ is updated and the LM parameter of each realization is decreased after each successful iteration. However, the truncated SVD of $G_{D,l}$ that is used in the inner loop (with iteration index of k at a fixed l), has already been used in the outer loop in an application of the LM algorithm to decrease $O_j(m)$. Changing the LM parameter, while using the same SVD parameters might even increase the computational cost. Note that the LM parameter has an impact on the weight of each of the right singular vectors in the search direction, and since the right singular vectors are the same, the LM parameter is kept fixed in the inner loop iterations.

In multiple iterations with the same SVD parameters, the cost of each successful iteration (with the same SVD parameters) is only one reservoir simulation run.

2.3.1 Steps of SVD-EnRML-MI

Now we present the steps of the SVD-EnRML-MI algorithm. Throughout l is the iteration index of SVD-EnRML-MI, where a truncated SVD of $G_{D,l}$ is computed using the Lanczos method; k denotes the iteration index of the inner loop at a fixed l.

1. Set $l = 0$ and assign the initial guess of $m_j^0 = m_{\mathrm{uc},j}$ for $j = 1, 2, \cdots N_e$ for RML realizations and an initial guess of $m_0^0 = m_{\mathrm{prior}}$ for the MAP estimate. Assign the initial value of the LM parameter, γ^0, and set $\gamma_j^1 = \gamma^0$ for $j = 0, 1, \cdots N_e$. Set the initial sv-cut and sv-cut after the convergence of the MAP estimate, μ_1 and μ_2, respectively. Set the final value of the singular cutoff, $\mu_{\min}$ and the maximum number of SVD parameters ($N_{\mathrm{SVD,max}}$) to be calculated by the Lanczos method; also set $\gamma_{\max}$ and $s_{\max}$.

2. Calculate the initial values of the objective functions by running the simulator to the final time with initial guesses of the model parameters for the MAP estimate and all realizations. Compute $O_j(m_j^l)$ with Eq. 2.20 for $j = 1, 2, \cdots N_e$ and compute

$O(m_0^l)$ from Eq. 2.19.

3. If $O_{N,\text{MAP}}(m_0^l) < 1 + 5\sqrt{2/N_d}$ or if Eqs. 2.37 and 2.38 are both satisfied, i.e., the MAP estimate has converged, use the Lanczos method to compute a truncated-SVD of the dimensionless sensitivity matrix pertaining to m_{base}^l where

$$O_{j_{\text{base}}}(m_{\text{base}}^l) = \max_{1 \le j \le N_e} O_j(m_j^l). \tag{2.41}$$

   In this case as the truncated SVD of $G_{D,l}$ is computed for m_{base}, in decreasing the maximum objective function, $O_{j_{\text{base}}}$, we want to have a large step for the change of the model; in addition, γ^l for the base realization may have a large value. Thus, set γ_{base}^l (γ^l pertaining to m_{base}^l) to a small number, e.g., 0.1.

   Otherwise, use the Lanczos method to compute a truncated SVD of the dimensionless sensitivity matrix associated with the MAP estimate at iteration l, m_0^l.

4. For $j = 0, 1, 2, \cdots N_e$ do the following steps of the LM algorithm, to decrease the objective function, $O_j(m)$.

   (a) Set $s = 0$, where s is the iteration index in the LM algorithm for decreasing the objective function of the jth realization, $O_j(m)$, at a fixed l.

   (b) Set the initial value for the LM parameter,

$$\gamma_{j,s} = \gamma_j^l.$$

   (c) Compute the LM search direction, $\delta\tilde{m}_j^{l+1}$, using $\gamma_{j,s}$ in Eq. 2.24 for an RML realization or in Eq. 2.15 for the MAP estimate, then use

$$m_{\text{temp},j}^{l+1} = m_j^l + L\delta\tilde{m}_j^{l+1}, \tag{2.42}$$

   to calculate a proposed new update of the jth realization. Run the simulator and calculate the objective function $O_j(m_{\text{temp},j}^{l+1})$ with Eq. 2.20.

(d) If $O_j(m^{l+1}_{\text{temp},j}) \geq O_j(m^l_j)$, then

i. If $s = s_{max}$ or $\gamma_{j,s} \geq \gamma_{\max}$, and $m_j \neq m_{\text{base}}$, set

$$\gamma^{l+1}_j = \gamma^l_j,$$

$$m^{l+1}_j = m^l_j, \tag{2.43}$$

and go to step 4(a) with the next realization, i.e., terminate the LM algorithm for the jth realization at iteration l. Note that the jth realization and its corresponding LM parameter are not updated at this iteration.

ii. Else, if $s < s_{\max}$ and $\gamma_{j,s} < \gamma_{\max}$, or $m_j = m_{\text{base}}$, set

$$\gamma_{j,s+1} = \max(10\gamma_{j,s}, 100),$$

$$s = s + 1, \tag{2.44}$$

and return to step (c).

Else, if $O_j(m^{l+1}_{\text{temp},j}) < O_j(m^l_j)$, set

$$m^{l+1}_j = m^{l+1}_{\text{temp},j},$$

$$\gamma^{l+1}_j = \gamma_{j,s}/10. \tag{2.45}$$

(e) If m_{base} is not the MAP estimate, and

$$\frac{O_j(m^l_j) - O_j(m^{l+1}_j)}{O_j(m^l_j)} > 0.01, \tag{2.46}$$

go to the next step which is the first step for the multiple iterations with the same SVD parameters. Otherwise, go to step 4(a) with the next realization.

The following steps are for multiple iterations with the same SVD parameters.

(a) Set $k = 1$, $m_j^k = m_j^{l+1}$, where k is the iteration index for the inner loop.

(b) Compute the LM search direction with $\gamma_{j,s}$ using Eq. 2.24, then use

$$m_j^{k+1} = m_j^k + L\delta\tilde{m}_j^{k+1}, \tag{2.47}$$

to calculate a proposed new update of the jth realization. Run the simulator and calculate the objective function $O_j(m_j^{k+1})$ given with Eq. 2.20.

(c) If $O_j(m_j^{k+1}) < O_j(m_j^k)$, then set

$$m_j^{l+1} = m_j^{k+1}, \tag{2.48}$$

if $k < 10$ and

$$\frac{O_j(m_j^k) - O_j(m_j^{k+1})}{O_j(m_j^k)} > 0.01, \tag{2.49}$$

then set $k = k + 1$ and return to the previous step.

Otherwise, go to step $4(a)$ with the next realization, i.e., end the multiple iterations for the jth realization.

5. Check for convergence using the criteria discussed before. If the algorithm has not converged, increase iteration index l by 1, replace sv-cut by $\max\left\{\frac{\text{sv-cut}}{2}, \mu_{\min}\right\}$ and go to step 3 ; otherwise, go to step 6.

6. End.

In the algorithm described, $k < 10$ is included to limit the number of iterations with the same SVD parameters.

In large scale history matching problems, large changes in the objective function at an early iteration may result in convergence to a model with a high objective function. If the objective function decreases by a factor of 10 or greater at the first iteration, the initial value of γ may be too small [28]. Based on this, in all SVD-EnRML algorithms, the change in the objective function of RML realizations is controlled at the first two iterations so that it does not decrease by more than 50%, i.e., at the first two iterations we require that

$$\frac{O_j(m_j^l) - O_j(m_j^{l+1})}{O_j(m_j^l)} \leq 0.5, \tag{2.50}$$

for $j = 1, 2, \cdots N_e$.

2.4 Rescaling Search Direction

Tavakoli and Reynolds [39] applied the SVD-EnRML method to simulate permeability fields. In this work, in addition to horizontal permeability fields, the method is applied to simulate porosity and vertical permeability fields as well. In estimating different rock property fields, scaling is an important issue. Since in SVD-EnRML, parameterization is performed in a LM algorithm, the Hessian matrix automatically scales the search direction. However, we do not want to change any property of a gridblock by several standard deviations at one iteration.

As mentioned before, the transformed model is defined as:

$$\tilde{m}^l = L^{-1}(m^l - m_{\text{prior}}). \tag{2.51}$$

For the transformed model in Eq. 2.51, and with the Cholesky decomposition of C_M, written as $C_M = LL^T$, the objective function for the MAP estimate can be written as:

$$\begin{aligned} O(m^l) = \frac{1}{2}(m^l - m_{\text{prior}})^T (LL^T)^{-1} (m^l - m_{\text{prior}}) \\ + \frac{1}{2}(g(m^l) - d_{\text{obs}})^T (C_D^{1/2} C_D^{1/2})^{-1} (g(m^l) - d_{\text{obs}}), \end{aligned} \tag{2.52}$$

which can also be written in the following form:

$$\begin{aligned} O(m^l) = \frac{1}{2}\{L^{-1}(m^l - m_{\text{prior}})\}^T \{L^{-1}(m^l - m_{\text{prior}})\} \\ + \frac{1}{2}\{C_D^{-1/2}(g(m^l) - d_{\text{obs}})\}^T \{C_D^{-1/2}(g(m^l) - d_{\text{obs}})\}, \end{aligned} \tag{2.53}$$

where L^{-1} denotes the inverse of L, and $C_D^{1/2}$ is a square root of the matrix C_D. $C_D^{-1/2}(g(m^l) - d_{\text{obs}})$ is defined as the dimensionless data mismatch. Thus, $O(\tilde{m})$ can be obtained as

$$O(\tilde{m}^l) \;=\; \frac{1}{2}(\tilde{m}^l)^T(\tilde{m}^l) \;+\; \frac{1}{2}\{C_D^{-1/2}(g(\tilde{m}^l) \;-\; d_{\text{obs}})\}^T\{C_D^{-1/2}(g(\tilde{m}^l) \;-\; d_{\text{obs}})\}. \quad (2.54)$$

By taking the gradient and the Hessian of $O(\tilde{m}^l)$ in Eq. 2.58, it is easy to obtain the LM search direction for the transformed model given in Eq. 2.6.

By using Eq. 2.51, although the model is dimensionless, and we are not worried about different types of model parameters, but still it is not clear how much a model parameter in the original space would change over an iteration. Following the work of Gao and Reynolds [13], we rescale the model parameters by defining the new rescaled variable $\hat{m}^l$ by

$$\hat{m}^l = D^{-1}(m^l - m_{prior}), \quad (2.55)$$

where D is a diagonal $N_m \times N_m$ matrix, and its ith diagonal entry is equal to the square root of the ith diagonal entry of C_M. In other words, the ith diagonal entry of D is the standard deviation of the prior uncertainty in the ith model parameter. With the new model defined in Eq. 2.55, the objective function can be written as:

$$\begin{aligned} O(m^l) = \frac{1}{2}(m^l - m_{\text{prior}})^T(D\hat{C}_M D)^{-1}(m^l - m_{\text{prior}}) \\ + \frac{1}{2}(g(m^l) - d_{\text{obs}})^T(C_D^{1/2}C_D^{1/2})^{-1}(g(m^l) - d_{\text{obs}}), \quad (2.56) \end{aligned}$$

which is equivalent to:

$$\begin{aligned} O(m^l) = \frac{1}{2}\{D^{-1}(m^l - m_{\text{prior}})\}^T\hat{C}_M^{-1}\{D^{-1}(m^l - m_{\text{prior}})\} \\ + \frac{1}{2}\{C_D^{-1/2}(g(m^l) - d_{\text{obs}})\}^T\{C_D^{-1/2}(g(m^l) - d_{\text{obs}})\}, \quad (2.57) \end{aligned}$$

where $\hat{C}_M$ denotes the prior correlation matrix. Thus, $O(\hat{m}^l)$ can be obtained with the following equation:

$$O(\hat{m}^l) = \frac{1}{2}(\hat{m}^l)^T \hat{C}_M^{-1}(\hat{m}^l) + \frac{1}{2}\{C_D^{-1/2}(g(m^l) - d_{\text{obs}})\}^T\{C_D^{-1/2}(g(m^l) - d_{\text{obs}})\}. \tag{2.58}$$

The gradient of $O(\hat{m}^l)$ can easily be obtained as

$$\nabla_{\hat{m}^l} O(\hat{m}^l) = \hat{C}_M^{-1}\hat{m}^l + D^{-1}G^T C_D^{-1/2}\{C_D^{-1/2}(g(m^l) - d_{\text{obs}})\}. \tag{2.59}$$

The Hessian for $O(\hat{m}^l)$ can be shown to be

$$H_l = \tilde{C}_M^{-1} + D^{-1}G_l^T C_D^{-1} G_l D^{-1}. \tag{2.60}$$

Thus, the equation of the GN search direction at iteration l for $\hat{m}^l$, is written as:

$$H_l \delta\hat{m}^{l+1} = -\nabla_{\hat{m}^l} O,$$

$$(\hat{C}_M^{-1} + D^{-1}G_l^T C_D^{-1} G_l D^{-1})\delta\hat{m}^{l+1} \;=\; -\hat{C}_M^{-1}\hat{m}^l - D^{-1}G_l^T C_D^{-1/2}\{C_D^{-1/2}(g(m^l) - d_{\text{obs}})\}. \tag{2.61}$$

With the Cholesky decomposition of the correlation matrix $\hat{C}_M$ given by $\hat{C}_M = \hat{L}\hat{L}^T$, Eq. 2.61 can be written as

$$(\hat{L}^{-T}\hat{L}^{-1} + D^{-1}G_l^T C_D^{-1} G_l D^{-1})\delta\hat{m}^{l+1} = -\hat{L}^{-T}\hat{L}^{-1}\hat{m}^l - D^{-1}G_l^T C_D^{-1/2}\{C_D^{-1/2}(g(\hat{m}^l) - d_{\text{obs}})\},$$

$$\begin{split}\hat{L}^{-T}(I_{N_m} + \hat{L}^T D^{-1}G_l^T C_D^{-1} G_l D^{-1}\hat{L})\hat{L}^{-1}\delta\hat{m}^{l+1} = \\ - \hat{L}^{-T}\hat{L}^{-1}\hat{m}^l - D^{-1}G_l^T C_D^{-1/2}\{C_D^{-1/2}(g(\hat{m}^l) - d_{\text{obs}})\},\end{split} \tag{2.62}$$

where $\hat{L}^{-T}$ denotes the inverse of the matrix $\hat{L}^T$. As $\delta\hat{m}^{l+1} = D^{-1}\delta m^{l+1}$, and $\hat{L}^{-1}D^{-1} = L^{-1}$, thus $\hat{L}^{-1}\delta\hat{m}^{l+1} = L^{-1}\delta m^{l+1} = \delta\tilde{m}^{l+1}$. Left multiplying both sides of Eq. 2.62 by

$\hat{L}^T$, the resulting equation is the GN search direction for the transformed model, $\tilde{m}$:

$$\begin{aligned}(I_{N_m} + \hat{L}^T D^{-1} G_l^T C_D^{-1} G_l D^{-1} \hat{L}) \hat{L}^{-1} \delta\hat{m}^{l+1} \\ = -\hat{L}^{-1}\hat{m}^l - \hat{L}^T D^{-1} G_l^T C_D^{-1/2}\{C_D^{-1/2}(g(\hat{m}^l) - d_{\text{obs}})\},\end{aligned}$$

$$(I_{N_m} + G_D^T G_D)\delta\tilde{m}^{l+1} = -\{\tilde{m}^l + G_D^T C_D^{-1/2}(g(\hat{m}^l) - d_{\text{obs}})\}. \tag{2.63}$$

Thus, minimizing the objective function for the rescaled vector of the model given by Eq. 2.55, is equivalent to minimizing the objective function for the transformed model, in Eq. 2.51. To use the rescaled model in Eq. 2.55, one can not directly use Eq. 2.61 with SVD parameterization. The minimization is performed in the transformed space, however after obtaining the search direction in the transformed space, $\delta\tilde{m}$, we calculate the rescaled search direction:

$$\delta\hat{m}^{l+1} = D^{-1}\delta m^{l+1} = D^{-1}L\delta\tilde{m}^{l+1}. \tag{2.64}$$

To calculate $\delta\hat{m}^{l+1}$, one can simply divide each entry of δm^{l+1} by the square root of the corresponding diagonal entry of C_M, and the matrix D is not formed.

If the infinity norm of $\delta\hat{m}^{l+1}$ corresponding to a search direction is greater than κ, we take a smaller step in the direction of δm^{l+1} with the following equation.

$$\delta m^{l+1} = \frac{\kappa}{\parallel \delta\hat{m}^{l+1} \parallel_{\text{inf}}} \delta m^{l+1}. \tag{2.65}$$

Note that $\delta\hat{m}^{l+1}$ is only used to obtain the infinity norm of the rescaled search direction. As the minimization is performed in the transformed space, we do not change the search direction, and we only take a smaller step in the direction of δm^{l+1} if necessary. Since $\delta m^{l+1} = L\delta\tilde{m}^{l+1}$, i.e., the transformation is linear, taking a step in the direction of δm^{l+1} in the actual space, is equivalent to take the same step in the direction of $\delta\tilde{m}^{l+1}$ in the transformed space. In the examples of this thesis, for the LM algorithm with SVD parameterization or SVD-EnRML, $\kappa = 2$ is used, and for the GN algorithm with

SVD-EnRML $\kappa = 1$ is used.

2.5 Reordering Model Parameters for Uncorrelated Layers

For a 3D reservoir with N_z layers, the covariance matrix can be very large, since there are many model parameters to be adjusted to history match data. However, if the model parameters of some layers are not correlated, one can reorder the vector of model parameters so that in the reordered m, the correlated parameter are grouped together. As an example, consider a 4 layer reservoir, where the model parameters in the 2 lower layers, have correlation with each other, and the model parameters in each of the 2 upper layers are uncorrelated with model parameters of any other layer. If the model parameters are porosity, horizontal and vertical log permeability of gridblocks of all layers, one can reorder the model parameters in the following form,

$$\begin{aligned} m &= \begin{pmatrix} m_1^T & m_2^T & m_{34}^T \end{pmatrix}^T \\ &= \begin{pmatrix} (\ln k_{h_1})^T & (\ln k_{z_1})^T & \phi_1^T & (\ln k_{h_2})^T & (\ln k_{z_2})^T & \phi_2^T & (\ln k_{h_{34}})^T & (\ln k_{z_{34}})^T & \phi_{34}^T \end{pmatrix}^T, \end{aligned} \tag{2.66}$$

where m_1 is the vector of the model parameters in the 1st layer, $\ln k_{h_1}$ denotes the vector of horizontal log permeability of gridblocks in the 1st layer; $\ln k_{z_1}$ denotes the vector of vertical log permeability of gridblocks in the 1st layer; ϕ_1 is the vector of porosity of gridblocks in the 1st layer. $\ln k_{h_{34}}$ denotes the vector of horizontal log permeability of gridblocks in the 3rd and the 4th layers. Other vectors are defined the same way.

In Eq. 2.66, the subscript 34 refers to layer 3 and 4, i.e.,

$$\ln k_{z_{34}} = \begin{pmatrix} (\ln k_{z_3})^T & (\ln k_{z_4})^T \end{pmatrix}^T. \tag{2.67}$$

The corresponding covariance matrix for vector m in Eq. 2.66 can be written as

$$C_M = \begin{pmatrix} C_{M_1} & 0 & 0 \\ 0 & C_{M_2} & 0 \\ 0 & 0 & C_{M_{34}} \end{pmatrix}. \tag{2.68}$$

In Eq. 2.68, each entry is a block matrix. The block matrices may have different dimensions. The matrix C_{M_1} has the following form

$$C_{M_1} = \begin{pmatrix} C_{\ln k_{h_1}} & C_{\ln k_{h_1}, \ln k_{z_1}} & C_{\ln k_{h_1}, \phi_1} \\ C_{\ln k_{z_1}, \ln k_{h_1}} & C_{\ln k_{z_1}} & C_{\ln k_{z_1}, \phi_1} \\ C_{\phi_1, \ln k_{h_1}} & C_{\phi_1, \ln k_{z_1}} & C_{\phi_1} \end{pmatrix}. \tag{2.69}$$

C_{M_2} has a similar form to C_{M_1}. $C_{M_{34}}$ which is the covariance matrix of model parameters in layers 3 and 4, has the following form:

$$C_{M_{34}} = \begin{pmatrix} C_{\ln k_{h_{34}}} & C_{\ln k_{h_{34}}, \ln k_{z_{34}}} & C_{\ln k_{h_{34}}, \phi_{34}} \\ C_{\ln k_{z_{34}}, \ln k_{h_{34}}} & C_{\ln k_{z_{34}}} & C_{\ln k_{z_{34}}, \phi_{34}} \\ C_{\phi_{34}, \ln k_{h_{34}}} & C_{\phi_{34}, \ln k_{z_{34}}} & C_{\phi_{34}} \end{pmatrix}. \tag{2.70}$$

Since in the SVD parameterization algorithms, we only need the products of G_D times a vector and G_D^T times a vector, the covariance matrix of Eq. 2.68 is not required to be formed, and only the nonzero block matrices are required. The dimensionless sensitivity matrix, G_D, for m in Eq. 2.66, can be written as

$$G_D = C_D^{-1/2} G C_M^{1/2} = \begin{pmatrix} C_D^{-1/2} G_1 C_{M_1}^{1/2} & C_D^{-1/2} G_2 C_{M_2}^{1/2} & C_D^{-1/2} G_{34} C_{M_{34}}^{1/2} \end{pmatrix}, \tag{2.71}$$

where G_1 is the sensitivity matrix of model parameters in the first layer, G_2 is the sensitivity matrix of model parameters in the second layer, and G_{34} is the sensitivity matrix of model parameters in the 2 bottom layers. Eq. 2.71 can be used to calculate G_D times a vector and G_D^T times a vector, which are required in the Lanczos method for computing a truncated SVD of G_D. This way, the square root or Cholesky decomposition of each of the covariance matrices can be directly used for computations, and there is no need to form the covariance matrix of Eq. 2.68. Note that in Eq. 2.71, in order to computed a truncated SVD, the dimensionless sensitivity matrix is not formed, and only the product of G_D times a vector and the product of G_D^T times a vector are required; thus the matrices $C_{M_1}^{1/2}$, $C_{M_2}^{1/2}$ and $C_{M_{34}}^{1/2}$ can be saved in sparse form.

2.6 SVD-EnRML with Levenberg-Marquardt and Gauss-Newton Algorithms

Tavakoli and Reynolds [37, 38] provided the derivation of the search direction for both the GN and LM algorithms with SVD parameterization; however, they only applied the LM algorithm and they did not use the GN algorithm to estimate the log permeability fields. In their examples, a good estimate of the true model was obtained with a fixed number of SVD parameters (20).

Dickstein et al. [8] used SVD parameterization in GN algorithm to condition the log permeability field to production and seismic data. In their synthetic example, in the absence of seismic data, when they used a fixed number of SVD parameters (25) in the GN algorithm, the resulting model was very rough, giving a poor representation of the reservoir, although the data matches were very good. However, they obtained good representations of the model with increasing N_{SVD} strategy, i.e., starting with a few SVD parameters and gradually increasing the number of SVD parameters as the GN iterations proceeds.

In summary, the results of Dickstein et al. [8] and Tavakoli and Reynolds [37, 38] show that with the LM algorithm one can obtain a good representation of the model with a fixed number of SVD parameters, while the GN algorithm may provide a rough and poor estimate of the model. Although, by gradually increasing the number of SVD parameters, both algorithms would provide a good estimate of the model. The literature also suggests that in the LM algorithm one should begin with a high initial value of γ [28, 2, 22].

The spectral analysis of the LM algorithm is provided in Oliver et al. [28]. They presented a theoretical argument that suggests a small LM parameter may result in large changes in the component of m that lies in the direction of eigenvectors of the GN Hessian associated with small eigenvalues. Noticing that in the transformed space, the eigenvectors of Hessian are the same as the right singular vectors of the dimensionless sensitivity matrix, G_D, the argument can be used to justify the failure of the GN algorithm with SVD parameterization in obtaining a reasonable estimate of the permeability field when a high number of SVD parameter are calculated at early iterations.

Although at late iterations of the LM algorithm, the LM parameter may have a small value, one should note that the data mismatch terms at late iterations are much smaller than early iterations of the algorithm. In addition, the important features of the model are resolved at early iterations. Further, we will discuss that even with a small LM parameter, e.g., 10, the eigenvectors associated with the small eigenvalues would have smaller components in the search direction.

As we will show, with a high value of the LM parameter, the major components of the search direction are along a few eigenvectors of the Hessian associated with the largest eigenvalues; the result is that at an early iteration of the LM algorithm, only a few important features of the true model are resolved; and as iteration proceeds, the weight of the eigenvectors associated with smaller eigenvalues increases in the search direction. Thus with the LM algorithm, the features of the true model are gradually resolved. Opposite to the LM algorithm, the GN search direction at an early iteration may have large components along eigenvectors of the Hessian associated with small eigenvalues. This means that the GN search direction tries to resolve many features of the true model from an early iteration. If the GN search direction, has large components along eigenvectors with very small eigenvalues, the resulting model can be very rough.

In this section, we discuss the difference of the GN and LM algorithms by looking at the search directions of the two methods. Further, the efficiency and applicability of SVD parameterization in the GN and LM algorithms are compared based on a synthetic example.

In SVD parameterization with GN algorithm, the search direction is given with the following equation:

$$\delta m_j^{l+1} = L \sum_{i=1}^{p} \left[\frac{-v_i^T \tilde{m}_j^l}{1+\lambda_i^2} - \frac{\lambda_i}{1+\lambda_i^2} u_i^T C_D^{-\frac{1}{2}} (g(m_j^l) - d_{\mathrm{uc},j}) \right] v_i, \tag{2.72}$$

while the search direction with the LM algorithm is as follows:

$$\delta m_j^{l+1} = L \sum_{i=1}^{p} \alpha_i v_i = L \sum_{i=1}^{p} \left[\frac{-v_i^T \tilde{m}_j^l}{1+\gamma_j^l+\lambda_i^2} - \frac{\lambda_i}{1+\gamma_j^l+\lambda_i^2} u_i^T C_D^{-\frac{1}{2}} (g(m_j^l) - d_{\mathrm{uc},j}) \right] v_i. \tag{2.73}$$

Comparing Eqs. 2.72 and 2.73, one can see the effect of the LM parameter on the search direction. With a high value of the LM parameter, e.g., 10^6, and a small singular value, the fraction $\frac{\lambda_i}{1+\gamma^l+\lambda_i^2}$ is very small. Note that the relative effect of the prior model for each singular vector in the two search directions are the same. The ratio of the coefficient of $v_i^T \tilde{m}_j^l$ to the coefficient of $u_i^T C_D^{-\frac{1}{2}}(g(m_j^l) - d_{\mathrm{uc},j})$ for the ith right singular vector in the GN search direction is:

$$(\frac{-1}{1+\lambda_i^2})/(\frac{-\lambda_i}{1+\lambda_i^2}) = \frac{1}{\lambda_i}, \tag{2.74}$$

and the same ratio in the LM search direction is:

$$(\frac{-1}{1+\gamma_j^l+\lambda_i^2})/(\frac{-\lambda_i}{1+\gamma_j^l+\lambda_i^2}) = \frac{1}{\lambda_i}. \tag{2.75}$$

Thus, the relative weight of the prior model for each of the right singular vectors is the same for the GN and LM search directions.

At early iterations of the GN or LM algorithm, the model mismatch term is small, while the data mismatch term is very large; therefore, the data mismatch term has the significant effect on the search direction. By taking the derivative of the term $\frac{\lambda}{1+\gamma^l+\lambda^2}$ with respect to λ, one can investigate the effect of the LM parameter on the search direction:

$$\frac{\partial}{\partial \lambda}\left\{\frac{\lambda}{1+\gamma^l+\lambda^2}\right\} = \frac{1+\gamma^l-\lambda^2}{(1+\gamma^l+\lambda^2)^2}, \tag{2.76}$$

therefore:

$$\frac{\partial}{\partial \lambda}\left\{\frac{\lambda}{1+\gamma^l+\lambda^2}\right\}\begin{cases} > 0, \text{ if } & 1+\gamma^l > \lambda^2 \\ < 0, \text{ if } & 1+\gamma^l < \lambda^2, \end{cases} \tag{2.77}$$

thus, if the value of γ^l is chosen such that $1+\gamma^l > \lambda_1^2$, where λ_1 is the first singular value, $\frac{\lambda}{1+\gamma^l+\lambda^2}$ is an increasing function of λ, which means that the right singular vectors corresponding to the largest singular values are likely to have the largest coefficients in the search direction; while in the GN search direction with $\gamma^l = 0$, and all retained singular values greater than 1, we have $1+0 < \lambda^2$, which means that the right singular vectors corresponding to the smallest retained singular values are likely to have the largest coefficients in the search direction. In addition, for a moderate value of γ^l, the maximum value of $\frac{\lambda}{1+\gamma^l+\lambda^2}$ is likely to be at a singular value, λ_i, that is close to $\sqrt{1+\gamma^l}$, as $\lambda = \sqrt{1+\gamma^l}$ gives a derivative of zero in Eq. 2.76.

If the LM parameter is small, say 100 or 10, for the singular values which are smaller than 1, the term $\frac{\lambda_i}{1+\gamma^l+\lambda_i^2}$ is approximately equal to $\frac{\lambda_i}{1+\gamma^l}$; while the term $\frac{\lambda_i}{1+\lambda_i^2}$ is close to λ_i. It means that with the LM parameter, the weight of the corresponding eigenvector is approximately reduced by a factor of γ^l.

In Example 1-2, the numerical values of $\frac{\lambda_i}{1+\gamma^l+\lambda_i^2}$ and $\frac{\lambda_i}{1+\gamma^l}$ are compared and the discussion is continued in Subsection 2.7.2.

As we will use the SVD parameterization with GN algorithm in Example 1-2, the step by step description of the algorithm is given here. In the following subsections, D denotes a diagonal $N_m \times N_m$ matrix, with the ith diagonal entry equal to the square root of the ith diagonal entry of C_M, as explained in Section 2.4. α denotes the step size. The step by step description of SVD-EnRML with the GN algorithm is introduced as it is used in Example 1 of this chapter; however we do not use it in any other example of this thesis.

Throughout this section l is the iteration index of the GN algorithm, where a truncated SVD of $G_{D,l}$ is computed using the Lanczos method.

2.6.1 Steps of Gauss-Newton Algorithm with SVD Parameterization

Here the specific steps of the GN algorithm with SVD parameterizaton for generating an estimate of the model is presented:

1. Set $l = 0$ and assign the initial guess of $m^0 = m_{\text{prior}}$ for the MAP estimate. Set the initial sv-cut, μ_1. Also, set the final value of the singular cutoff, μ_{min}, and the maximum number of SVD parameters ($N_{\text{SVD,max}}$) to be calculated by the Lanczos method.

2. Run the simulator with initial guess of the model parameters for the MAP estimate, and Compute $O(m^l)$ from Eq. 2.19.

3. Use the Lanczos method with the inputs sv-cut and $N_{\text{SVD,max}}$, to compute a truncated SVD of the dimensionless sensitivity matrix associated with the MAP estimate at iteration l, m^l. Set $\alpha = 1.0$.

4. (a) Compute the GN search direction:

$$\delta\tilde{m}^{l+1} = \sum_{i=1}^{p} \Big[\frac{-v_i^T \tilde{m}^l - \lambda_i u_i^T C_D^{-\frac{1}{2}}(g(m^l) - d_{obs})}{1 + \lambda_i^2}\Big] v_i, \tag{2.78}$$

then use

$$\delta m^{l+1} = L\delta\tilde{m}^{l+1}. \tag{2.79}$$

Compute $\delta\hat{m}^{l+1}$ by

$$\delta\hat{m}^{l+1} = D^{-1}\delta m^{l+1}. \tag{2.80}$$

For the GN algorithm, we use Eq. 2.65 with $\kappa = 1$; thus, if $\|\delta\hat{m}^l\|_{\text{inf}} > 1.0$ then set

$$\delta m^{l+1} = \delta m^{l+1}/\|\delta\hat{m}^{l+1}\|_{\text{inf}}. \tag{2.81}$$

(b) Use the following equation to calculate a proposed new update of the MAP estimate:

$$m_{\text{temp}}^{l+1} = m^l + \alpha \delta m^{l+1}. \tag{2.82}$$

Run the simulator with m_{temp}^{l+1} and calculate the objective function $O(m_{\text{temp}}^{l+1})$ given by Eq. 2.19.

(c) If $O(m_{\text{temp}}^{l+1}) < O(m^l)$, then set

$$m^{l+1} = m_{\text{temp}}^{l+1}. \tag{2.83}$$

Else if $O(m_{\text{temp}}^{l+1}) > O(m^l)$, then set $\alpha = \alpha/2$ and go to step b.

5. Check for convergence using the criteria in Eqs. 2.37 and 2.38. If the algorithm has not converged, increase iteration index l by 1, replace sv-cut by $\max\left\{\frac{\text{sv-cut}}{2}, \mu_{\min}\right\}$ and go to step 3; otherwise, if the convergence criteria are satisfied, go to step 6.
6. End.

2.6.2 Steps of SVD-EnRML with Gauss-Newton

Here the steps of the GN algorithm with SVD parameterizaton for simulation of rock property fields is presented. The steps of this algorithm are the same as the SVD-EnRML-SMM algorithm, with a difference that the LM algorithm is replaced with a simple line search.

1. Set $l = 0$ and assign the initial guess of $m_j^0 = m_{\text{uc,j}}$ for $j = 1, 2, \cdots N_e$ for RML realizations and an initial guess of $m_0^0 = m_{\text{prior}}$ for the MAP estimate. Set the initial value of the step size, $\alpha_j^l = 1.0$ for $j = 0, 1, \cdots N_e$. Set the initial sv-cut and sv-cut after the convergence of the MAP estimate, μ_1 and μ_2, respectively. Set the final value of the singular cutoff, $\mu_{\min}$ and the maximum number of SVD parameters ($N_{\text{SVD,max}}$) to be calculated by the Lanczos method; also set $\alpha_{\min}$ and

$s_{\max}$, where $\alpha_{\min}$ is the minimum step size and $s_{\max}$ is the maximum iterations in a line search.

2. Calculate the initial values of the objective functions by running the simulator to the final time with initial guesses of the model parameters for the MAP estimate and all realizations. Compute $O_j(m_j^l)$ with Eq. 2.20 for $j = 1, 2, \cdots N_e$ and compute $O(m_0^l)$ from Eq. 2.19.

3. If $O_{N,\text{MAP}}(m_0^l) < 1 + 5\sqrt{2/N_d}$ or if Eqs. 2.37 and 2.38 are both satisfied, i.e., the MAP estimate has converged, use the Lanczos method to compute a truncated SVD of the dimensionless sensitivity matrix pertaining to m_{base}^l where

$$O_{j_{\text{base}}}(m_{\text{base}}^l) = \max_{1 \le j \le N_e} O_j(m_j^l). \tag{2.84}$$

   In this case set α_{base}^l (α^l pertaining to m_{base}^l) to a 1.0, i.e., take a full step for the base realization.

   Otherwise, use the Lanczos method to compute a truncated SVD of the dimensionless sensitivity matrix associated with the MAP estimate at iteration l, m_0^l.

4. For $j = 0, 1, 2, \cdots N_e$

   (a) Set $s = 0$, where s is the iteration index in the line search for decreasing the objective function of the jth realization, at a fixed l. If $\alpha_j^l < 0.1$, set $\alpha_j^l = 0.1$

   (b) Compute the GN search direction, $\delta\tilde{m}_j^{l+1}$, using Eq. 2.78 for the MAP estimate, or using the following equation for an RML realization:

$$\delta\tilde{m}_j^{l+1} = \sum_{i=1}^{p} \left[\frac{-v_i^T \tilde{m}_j^l - \lambda_i u_i^T C_D^{-\frac{1}{2}}(g(m_j^l) - d_{\text{uc, j}})}{1 + \lambda_i^2} \right] v_i, \tag{2.85}$$

   then use

$$\delta m_j^{l+1} = L\delta\tilde{m}_j^{l+1}. \tag{2.86}$$

Compute $\delta\hat{m}_j^{l+1}$ by

$$\delta\hat{m}_j^{l+1} = D^{-1}\delta m_j^{l+1}. \tag{2.87}$$

For the GN algorithm, we use Eq. 2.65 with $\kappa = 1$; thus, if $\|\delta\hat{m}_j^l\|_{\text{inf}} > 1.0$ then set

$$\delta m_j^{l+1} = \delta m_j^{l+1}/\|\delta\hat{m}_j^{l+1}\|_{\text{inf}}. \tag{2.88}$$

(c) Use the following equation to calculate a proposed new update of the jth realization:

$$m_{\text{temp},j}^{l+1} = m_j^l + \alpha_j^l\delta m_j^{l+1}. \tag{2.89}$$

Run the simulator with $m_{\text{temp},j}^{l+1}$ and calculate the objective function $O_j(m_{\text{temp},j}^{l+1})$.

(d) If $O_j(m_{\text{temp},j}^{l+1}) \geq O_j(m_j^l)$, then

i. If $s = s_{\max}$ or $\alpha_j^l = \alpha_{\min}$, and $m_j \neq m_{\text{base}}$, then set

$$\alpha_j^{l+1} = \alpha_j^l,$$

$$m_j^{l+1} = m_j^l, \tag{2.90}$$

and terminate the line search for the jth realization at iteration l. Note that the jth realization is not updated at this iteration.

ii. Else, if $s < s_{\max}$ and $\alpha_j^l > \alpha_{\min}$, or $m_j = m_{\text{base}}$, set

$$\alpha_j^l = \max(\frac{\alpha_j^l}{2}, \alpha_{\min}),$$

$$s = s + 1, \tag{2.91}$$

and return to step (c).

Else, if $O_j(m^{l+1}_{\text{temp},j}) < O_j(m^l_j)$, set

$$m^{l+1}_j = m^{l+1}_{\text{temp},j},$$

$$\alpha^{l+1}_j = \min(2\alpha^l_j, 1). \tag{2.92}$$

5. Check for convergence using the criteria in Eqs. 2.37 and 2.38. If the algorithm has not converged, increase iteration index l by 1, replace sv-cut by $\max\{\frac{\text{sv-cut}}{2}, \mu_{\min}\}$ and go to step 3; otherwise, go to step 6, i.e., terminate the algorithm.

6. End.

In our implementation, we used $\alpha_{\min} = 0.005$.

2.7 Example 1

This example pertains to a two-dimensional horizontal reservoir model with 28×30 uniform grid. True porosity and log permeability fields are shown in Fig. 2.1. The key geostatistical parameters used to generate the true model are listed in Table 2.1. In this table, ϕ_{mean} and $[\ln(k)]_{\text{mean}}$, denote the prior mean of porosity and log-permeability, respectively. The standard deviations of porosity and log-permeability are denoted by σ_ϕ and $\sigma_{\ln(k)}$, respectively; $\rho_{\phi,\ln(k)}$ denotes the correlation coefficient between porosity and log-permeability; α is the angle measured counterclockwise from the x-axis to the principal correlation direction of the covariance function; r_1 is the correlation range in the principal direction and r_2 is the correlation range in the orthogonal direction.

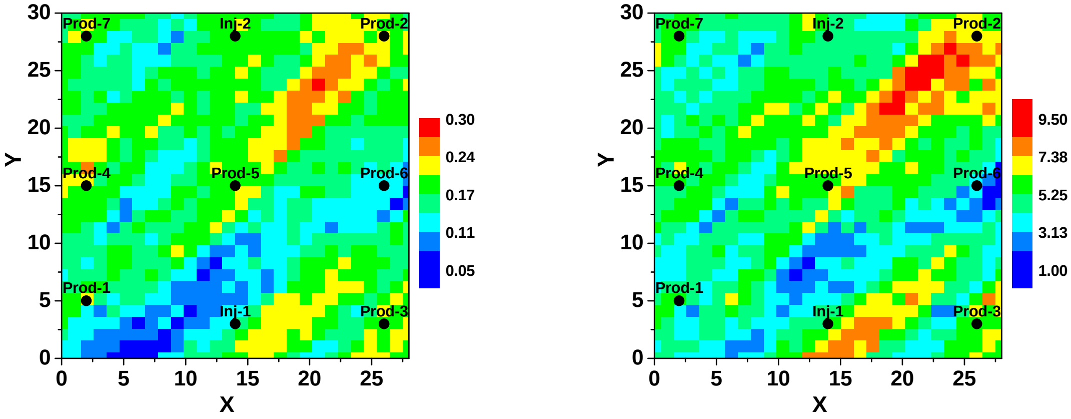
The gridblock dimensions are:

$$\triangle x = \triangle y = 200\,\text{ft},\ \triangle z = 10\,\text{ft}.$$

The initial reservoir pressure is 4800 psi. Initially the reservoir is at irreducible water saturation. There are 7 producers and 2 injectors in this reservoir. Table 2.2 shows the summary of well controls. The total history matching period is 1800 days.

Table 2.1: Geostatistical parameters of Example 1.

Parameters	Values
ϕ_{mean}	0.15
$[\ln(k)]_{\text{mean}}$	4.50
σ_ϕ	0.0016
$\sigma_{\ln(k)}$	1.414
$\rho_{\phi,\ln(\text{k})}$	0.80
α	45^o
r_1	$25\triangle x$
r_2	$7\triangle x$



(a) True porosity field

(b) True log permeability field

Figure 2.1: True porosity and log permeability fields, Example 1.

At 1800 days, only Prod-1, Prod-3 and Prod-7 have experienced water breakthrough. Observed data include the flowing bottom hole pressure (BHP) of all wells and water rates of producers which had water breakthrough, at 30 day intervals.

Synthetic observed data are generated by adding Gaussian random noise to the true data, where the true data are the simulator output when it is run with the true model. The standard deviation of noise (measurement error) is 2% of rates for water rate data and 3 psi for pressure data, i.e., $\sigma_{\text{BHP}} = 3$ psi and $\sigma_{q_w} = 0.02q_w$. The minimum measurement error for water rate is specified to 0.5 STB/D while the maximum measurement error is 3 STB/D.

Table 2.2: Well controls of Example 1. Total liquid rate is specified at producers and injection rate is specified at injectors. The rates are in STB/D.

Time Period	Inj-1	Inj-2	Prod-1	Prod-2, 3	Prod-4,5	Prod-6	Prod-7
0 – 90 Days	450	450	300	300	200	200	200
90 – 5000 Days	1050	1000	350	300	300	200	300

Model parameters include porosity and log permeability of all gridblocks. Fluid properties are assumed to be known. The number of model parameters and observed data are

$$N_m = 2N_x \times N_y = 1680, \; N_d = 720.$$

In Example 1-1 and 1-2, the objective is to simulate porosity and log permeability fields with SVD parameterization in LM and GN algorithms and compare the results in terms of the final values of objective functions, the rate of convergence and details of search direction. Only 5 conditional realizations in addition to the MAP estimate are generated. In both the LM and GN algorithms, the change in the objective functions are controlled over the first two iterations, such that

$$\frac{O(m_j^l) - O(m_j^{l+1})}{O(m_j^l)} \leq 0.5, \tag{2.93}$$

for $j = 1, \cdots N_e$. In both the LM and GN algorithms, the parameter sv-cut is used to control the number of SVD parameters at each iteration with a maximum of 55 singular triplets. The same unconditional realizations are used in the two algorithms.

For the LM algorithm, SVD-EnRML-SMM with the step by step description given this chapter, is used. For the GN algorithm, we follow the step by step description given in Subsection 2.6.2.

Throughout, μ_1 refers to the initial value of sv-cut in SVD-RML algorithms; μ_2 refers to the value of sv-cut at the first iteration after the convergence of the MAP estimate; $\mu_{\min}$ is the minimum sv-cut. $N_{\text{SVD,max}}$ is the maximum number of SVD parameters which can be calculated at an iteration.

In tables or figures, “MI” refers to SVD-EnRML-MI; “SMM” refers to SVD-EnRML-SMM, “AG” refers to SVD-EnRML-AG, “Sim.” refers to reservoir simulation runs; and Direct refers to direct method (gradient simulator method). The computational costs are compared based on equivalent simulation runs which is denoted by “Equ. Sim. Runs”. In computing the computational cost in terms of equivalent simulation runs, 4 adjoint solutions or 4 direct method applications are roughly assumed to be equivalent to 1 reservoir simulation run. N_{iter} refers to the number of iterations until convergence.

We consider two cases. In the first case, we use $\mu_1 = 0.5$. In the second case, we choose $\mu_1 = 0.01$. In both cases, the results of the GN and LM algorithms are compared.

Note that in all the figures of this example, at iteration l, the truncated SVD of $G_{D,l}$ is updated by the Lanczos method.

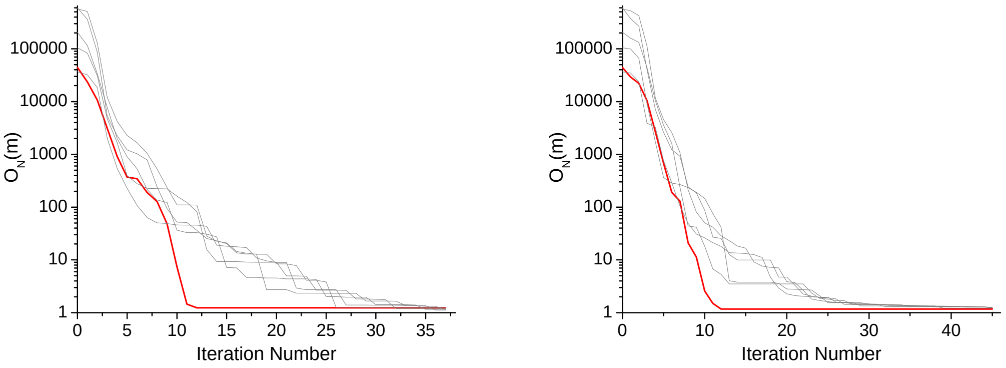
2.7.1 Example 1-1

Table 2.3 shows the input parameters of the algorithms. Fig. 2.2 shows the values of normalized objective functions of all realizations versus iterations for the two algorithms, and Fig. 2.3 shows the values of the $\max\{O_N(m)\}$ versus iterations of the two algorithms. SVD-EnRML with GN algorithm converged in 37 iterations, while the SVD-EnRML-SMM algorithm converged in 45 iterations. At convergence of both algorithms, all realizations have normalized objective function of less than $1 + 5\sqrt{2/N_d} = 1.263$.

One may conclude that the GN has better convergence properties as the algorithm has converged in fewer iterations; however, it should also be noticed that in Fig. 2.3, the value of $\max\{O_N(m)\}$, between iteration $l = 8$ and $l = 33$ for the LM algorithm is less than the one from the GN algorithm.

Table 2.3: Input parameters of the SVD-EnRML with GN and LM algorithm, Example 1-1.

Algorithm	μ_1	μ_2	$N_{\text{SVD,max}}$	$\mu_{\min}$
GN	0.5	0.05	55	0.0002
LM	0.5	0.05	55	0.0002



(a) GN

(b) LM

Figure 2.2: $O_N(m)$ of realizations versus iterations of SVD-EnRML with the GN and the LM algorithms, Example 1-1.

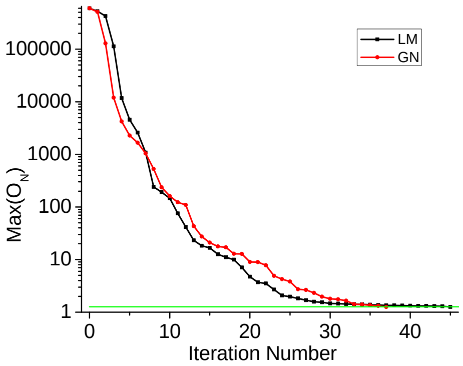
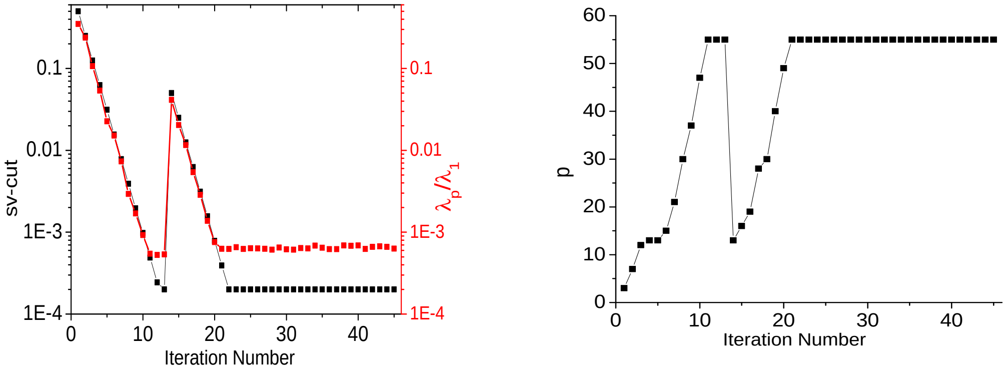


Figure 2.3: Max$\{O_N(m)\}$ versus iterations of SVD-EnRML with GN and the LM algorithms, the green line shows the value of $1 + 5\sqrt{2/N_d} = 1.263$, Example 1-1.

The number of retained SVD parameters versus iteration of the SVD-EnRML-SMM are shown in Fig. 2.4(b). The jump in this figure happens at the convergence of the algorithm, where we set sv-cut= $\mu_2 = 0.05$. Fig. 2.4 shows the values of sv-cut and λ_p/λ_1 versus iteration of the SVD-EnRML-SMM algorithm. According to this figure, at late iterations the Lanczos algorithm computed 55 singular triplets of $G_{D,l}$, since we specified $N_{\text{SVD,max}} = 55$. Note that the value of λ_{55}/λ_1 is greater than the minimum sv-cut which is 0.0002.



(a) sv-cut

(b) N_{SVD}

Figure 2.4: Values of sv-cut and λ_p/λ_1, and the number of retained singular triplets versus iterations of SVD-EnRML-SMM, Example 1-1.

This example shows that the SVD-EnRML with GN algorithm and gradually increasing the number of SVD parameters, can be used to efficiently generate the MAP estimate and RML realizations. In both the GN and LM algorithms, the MAP estimate converged in the same number of iterations, with comparable values of normalized objective functions.

In Example 1-2, we compare the two algorithms when using a high number of singular triplets at early iterations.

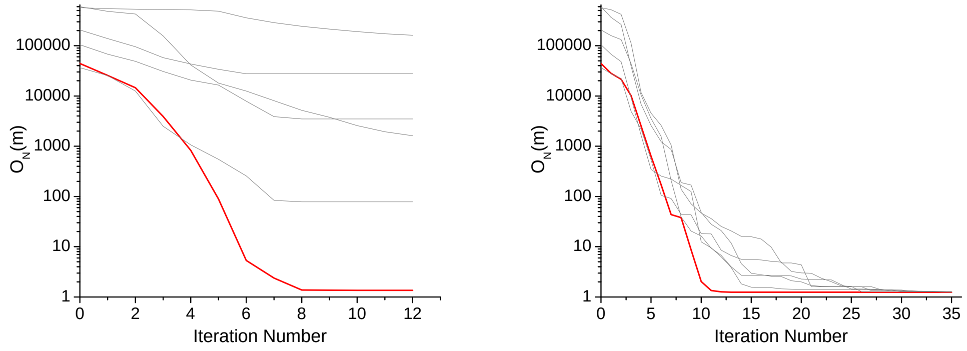
2.7.2 Example 1-2

In this case, both the SVD-EnRML-SMM algorithm and SVD-EnRML with GN algorithm start with $\mu_1 = 0.01$, i.e., more singular triplets are calculated at early iterations compared to Example 1-1. We already know that computing many singular triplets at early iterations is neither necessary, nor beneficial; however our objective is to compare the results of the GN and LM algorithms in this situation.

Table 2.4: Input parameters of SVD-EnRML with GN and LM algorithms, Example 1-2.

Algorithm	μ_1	μ_2	$N_{\text{SVD,max}}$	μ_{min}
GN	0.01	0.01	55	0.0002
LM	0.01	0.01	55	0.0002

Table 2.4 shows the input parameters of the algorithms. The SVD-EnRML with GN converged in 12 iterations, as both Eqs. 2.37 and 2.38 were satisfied. At convergence, $O_N(m)$ of the MAP estimate from GN is slightly greater than $1+\sqrt{2/N_D} = 1.264$. With the LM method, the algorithm converged in 35 iterations, with small values of $O_N(m)$ for all realizations.



(a) GN

(b) LM

Figure 2.5: $O_N(m)$ of realizations versus iterations of SVD-EnRML with GN and LM algorithms, Example 1-2.

Table. 2.5 contains the final values of $O_N(m)$ of realizations at convergence of the algorithms. Fig. 2.5 shows $O_N(m)$ of all realizations versus iterations for the GN and LM algorithms. The convergence properties of the LM algorithm over iterations in this case that more singular triplets were calculated at early iterations, is very similar to Example 1-1. However, convergence properties of the GN algorithm for RML realizations is much worse than Example 1-1. At convergence of the GN algorithm, only the MAP estimate has O_N close to a reasonable value.

In this example, the rate of decrease of objective functions of RML realizations in SVD-EnRML with GN is very slow at early iterations. One reason can be that by retaining a high number of SVD parameters at early iterations, roughness is added to the models. Note that we are using a truncated SVD of $G_{D,l}$ computed for the MAP estimate for decreasing $O_j(m)$ of RML realizations; at early iterations, the right singular vectors corresponding to small singular values of $G_{D,l}$ computed for the MAP estimate, contain the less important features of the model, and they are very noisy. The GN search direction composed of these noisy vectors for an RML realization adds roughness to the model.

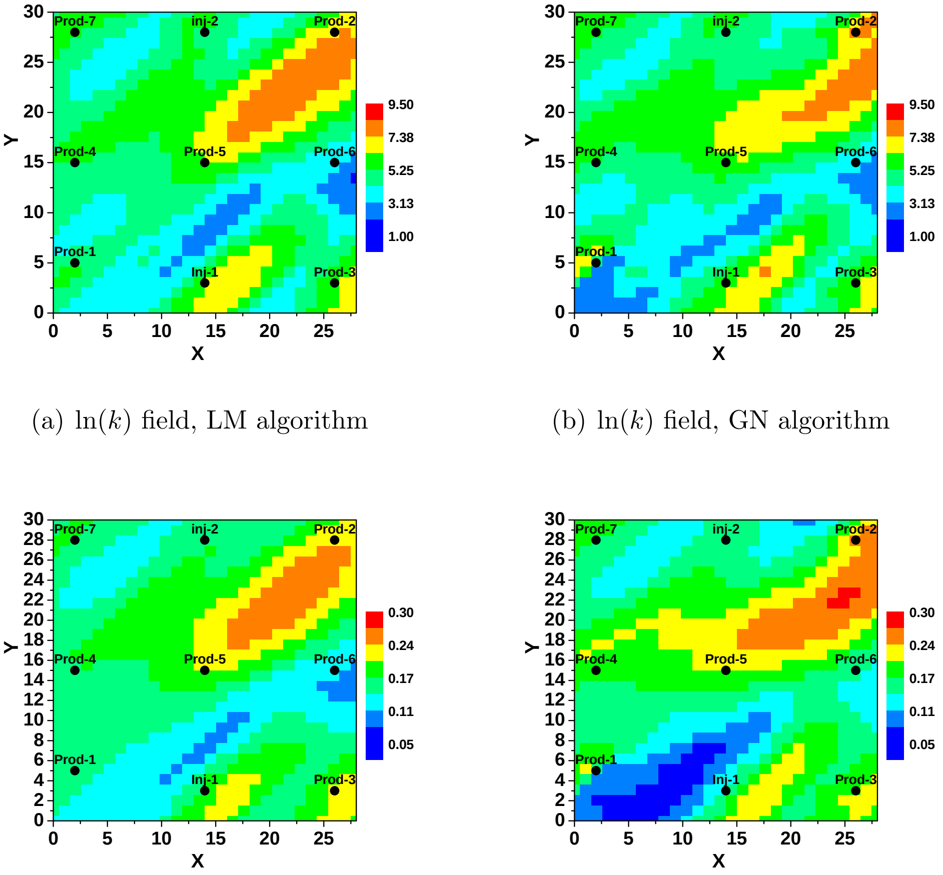
Table 2.5: Final values of normalized objective functions at convergence of the GN and LM algorithms with SVD parameterization, Example 1-2.

Algorithm	$j = 0$	$j = 1$	$j = 2$	$j = 3$	$j = 4$	$j = 5$
GN	1.355	1612.9	3484.2	78.2	161158.0	27540.1
LM	1.244	1.280	1.280	1.272	1.255	1.257

Table 2.6: Comparison of objective functions of the MAP estimate at convergence, generated from the GN and LM algorithms with SVD parameterization, Example 1-2.

Algorithm	$O_d(m)$	$O_m(m)$	$O_N(m)$
GN	398	90.14	1.355
LM	413.7	34.018	1.244

Fig. 2.6 shows the log permeability and porosity fields of the MAP estimate with the two algorithms. The rock property fields obtained from both the GN and LM algorithms, display the main characteristic of the true rock property fields shown in Fig. 2.1. Unlike the results of Dickstein et al. [8], the MAP estimate generated from SVD parameterization in GN algorithm, is not rough and it has captured the main features of the true model.



(a) ln(k) field, LM algorithm

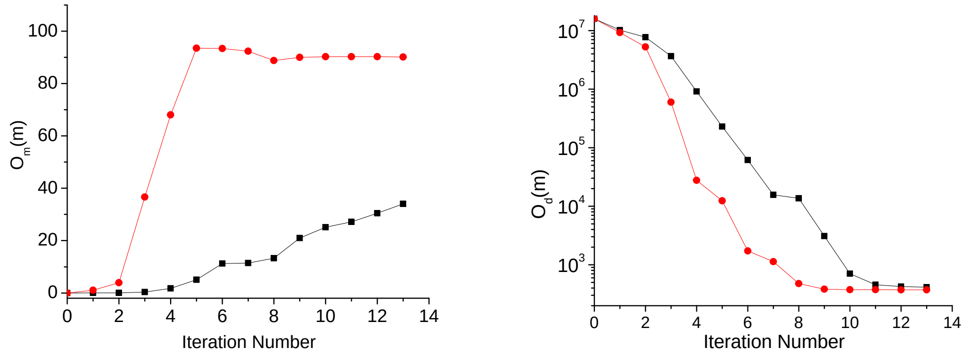
(b) ln(k) field, GN algorithm

(c) Porosity field, LM algorithm

(d) Porosity field, GN algorithm

Figure 2.6: Log permeability and porosity fields of the MAP estimate from the LM and GN algorithms with SVD parameterization, Example 1-2.

Fig. 2.7 shows the model mismatch and data mismatch terms of the objective function of the MAP estimate versus iterations of the two algorithms. It can be seen that in the GN algorithm with SVD parameterization, the model mismatch term increased to a high value in a few iterations and then it has small changes, while in the LM algorithm, the model mismatch term increases by a slow rate during all iterations. Table 2.6 shows the values of model mismatch part and data mismatch part of the objective function of the MAP estimate for the two algorithms. The model mismatch term of the MAP estimate with the GN algorithm is noticeably higher than the one from the LM algorithm.



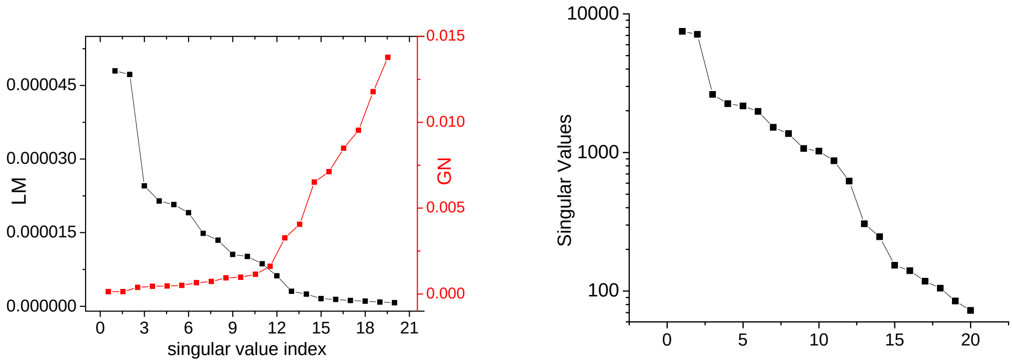
(a) $O_m(m)$

(b) $O_d(m)$

Figure 2.7: $O_d(m)$ and $O_m(m)$ of the MAP estimate versus iterations of SVD-EnRML with the GN (red) and LM (black) algorithms, Example 1-2.

At the first iteration, with sv-cut of 0.01, 20 singular triplets were calculated. As can be seen in Eq. 2.73, the ratio $\frac{\lambda_i}{1+\gamma^l+\lambda_i^2}$ has a major effect on the search direction. Note that for a large γ^l and small λ_i, the ratio $\frac{\lambda_i}{1+\gamma^l+\lambda_i^2}$ would have a small value. Here, some details of the search direction for the MAP estimate at the first iteration of the GN and LM algorithms are investigated.

In the LM algorithm, a value of 10^7 was used for γ^0. In Fig. 2.8(a) the left column shows the value of $\frac{\lambda_i}{1+\gamma_1+\lambda_i^2}$ for the ith largest singular value with $\gamma_1 = 10^7$, while the right column shows the value of $\frac{\lambda_i}{1+\lambda_i^2}$. Fig. 2.8(b) shows the singular values of G_D calculated at the first iteration. The largest 2 singular values are 7474 and 7108, and then the 3rd singular value is 2621 which is much lower than λ_1 and λ_2.



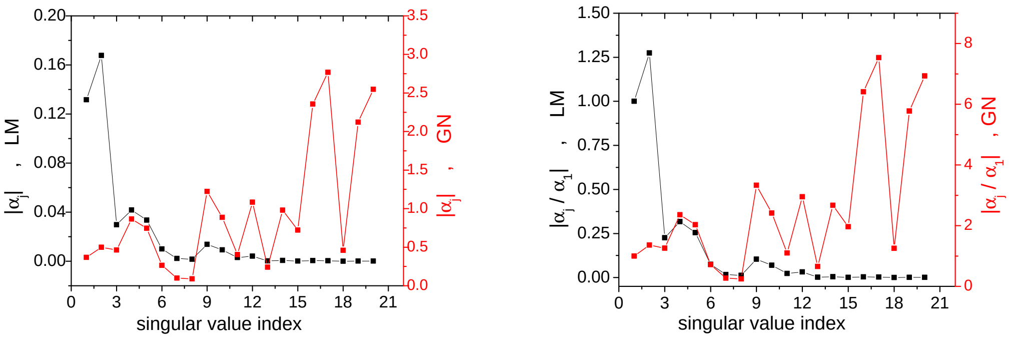
(a) $\frac{\lambda_i}{1+\gamma_1+\lambda_i^2}$

(b) λ_i

Figure 2.8: Singular values, λ_i, and $\frac{\lambda_i}{1+\gamma_1+\lambda_i^2}$ at the first iteration. For GN, the column shows the values of $\frac{\lambda_i}{1+\lambda_i^2}$. Example 1-2.

Fig. 2.9(a) shows the absolute values of $\alpha_i, i = 1, \cdots p$ from Eq. 2.73. Note that α_i is the coefficient of v_i in the expansion of $\delta\tilde{m}$ in terms of the right singular vectors, $v_i, i = 1, \cdots p$. Fig. 2.9(b) shows the ratio $|\alpha_i/\alpha_1|$ for the retained singular values. α_1 is the coefficient of v_1 corresponding to the largest singular value. This figure shows that the LM search direction has its main components in the direction of v_1 and v_2, corresponding to the two largest singular values. In addition, $|\alpha_i/\alpha_1|$ for all $i > 12$, is less than 0.005. For the smallest retained λ which is the $20th$ singular value, the ratio $|\alpha_{20}/\alpha_1|$ is 0.001. Note that the first 2 singular values in fig. 2.8(b) form a cluster. Returning to Fig. 2.9(b), we see that the GN search direction has large components along several right singular vectors, including the right singular vectors corresponding to the small singular values. This is the opposite of the LM search direction. If we were to retain more singular triplets, LM search direction will not highly change, as the right singular vectors corresponding to the small singular values would have very small components in the search direction, $\delta\tilde{m}$, but the GN search direction is very sensitive to the number of retained singular triplets.

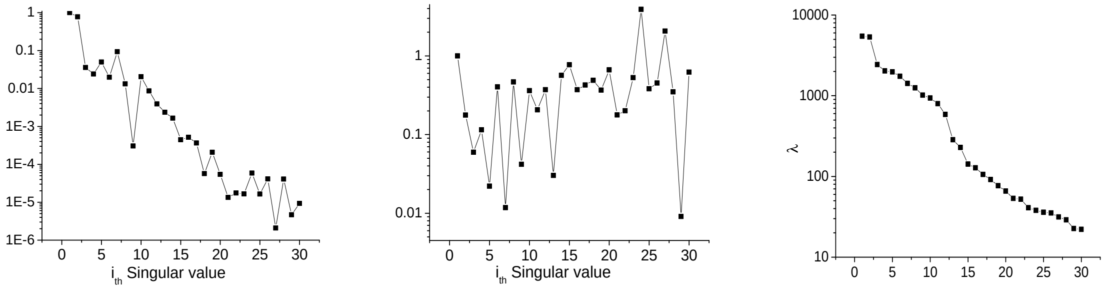
Now we consider the search direction of an RML realization. Fig. 2.10(c) shows the singular values of G_D at iteration $l = 3$. Figs. 2.10(a) and 2.10(b) show the absolute



(a) $|\alpha_i|$

(b) $|\alpha_i|/|\alpha_1|$

Figure 2.9: Parameters of the search direction at the first iteration of SVD-EnRML with the GN and LM, corresponding to the MAP estimate. The index in the horizontal axis represents ith largest singular value, Example 1-2.



(a) $|\alpha_i|/|\alpha_1|$, LM algorithm

(b) $|\alpha_i|/|\alpha_1|$, GN algorithm

(c) Singular Values

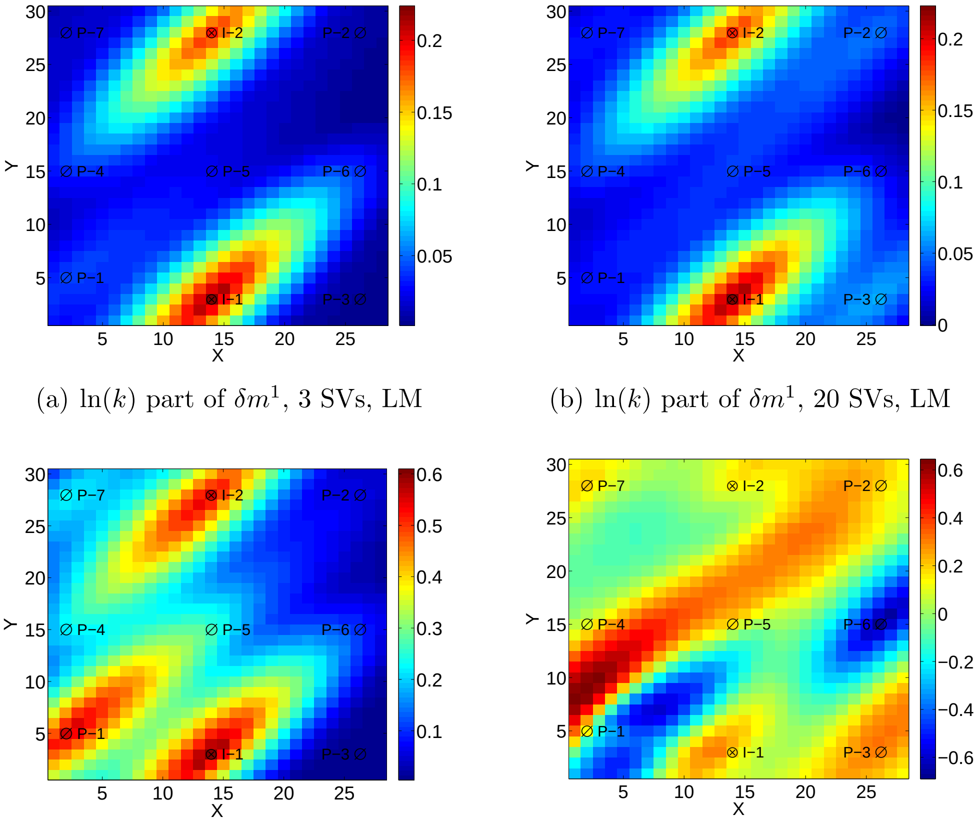
Figure 2.10: (a,b) Values of $|\alpha_i|/|\alpha_1|$ for search direction, $\delta\tilde{m}_j^l$, for realization $j = 2$ at iteration $l = 3$. (c) Retained singular values of G_D at iteration 3, corresponding to the MAP estimate. Example 1-2.

values of α_i/α_1 for the ith singular value, in the search direction, $\delta\tilde{m}$ of realization #2 for the GN and LM algorithms. The value of γ_2 was 10^8 in calculating the search direction at this iteration of the LM algorithm. Note that in Fig. 2.10(a), for the LM search direction, the values are on a log scale and the absolute values α_i/α_1 are all less than 0.001 for $i > 14$, while in Fig. 2.10(b), for the GN search direction, some α_i, corresponding to small singular values, are even higher than α_1.

Here, we investigate more on the effect of the number of retained singular triplets on the search direction. In Example 1-1, at the first iteration of the algorithm ($l = 1$), with sv-cut= 0.5, 3 singular triplets were retained. The $\ln(k)$ part of the search direction for the MAP estimate in the actual space, δm^1, with the LM and GN algorithms are shown in Figs. 2.11(a) and 2.11(c), respectively.

In Example 1-2, at the first iteration of the algorithm ($l = 1$), with sv-cut= 0.01, 20 singular triplets were retained. The $\ln(k)$ part of the search direction for the MAP estimate in the actual space, δm^1, with the LM and GN algorithms are shown in Figs. 2.11(b) and 2.11(d), respectively. Comparing Figs. 2.11(b) and 2.11(a), one can see that the search direction for the LM algorithm with 3 singular triplets, look very similar to the one with 20 singular triplets. Opposite to the LM search direction, the GN search direction with 3 singular triplets in Fig. 2.11(c), is very different from the one with 20 singular triplets, in Fig. 2.11(d).

As one can see in Figs. 2.11(b) and 2.11(a), the LM search direction is mainly changing the permeability of two important regions of the true model, which are around the two injectors. As can be seen in Fig. 2.15, these two high permeability zones are from the two eigenvectors associated with the largest eigenvalues. This is in agreement with the results of Fig. 2.9(b), where only the first 2 eigenvectors have large components in the search direction of the MAP estimate at $l = 1$. Except these two eigenvectors, the search direction is not mainly affected by other eigenvectors which contain high frequency components. This effectively means that the LM algorithm is gradually resolving the important characteristics of the model. The GN search direction in Figs. 2.11(d)



(a) ln(k) part of δm^1, 3 SVs, LM

(b) ln(k) part of δm^1, 20 SVs, LM

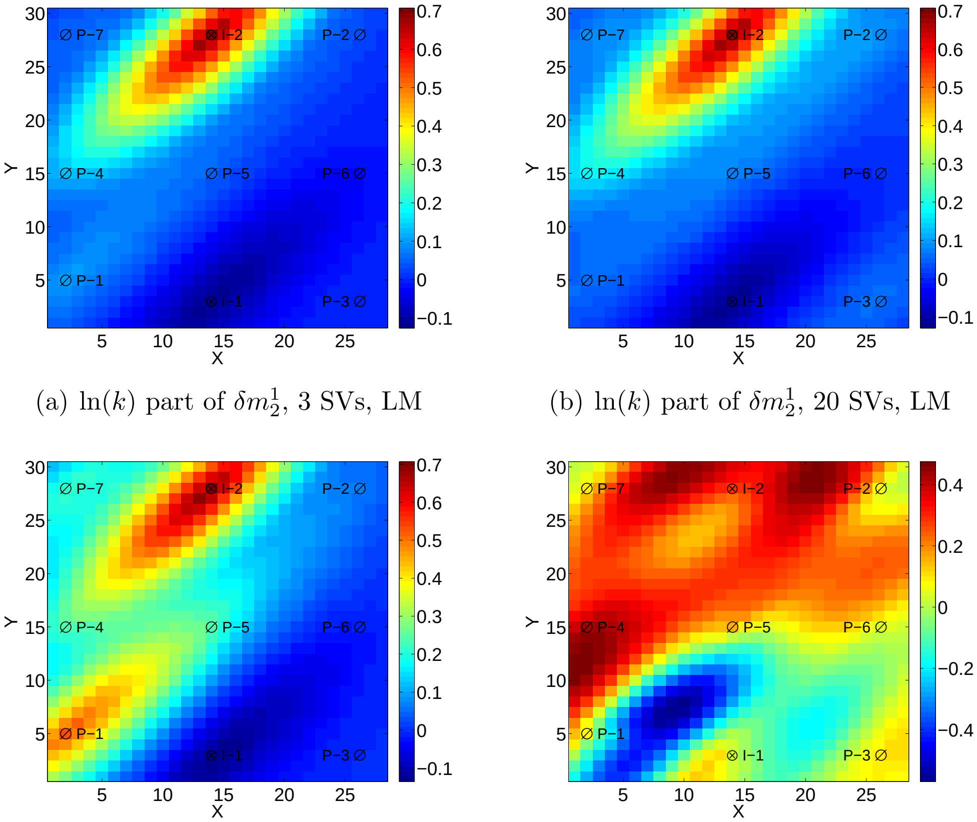
(c) ln(k) part of δm^1, 3 SVs, GN

(d) ln(k) part of δm^1, 20 SVs, GN

Figure 2.11: ln(k) part of δm^1 in the LM and GN algorithms with 3 and 20 singular triplets, Example 1-2.

and 2.11(c) is affected by all the eigenvectors that were used in calculating δm^{l+1}, and the algorithm is trying to change many parts of the model at the same time.

The search direction for an RML realization ($j = 2$) with both the GN and LM algorithms at iteration $l = 1$ of the algorithms are shown in Fig. 2.12. As can be seen in this figure, the search direction in the LM algorithm with retaining either 3 or 20 singular triplets look very similar, however the search direction in the GN algorithm with 3 singular triplets is far different from the one with 20 singular triplets.



(a) ln(k) part of δm_2^1, 3 SVs, LM

(b) ln(k) part of δm_2^1, 20 SVs, LM

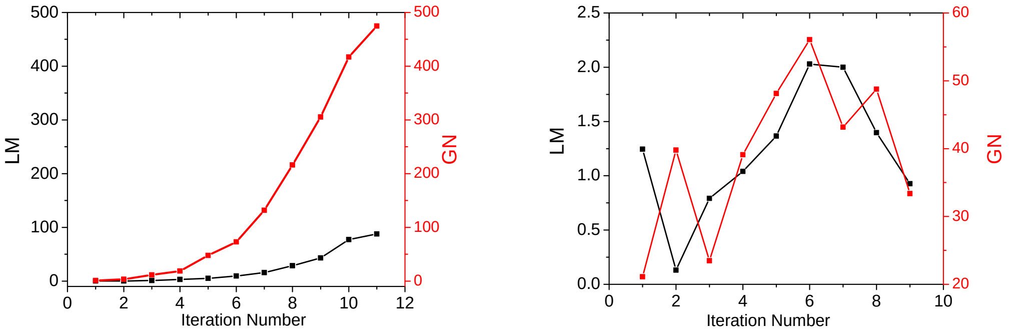
(c) ln(k) part of δm_2^1, 3 SVs, GN

(d) ln(k) part of δm_2^1, 20 SVs, GN

Figure 2.12: ln(k) part of δm_2^1 in the LM and GN algorithms with 3 and 20 singular triplets, Example 1-2.

We also would like to compare the model mismatch terms versus iterations for the two algorithms. However, since both algorithms are performed in a transformed space, we investigate the behavior of the norm of $\delta\tilde{m}$. Since the right singular vectors are orthogonal, this norm can be computed as

$$\|\delta\tilde{m}^{l+1}\| = \|V_p\alpha^{l+1}\| = \sqrt{(\alpha^{l+1})^T V_p^T V_p(\alpha^{l+1})} = \sqrt{(\alpha^{l+1})^T(\alpha^{l+1})}. \tag{2.94}$$



(a) $O_m(m)$ versus iterations

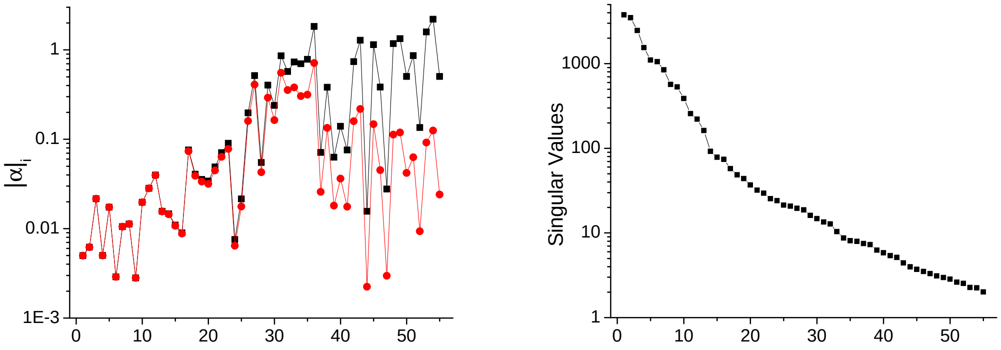
(b) $\|\delta\tilde{m}\|$ versus the first 9 iterations

Figure 2.13: $O_m(m)$ and $\|\delta\tilde{m}\|$ of realization #2 versus iterations of SVD-EnRML-SMM with GN and LM algorithms, Example 1-2.

Fig. 2.13(b) shows $\|\delta\tilde{m}^{l+1}\|$ of realization #2 versus iterations l in the GN and LM algorithms. Although not shown, the infinity norm of $\delta\hat{m}_2^{l+1}$, given by Eq. 2.65, in the GN was greater than 6 during the first 9 iterations of the GN, while the infinity norm of $\delta\hat{m}_2^{l+1}$ in the LM algorithm was less than 2 during the first 9 iterations. It means that in the GN algorithm, to avoid large changes in model parameters, a small step along the search direction for this realization was used at all these 9 iterations. Fig. 2.13(a) shows the model mismatch term for realization #2 versus the first 9 iterations. By comparing Figs. 2.13(a) and 2.13(b), one can see that in the GN, the model mismatch term at the first 2 iterations is small (1.158 and 3.267, at $l = 1$ and $l = 2$ respectively), but $\|\delta\tilde{m}_2^l\|$ is high from the first iteration. In the LM algorithm, γ decreases with iterations and it reached a value of 1.0 at iteration $l = 9$, but $\|\delta\tilde{m}_2^l\|$ does not increase to a high value compared to the GN algorithm. The reason is that, at early iterations of GN, the high data mismatch term in Eq. 2.72 makes the α_i corresponding to small singular values even higher, and the search direction would have large components along the noisy right singular vectors. But in the LM algorithm, as γ decreases with iterations, by the time γ

reaches a relatively small value, the data mismatch term is much smaller and the values of α_i corresponding to small singular values, are not high.

At late iterations where the LM parameter is small, the distribution of the values of $|\alpha_i|$ in the LM search direction are different from the one at early iteratiuons. At iteration $l = 10$ of SVD-EnRML algorithm, the value of γ for the MAP estimate was very small ($\gamma = 0.01$), but since the LM search direction did not decrease the objective function, γ increased to 100, and with this value the objective function decreased. Fig. 2.14(a) shows the values of $|\alpha_i|$ for the MAP estimate at iteration $l = 10$. In this figure, black shows the values of $|\alpha_i|$ with $\gamma = 0.01$ and red shows the values of $|\alpha_i|$ with $\gamma = 100$. Note that increasing γ to 100, had a negligible effect on α_i corresponding to large singular values, but it highly reduced the values of $|\alpha_i|$ corresponding to small singular values.



(a) $|\alpha_i|$, LM algorithm

(b) retained Singular Values of G_D at $l = 10$

Figure 2.14: (a) The values of $|\alpha_i|$ in the search direction with the LM algorithm corresponding to the MAP estimate, red is with $\gamma = 100$, black is with $\gamma = 0.01$. (b) The retained singular values of $G_{D,l}$ at $l = 10$, Example 1-2.

2.7.3 Conclusions

The results presented, show that with a large value of γ, the search direction $\delta\tilde{m}$ in the LM algorithm has components mainly in the direction of the right singular vectors corresponding to the largest singular values, while the search direction $\delta\tilde{m}$ in the GN algorithm has large components mainly in the direction of the right singular vectors corresponding to the small singular values. These results are consistent with the discussion about spectral analysis of the LM algorithm given in Oliver et al. [28], that with a γ close to zero, the search direction in the LM algorithm may have large components in the direction of eigenvectors of the Hessian associated with the small eigenvalues; where for our LM algorithm, the eigenvectors of the Hessian in the transformed space are the eigenvectors of $G_D^T G_D$, which are the same as the right singular vectors of the dimensionless sensitivity matrix, G_D.

The LM algorithm does not only damp the change of model parameters at early iterations. At early iterations, the LM algorithm gradually resolves the important features of the model which are contained in eigenvectors of the Hessian associated with the largest eigenvalues.

The results of Tavakoli and Reynolds [37] shows that with SVD parameterization, the behavior of the normalized objective function with a fixed sv-cut of 0.003125 and with sv-cut changing from 0.1 to 0.003125 by dividing by 2 at each iteration, are almost the same. In addition, in their results, the behavior of the normalized objective function at early iterations has no visible difference for 3 different cases, which are fixed sv-cut of 0.1, fixed sv-cut of 0.003125, and changing sv-cut from 0.1 to 0.003125. Their results can be justified with the results of this example. No matter how many singular triplets are calculated at early iterations, with a high LM parameter, the main components of the search direction are along a few right singular vectors associated with the largest singular values. Thus in the LM algorithm, the behavior of objective function at early iterations are almost identical in cases that a few singular triplets be calculated or many singular triplets be calculated.

The results of Dickstein et al. [8] can also be justified with our discussion. According to their results, in the absence of seismic data, by using a fixed number of SVD parameters (25) with the GN algorithm, the resulting model was very rough, and a poor representation of the reservoir. In this example, we showed that in SVD parameterization, if many singular triplets are calculated at early iterations, the GN search direction would have large components along the right singular vectors corresponding to some small singular values.

2.7.4 Right and Left Singular Vectors of $G_{D,l}$

In this part, we show some of the right and left singular vectors of $G_{D,l}$, corresponding to the SVD-EnRML-SMM algorithm in Example 1-2. The singular vectors of $G_{D,l}$ are corresponding to iterations $l = 1$ and $l = 12$. At these two iterations, the truncated SVD of $G_{D,l}$ is computed for the MAP estimate.

The right singular vectors of the dimensionless sensitivity matrix are used to parameterize the change in the model in the transformed space, and the left singular vectors can be used to parameterize the change in the predicted data. Each right singular vector of $G_{D,l}$ is an N_m-dimensional column vector. The search direction in the actual space, δm^{l+1}, is

$$\delta m^{l+1} = L\delta\tilde{m}^{l+1} = L\sum_{i=1}^{p}\alpha_i v_i, \tag{2.95}$$

where L is from the Cholesky decomposition of C_M, and v_i is the ith right singular vector of $G_{D,l}$. Thus, we plot Lv_i, since Lv_i is in the actual space, and it shows the features that the ith Lv_i contains. As model parameters include log permeability and porosity of gridblocks, each right singular vector should be plotted for both log permeability and porosity.

The left singular vectors can be plotted versus time. Each left singular vector is an N_d dimensional column vector and its ith entry, is corresponding to the ith entry of the predicted data vector.

As $G_D v_i = \lambda_i u_i$, if the ith right singular vector contains a feature around a particular well, we expect that the corresponding left singular vector have the most pronounced components related to the predicted data of that well.

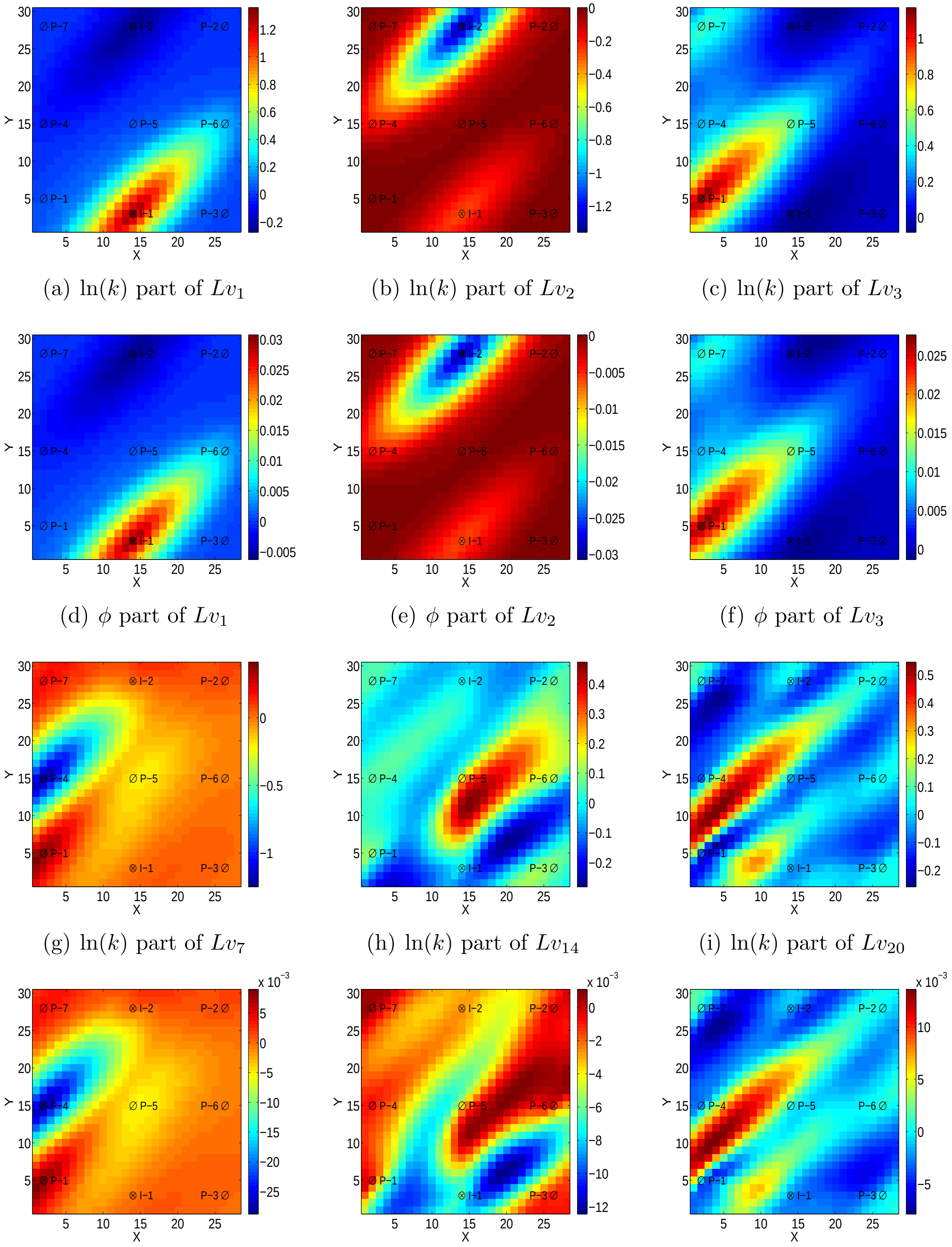
Fig. 2.15 shows the $\ln(k)$ and ϕ part of some of the Lv_i, where v_i is the ith right singular vector of $G_{D,l}$ corresponding to the MAP estimate, at iteration $l = 1$ of the algorithm. Since the porosity and log permeability fields are highly correlated, the corresponding porosity and log permeability parts of each Lv_i, are very similar. In addition, all features of the Lv_i are in the principal correlation direction, with a correlation length affected by the correlation length of the covariance matrix; it means that if the C_M we used, was not correct, the features would be in wrong direction.

In Figs. 2.15(a) and 2.15(d), the main feature of the right singular vector is the high permeability and high porosity region around Inj-1. Note that the feature is extended around the gridblock of Inj-1. There is also a low permeability and porosity region around Inj-2. Since the prior model is uniform, the features in the Lv_i seem to be symmetric with respect to the principal correlation direction. As one can see in Fig. 2.1, in the true model, there is high permeability region close to Inj-1, and a low permeability region around Inj-2, which justifies the features in Lv_1. Fig. 2.16 shows the components of u_1, the first left singular vector of $G_{D,1}$, for some of the wells. As can be seen in Fig. 2.16, the first left singular vector of $G_{D,1}$ has large components corresponding to p_{wf} of Inj-1 and Inj-2, and the components of u_1 corresponding to other wells in Fig. 2.16, are small or negligible.

In Figs. 2.15(b) and 2.15(e), the main feature of the 2nd right singular vector is the low permeability and low porosity region around Inj-2. This feature is extended around the gridblock of Inj-2. The components of u_2, the second left singular vector of $G_{D,l}$, are shown in Fig. 2.17, for some of the wells. As one can see in Fig.2.17(b), the 2nd left singular vector of $G_{D,l}$ has large components corresponding to p_{wf} of Inj-2, and other components of u_2 are small.

In Figs. 2.15(h) and 2.15(k), the 14th right singular vector contains several features. The components of u_{14}, the corresponding left singular vector of $G_{D,l}$, are shown in Fig. 2.18, for some of the wells. As one can see in this figure, the 14th left singular vector of $G_{D,l}$ has large components for the data of several wells.

Fig. 2.19 shows the $\ln(k)$ part of some of Lv_i at iteration $l = 12$ of the algorithm, where the truncated SVD of $G_{D,l}$ is computed for the MAP estimate. In this figure, as the singular values get smaller, the corresponding right singular vector is noisier.



(a) $\ln(k)$ part of Lv_1

(b) $\ln(k)$ part of Lv_2

(c) $\ln(k)$ part of Lv_3

(d) ϕ part of Lv_1

(e) ϕ part of Lv_2

(f) ϕ part of Lv_3

(g) $\ln(k)$ part of Lv_7

(h) $\ln(k)$ part of Lv_{14}

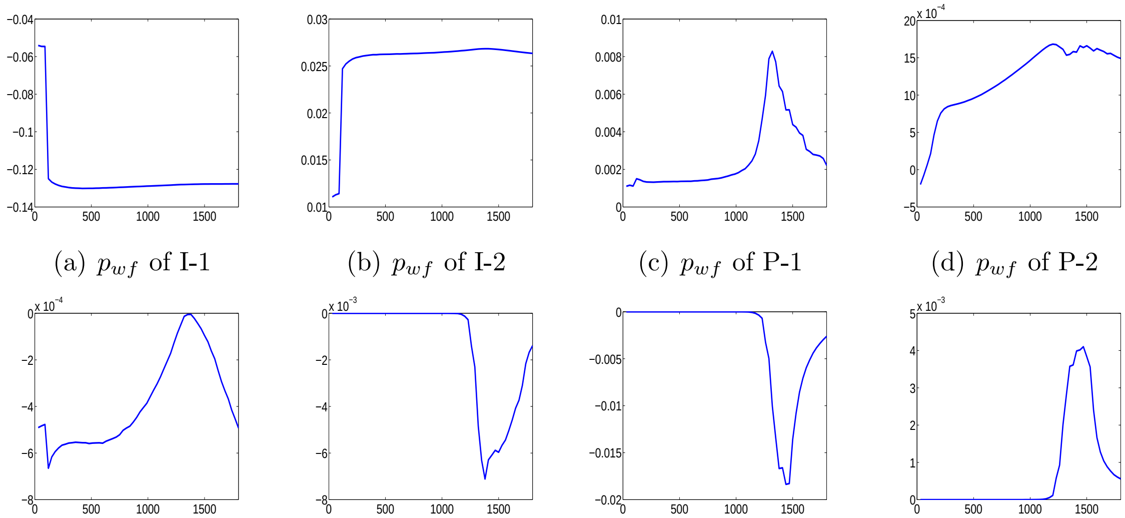
(i) $\ln(k)$ part of Lv_{20}

(j) ϕ part of Lv_7

(k) ϕ part of Lv_{14}

(l) ϕ part of Lv_{20}

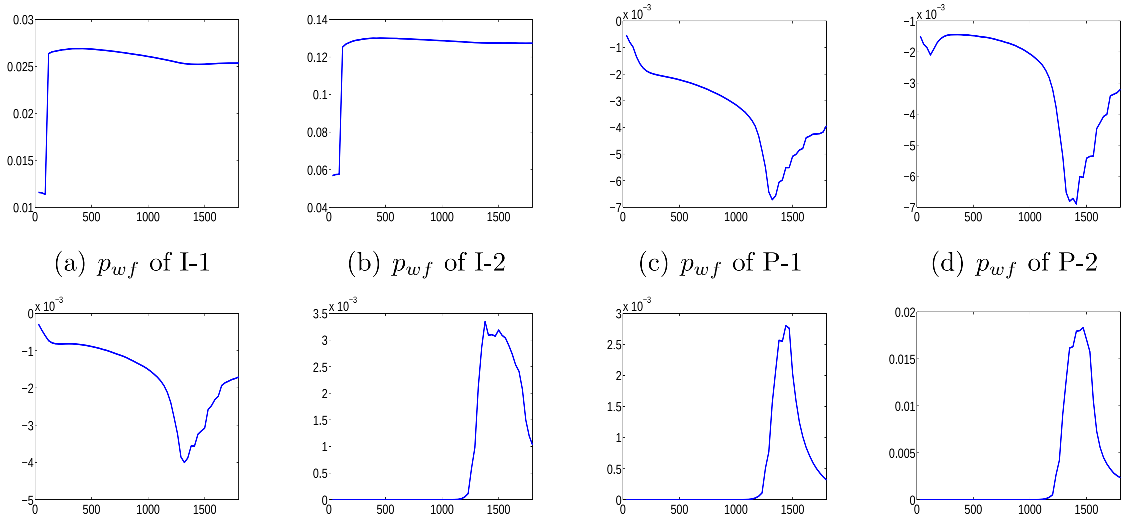
Figure 2.15: $\ln(k)$ and ϕ parts of 6 of Lv_i, where v_i is the ith right singular vector of $G_{D,l}$ at iteration $l = 1$, Example 1-2.



(a) p_{wf} of I-1 (b) p_{wf} of I-2 (c) p_{wf} of P-1 (d) p_{wf} of P-2

(e) p_{wf} of P-4 (f) q_w of P-1 (g) q_w of P-3 (h) q_w of P-7

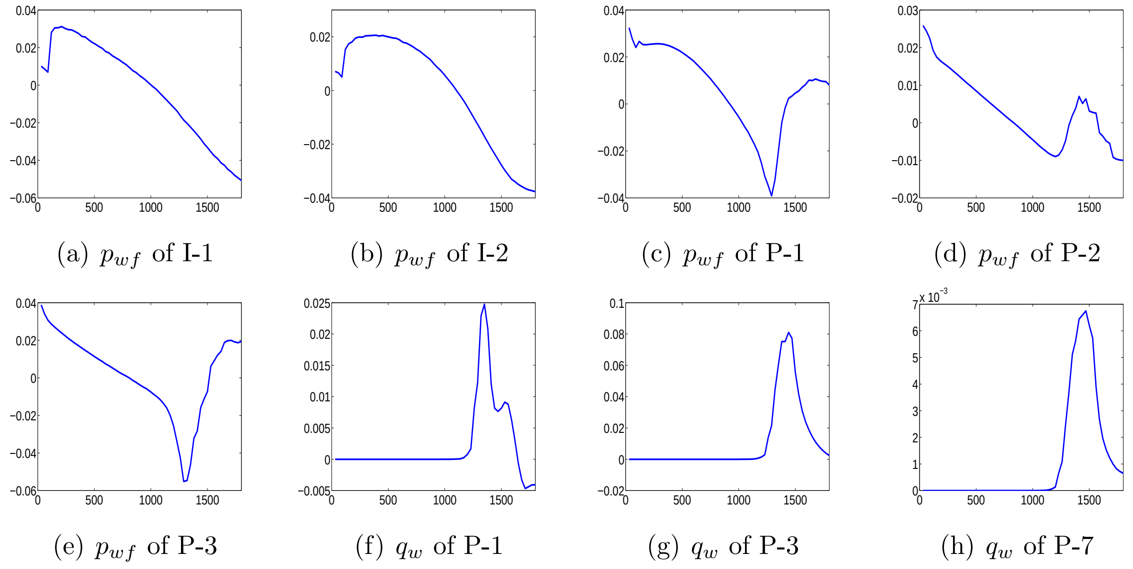
Figure 2.16: The components of the 1st left singular vector of $G_{D,l}$ versus time, at iteration $l = 1$, for some of the wells, Example 1-2.



(a) p_{wf} of I-1 (b) p_{wf} of I-2 (c) p_{wf} of P-1 (d) p_{wf} of P-2

(e) p_{wf} of P-3 (f) q_w of P-1 (g) q_w of P-3 (h) q_w of P-7

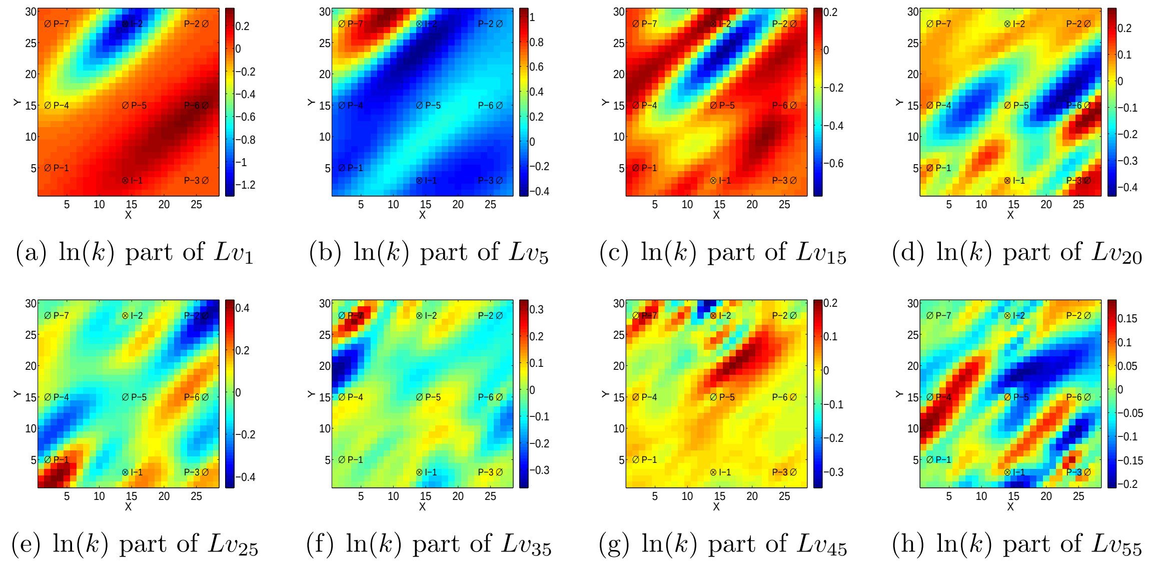
Figure 2.17: The components of the 2nd left singular vector of $G_{D,l}$ versus time, at iteration $l = 1$, for some of the wells, Example 1-2.



(a) p_{wf} of I-1 (b) p_{wf} of I-2 (c) p_{wf} of P-1 (d) p_{wf} of P-2

(e) p_{wf} of P-3 (f) q_w of P-1 (g) q_w of P-3 (h) q_w of P-7

Figure 2.18: The components of the 14th left singular vector of $G_{D,l}$ versus time, at iteration $l = 1$, for some of the wells, Example 1-2.



(a) $\ln(k)$ part of Lv_1 (b) $\ln(k)$ part of Lv_5 (c) $\ln(k)$ part of Lv_{15} (d) $\ln(k)$ part of Lv_{20}

(e) $\ln(k)$ part of Lv_{25} (f) $\ln(k)$ part of Lv_{35} (g) $\ln(k)$ part of Lv_{45} (h) $\ln(k)$ part of Lv_{55}

Figure 2.19: $\ln(k)$ part of 8 of Lv_i, where v_i is the ith right singular vector of $G_{D,l}$ at iteration $l = 12$, Example 1-2.

2.7.5 Example 1-3, The Maximum Number of Required SVD Parameters

In the SVD parameterization algorithm, one important source of computational cost is the number of SVD parameters that is calculated with the Lanczos method. For a synthetic example, Tavakoli and Reynolds [37] compared the results of the LM algorithm with SVD parameterization for a fixed 10, 15 and 20 SVD parameters. They generated only the MAP estimate. According to their results, with a fixed 10 singular triplets the algorithm converged in 48 iterations with $O_N(m) = 3.98$ which is high. With 15 SVD parameters, the MAP estimate converged in 34 iterations with $O_N(m) = 1.34$ which is slightly larger than $1 + 5\sqrt{2/N_d} = 1.27$. They obtained the best results in terms of both computational efficiency and final value of objective function by using a fixed 20 SVD parameters. In the latter case, the MAP estimate converged in 29 iterations with $O_N(m) = 0.99$. Their results show that if one uses a few SVD parameters, the normalized objective function at convergence does not decrease to an acceptable value, i.e., to a value less than $1 + 5\sqrt{2/N_d}$.

Later, Tavakoli and Reynolds [38] used the parameter *sv-cut* to control the number of parameters at each iteration. Their results showed that using a high sv-cut of 0.1, the final value of normalized objective function is more than 100, however with sv-cut of 0.003125 they obtained a small value of the objective function at convergence of the algorithm.

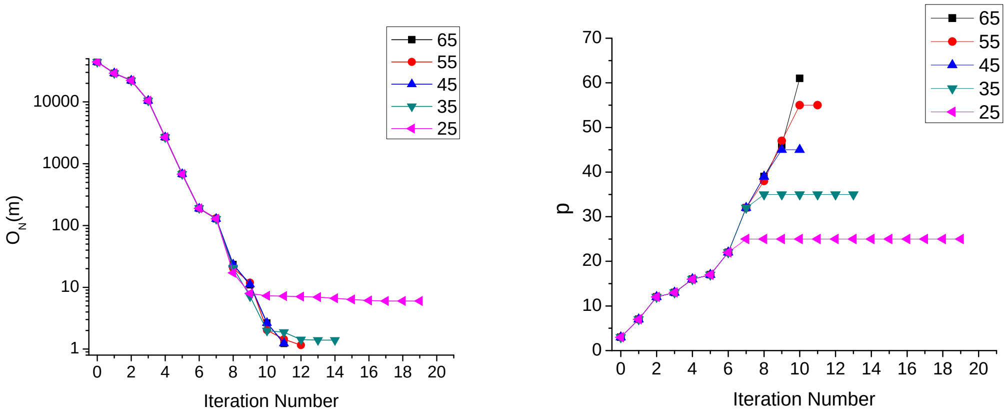
Based on several examples that we have done with SVD parameterization in generating a single estimate of the model, and the examples that are presented in the next chapters, it seems that in order to obtain fast convergence rate, small value of objective function and computational efficiency, the number of retained singular triplets should be high enough and the smallest retained singular value at late iterations should be reasonably close to 1. In other words, in addition to the final sv-cut, the smallest retained singular value and the number of singular values should also be considered. This is one reason that we modified the algorithm to specify $N_{\text{SVD,max}}$ as well as final sv-cut.

As we will show in some examples, after the first few iterations, the largest singular value of $G_{D,l}$ does not change significantly. The value of the largest singular value at late iterations differs from example to example. If the maximum singular value is around 10000 for a case, then with a sv-cut of 0.001, the minimum retained singular value will be around 10, and if there are many singular values between 10 and 1, it is less likely to obtain a fast convergence rate and small value of the objective function. Thus by specifying a minimum *sv-cut* of 0.003 or 0.001, the minimum singular value and the number of retained singular values can be different for any case.

Table 2.7: Summary of computational costs of SVD parameterization algorithm to the generate porosity and log permeability fields of the MAP estimate, Example 1-3.

$N_{\text{SVD,max}}$	Simulations	Direct	Adjoint	Equ. Sim. Runs	N_{iter}	O_N
65	13	367	356	194	11	1.244
55	12	440	428	229	12	1.158
45	12	350	339	189	11	1.241
35	16	469	455	247	14	1.395
25	21	634	614	333	19	6.0

In order to study the effect of the number of singular triplets on convergence properties and computational costs of SVD parameterization, we apply the algorithm with different numbers for $N_{\text{SVD,max}}$. We specify a $\mu_{\min}$ of 0.0001, which is fairly small, and we expect that the number of retained singular values be around $N_{\text{SVD,max}}$ at late iterations. Table 2.7 shows the computational cost and the final values of $O_N(m)$ for different cases. Note that using a small number of singular triplets ($N_{\text{SVD,max}} = 25$), gives the worst results in terms of both computational cost and final value of objective function. Fig. 2.20(a) shows the value of O_N of the MAP estimate versus iterations for different specified $N_{\text{SVD,max}}$. Fig. 2.20(b) shows the corresponding number of singular triplets versus iterations for each case. Note that with $N_{\text{SVD,max}}$ of 35 and 25, the final value of $O_N(m)$ is larger than $1 + 5\sqrt{2/N_D} = 1.264$. Although for $N_{\text{SVD,max}} = 35$, the value of λ_{35}/λ_1 was less than 0.0025 during iterations, this number of singular triplets was not enough to achieve a small value of objective function at convergence of the algorithm. However with $N_{\text{SVD,max}}$ of 45 or more, we could obtain a small value of the objective function. Fig. 2.22 shows the largest 71 singular values, λ_i, of G_D and the corresponding ratio of λ_i/λ_1 at a late iteration of the algorithm. Note that a noticeable number of singular triplets (35 of them) are between 10 and 1.



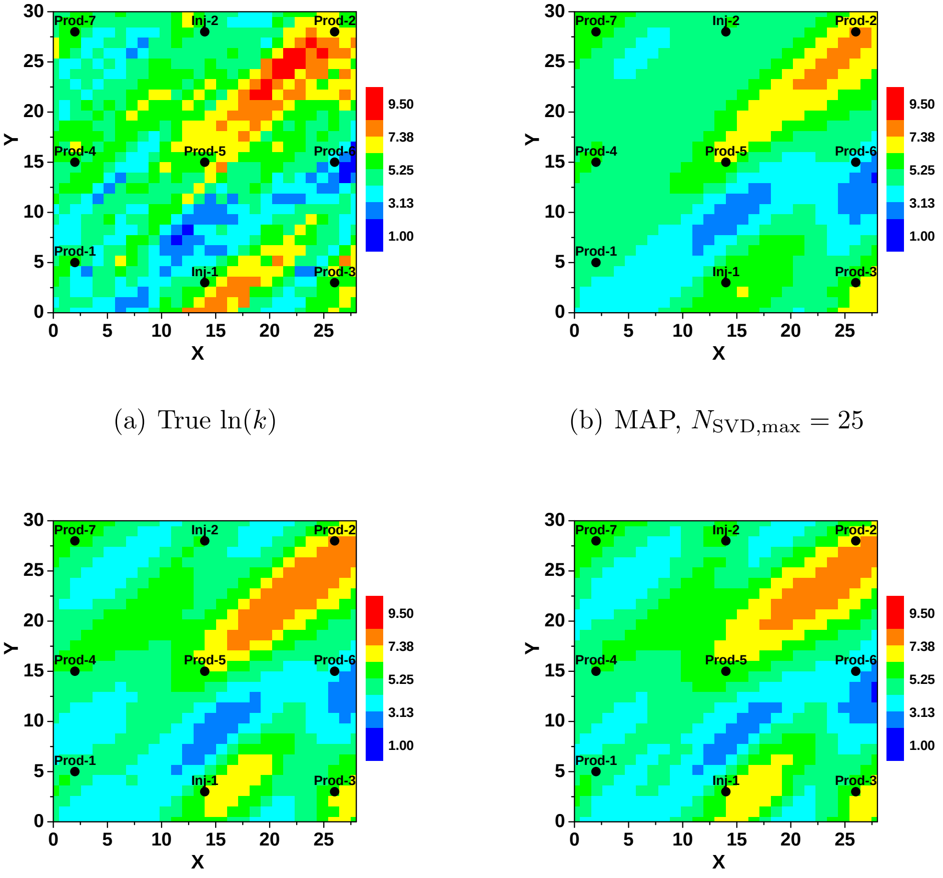
(a) O_N of the MAP estimate

(b) Number of retained SVD parameters

Figure 2.20: (a) The Values of O_N of the MAP estimate for different $N_{\text{SVD,max}}$. (b) The number of retained singular triplets of G_D versus iterations, Example 1-3.

Fig. 2.21 shows the log permeability fields of the MAP estimate with 3 different $N_{\text{SVD,max}}$. True log permeability field is also shown for comparison. The MAP estimate with $N_{\text{SVD,max}} = 25$ is very smooth and is distinctly different from the other two. There is not a distinct difference between the MAP estimate from $N_{\text{SVD,max}} = 45$ with the one from $N_{\text{SVD,max}} = 35$, and both of them contain the main features of the true model, although the MAP estimate with $N_{\text{SVD,max}} = 45$ has a smaller objective function.

This example shows that in order to obtain a small value of the objective function at convergence of the algorithms and computational efficiency, in addition to a small singular cut off, the number of singular triplets should be reasonably high at late iterations of the algorithm. Computing a small number of singular triplets may decrease the computational effort in the Lanczos method over one iteration, but the algorithm would converge in more iterations to a model with a high value of objective function which is not desired.



(a) True $\ln(k)$

(b) MAP, $N_{\mathrm{SVD,max}} = 25$

(c) MAP, $N_{\mathrm{SVD,max}} = 35$

(d) MAP, $N_{\mathrm{SVD,max}} = 45$

Figure 2.21: Log permeability fields of the MAP estimate with 3 different $N_{\mathrm{SVD,max}}$ compared with the truth, Example 1-3.

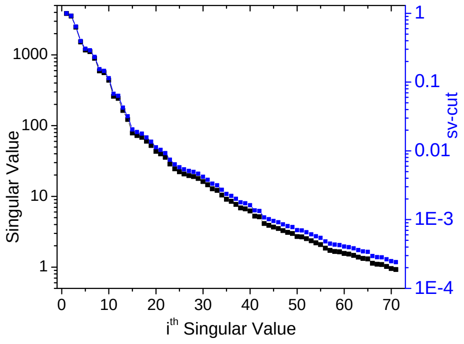


Figure 2.22: λ_i, singular values of G_D for the MAP estimate, and the corresponding ratio of λ_i/λ_1 at iteration $l = 11$, Example 1-3.

2.7.6 Example 1-4, Results of SVD-EnRML-MI and SVD-EnRML-SMM

In this part, the results of SVD-EnRML-SMM and SVD-EnRML-MI for generating the MAP estimate and $N_e = 15$ RML realizations are compared in terms of the computational cost and final values of objective functions. In both algorithms, the same prior realizations of porosity and log permeability fields are used.

We applied the two algorithms with $N_{\mathrm{SVD,max}} = 65$. The 1st and 2nd rows of Table 2.8 show the computational costs of the two algorithms and the maximum O_N at convergence. The SVD-EnRML-SMM algorithm did not converge in 57 iterations, and it was terminated at 57 iterations. As one can see in the 1st row of Table 2.8, the maximum O_N after 57 iterations is 1.482. After 57 iterations, only 3 out of 16 conditional realizations have $O_N < 1+5\sqrt{2/N_d} = 1.264$. The SVD-EnRML-MI algorithm converged in 40 iterations, and at convergence the $O_N(m)$ of all realizations is less than 1.264. Fig. 2.23 shows the values of $O_N(m)$ for all realizations versus iterations of the SVD-EnRML-SMM algorithm, and Fig. 2.25(b) shows the values of $O_N(m)$ for all realizations versus iterations l of the SVD-EnRML-MI algorithm with $N_{\mathrm{SVD,max}} = 65$.

We also applied the SVD parameterization algorithm with $N_{\mathrm{SVD,max}} = 65$, to generate the MAP estimate and $N_e = 15$ RML realizations one by one. Fig. 2.24(a) shows the values of $O_N(m)$ for all realizations versus iterations of the SVD parameterization algorithm; note that each realization is minimized separately with the step by step description given in this chapter. Fig. 2.24(b) shows the computational cost in terms of equivalent simulation runs for generating each of the 16 conditional realizations. Generating the MAP estimate has the lowest computational cost, which is 194 Equ. Sim. Runs, while generating some of the RML realizations, takes a computational cost of more than 500 equivalent simulation runs. The last row of Table 2.8 shows the total computational cost of generating 16 conditional realizations, with SVD parameterization algorithm. The number of simulation runs with SVD parameterization algorithm is less than SVD-EnRML algorithms; but, the total computational cost in terms of equivalent simulation runs, for minimizing the objective functions one by one, is far more than

SVD-EnRML algorithms. With SVD parameterization algorithm, 2 of RML realizations converged with $O_N(m)$ greater than $1 + 5\sqrt{2/N_d} = 1.264$.

Table 2.8: Computational costs of SVD-EnRML-SMM and SVD-EnRML-MI algorithms for generating 16 conditional realizations of porosity and log permeability, and the $\max\{O_N\}$ at convergence, Example 1-4.

Method	$N_{\text{SVD,max}}$	Sim.	Direct	Adjoint	Equ. Sim. Runs	N_{iter}	$\max\{O_N\}$
SMM	65	1811	3870	3813	3732	57	1.482
MI	65	1411	2479	2439	2640	40	1.26
MI	55	1356	2358	2316	2524	42	1.38
MI	75	1131	2020	1987	2133	33	1.26
SVD	65	301	12651	12431	6571	-	1.856

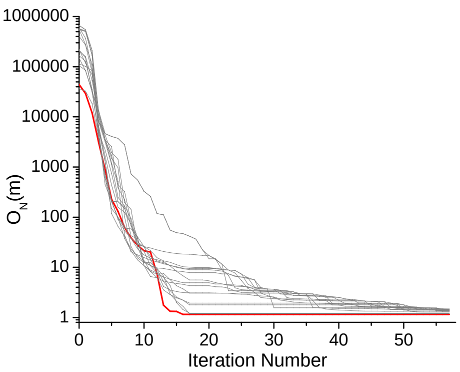
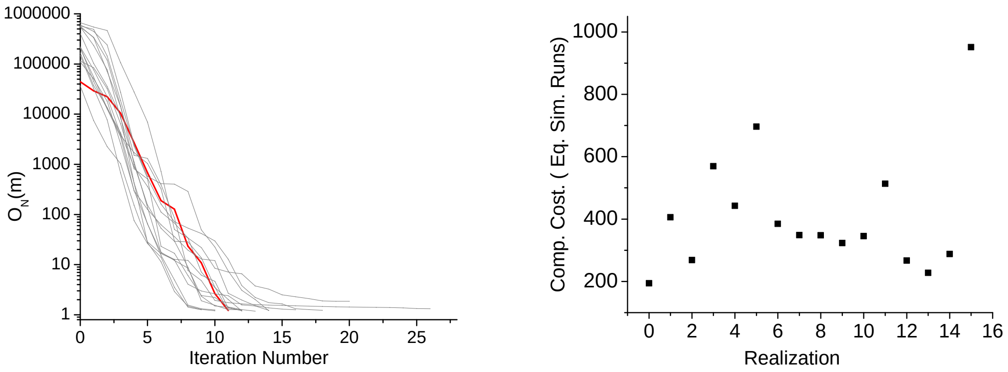


Figure 2.23: $O_N(m)$ of all realizations versus iterations of the SVD-EnRML-SMM algorithm, with $N_{\text{SVD,max}} = 65$, red curve is for the MAP estimate, gray curves are from 15 RML realizations, Example 1-4.

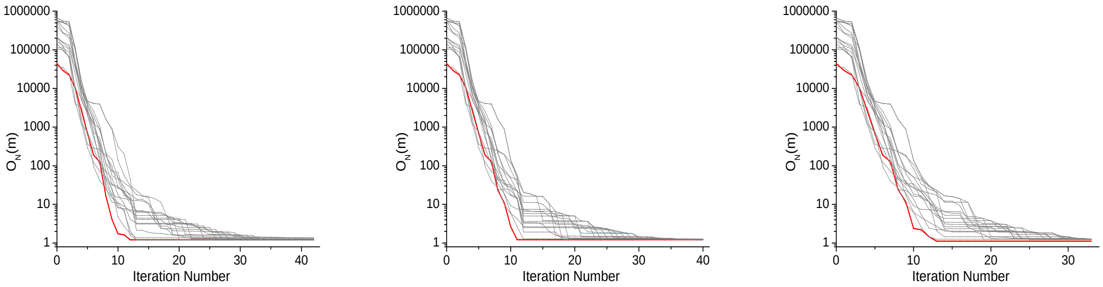
In order to investigate the effect of $N_{\text{SVD,max}}$ on the computational efficiency of SVD-EnRML-MI, we applied the algorithm with $N_{\text{SVD,max}}$ of 75 and 55 as well. Fig. 2.25 shows the values of $O_N(m)$ for all realizations versus iterations l of the SVD-EnRML-MI algorithm with $N_{\text{SVD,max}}$ of 55, 65 and 75. The computational costs of SVD-EnRML-MI and the maximum O_N at convergence for 3 different $N_{\text{SVD,max}}$ are shown in Table 2.8. As one can see in this table, the computational cost of SVD-EnRML-MI in terms of equivalent simulation runs, with $N_{\text{SVD,max}} = 75$ is about 20% less than the one with $N_{\text{SVD,max}} = 65$. With $N_{\text{SVD,max}} = 55$, the SVD-EnRML-MI algorithm converged in 42



(a) $O_N(m)$ versus l

(b) Computational cost

Figure 2.24: (a) $O_N(m)$ of the MAP estimate (red) and 15 RML realizations (gray) versus iterations of the SVD parameterization algorithm, with $N_{\text{SVD,max}} = 65$. (b) Computational cost in terms of equivalent simulation runs for generating each of the 16 conditional realizations, Example 1-4.



(a) $N_{\text{SVD,max}} = 55$

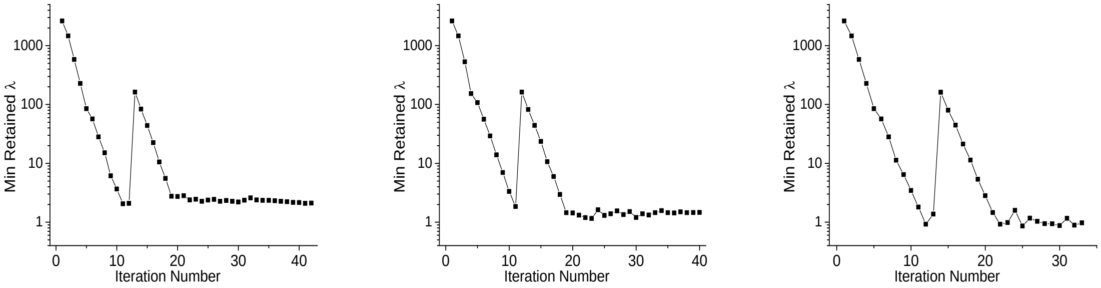
(b) $N_{\text{SVD,max}} = 65$

(c) $N_{\text{SVD,max}} = 75$

Figure 2.25: $O_N(m)$ of all realizations versus iterations of the SVD-EnRML-MI algorithm. Red curve is for the MAP estimate, gray curves are from 15 RML realizations, Example 1-4.

iterations, as both the change in the model and the change in the objective functions of all realizations were small. At convergence, only 3 of the realizations have $O_N(m) < 1.264$, however, the maximum O_N is 1.38 which is fairly small. Note that in this example, the least computational cost of SVD-EnRML-MI for generating the MAP estimate and 15 conditional realizations is obtained with $N_{\text{SVD,max}} = 75$.

Fig. 2.26 shows the minimum retained singular value of $G_{D,l}$ versus iterations of the SVD-EnRML-MI algorithm, for 3 different numbers of $N_{\text{SVD,max}}$. The 55th singular value of $G_{D,l}$ is around 2.2, while the 75th singular value of $G_{D,l}$ is around 0.9. It means that 20 of the singular values are very close, and retaining these singular values decreases the computational cost of the algorithm. Fig. 2.27 shows the first 75 singular values of $G_{D,l}$ at a late iteration of the algorithm.



(a) $N_{\text{SVD,max}} = 55$

(b) $N_{\text{SVD,max}} = 65$

(c) $N_{\text{SVD,max}} = 75$

Figure 2.26: The smallest retained singular value of $G_{D,l}$ versus iterations of the SVD-EnRML-MI algorithm, Example 1-4.

Fig. 2.28 shows the number of retained singular triplets of $G_{D,l}$ versus iterations of the SVD-EnRML-MI algorithm, for 3 different numbers of $N_{\text{SVD,max}}$. By retaining a higher number of singular values, the algorithm converges in fewer iterations.

Now we show some of the conditional realizations from SVD-EnRML-MI with $N_{\text{SVD,max}} = 75$. In Figs. 2.30 and 2.29, respectively, 3 unconditional and conditional realizations of log permeability field are shown. In Figs. 2.32 and 2.31, respectively, 3 unconditional and conditional realizations of porosity field are shown. All conditional realizations have the main features of the truth shown in Fig. 2.1.

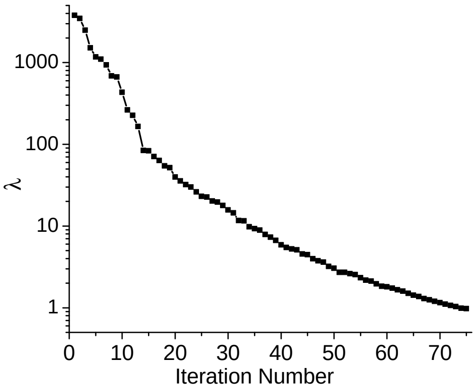
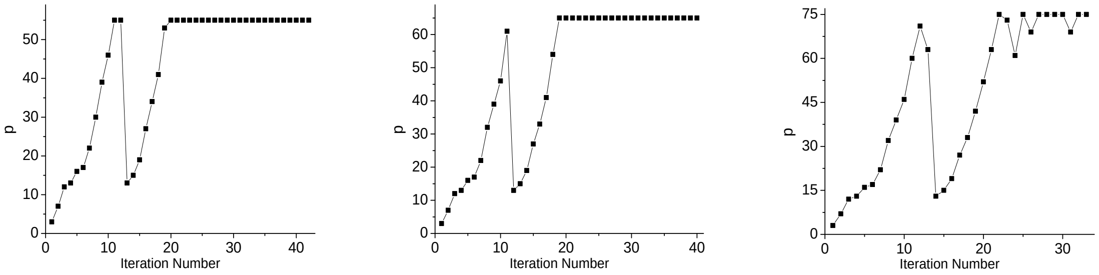
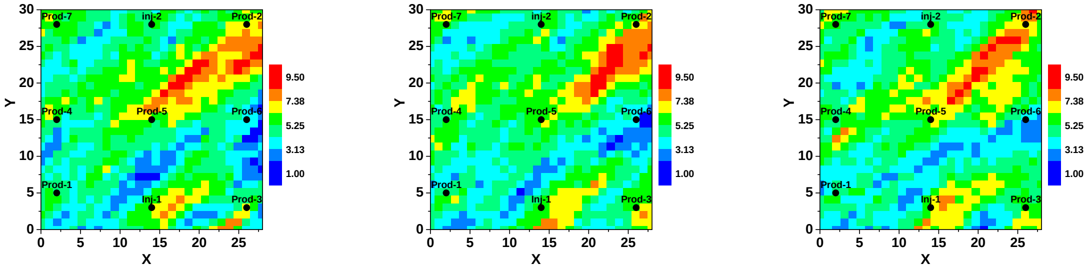


Figure 2.27: Singular values of $G_{D,l}$ at iteration $l = 33$ of SVD-EnRML-MI with $N_{\text{SVD,max}} = 75$, Example 1-4.



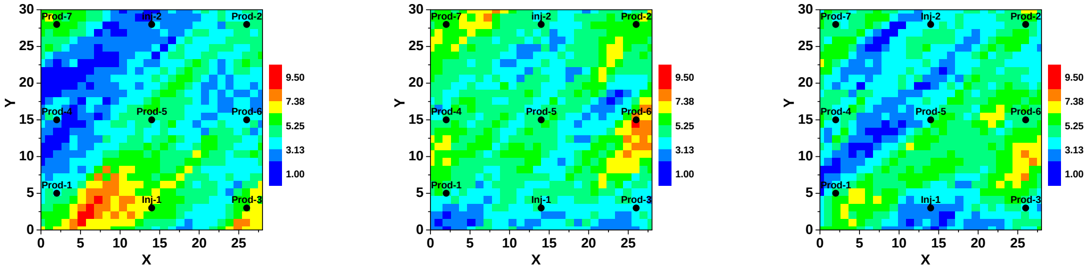
(a) $N_{\text{SVD,max}} = 55$ (b) $N_{\text{SVD,max}} = 65$ (c) $N_{\text{SVD,max}} = 75$

Figure 2.28: The number of retained singular values of $G_{D,l}$ versus iterations of the SVD-EnRML-MI algorithm, Example 1-4.



(a) Realization 2 (b) Realization 8 (c) Realization 15

Figure 2.29: 3 conditional realizations of log-permeability field, generated with SVD-EnRML-MI, Example 2.

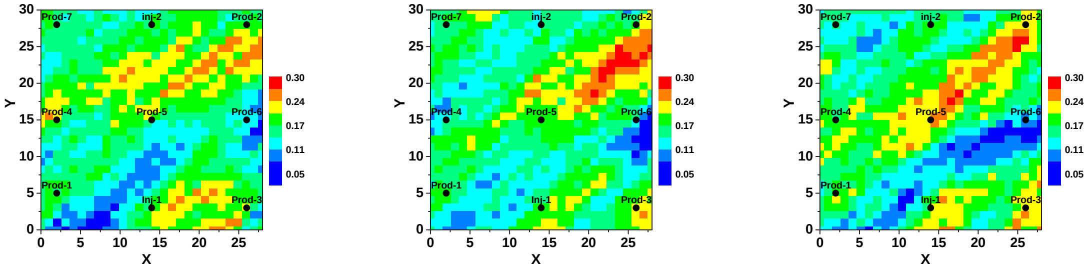


(a) Realization 2

(b) Realization 8

(c) Realization 15

Figure 2.30: 3 unconditional realizations of log-permeability field, Example 2.

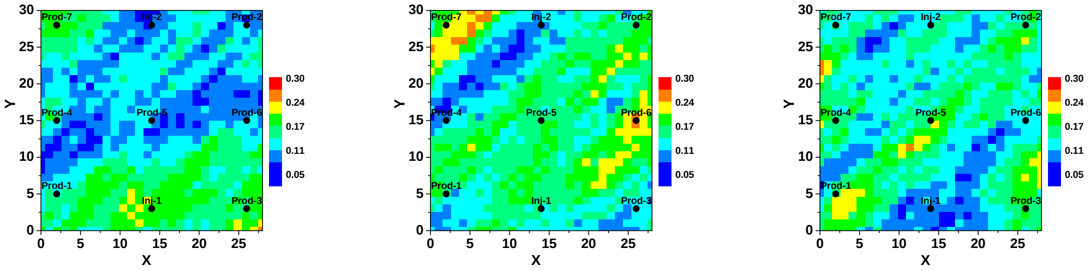


(a) Realization 2

(b) Realization 8

(c) Realization 15

Figure 2.31: 3 conditional realizations of porosity field generated with SVD-EnRML-MI,, Example 2.



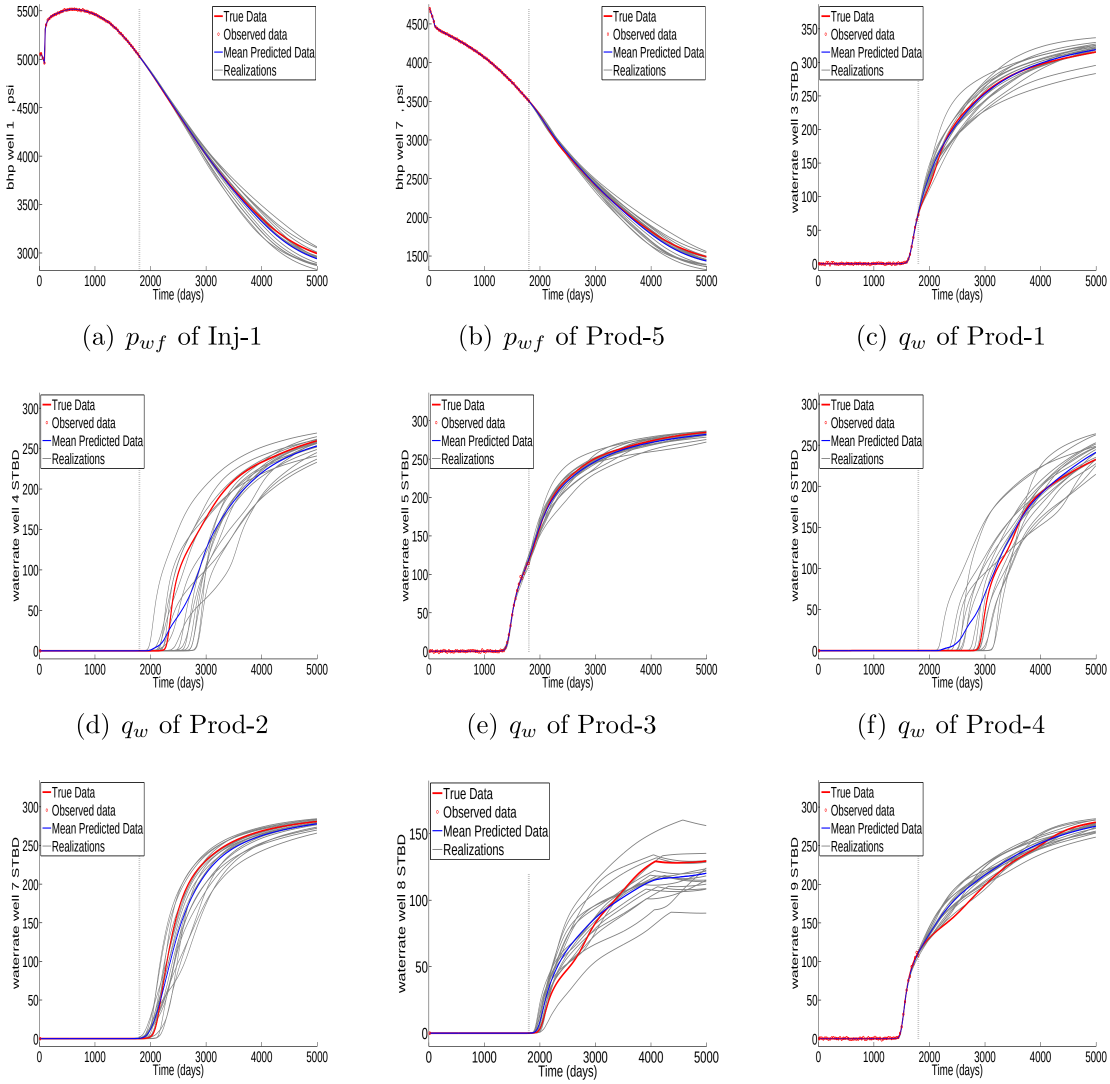
(a) Realization 2

(b) Realization 8

(c) Realization 15

Figure 2.32: 3 unconditional realizations of porosity field, Example 2.

Fig. 2.33 shows the data matches and predictions of water rates of all producers and p_{wf} for some of the wells, from SVD-EnRML-MI with $N_{\mathrm{SVD,max}} = 75$. The mean of the predictions is very close to the truth; in addition, the predictions of water rate even for the wells that experienced the water breakthrough after the end of history matching, are noticeably close to the truth. As expected, the uncertainty range in predictions of q_w for a well that has experienced the water breakthrough before the end of history matching, e.g., in Fig. 2.33(c), is smaller than the uncertainty range in predictions of q_w for a well with water breakthrough after the end of history matching period, e.g., Fig. 2.33(d). Data matches and predictions of p_{wf} for the wells not shown are similar.



(a) p_{wf} of Inj-1 (b) p_{wf} of Prod-5 (c) q_w of Prod-1

(d) q_w of Prod-2 (e) q_w of Prod-3 (f) q_w of Prod-4

(g) q_w of Prod-5 (h) q_w of Prod-6 (i) q_w of Prod-7

Figure 2.33: Data matches and predictions from the conditional realizations generated by SVD-EnRML-MI with $N_{\text{SVD,max}} = 75$. The dashed vertical line shows the end of history matching (1800 days), Example 1-4.

CHAPTER 3

2D EXAMPLES

In this chapter, some synthetic 2D examples are presented. We compare the results of SVD-EnRML algorithms. We also compare the efficiency and computational cost of some other parameterization methods based on the results of examples. The algorithms which are compared with SVD-EnRML, are the subspace method and ensemble Kalman filter (EnKF). As the formulation of SVD-EnRML algorithms was discussed in the previous chapter, here, only the subspace method and the EnKF are briefly discussed and then examples are presented.

In examples of this chapter, we used the following parameters for controlling sv-cut in SVD-EnRML algorithms, unless stated otherwise.

$$\mu_1 = 0.5,$$

$$\mu_2 = 0.05,$$

$$\mu_{\min} = 0.0002\,. \tag{3.1}$$

These parameters are discussed in Section 2.2. Note that regardless of sv-cut, the Lanczos algorithm computes a minimum number of 3 singular triplets at each iteration l and a maximum number of $N_{\text{SVD,max}}$ singular triplets.

In tables, ϕ_{mean} and $[\ln(k)]_{\text{mean}}$, denote the prior mean of porosity and log-permeability, respectively. The standard deviations of porosity and log-permeability are denoted by σ_ϕ and $\sigma_{\ln(k)}$, respectively; $\rho_{\phi,\ln(k)}$ denotes the correlation coefficient between porosity and log-permeability; α is the angle measured counterclockwise from the x-axis to the principal correlation direction of the covariance function; r_1 is the correlation range in the principal direction and r_2 is the correlation range in the orthogonal direction.

All examples are for two phase flow (oil and water).

3.1 Subspace Method

The subspace method is a parameterization method that significantly reduces the size of the matrix problem solved at each iteration of the Gauss-Newton or LM algorithm. In a subspace method, the search direction vector is expanded as a linear combination of a few basis vectors. The dimension of the subspace is much smaller than the number of model parameters.

Reynolds et al. [30] and Abacioglu et al. [2] applied the subspace method with the LM algorithm to condition 2D and 3D single phase reservoirs to pressure data. They investigated many details of the application of this method to history matching problems, including the type and the number of subspace vectors that are required to obtain fast convergence and a small value of objective function. In this and the next chapter, the application of the subspace method to some synthetic 2D and 3D two phase reservoirs is investigated. In our implementation, we follow the work of Abacioglu et al. [2] which is explained in more detail in Abacioglu [3]. At first we briefly explain the method.

In the Gauss-Newton method of minimizing objective function $O(m)$, given by Eq. 1.8, the search direction δm^{l+1} at the lth iteration is generated from

$$H_l \delta m^{l+1} = -\nabla O^l. \tag{3.2}$$

The basic idea of the subspace method is that at the lth iteration of the Gauss-Newton or LM algorithms, the search direction vector δm^{l+1} is parameterized as a linear combination of a relatively small number of subspace vectors, a_j for $j = 1, \cdots, N_B$. The change in the model estimate or search direction vector at the lth iteration of the

Gauss-Newton method is then written as

$$\delta m^{l+1} = \sum_{j=1}^{N_B} \alpha_j a_j = A_l \alpha^{l+1}, \tag{3.3}$$

where A_l is the $N_m \times N_B$ matrix of the subspace vectors at the lth iteration of the algorithm and it's jth column is a_j^l; α^{l+1} is an N_B-dimensional column vector. Using Eq. 3.3 in Eq. 3.2, and multiplying both sides by A_l^T, the following equation is obtained:

$$(A_l^T H_l A_l)\alpha^{l+1} = -A_l^T \nabla O^l. \tag{3.4}$$

Note that with this formulation the size of the Hessian matrix changes from $N_m \times N_m$ to $N_B \times N_B$.

From Eq. 3.4, the GN equation for the coefficients of the subspace vectors is

$$(A_l^T (C_M^{-1} + G_l^T C_D^{-1} G_l) A_l)\alpha^{l+1} = -A_l^T \nabla O^l. \tag{3.5}$$

The gradients of the partitioned data objective functions, multiplied by C_M, are used to compute the subspace vectors; note that we calculate an orthogonal basis for the subspace vectors.

Assuming that the measurement errors are either independent, or that the partitioning is done in such a way that data with correlated measurement errors are included in the same group, the data misfit objective function can be partitioned as

$$O_d(m) = \sum_k O_d^k(m) = \sum_k \frac{1}{2}(g^k(m) - d^k)^T [C_D^k]^{-1} (g^k(m) - d^k), \tag{3.6}$$

where d^k is the kth set of data and C_D^k is the data covariance matrix for the kth set of data. The vector of model mismatch can also be used in 2 or 3 partial objective functions, one for porosity, one for horizontal log permeability and one for vertical log permeability. For now, consider the model mismatch term, given by the following equation, as a single subobjective function.

$$O_m(m) = \frac{1}{2}(m - m_{\text{prior}})^T C_M^{-1}(m - m_{\text{prior}}). \tag{3.7}$$

The gradients of $O_d^k(m)$'s are given by

$$\nabla O_d^k(m) = G_k^T [C_D^k]^{-1}(g^k(m) - d^k), \tag{3.8}$$

where G_k is the sensitivity matrix of data in the kth subobjective function with respect to all model parameters, and

$$\nabla O_m(m) = C_M^{-1}(m - m_{\text{prior}}). \tag{3.9}$$

A partitioning of the objective function is done at each iteration of the GN or LM algorithms. Here, we partition the data objective function based on the type of data, the well at which data is observed, and time periods. We also gradually increase the number of subspace vectors. The partitioning is performed such that at first, each data type from each well is used in one subobjective function, e.g., data mismatch terms containing q_w of Prod-1 are grouped into one subobjective function. The number of subspace vectors is increased by partitioning each data type at each well into smaller time intervals.

Let b_k be the gradient of the kth partial data objective function, and let

$$a_k = C_M b_k, \tag{3.10}$$

be the kth subspace vector. Two (or three) additional subspace vectors that are based on the model mismatch vector are also included. Let B_l denote the $N_m \times N_B$ matrix that contains all the presubspace vectors at the lth iteration of the algorithm. Using $A_l = C_M B_l$ in Eq. 3.5, it can be obtained

$$(B_l^T C_M B_l + B_l^T C_M G_l^T C_D^{-1} G_l C_M B_l)\alpha^{l+1} = -B_l^T C_M \nabla O^l. \tag{3.11}$$

Pre-subspace vectors, which are the gradients of subobjective functions, are obtained with the adjoint method, while the product $G_l C_M B_l$ or $G_l A_l$ is obtained with the gradient simulator method. In obtaining ∇O_d^k, the product of G_k^T times the vector $[C_D^k]^{-1}(g^k(m) - d^k)$ should be calculated. This product can simply be calculated by obtaining the product of G^T times vector $w_k \circ C_D^{-1}(g(m) - d)$, where $\circ$ denotes the Schur product of the vectors w_k and $C_D^{-1}(g(m) - d)$; w_k is an N_d dimensional column vector with all entries equal to zero except entries corresponding to the data of the kth subobjective function in the data vector, d_{obs}, which are 1.

Note that if a subobjective function only contains data at early times, since these data are not a function of pressures and saturations at later times, the adjoint solution for computing its gradient can be much faster than a typical adjoint solution.

In this work, we use the subspace method with the modified LM algorithm, where at iteration l of the algorithm, the following equation is solved:

$$(A_l^T((1 + \gamma^l)C_M^{-1} + G_l^T C_D^{-1} G_l)A_l)\alpha^{l+1} = -A_l^T \nabla O^l, \tag{3.12}$$

where γ^l is the LM parameter. After solving Eq. 3.5, the search direction vector is calculated from Eq. 3.3 and then the new candidate for the updated model is obtained from $m^{l+1} = m^l + \delta m^{l+1}$; if $O(m^{l+1}) < O(m^l)$, m^{l+1} is accepted as the new updated model and γ^l is decreased by a factor of 10 for the next iteration, otherwise γ^l is increased by a factor of 10 and the iteration is redone.

3.1.1 Applying Subspace Method to Perform RML

We explore the idea of using the subspace method to perform RML. The idea is motivated from the SVD-EnRML. In SVD-RnRML, at iteration l of the LM algorithm, right singular vectors of a truncated SVD of $G_{D,l}$ computed for a particular realization, are used to parameterize the vector of the change in model parameters of all realizations. Similarly, subspace vectors can be calculated based on a particular realization, and be used to decrease the objective functions of the MAP estimate and N_e RML realizations.

In generating the MAP estimate and N_e RML realizations m_j, $j = 1, \cdots N_e$, in Eq. 3.11, the matrix of pre-subspace vectors, B_l, and the Hessian matrix in the left-hand-side, are calculated based on a particular realization, m_{base}. For each realization, we calculate its gradient with an adjoint solution and use it in the right-hand-side of Eq. 3.11. Thus, Eq. 3.11 is modified for the jth realization to

$$(B_l^T C_M B_l + B_l^T C_M G_l^T C_D^{-1} G_l C_M B_l)\alpha_j^{l+1} = -B_l^T C_M \nabla O_j^l, \tag{3.13}$$

where α_j^{l+1} is an N_B dimensional vector, which determines the combination of the subspace vectors in calculating the search direction, δm_j^{l+1}. After solving Eq. 3.13, by using Eq. 3.3, the vector of the change in model parameters for the jth realization, δm_j^{l+1}, is calculated. The subspace vectors are computed based on the MAP estimate; and after the convergence of the MAP estimate, the subspace vectors are computed for the realization with the maximum objective function. Although the subspace vectors are computed based on a particular realization, one advantage of this formulation is that the search direction is always downhill, since the actual gradient of each realization is computed. To further explain the reason that the search direction is downhill, let us return to the basic equation of the search direction. Here, for applying RML, we are using the following equations for calculating the search direction for each realization:

$$H_{l,\text{base}} \delta m_j^{l+1} = -\nabla O_j^l, \tag{3.14}$$

$$\delta m_j^{l+1} = A_{l,\text{base}} \alpha_j^{l+1}, \tag{3.15}$$

where $A_{l,\text{base}}$ is the matrix of the subspace vectors, which is calculated for a particular basis realization; $H_{l,\text{base}}$ is the positive definite Hessian matrix that is also calculated for the same particular basis realization. Since the actual gradient for each realization is calculated with an adjoint solution, as long as the Hessian is positive definite, the search direction is downhill. Although in the subspace method, both sides of Eq. 3.14 are multiplied by $A_{l,\text{base}}^T$, the rank of the resulting matrix, $A_{l,\text{base}}^T H_{l,\text{base}} A_{l,\text{base}}$, is equal to the rank of $A_{l,\text{base}}$ which is N_B. Note that as explained in Abacioglu [3], we use an orthogonal set of basis vectors, which guarantees that the rank of the matrix $A_{l,\text{base}}$ is equal to the number of columns, N_B.

3.2 Ensemble Kalman Filter (EnKF)

The ensemble Kalman filter (EnKF) is a Monte Carlo data assimilation method that is able to incorporate available observations sequentially in time. The probability distribution of a model state (including both model parameters and state variables) is represented empirically by an ensemble of realizations. The state variables are propagated forward in time based on the model dynamics. This is called the forecasting step. The ensemble of the model state is adjusted by assimilating available observed data. This is called the updating step [5]. The EnKF was originally introduced by Evensen [12].

In the EnKF scheme, both the model statistics and the sensitivity can be approximately obtained from the ensemble in a straightforward and computationally efficient manner. By approximating the probability density function by an ensemble, EnKF reduces the dimension of the inverse problem from the number of the unknown variables to the number of realizations [5].

EnKF has several advantages, including ease of implementation and low computational costs. It can be easily coupled with any reservoir simulator. No specific knowledge of the simulation numerics is required for implementation of the EnKF.

In the EnKF, an augmented state vector, y^n, includes reservoir model parameters m, and state variables p^n; m is an N_m dimensional column vector, and p^n is an N_p dimensional column vector. p^n is the state vector which includes the parameters that describe the state of the dynamical system. The augmented state vector, y^n, is a vector with dimension $N_y = N_m + N_p$. In black oil reservoir applications, the state vector can include grid block pressures, two saturations and the solution gas-oil ratio. The state vector can be written as:

$$y^n = \begin{bmatrix} m \\ p^n \end{bmatrix}. \tag{3.16}$$

The superscript n denotes the assimilation time. The update equation for each ensemble member in the EnKF is given by

$$y_j^{n,u} = y_j^{n,f} + C_{Y^{n,f}D^{n,f}}(C_{D^{n,f}D^{n,f}} + C_{D^n})^{-1}(d_{\text{uc},j}^n - d_j^{n,f}) \quad \text{for } j = 1, 2, \cdots N_e, \tag{3.17}$$

where $C_{Y^{n,f}D^{n,f}}$ is the covariance between the forecast state vector $y^{n,f}$ and the predicted data at the nth assimilation time, $d^{n,f}$; $C_{D^{n,f}D^{n,f}}$ is the auto-covariance for predicted data; $d_j^{n,f}$ denotes the predicted data corresponding to the jth ensemble member at time n; $d_{\text{uc},j}^n$ is a sample from the Gaussian distribution $N(d_{\text{obs}}^n, C_{D^n})$, where d_{obs}^n denotes the vector of observed data and C_{D^n} denotes the covariance of the corresponding vector of measurement errors at the nth data assimilation step. The superscript f denotes forecast and the superscript u refers to the result of the update step, e.g., $y_j^{n,u}$ is the jth updated state vector at the nth time by assimilating data $d_{\text{uc},j}^n$.

The covariance matrices $C_{Y^{n,f}D^{n,f}}$ and $C_{D^{n,f}D^{n,f}}$ are approximated from the ensemble; for details see Aanonsen et al. [1] and Zafari and Reynolds [42]. The relatively small ensemble sizes employed to represent covariances in EnKF, introduce sampling errors that tend to produce spurious non-zero long-distance correlations between elements of the state vector and between elements of the state vector and predicted data, whereas in reality, there is not a correlation between variables and data at gridblocks far apart. Spurious correlations can make large changes in components of the augmented state vector due to assimilating data at a location far away from the spatial location of this component, whereas if covariances were accurately represented, no change in the component would occur [9].

Representing covariances with finite samples also limits the degrees of freedom available to assimilate data and update the state vector. Each vector of updated realization of model parameters is a linear combination of the N_e vectors of prior realizations.

It means that there is at most N_e degrees of freedom to assimilate data [9].

Covariance localization is a standard procedure for eliminating or reducing the spurious correlations caused by sampling errors in EnKF [15, 9, 10, 11]. In this approach, each of the auto-covariance of the predicted data vector, $C_{D^{n,f}D^{n,f}}$, and the cross-covariance between the state vector and the predicted data vector, $C_{Y^{n,f}D^{n,f}}$, are replaced with their Schur product with a correlation matrix ρ, e.g., the matrix $C_{Y^{n,f}D^{n,f}}$ is replaced with

$$C_{Y^{n,f}D^{n,f},\rho} = \rho_{YD} \circ C_{Y^{n,f}D^{n,f}}, \tag{3.18}$$

where ρ_{YD} is the correlation matrix. The dimension of the correlation matrix ρ_{YD} is the same as $C_{Y^{n,f}D^{n,f}}$ and equal to $N_y \times N_d$, where N_d is the number of observed data at the nth assimilation step. By replacing $C_{Y^{n,f}D^{n,f}}$ with $\rho_{YD} \circ C_{Y^{n,f}D^{n,f}}$, long-distance correlations are set to zero, in order to remove spurious correlations. Note that the (i, j) entry of the matrix $\rho_{YD} \circ C_{Y^{n,f}D^{n,f}}$ is the product of the (i, j) entry of ρ_{YD} and the (i, j) entry of $C_{Y^{n,f}D^{n,f}}$.

With covariance localization, the EnKF update equation of Eq. 3.17 is replaced with

$$y_j^{n,a} = y_j^{n,f} + (\rho_{YD} \circ C_{Y^{n,f}D^{n,f}})(C_{D^n} + \rho_{DD} \circ C_{D^{n,f}D^{n,f}})^{-1}(d_{\mathrm{uc},j}^{n} - d_j^{n,f}), \tag{3.19}$$

for $j = 1, 2, \cdots N_e$, where ρ_{YD} denotes the correlation matrix between the state vector and data and ρ_{DD} is a $N_d \times N_d$ auto-correlation matrix for predicted data.

In our application of covariance localization, we use the fifth-order compact anisotropic correlation function defined by Gaspari and Cohn [15] to calculate the elements of correlation matrix, so that its principal directions are aligned with the prior covariance function. The correlation function is given by

$$\rho(L_D) = \begin{cases} -\frac{1}{4}L_D^5 + \frac{1}{2}L_D^4 + \frac{5}{8}L_D^3 - \frac{5}{3}L_D^2 + 1, \ 0 \leq L_D \leq 1; \\ \frac{1}{12}L_D^5 - \frac{1}{2}L_D^4 + \frac{5}{8}L_D^3 + \frac{5}{3}L_D^2 - 5L_D + 4 - \frac{2}{3}L_D^{-1}, \quad 1 < L_D \leq 2; \\ 0, \quad L_D > 2, \end{cases} \tag{3.20}$$

where L_D is the dimensionless correlation length which is equal to δ/L. L is the length scale of the correlation function; δ is the Euclidean distance between any grid point and an observation location when computing ρ_{YD}, and the Euclidean distance between two data observation points when computing ρ_{DD}. For more details on covariance localization, see Emerick and Reynolds [9, 10].

3.3 Example 1

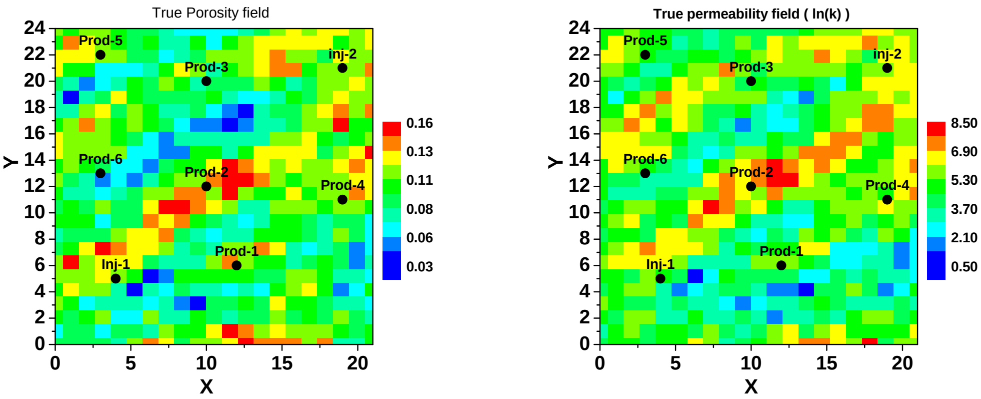
This example pertains to a two-dimensional horizontal reservoir model with 21 × 24 uniform grid. True porosity and log permeability fields which are shown in Fig. 3.1 are generated from an exponential covariance matrix. There are 6 producers and 2 injectors in this reservoir. As can bee seen in Fig. 3.1, there is a high permeability channel which connects Prod-2 to the injectors. There is also a barrier with low permeability and porosity values close to Prod-3 and Prod-6.

The key geostatistical parameters used to generate the true model are listed in Table 3.1. The prior mean of vector of model parameters which contains log-permeability and porosity of gridblocks is uniform and equal to 4.5 for ln(k) and 0.1 for ϕ.

The gridblock dimensions are:

$$\triangle x = \triangle y = 250\,\text{ft}, \ \triangle z = 10\,\text{ft}.$$

The initial reservoir pressure is 4800 psi. Initially the reservoir is at irreducible water saturation, which is $s_{wc} = 0.1$. The total history matching period is 1005 days. At 1005 days, only Prod-2 and Prod-6 have experienced water breakthrough. Observed



(a) True porosity

(b) True $\ln(k)$

Figure 3.1: True porosity and log permeability fields, Example 1.

Table 3.1: Geostatistical parameters of Example 1.

Parameters	Values
ϕ_{mean}	0.1
$[\ln(k)]_{\text{mean}}$	4.50
σ_ϕ	0.0265
$\sigma_{\ln(k)}$	2.0
$\rho_{\phi,\ln(\text{k})}$	0.80
α	30^o
r_1	$18\triangle x$
r_2	$6\triangle x$

data include the flowing bottom hole pressure (BHP) of all wells and water rates of Prod-2 and Prod-6 at 15 day intervals.

Synthetic observed data are generated by adding Gaussian random noise to the true data, where the true data are the simulator output when it is run with the true model. The standard deviation of the noise (measurement error) is 2% of rates for rate data and 1.4 psi for pressure data, i.e., $\sigma_{\text{BHP}} = 1.4$ psi and $\sigma_{q_w} = 0.02q_w$, where q_w is the true data. The minimum measurement error for the water rate is specified to 0.5 STB/D while the maximum measurement error is 3 STB/D.

Well controls are as follows. Oil rate is specified at all producers, such that Prod-

1, Prod-3, Prod-4 and Prod-5 produce at 150 STB/D and Prod-2 and Prod-6 produce at 200 STB/D. The injection rates are 650 and 350 STB/D for Inj-1 and Inj-2, respectively.

Model parameters include porosity and log permeability of all gridblocks. Fluid properties are assumed to be known. The number of model parameters and observed data are:

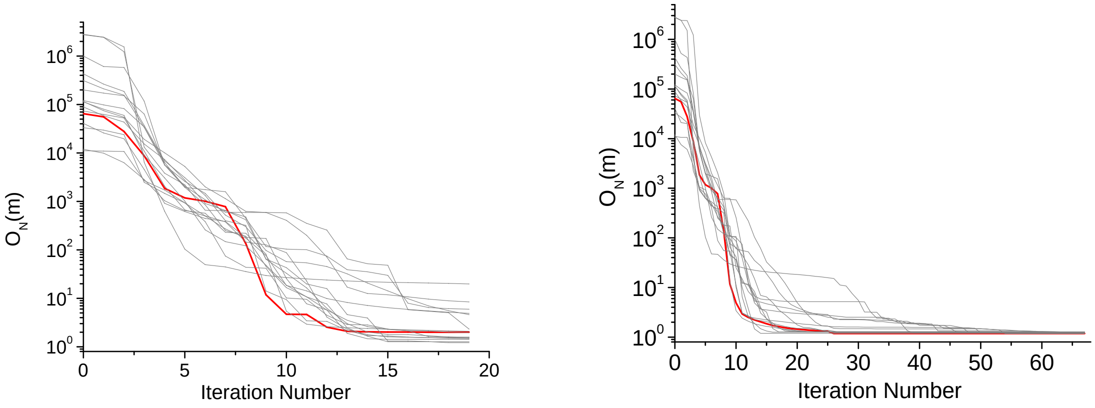
$$N_m = 2N_x \times N_y = 1008,\ N_{\mathrm{d}} = 670.$$

In the following cases, the MAP estimate and $N_e = 15$ RML realizations of porosity and log permeability are generated with SVD-EnRML methods, i.e., a total of 16 conditional realizations are generated. The objective is to compare the computational cost and final values of the normalized objective function for SVD-EnRML-SMM and SVD-EnRML-MI. The discussion about the normalized objective function can be found in Subsection 1.3.1. We also investigate the effect of the maximum number of SVD parameters, $N_{\mathrm{SVD,max}}$, and the effect of the growth and decay factor of the LM parameter.

3.3.1 SVD-EnRML-SMM Results

We applied the SVD-EnRML-SMM algorithm with $N_{\mathrm{SVD,max}} = 30$. Fig. 3.2(a) shows the values of the normalized objective functions of all realizations versus iterations. The algorithm converged in 19 iterations with $O_N(m) = 1.995$ for the MAP estimate. Because the number of SVD parameters was not enough, the normalized objective function of the MAP estimate did not decrease to a value less than $1 + 5\sqrt{2/N_d} = 1.273$. The computational cost are shown in the first row of Table 3.2. As $N_{\mathrm{SVD,max}} = 30$ was specified, from iteration 10 to 19, the Lanczos method computed 30 SVD parameters. Fig. 3.3(c) shows the number of retained singular values versus iterations. Fig. 3.3(b) shows the smallest retained singular value of $G_{D,l}$ versus the iteration index l for this case. The 30th singular value of G_D of the MAP estimate is between 8.4 and 7.8 from iteration $l = 10$ to $l = 19$. The value of λ_{30}/λ_1 is around 0.0012.

Since with a maximum 30 singular triplets, SVD-EnRML-SMM converged with a slightly high value of objective function, we applied the method with $N_{\mathrm{SVD,max}} = 35$, i.e.,



(a) $N_{\text{SVD,max}} = 30$

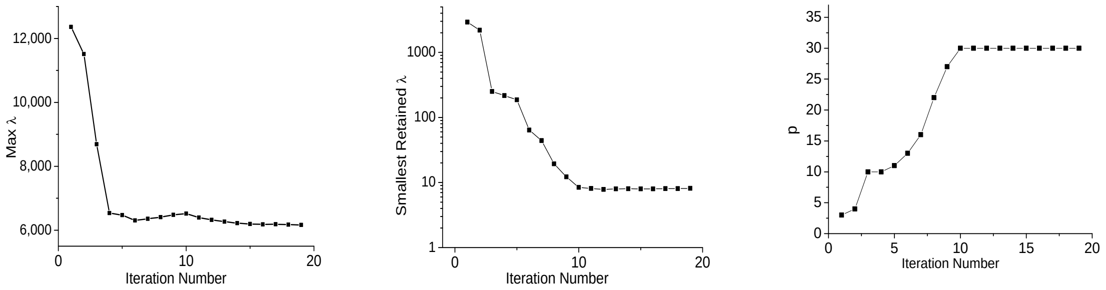
(b) $N_{\text{SVD,max}} = 35$

Figure 3.2: The normalized objective functions of 16 realizations versus iterations of SVD-EnRML-SMM, with $N_{\text{SVD,max}}$ of 30 and 35. The red curve is for the MAP estimate, while gray curves are for 15 RML realizations, Example 1.

we repeated the previous case with $N_{\text{SVD,max}} = 35$. This time, the algorithm converged in 66 iterations. At convergence of the algorithm, the normalized objective functions of all 16 realizations decreased to less than $1 + 5\sqrt{2/N_d} = 1.273$. Fig. 3.2(b) shows the values of the normalized objective functions of realizations versus iterations. The computational cost are shown at the 2nd row of Table 3.2.

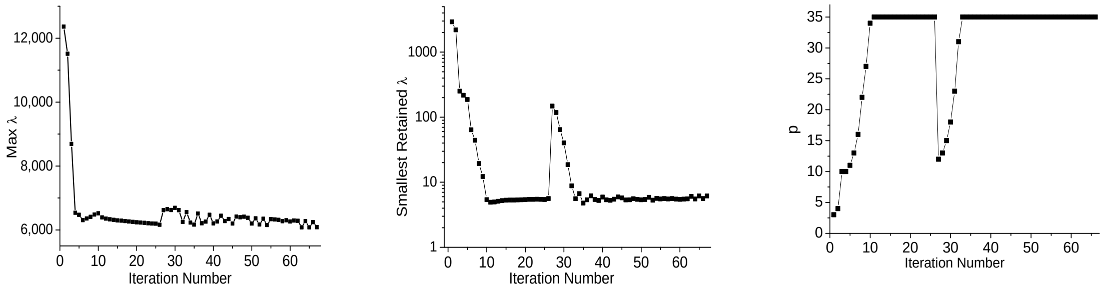
Figs. 3.3 and 3.4 show some details of the SVD-EnRML-SMM algorithm with $N_{\text{SVD,max}}$ of 30 and 35, respectively. Fig. 3.4(c) shows the number of retained singular values versus iterations. The jumps in Figs. 3.4(b) and 3.4(c) happen at the convergence of the MAP estimate where sv-cut is set to $\mu_2 = 0.05$ and the Lanczos algorithm computes a truncated SVD of $G_{D,l}$ corresponding to the realization with the maximum objective function.

Figs. 3.4(a) and 3.3(a) show the first singular value of $G_{D,l}$ versus iterations l of SVD-EnRML-SMM, for the two cases. As one can see, after a few iterations, the first singular value does not change significantly, even after the convergence of the MAP estimate, where at each iteration, a truncated SVD of $G_{D,l}$ for the realization with the maximum objective function is calculated. Fig. 3.4(b) shows the smallest retained singular value of $G_{D,l}$ versus iterations l for this case. The 35th singular value of G_D is



(a) First SV (b) Smallest retained SV (c) # of retained SVs

Figure 3.3: The first singular value and the smallest retained singular value of G_D and the number of retained SVD parameters versus iterations of SVD-EnRML-SMM, with $N_{\mathrm{SVD,max}} = 30$, Example 1.



(a) First SV (b) Smallest retained SV (c) # of retained SVs

Figure 3.4: The first singular value and the smallest retained singular value of G_D and the number of retained SVD parameters versus iterations of SVD-EnRML-SMM, with $N_{\mathrm{SVD,max}} = 35$, Example 1.

between 4.6 and 6, and the value of λ_{35}/λ_1 is around 0.0008.

According to the results presented here, at convergence of the SVD-EnRML-SMM algorithm with $N_{\text{SVD,max}} = 35$, $O_N(m)$ of all realizations decreased to small values, however 30 SVD parameters at late iterations was not enough to obtain values of $O_N(m)$ that satisfy Eq. 1.12. Note that if we were to specify a final sv-cut of 0.0012 or higher, the algorithm would have converged with slightly high values of the normalized objective function. Further, it shows that in general, a small sv-cut at late iterations ($\mu_{\min}$) does not guarantee small values of objective functions at convergence of the algorithm.

3.3.2 SVD-EnRML-MI Results

Here the SVD-EnRML-MI algorithm is applied to generate a total of 16 conditional realizations. Since we want to compare the results of SVD-EnRML-MI with the results of SVD-EnRML-SMM, at first we apply the algorithm with $N_{\text{SVD,max}} = 35$.

With $N_{\text{SVD,max}} = 35$, the SVD-EnRML-MI algorithm converged in 51 iterations. At convergence, normalized objective functions of all realizations decreased to less than $1 + 5\sqrt{2/N_d} = 1.273$. The computational costs are shown at the 3rd row of Table 3.2. By comparing the 2nd and the 3rd rows of Table 3.2, one can see that the computational cost of SVD-EnRML-MI in terms of equivalent simulation runs is less than the one from SVD-EnRML-SMM.

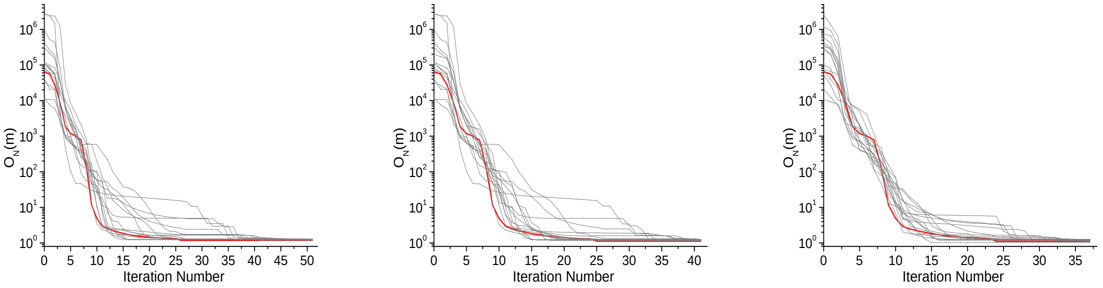
According to the results presented, the SVD-EnRML-MI is a more efficient algorithm than SVD-EnRML-SMM. In addition, with our modifications in applying the modified Levenberg-Marquardt algorithm, SVD-EnRML-SMM algorithm required far less computational cost compared to the results of Tavakoli and Reynolds [39].

In order to investigate the effect of the maximum number of SVD parameters, $N_{\text{SVD,max}}$, on the computational cost and convergence behavior, we applied the SVD-EnRML-MI algorithm with $N_{\text{SVD,max}}$ of 45 and 55. The same prior realizations, and the same parameters for sv-cut are used in all cases. The last 3 rows of Table 3.2, show the computational cost of SVD-EnRML-MI with 35, 45 and 55 maximum SVD parameters. As one can see in this table, increasing the maximum number of SVD parameters from

35 to either 45 and 55, actually decreased the computational costs. Fig. 3.5 shows the values of $O_N(m)$ of all realizations versus iterations of SVD-EnRML-MI with 3 different numbers of $N_{\text{SVD,max}}$. By increasing $N_{\text{SVD,max}}$, although the cost of computing the truncated SVD of $G_{D,l}$ at each of the late iterations increases, the convergence is obtained in fewer iterations. We also see that, by increasing $N_{\text{SVD,max}}$ from 45 to 55, the convergence is obtained in slightly fewer iterations, and the total computational cost is close for the two cases. The 45th retained singular value of $G_{D,l}$ is around 2.6 while the 55th singular value is around 1.3. With

Table 3.2: Summary of the computational costs of SVD-EnRML algorithms for generating 16 conditional realizations of porosity and log permeability, and the $\max\{O_N(m)\}$ at convergence, Example 1.

Method	$N_{\text{SVD,max}}$	Sim.	Direct	Adjoint	Equ. Sim. Runs	N_{iter}	$\max\{O_N(m)\}$
SMM	30	471	586	567	759	19	19.82
SMM	35	1345	2727	2660	2692	67	1.27
MI	35	1287	2020	1968	2284	51	1.272
MI	45	933	1825	1784	1835	41	1.266
MI	55	875	2060	2023	1896	37	1.272



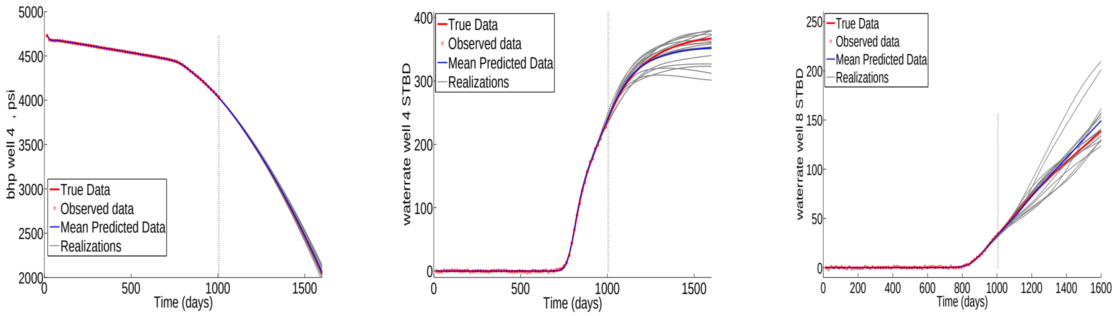
(a) $N_{\text{SVD,max}} = 35$

(b) $N_{\text{SVD,max}} = 45$

(c) $N_{\text{SVD,max}} = 55$

Figure 3.5: The normalized objective functions of 16 realizations versus iterations of SVD-EnRML-MI, with 3 different $N_{\text{SVD,max}}$. The red curve is for the MAP estimate, while gray curves are frrom 15 RML realizations.

Now we show some of the results of SVD-EnRML-MI with $N_{\text{SVD,max}} = 55$. Data matches and predictions for 2 of the wells generated from the history matched realizations are shown in Fig. 3.6. The quality of data matches and predictions of p_{wf} of the wells not shown, are similar. The true prediction is between the uncertainty bounds and close to the mean. Fig. 3.7 shows the MAP estimate for this case. The porosity and log permeability fields of the MAP estimate are smooth and they contain the main features of the truth. Fig. 3.8 shows 3 unconditional realizations of log permeability fields, and Fig. 3.9 shows the corresponding updated realizations of log permeability fields. All of the conditional realizations match the data equally well and they display some of the main large scale features of the truth shown in Fig. 3.1, however they are distinct as each one is a sample of the posterior pdf. As the quality of conditional porosity fields were similar, they are not shown here.

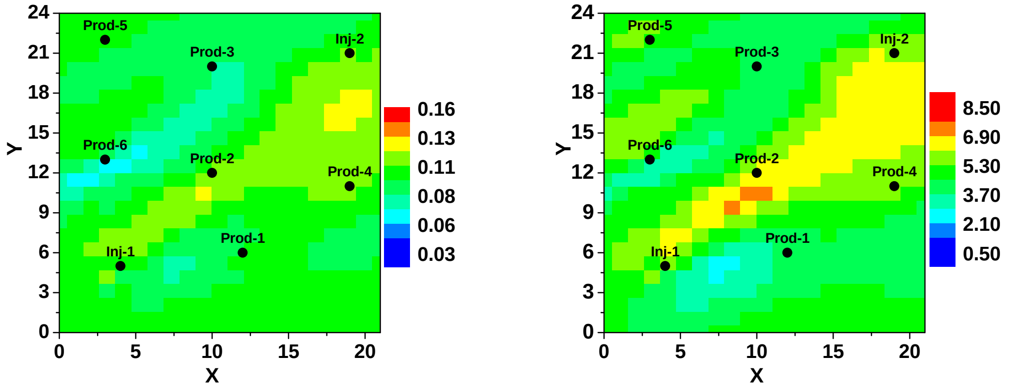


(a) p_{wf} of Prod-2 (b) q_w of Prod-2 (c) q_w of Prod-6

Figure 3.6: Data matches and predictions from realizations of SVD-EnRML-MI with $N_{\text{SVD,max}} = 55$ for two of the wells. The dashed vertical line shows the end of the history matching period (1005 days), Example 1.

3.3.3 Growth and Decay Factor of the LM Parameter

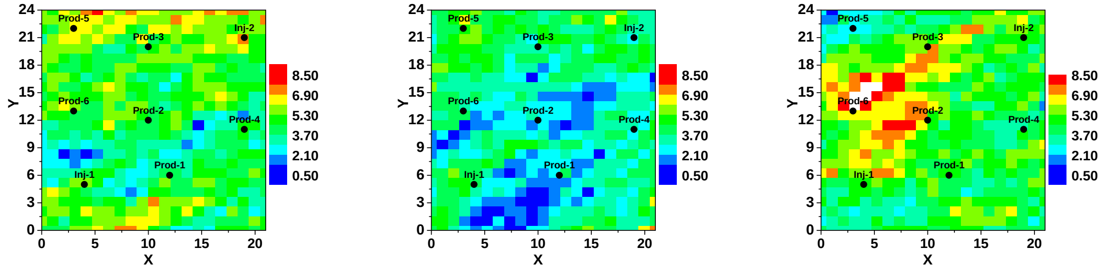
With the improvement in the computational efficiency obtained by using more SVD parameters, here the effect of the growth and decay factor of the LM parameter is investigated. The growth and decay factor of the LM parameter is defaulted to 10. We applied the SVD-EnRML-MI algorithm with a growth and decay factor of 5 for two cases, one with $N_{\text{SVD,max}} = 45$, and the other one with $N_{\text{SVD,max}} = 55$. The summary of the computational costs are shown in Table 3.3. In all 4 cases of this table, at convergence, $O_N(m)$ of all realizations decreased to less than $1+5\sqrt{2/N_d} = 1.273$. The 3rd column of this table shows the factor by which the LM parameter is decreased or increased (growth and decay factor). As one can see, depending on $N_{\text{SVD,max}}$, using a factor of 5 for the LM



(a) porosity field

(b) ln(k) field

Figure 3.7: Porosity and log permeability fields of the MAP estimate, Example 1.

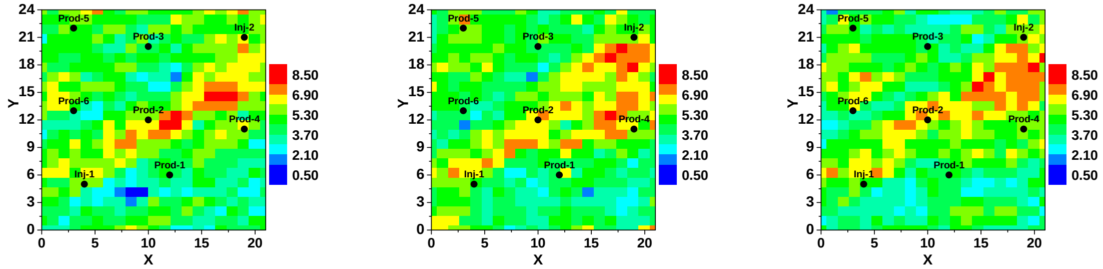


(a) Realization 2

(b) Realization 8

(c) Realization 15

Figure 3.8: 3 unconditional realizations of log-permeability field, Example 1.



(a) Realization 2

(b) Realization 8

(c) Realization 15

Figure 3.9: 3 conditional realizations of log-permeability field, generated with SVD-EnRML-MI, Example 1.

parameter may either decrease or increase the computational cost. As using a smaller factor for changing γ does not necessarily decrease the computational costs, for the rest of this work we use the default value of 10 for changing the LM parameter.

With a growth and decay factor of 5, the MAP estimate converged in 20 iterations for both $N_{\mathrm{SVD,max}}$ of 55 and 45, while with a growth and decay factor of 10, it required 24 iterations or more to obtain convergence. One can conclude that in generating only the MAP estimate, using a growth and decay factor of 5 instead of 10 may decrease the computational cost; however, this is not necessarily correct for all examples.

Table 3.3: Summary of the computational costs of SVD-EnRML-MI algorithm with different input parameters for generating 16 conditional realizations of porosity and log permeability fields, Example 1.

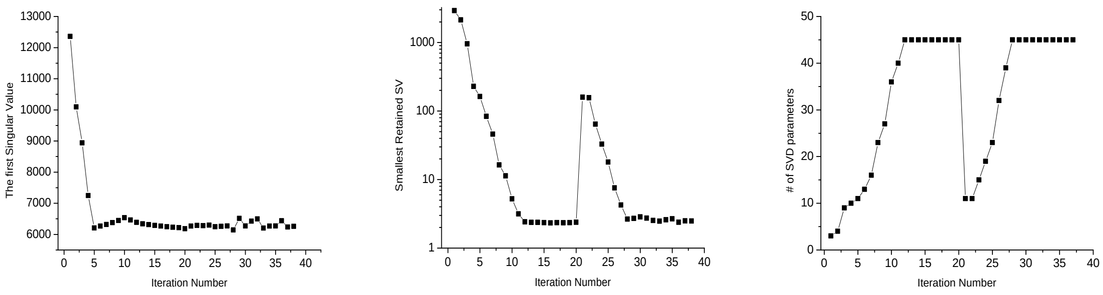
Algorithm	$N_{\mathrm{SVD,max}}$	γ factor	Sim.	Direct	Adjoint	Equ. Sim. Runs	N_{iter}
MI	45	10	933	1825	1784	1835	41
MI	45	5	888	1658	1620	1708	38
MI	55	10	875	2060	2023	1896	37
MI	55	5	1042	2019	1979	2042	39

Some details on the singular values versus iteration corresponding to the 2nd and 4th rows of Table 3.3 are shown in Figs. 3.10 and 3.11, respectively. Fig. 3.10(a) shows how the first singular value of $G_{D,l}$ changes versus iterations l of SVD-EnRML-MI. According to this figure, after a few iterations, the first singular value does not change significantly. Figs. 3.10(b) and 3.11(b) show the smallest retained singular value of $G_{D,l}$ versus iterations l. In Fig. 3.10(b), the 45$^{\mathrm{th}}$ retained singular value is around 2.33 for the MAP estimate, and it changes between 2.85 and 2.38 after the convergence of the MAP estimate. In Fig. 3.11(b), the 55th retained singular value is around 1.22 . Figs. 3.10(c) and 3.11(c) display the number of computed singular triplets versus iterations. The jumps in Figs. 3.10(c) and 3.10(b) happen at the convergence of the MAP estimate, where sv-cut is set to $\mu_2 = 0.05$ and the algorithm computes a truncated SVD of $G_{D,l}$ for the realization with the maximum objective function.

Fig. 3.12 shows the value of the LM parameter for the MAP estimate and an

RML realization versus iteration number of SVD-EnRML-MI with $N_{\text{SVD,max}} = 45$ and a growth and decay factor of 5 for the LM parameter. The 12th RML realization has converged in 16 iterations with $O_N(m) = 1.243$. Note that at each iteration, when the objective function decreased, the value of γ^l is divided by a factor (which is 5 in this case) for the next iteration. According to the results of Fig. 3.12, the value of the LM parameter is greater than 25 even at late iterations. A LM parameter in the order of 100, reduces the weight of the right singular vectors corresponding to small singular values in the search direction, however, it does not eliminate them. They affect the search direction, but with a lower weight compared to the one with a LM parameter of zero.

In Example 1, the lowest computational cost is obtained with $N_{\text{SVD,max}} = 45$ and a factor of 5 for changing the LM parameter. Using a higher number of 55 SVD parameters did not change the computational cost significantly; while with $N_{\text{SVD,max}} = 35$, the computational cost was the highest.

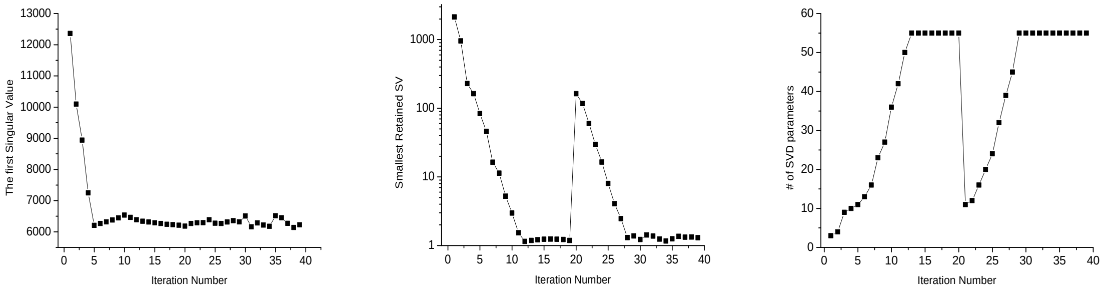


(a) First SV (b) Smallest retained SV (c) # of retained SVs

Figure 3.10: The largest singular value (SV) and the smallest retained singular value of G_D and the number of retained SVD parameters versus iterations of SVD-EnRML-MI with $N_{\text{SVD,max}} = 45$, corresponding to the 2nd row of Table 3.3, Example 1.

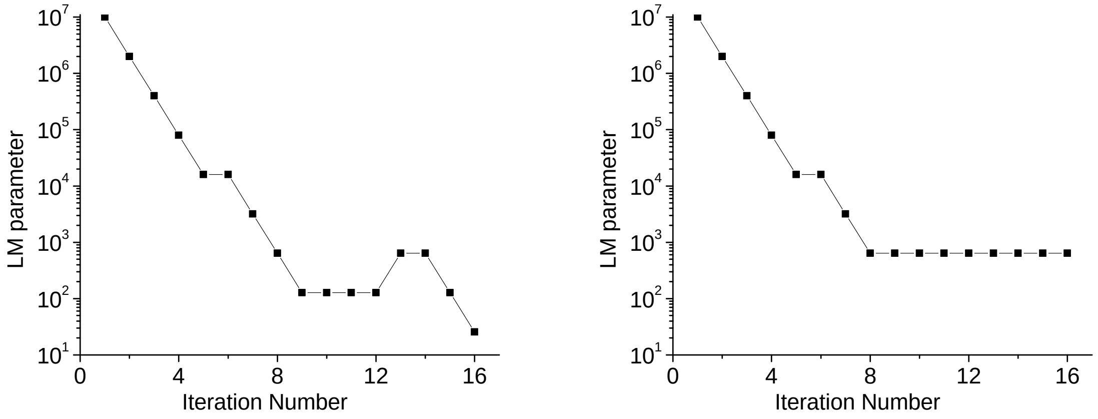
3.3.4 Controlling the Change in the Objective Function at Early Iterations

In Example 1, $\gamma^0 = 10^7$ was used for all realizations. Even with this large LM parameter, we require the change in the objective function to be less than 50%, i.e., if the objective function of a realization decreased by more than 50% at iteration $l = 1$ and $l = 2$, the LM parameter is increased by the specified growth factor and the iteration is



(a) First SV (b) Smallest retained SV (c) # of retained SVs

Figure 3.11: The largest singular value and the smallest retained singular value of G_D and the number of retained SVD parameters versus iterations of SVD-EnRML-MI with $N_{\mathrm{SVD,max}} = 55$, corresponding to the 4th row of Table 3.3, Example 1.



(a) γ_0^l (b) γ_{12}^l

Figure 3.12: The LM parameter of the MAP estimate (a) and an RML realization (b) versus iteration number l of SVD-EnRML-MI with $N_{\mathrm{SVD,max}} = 45$ and a growth and decay factor of 5 for the LM parameter.

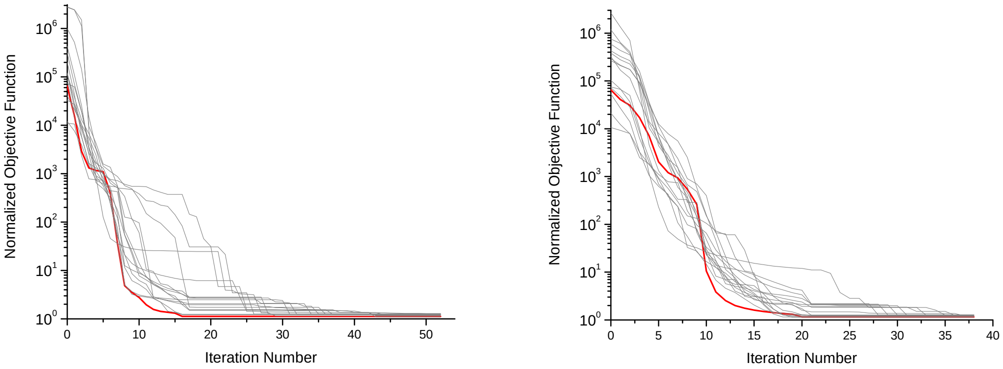
redone until the change is less than 50%.

After obtaining the results corresponding to the 2nd row of Table 3.3, we repeated that case using no control on the change in the objective functions at the first two iterations. Table 3.4 shows the comparison between the two cases. In both cases, $N_e = 15$, $N_{\text{SVD,max}} = 45$, $\mu_1 = 0.5$, $\mu_2 = 0.05$, $\mu_{\min} = 0.0001$, $\gamma_0 = 10^7$ and the factor by which the LM parameter changes is 5. In both cases, at convergence, $O_N(m)$ of all realizations decreased to less than $1 + 5\sqrt{2/N_d}$. The first row of Table 3.4 shows the computational costs when the change in the objective function was controlled at the first 2 iterations, while the second row is for the case where the objective function could decrease by any value. Note that when the change in the objective function is not controlled over the first two iterations, the total computational cost increases.

Fig. 3.13 shows the normalized objective functions of realizations versus iterations for the two cases. According to the results of this figure, when the change in the objective function is not controlled over the first two iterations, the MAP estimate converged in 16 iterations, while by controlling the change in the objective function, it converged in 20 iterations. The conclusion is that, in generating a single realization with SVD parameterization algorithm, with a high starting value of the LM parameter, e.g, 10^6, there is no need to control the change in the objective function over the first iterations; however in generating multiple realizations with SVD-EnRML algorithms, avoiding large changes of objective functions at the first two iterations may decrease the total computational costs. One reason is that at early iterations of SVD-EnRML, the search directions for RML realizations are computed using a truncated SVD of $G_{D,l}$ corresponding to the MAP estimate which is different from the actual truncated SVD of $G_{D,l}$ for each realization. For the MAP estimate, since a truncated SVD of its actual $G_{D,l}$ is computed, with a high starting LM parameter, there is no need to control the change of its objective function; while a large decrease in the objective function of an RML realization is not desired.

Table 3.4: Comparison between the computational costs of SVD-EnRML-MI algorithm with and without controlling the change of $O_j(m)$ at the first 2 iterations, for generating 16 conditional realizations of porosity and log permeability.

Algorithm	Simulations	Direct	Adjoint	Equ. Sim. Runs	N_{iter}
MI	888	1658	1620	1708	38
MI (no control)	1381	2473	2421	2604	52



(a) No control in change of $O(m)$

(b) Controlling change of $O(m)$ at the first two iterations

Figure 3.13: $O_N(m)$ versus iterations of SVD-EnRML-MI. Red curve is for the MAP estimate while the gray curves are for 15 conditional realizations. Example 1.

3.4 Example 2

The true model for this example is the same as the one for the previous 21 × 24 synthetic reservoir. True porosity and log permeability fields are shown in Fig. 3.1. The details of the covariance matrix are explained in Example 1.

The difference between this example and Example 1 is the well controls. In this example, producers are controlled with pressure specified. The total history matching period is 1200 days. At 1200 days, only Prod-2 and Prod-6 have experienced water breakthrough. Observed data include the flowing bottom hole pressure (BHP) of injectors and oil and water rates of all producers at 30 day intervals. Although four of the producers have not experienced water breakthrough at 1200 days, the zero water rates of these wells after adding synthetic measurement error are included in the vector of observed data.

Synthetic observed data are generated by adding Gaussian random noise to the true data. The standard deviation of the noise (measurement errors) is 2% of rates for the rate data and 1.4 psi for pressure data, i.e., $\sigma_{\text{BHP}} = 1.4$ psi and $\sigma_q = 0.02q$, where q is the true rate. The minimum measurement error for the rates is specified to 0.5 STB/D while the maximum measurement error is 3 STB/D.

The initial reservoir pressure is 4800 psi. All 6 producers are controlled with the same specified p_{wf}. The summary of well controls are given in Table 3.5.

Table 3.5: Well controls of Example 2. p_{wf} is specified at producers and injection rate is specified at injectors. Rates are in STB/D.

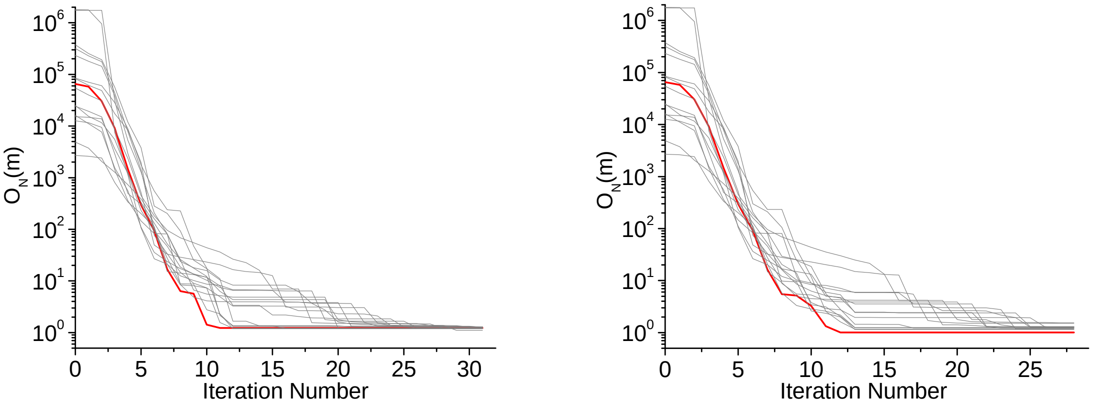
Time Period, days	q_w of Inj-1	q_w of Inj-2	p_{wf} of producers, psi
0 – 300	650	350	4500
300 – 600	650	350	4200
600 – 900	650	350	4000
900 – 1200	650	350	3800
1200 – 1500	650	350	3500
1500 – 1800	650	350	3200
1800 – 2100	650	350	3000
2100 – 3000	650	350	2500

Model parameters include porosity and log permeability of all gridblocks. Fluid properties are assumed to be known. The number of model parameters and observed data are:

$$N_m = 2N_x \times N_y = 1008, \; N_{\text{d}} = 560.$$

3.4.1 Results of SVD-EnRML Algorithms

We applied the SVD-EnRML-SMM, SVD-EnRML-AG and SVD-EnRML-MI algorithms to generate the MAP estimate and 15 RML realizations. The objective is to compare the results based on the computational cost, final values of the objective functions and convergence behavior. The maximum number of SVD parameters, $N_{\text{SVD,max}}$, was specified to be equal to 60 for all three methods. The first, second and the forth rows of Table 3.6 show the computational costs and the results of the three algorithms mentioned, where we used the same input parameters for all. At convergence of both the SVD-EnRML-SMM and SVD-EnRML-MI algorithms, $O_N(m)$ of all realizations decreased to less than $1 + 5\sqrt{2/N_d} = 1.299$. $O_N(m)$ of realizations versus iterations for SVD-EnRML-SMM and SVD-EnRML-MI are shown in Figs. 3.15(a) and 3.14(a), respectively. $O_N(m)$ of realizations versus iterations for SVD-EnRML-AG is shown in Fig. 3.15(b). Not that in Fig. 3.15(b), an even iteration number is for the result of the inner loop, while an odd iteration number is for the outer or main loop. SVD-EnRML-MI converged in fewer iterations and its computational cost is less than SVD-EnRML-SMM in terms of equivalent simulation runs. The computational cost of SVD-EnRML-AG is far more than the two other algorithms. The convergence rate of VD-EnRML-AG is slower than the two other algorithms.

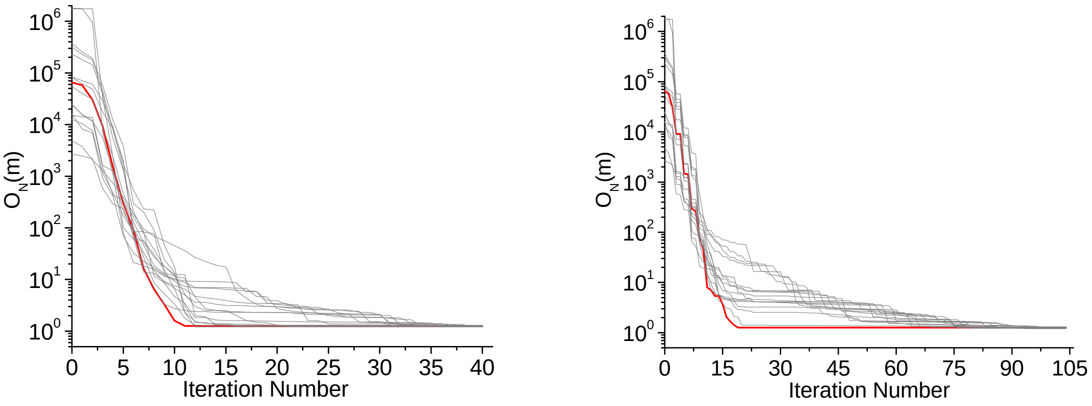


(a) MI, $N_{\text{SVD,max}} = 60$

(b) MI, $N_{\text{SVD,max}} = 45$

Figure 3.14: $O_N(m)$ versus iterations of SVD-EnRML-MI algorithm. Red curve is for the MAP estimate while the gray curves are for 15 RML realizaions. Example 2.

In order to see the effect of $N_{\text{SVD,max}}$ on convergence properties, we also ran the SVD-EnRML-MI algorithm with a maximum number of 45 SVD parameters. Fig. 3.14(b) shows $O_N(m)$ of realizations versus iterations for this case. As can be seen from the results in the 3rd row of Table 3.6, the computational efficiency has been improved by decreasing the maximum number of singular triplets. However at convergence, the max-



(a) SMM, $N_{\text{SVD,max}} = 60$

(b) AG, $N_{\text{SVD,max}} = 60$

Figure 3.15: $O_N(m)$ versus iterations of SVD-EnRML-SMM and SVD-EnRML-AG algorithms. Red curve is for the MAP estimate while the gray curves are for 15 RML realizaions. Example 2.

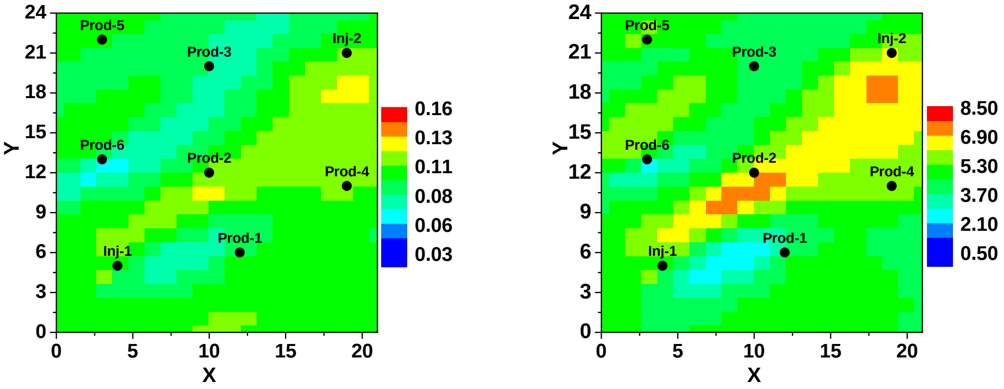
imum $O_N(m)$ of realizations is 1.53, and 4 of the realizations out of 16 have $O_N(m)$ higher than 1.299. This result shows that when using a small number of singular triplets, at convergence of the algorithm, the objective functions might have slightly high values. The 45th singular value of G_D varied from 2.2 for the MAP estimate to 4.6 at the last iteration. The 60th singular value varied from 0.98 to 1.8.

The comparison in Table 3.6, shows that SVD-EnRML-MI improves the computational efficiency over SVD-EnRML-SMM. In addition, SVD-EnRML-AG does not improve the results compared to SVD-EnRML-SMM.

Table 3.6: Summary of computational costs of SVD-EnRML algorithms for generating 16 conditional realizations of porosity and log permeability, and the $\max\{O_N\}$ at convergence, Example 2.

Method	$N_{\text{SVD,max}}$	Sim.	Direct	Adjoint	Equ. Sim. Runs	N_{iter}	$\max\{O_N\}$
SMM	60	837	2157	2117	1905	40	1.297
MI	60	953	1452	1421	1681	31	1.296
MI	45	823	1066	1038	1349	28	1.533
AG	60	4527	3177	4567	6463	52	1.297

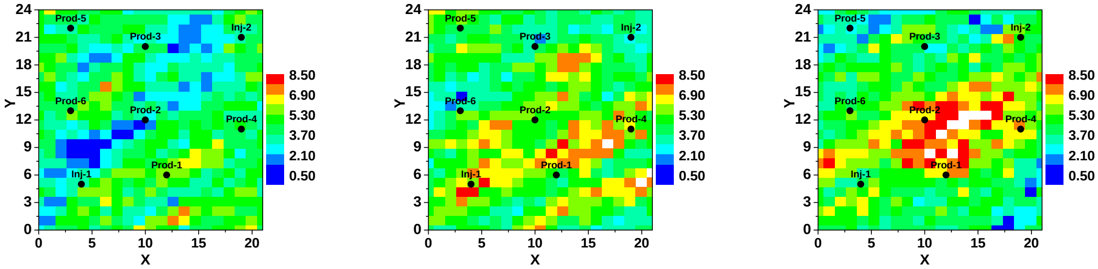
Here we present some of the results of SVD-EnRML-MI with $N_{\text{SVD,max}} = 60$, corresponding to the 2nd row of Table 3.6. Fig. 3.16 shows the porosity and log permeability fields of the MAP estimate. For the MAP estimate, the prior porosity is uniform and equal to 0.1 and the prior log permeability is uniform 4.5. In Figs. 3.17 and 3.18, respectively, 3 unconditional and conditional realizations of the log permeability field are shown. In Figs. 3.19 and 3.20, respectively, 3 unconditional and conditional realizations of the porosity field are shown. All conditional realizations of the permeability field have similar features of the truth shown in Fig. 3.1, including the channel through the injectors and Prod-2; but the porosity field is not well resolved by the data.



(a) Porosity field

(b) ln(k) field

Figure 3.16: Porosity and log permeability fields of the MAP estimate, Example 2.



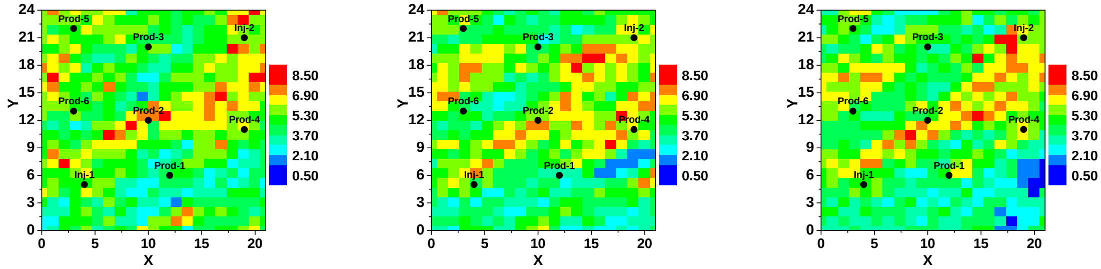
(a) Realization 2

(b) Realization 8

(c) Realization 15

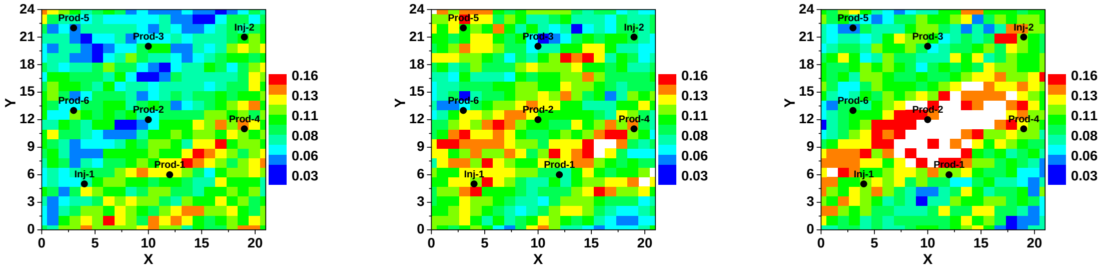
Figure 3.17: 3 unconditional realizations of log-permeability field, Example 2.

Fig. 3.21 shows the water saturation distribution at the end of history matching period for the true model, the MAP estimate and one of the conditional realizations.



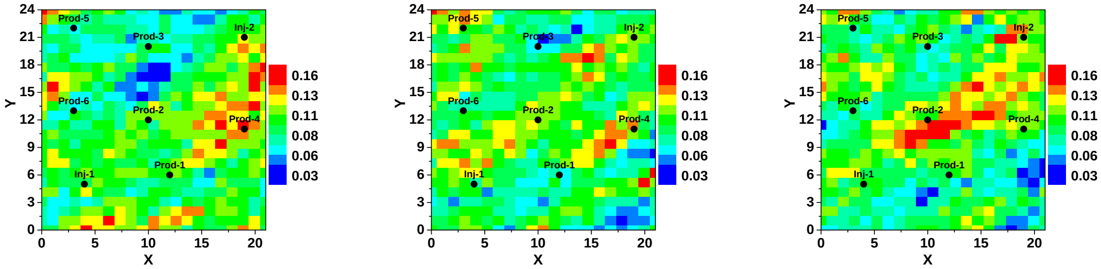
(a) Realization 2 (b) Realization 8 (c) Realization 15

Figure 3.18: 3 conditional realizations of log-permeability field, generated with SVD-EnRML-MI, Example 2.



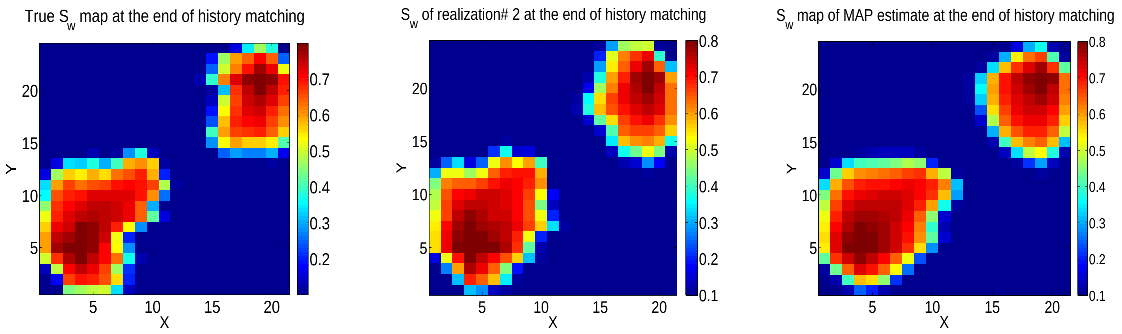
(a) Realization 2 (b) Realization 8 (c) Realization 15

Figure 3.19: 3 unconditional realizations of porosity field, Example 2.



(a) Realization 2 (b) Realization 8 (c) Realization 15

Figure 3.20: 3 conditional realizations of porosity field generated with SVD-EnRML-MI, Example 2.



(a) True S_w (b) S_w of realization 2 (c) S_w of the MAP estimate

Figure 3.21: Distribution of S_w at the end of the history matching period (1200 days) for the true model, MAP estimate and a conditional realization, Example 2.

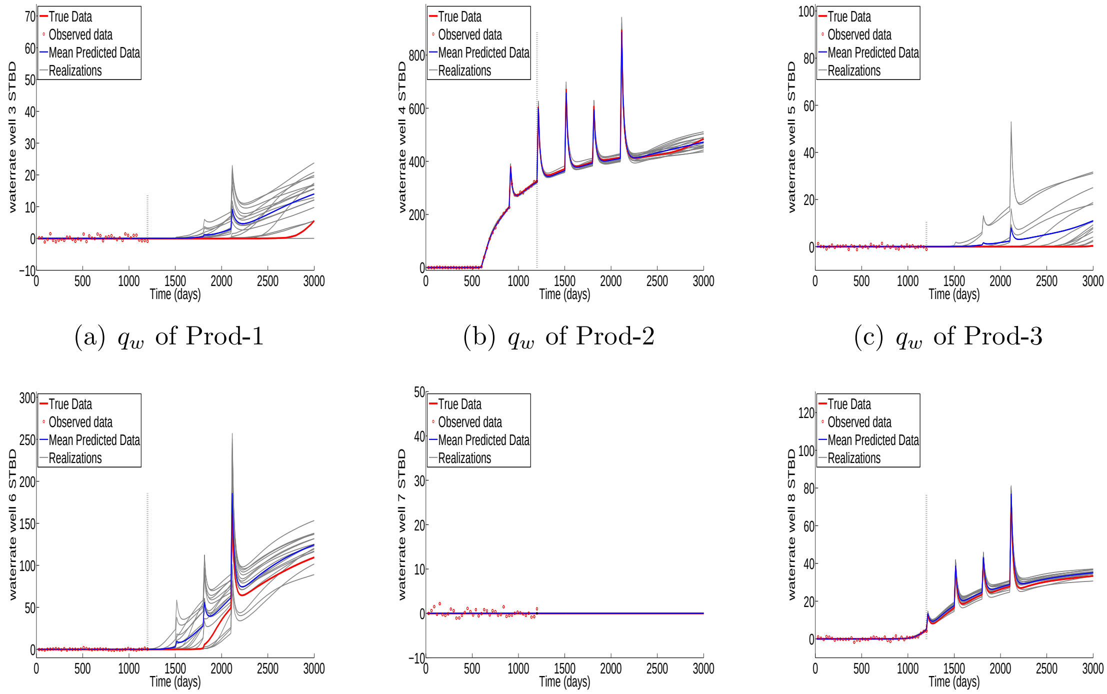
Figs. 3.22 and 3.23, respectively, show the data matches ($0 < t < 1200$ days) and predictions ($1200 < t < 3000$ days) of water rates and oil rates for some of the producers. Data matches and predictions of the wells not shown are similar to those shown. Since p_{wf} of producers are reduced at certain times, oil rates and water rates of the wells have a peak at those times. In Figs. 3.22(a) and 3.22(c) the mean of the conditional sample does not predict the water breakthrough time correctly. This may be due to the fact that the number of realizations is small and 16 realizations is not enough to perform reliable predictions of breakthrough time for this example.

Figs. 3.25 and 3.26 show the cumulative oil and water production of the field from the unconditional and conditional realizations, respectively. The uncertainty in cumulative oil and water production is significantly reduced by conditioning the realizations to production data.

3.4.2 Comparison of The Subspace Method and SVD Parameterization

In this subsection, we apply the subspace method to generate the MAP estimate for porosity and log permeability fields, and we compare the results with SVD parameterization. Then we compare the subspace method when applying RML with SVD-EnRML.

The subspace method for generating the MAP estimate converged in 11 iterations. The initial value of the LM parameter is 10^7.



(a) q_w of Prod-1

(b) q_w of Prod-2

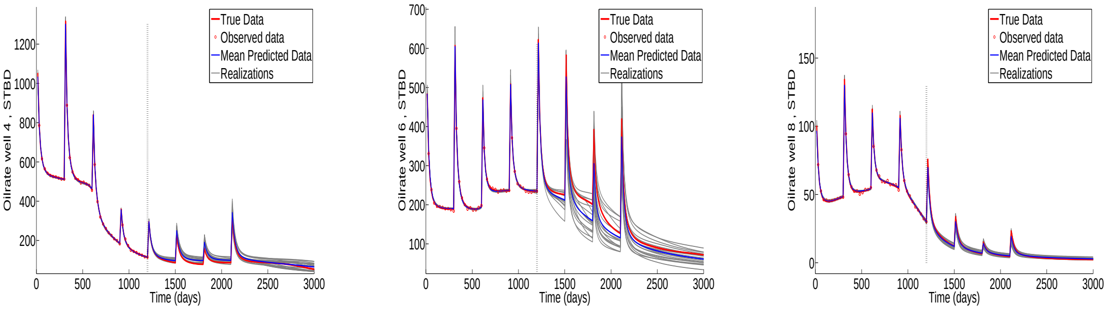
(c) q_w of Prod-3

(d) q_w of Prod-4

(e) q_w of Prod-5

(f) q_w of Prod-6

Figure 3.22: Data matches and predictions of q_w. The dashed vertical line shows the end of the history matching period (1200 days), Example 2.

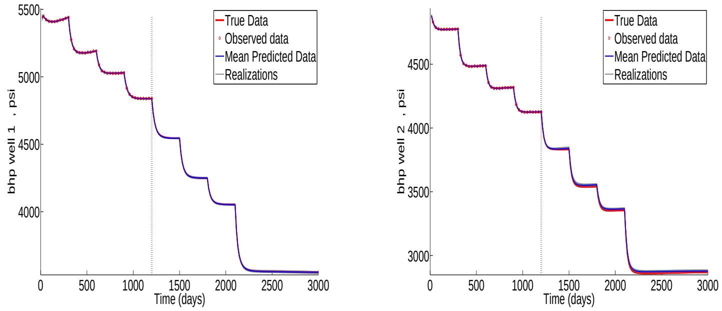


(a) q_o of Prod-2

(b) q_o of Prod-4

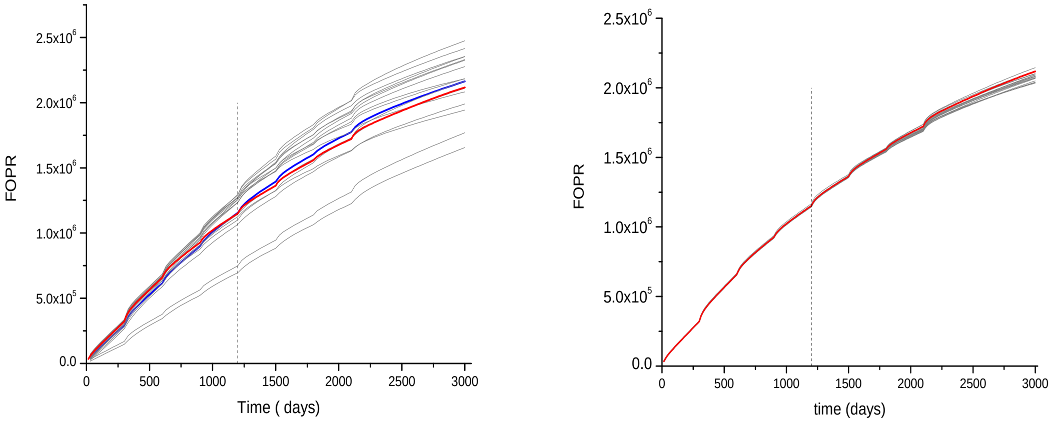
(c) q_o of Prod-6

Figure 3.23: Data matches and predictions of q_o for 3 of producers. The dashed vertical line shows the end of the history matching period (1200 days), Example 2.



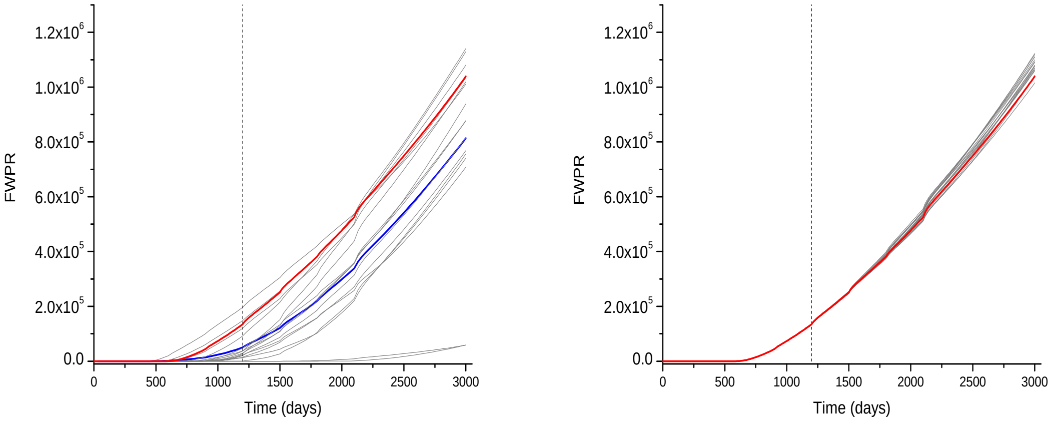
(a) p_{wf} of Inje-1 (b) p_{wf} of Inje-2

Figure 3.24: Data matches and predictions of p_{wf} of the two injectors. The dashed vertical line shows the end of the history matching period (1200 days), Example 2.



(a) prior FOPR (b) posterior FOPR

Figure 3.25: Field oil production (in STB) from the prior and the posterior samples, red curve is from the truth, gray curves are from 16 realizations, blue shows the mean of gray curves. The dashed line shows the end of the history matching period (1200 days), Example 2.



(a) prior FWPR

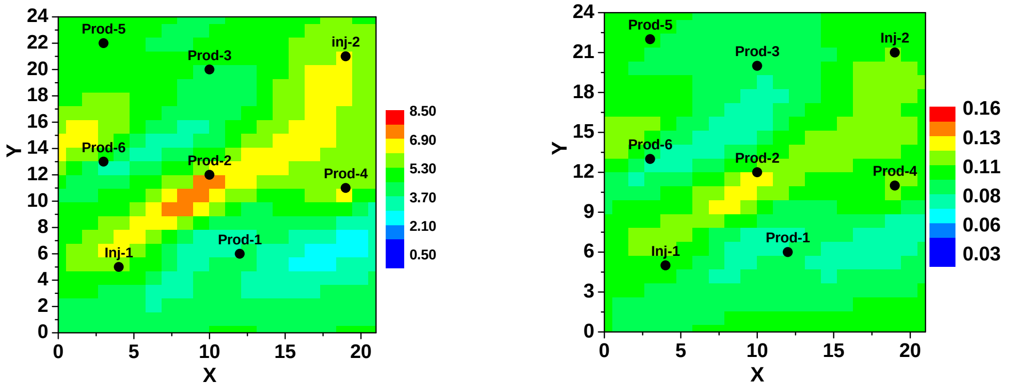
(b) posterior FWPR

Figure 3.26: Field water production (in STB) from the prior and the posterior samples, red curve is from the truth, gray curves are from 16 realizations, blue shows the mean of gray curves. The dashed vertical line shows the end of the history matching period (1200 days), Example 2.

As discussed before, the subspace vectors are the gradients of subobjective functions multiplied by C_M. The observed data in this example include p_{wf} of injectors, and oil rates and water rates of the producers. At the first iteration, each subobjective function contains one data type from one well, e.g., oil rate data from Prod-3 are used in one subobjective function while water rate data from Prod-3 are used in another subobjective function. Although the zero water rates after adding noise were added to observed data, if water rate of a well is zero during the whole history matching period, it is not used in a subobjective function.

The number of the subspace vectors is increased simply by partitioning each data type of each well into smaller time intervals. If a well has nonzero water rate data during the history matching period, they are used together in one subobjective function during all iterations, i.e., water rate data of a well is not partitioned into smaller time intervals. Note that if we partition water rate data into time intervals, and a subobjective function includes only zero rates, it will not provide an appropriate subspace vector.

Now we further explain the partitioning of data. In this example, there is 1200 days of data history with observed data at 30 day intervals. At iteration 5, each data type from each well (except q_w data) is divided into 2 groups, one containing data until 600 days and the other one containing data after 600 days until 1200 days. Each group is used in a separate subobjective function. From iteration 7 until convergence, each data type from each well is divided into 6 uniform time intervals, each of length 200 days. At this point, some of the subobjective functions contain 6 observed data and some contain 7. The choice was arbitrary. After the first iteration, the model mismatch vectors for each of the porosity and log permeability parameters are used in 2 additional subspace vectors.



(a) ln(k) field

(b) porosity field

Figure 3.27: Porosity and log permeability fields of the MAP estimate by the subspace method, Example 2.

Table 3.7: Computational cost of the LM algorithm with the subspace method and SVD parameterization for generating the MAP estimate, Example 2.

Algorithm	Simulations	Direct	Adjoint	Equ. Sim. Runs	N_{iter}
Subspace method	15	362	353	193	11
SVD parameterization	15	280	275	153	11

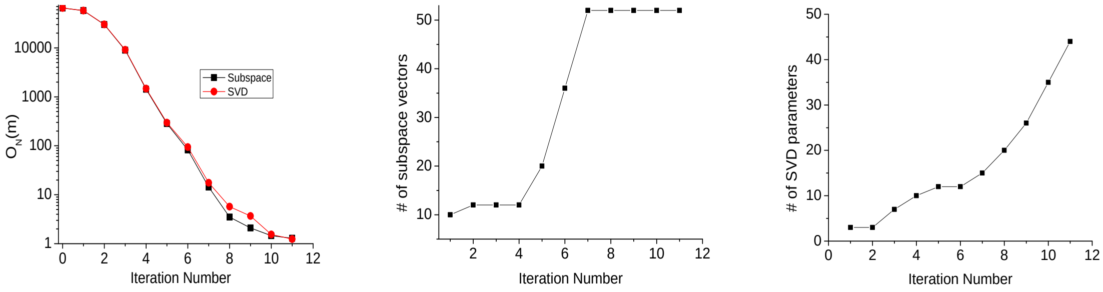
Fig. 3.27 shows the porosity and log permeability fields of the MAP estimate obtained with the subspace method. The MAP estimate is smooth and it contains the main features of the truth. Table 3.7 shows the computational costs of the subspace method

Table 3.8: Final values of objective function of the MAP estimate with the subspace and SVD parameterization methods.

Algorithm	$O_d(m)$	$O_m(m)$	$O_N(m)$
Subspace method	350.1	10.5	1.288
SVD parameterization	338.5	9.9	1.244

and SVD parameterization algorithm in generating the MAP estimate. Fig. 3.28(a) shows the value of $O_N(m)$ versus iterations of the subspace method and SVD parameterization. It shows that both algorithms have similar convergence properties and they converged in the same number of iterations. Table 3.8 shows the final values of the normalized objective function at convergence. Note that the final values of the normalized objective functions are less than $1 + 5\sqrt{2/N_d} = 1.299$.

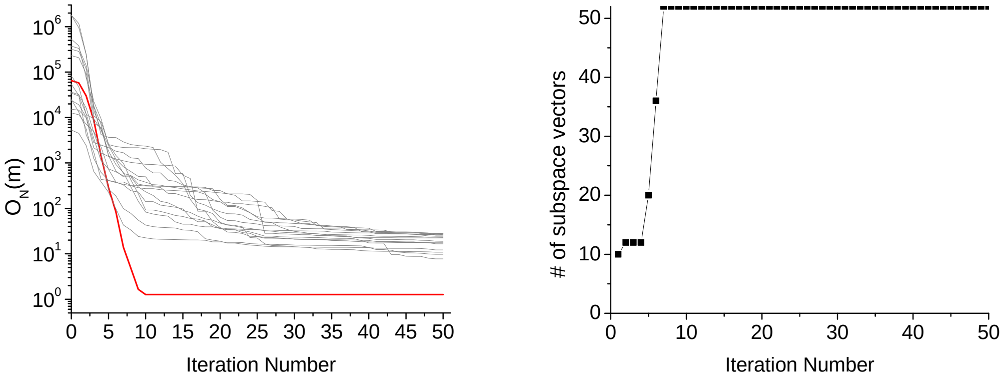
Figs. 3.28(b) and 3.28(c) show the number of subspace vectors and number of SVD parameters versus iterations of the two algorithms, respectively. At each iteration of the SVD parameterization algorithm, the number of SVD parameters is determined based on sv-cut, while in the subspace method, the number of subspace vectors at each iteration is directly specified as input.



(a) $O_N(m)$ (b) # of subspace vectors (c) # of SVD parameters

Figure 3.28: Number of subspace vectors and SVD parameters versus iterations of the subspace method and SVD parameterization algorithm, and $O_N(m)$ versus iterations of the two algorithms, Example 2.

Now, we use the subspace method to perform RML. In performing RML, the subspace vectors and the Hessian matrix are generated based on a particular realization, while for each realization its actual gradient is computed with adjoint method. The information (primary variables of simulation) needed to compute the gradient from the adjoint solution are saved during the previous simulation run on disk using unformatted direct access I/O which uses the binary format to store data record by record.



(a) $O_N(m)$ of 16 realization

(b) Number of subspace vectors

Figure 3.29: (a) $O_N(m)$ of 16 realizations versus iterations of the subspace method. Red curve is for the MAP estimate while the gray curves are for 15 RML realizations, (b) The number of subspace vectors versus iterations of the subspace method, Example 2.

Fig. 3.29(a) shows the values of normalized objective function of all realizations versus iterations. Although the MAP estimate converged in a few iterations, the convergence of RML realizations is much slower than the results of SVD-EnRML shown in Fig. 3.14. Number of subspace vectors versus iterations is shown in Fig. 3.29(b). We used a reasonable number of subspace vectors for this example.

3.5 Example 3

This example pertains to a two-dimensional horizontal reservoir model with a 29×35 uniform grid. The true porosity and log permeability fields, which are shown in Fig. 3.30, are generated from a spherical covariance matrix. The prior parameters are listed in Table 3.9.

The objective is to simulate porosity and log permeability fields with the SVD-EnRML-MI algorithm and compare the results with SVD-EnRML-SMM in terms of computational cost and final values of objective functions. We also compare the results with EnKF in terms of the quality of predictions and data matches.

There are 7 producers and 2 injectors in this reservoir. The summary of well controls are presented in Table 3.10. The total history matching period is 2250 days and is followed by 2750 days of predictions. At 2250 days, 4 of the producers have experienced water breakthrough. These wells are Prod-1, Prod-2, Prod-3 and Prod-7. Observed data include p_{wf} of injectors, oil rates of all producers and water rates of the 4 producers which had water breakthrough, at 30 day intervals.

Observed data are generated by adding synthetic measurement errors to the true data. The standard deviation of the measurement error is 2% of rates for rate data and 3 psi for pressure data, i.e., $\sigma_{\mathrm{BHP}} = 3$ psi and $\sigma_q = 0.02q$. The minimum measurement error for rates is specified to 0.5 STB/D while the maximum measurement error is 3 STB/D.

The number of observed data and model parameters are

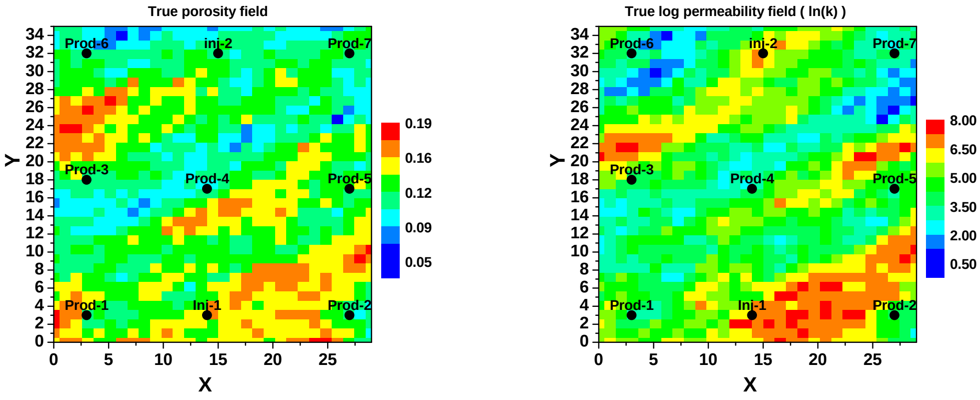
$$N_m = 2N_x \times N_y = 2030, \; N_{\mathrm{d}} = 975.$$

The gridblock dimensions are

$$\triangle x = \triangle y = 250\,\mathrm{ft}, \; \triangle z = 10\,\mathrm{ft}.$$

Table 3.9: Geostatistical parameters of Example 3.

Parameters	Values
ϕ_{mean}	0.12
$[\ln(k)]_{\mathrm{mean}}$	4.50
σ_ϕ	0.0265
$\sigma_{\ln(k)}$	2.0
$\rho_{\phi,\mathrm{ln(k)}}$	0.80
α	30^o
r_1	$18\triangle x$
r_2	$8\triangle x$



(a) True porosity

(b) True ln(k)

Figure 3.30: True porosity and log permeability fields, Example 3.

Table 3.10: Well controls of Example 3. p_{wf} is specified at producers and injection rate is specified at injectors.

Time Period, days	injection rates, STB/D	p_{wf} of producers, psi
$0 - 300$	200	4500
$300 - 600$	800	4200
$600 - 900$	1200	4000
$900 - 1200$	1200	3800
$1200 - 1500$	1200	3500
$1500 - 1800$	1200	3200
$1800 - 2100$	1200	3000
$2100 - 5000$	1200	2500

3.5.1 Results of SVD-EnRML-SMM and SVD-EnRML-MI

In this part, the results of SVD-EnRML-SMM and SVD-EnRML-MI for generating the MAP estimate and $N_e = 15$ RML realizations are compared in terms of the computational cost and final values of objective functions. In both algorithms, the same prior realizations of porosity and log permeability fields are used. We applied both of the algorithms with $N_{\text{SVD,max}} = 55$.

The first two rows of Table 3.11 show the computational costs of the two algorithms. Comparing the first and second rows, one can see that the computational cost in terms of equivalent simulation runs for SVD-EnRML-MI is far less than SVD-EnRML-SMM. Further, at convergence of the algorithms, the maximum value of the normalized objective function of realizations for SVD-EnRML-MI, is less than that of SVD-EnRML-SMM.

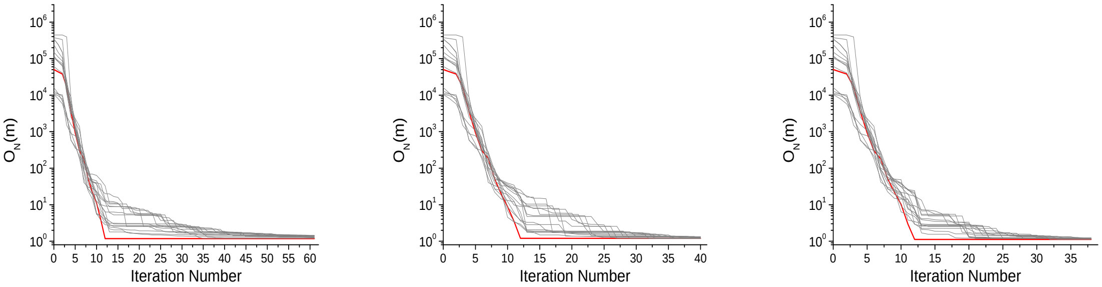
With $N_{\text{SVD,max}} = 55$, the SVD-EnRML-MI algorithm converged in 40 iterations. The MAP estimate converged in 12 iterations. At convergence of SVD-EnRML-MI, $O_N(m)$ of 8 realizations (out of 16) are less than $1+5\sqrt{2/N_d} = 1.23$, while the maximum $O_N(m)$ is 1.33 which is fairly small. At convergence of the SVD-EnRML-SMM, $O_N(m)$ is less than $1+5\sqrt{2/N_d}$ for only 2 of the realizations. Figs. 3.31(a) and 3.31(b) show the values of O_N of all realizations versus iterations l (where SVD parameters are updated with the Lanczos method).

Fig. 3.32(b) shows some important details about the convergence properties of SVD-EnRML-MI algorithm. According to this figure, $O_N(m)$ of all 16 realizations are less than 2 after 28 iterations with a computational cost of 1770 equivalent simulation runs; or at iteration 17 with 840 equivalent simulation runs, $O_N(m)$ of all 16 realizations are less than 10.

Table 3.11: Summary of the computational costs of SVD-EnRML-SMM and SVD-EnRML-MI algorithms for generating 16 conditional realizations of porosity and log permeability, and the max$\{O_N\}$ at convergence, Example 3.

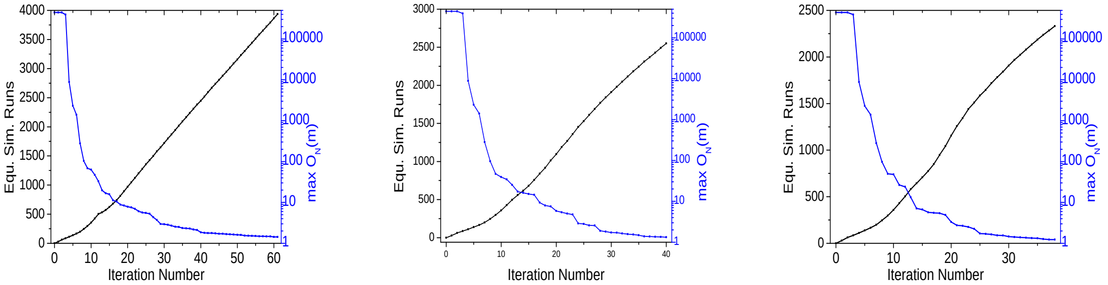
Method	$N_{\text{SVD,max}}$	Sim.	Direct	Adjoint	Equ. Sim. Runs	N_{iter}	max$\{O_N\}$
SMM	55	2070	3763	3702	3936	61	1.433
MI	55	1431	2258	2221	2550	40	1.331
MI	65	1159	2363	2325	2331	38	1.23

In order to investigate the effect of $N_{\text{SVD,max}}$ on the computational efficiency of SVD-EnRML-MI, we applied the algorithm with $N_{\text{SVD,max}} = 65$. Fig. 3.31(c) shows the values of O_N of all realizations versus iterations l for this case. The 3rd row of Table 3.11, shows the computational cost of SVD-EnRML-MI with $N_{\text{SVD,max}} = 65$. Comparing the 2nd and the 3rd rows of Table 3.11, one can see that increasing $N_{\text{SVD,max}}$ from 55 to 65, reduces the computational costs. With $N_{\text{SVD,max}} = 65$, in addition to the lower



(a) SMM, $N_{\text{SVD,max}} = 55$ (b) MI, $N_{\text{SVD,max}} = 55$ (c) MI, $N_{\text{SVD,max}} = 65$

Figure 3.31: $O_N(m)$ of all realizations versus iterations. Red curve is for the MAP estimate, gray curves are from 15 RML realizations from (a) SVD-EnRML-SMM, (b) SVD-EnRML-SMM, (c) SVD-EnRML-MI, Example 3.



(a) SMM, $N_{\text{SVD,max}} = 55$ (b) MI, $N_{\text{SVD,max}} = 55$ (c) MI, $N_{\text{SVD,max}} = 65$

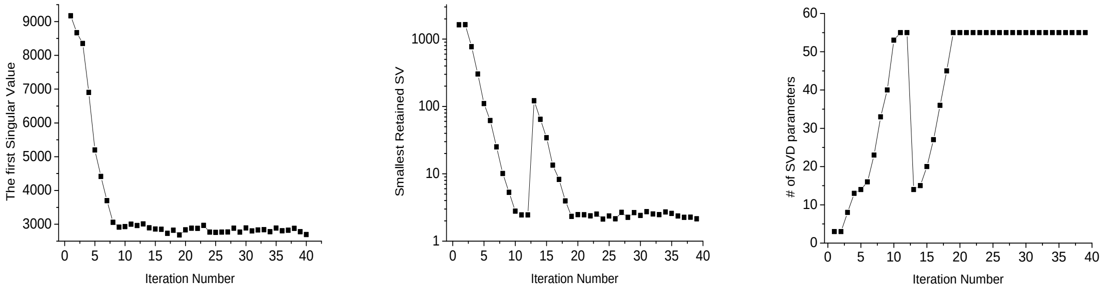
Figure 3.32: Computational cost in terms of equivalent simulation runs and max $\{O_N(m)\}$ versus iterations, (a) SVD-EnRML-SMM, (b) SVD-EnRML-MI, (c) SVD-EnRML-MI, Example 3.

computational cost, $O_N(m)$ for all 16 realizations is less than $1 + 5\sqrt{2/N_d} = 1.23$ at convergence.

With $N_{\text{SVD,max}} = 65$, the value of λ_{65}/λ_1 varied between 0.0005 to 0.0006, while with $N_{\text{SVD,max}} = 55$, the value of λ_{55}/λ_1 varied from 0.00077 to 0.001.

Fig. 3.32 shows the computational cost and the maximum $O_N(m)$ versus iterations of the three cases we discussed. Since multiple iterations are performed after the convergence of the MAP estimate, all 3 cases in Fig. 3.32 are similar at early iterations. However, after the convergence of the MAP estimate, the $\max(O_N)$ decreases faster with the SVD-EnRML-MI method.

Figs. 3.33(a) and 3.34(a) show the first singular value of G_D versus iterations of SVD-EnRML-MI with $N_{\text{SVD,max}}$ of 55 and 65, respectively. Similar to the results of other examples, after the first few iterations, the largest singular value does not change significantly. Figs. 3.33(b) and 3.34(b) show the smallest retained singular value of $G_{D,l}$ versus iterations. According to the results in Figs. 3.33(b) and 3.33(c), the 55th retained singular value is around 2.45 for the MAP estimate, and after the convergence of the MAP estimate it changes from a maximum of 2.67 to a minimum of 2.1. According to Fig. 3.34(b), when we increased $N_{\text{SVD,max}}$ from 55 to 65, the 65th singular value varied from 1.45 to 1.75.



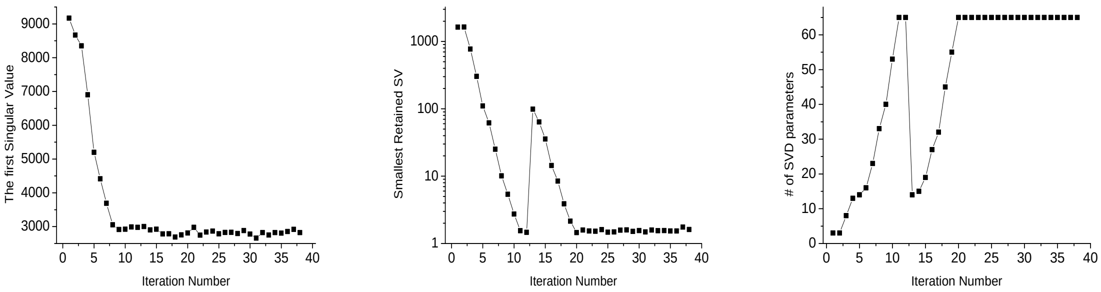
(a) First SV

(b) Smallest retained SV

(c) # of retained SV

Figure 3.33: The first singular value and the smallest retained singular value of $G_{D,l}$ and the number of retained SVD parameters versus iterations of SVD-EnRML-MI algorithm with $N_{\text{SVD,max}} = 55$, Example 3.

Figs. 3.33(c) and 3.34(c) display the number of computed singular triplets versus



(a) First SV

(b) Smallest retained SV

(c) # of retained SVs

Figure 3.34: The largest singular value and the smallest retained singular value of G_D and the number of SVD parameters versus iterations of SVD-EnRML-MI algorithm with $N_{\text{SVD,max}} = 65$, Example 3.

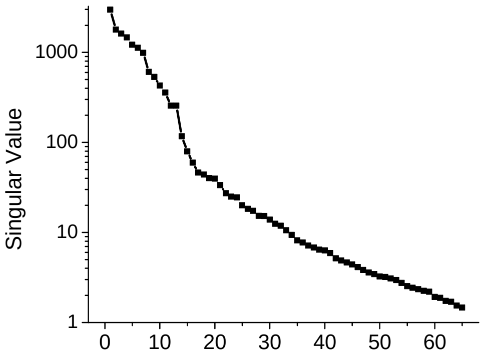
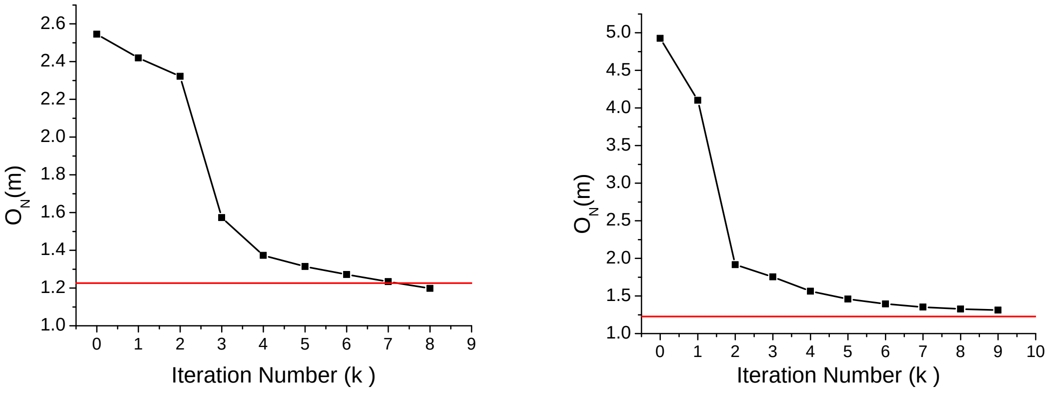


Figure 3.35: Singular values of G_D corresponding to the MAP estimate at iteration $l = 12$, Example 3.

iterations. The jumps in these figures happen at the convergence of the MAP estimate, where sv-cut is set to $\mu_2 = 0.05$ and the algorithm computes a truncated SVD of $G_{D,l}$ for the realization with the maximum objective function. Fig. 3.35 shows 65 singular values of $G_{D,l}$ corresponding to the MAP estimate, at iteration $l = 12$. According to this figure, about half of these 65 singular values are less than 10. In addition, the distribution does not show a sharp jump. The results in Table 3.11 show that retaining more singular values which are between 1 and 10, decreased the computational costs of SVD-EnRML-MI.



(a) Realization #6, $l = 23$

(b) Realization #9, $l = 20$

Figure 3.36: O_N versus outer and inner loop iterations, for two cases. Red line shows the value of $1 + 5\sqrt{2/N_d} = 1.23$. (a) O_N of realization #6 at iteration $l = 23$, (b) O_N of realization #9 at iteration $l = 20$, Example 3.

Fig. 3.36 shows how multiple iterations with the same SVD parameters decreases the objective function. These results are corresponding to the case with $N_{\text{SVD,max}} = 65$. In Fig. 3.36(a), the value of O_N at $k = 0$ is the value of the normalized objection function of realization #6 at the beginning of iteration $l = 23$. A truncated SVD of $G_{D,l}$ is computed for $j_{\text{base}} = 15$ at iteration $l = 23$. Based on the results (not shown here), the search direction with $\gamma = 1.0$ did not decrease the objective function. With $\gamma = 100$, the normalized objective function, O_6/N_d, decreased from 2.54 to 2.42. As the decrease of O_6 at the first application of the LM algorithm was more than 1%, the realization entered a loop where the same SVD parameters and the same $\gamma = 100$ are used in the search direction. As one can see in Fig. 3.36(a), with multiple iterations at iteration $l = 23$ of the algorithm, O_6/N_d decreased from 2.42 to 1.198 which is less than $1 + 5\sqrt{2/N_d}$.

In Fig. 3.36(b), the value of O_N at $k = 0$ is the value of O_N of realization #9 at the beginning of iteration $l = 20$. Based on the results, the search direction with $\gamma = 0.01$ did not decrease the objective function. With $\gamma = 100$, the normalized objective function, O_9/N_d, decreased from 4.926 to 4.102. Again, as the decrease in O_9 at $k = 1$ was more than 1%, the realization entered a loop where the same SVD parameters and the same $\gamma = 100$ are used in the search direction. Note that in the inner loop, O_9/N_d decreased from 4.102 to 1.312. The inner loop ended at $k = 9$, as the maximum number of iterations with the same SVD parameters was specified to 9.

In summary, the results of this example show that SVD-EnRML-MI is computationally more efficient that SVD-EnRML-SMM. The results also show that using more singular triplets at late iterations, can actually improve the computational efficiency of SVD-EnRML-MI. Many of the retained singular values were smaller than 10, but all of them were greater than 1.

3.5.2 Comparison of The Results of EnKF With The Results of SVD-EnRML-MI

In this subsection, the results of EnKF and SVD-EnRML-MI are compared. For SVD-EnRML-MI, we used the results corresponding to the second row of Table 3.11.

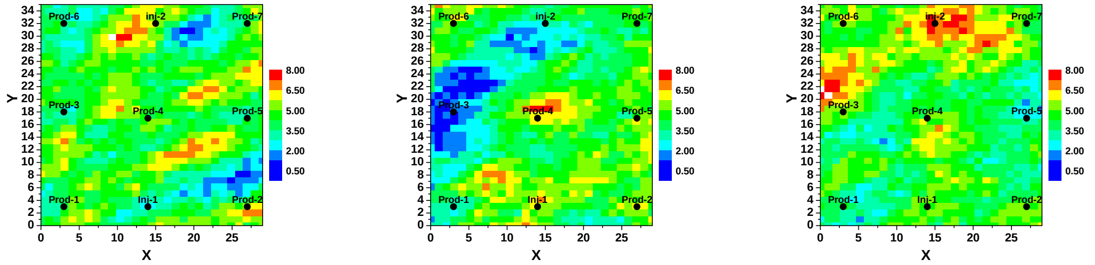
With the ensemble Kalman filter (EnKF), observed data are sequentially assimilated. The covariance localization is performed using anisotropic Gaspari and Cohn correlation function [15]. The anisotropic Gaspari and Cohn correlation function is generated with a principal correlation length of $20\triangle x$ aligned with the principal direction of the prior covariance matrix and a critical length of $15\triangle x$ in the perpendicular direction. Note that we used a fixed correlation lengths in generating the correlation matrices. Although not shown, the results of EnKF with localization is much better than EnKF without localization in terms of data matches and predictions.

In our implementation of EnKF, unrealistic values of porosity, log permeability and saturations after each assimilation step are simply truncated. The minimum bounds for $\ln(k)$ and ϕ were specified to -2 and 0.01 while the maximum bounds were 10 and 0.32, respectively. Fig. 3.37 shows 3 prior realizations of the log permeability field, and Fig. 3.38 shows 3 updated realizations of log permeability field with the EnKF and SVD-EnRML-MI.

Fig. 3.39 shows 3 prior realizations of porosity field, and Fig. 3.40 shows 3 updated realizations of porosity fields of EnKF and SVD-EnRML-MI. Note that the variation of porosity in the updated models of EnKF is higher than the true model. Some gridblocks have porosity values around 0.25.

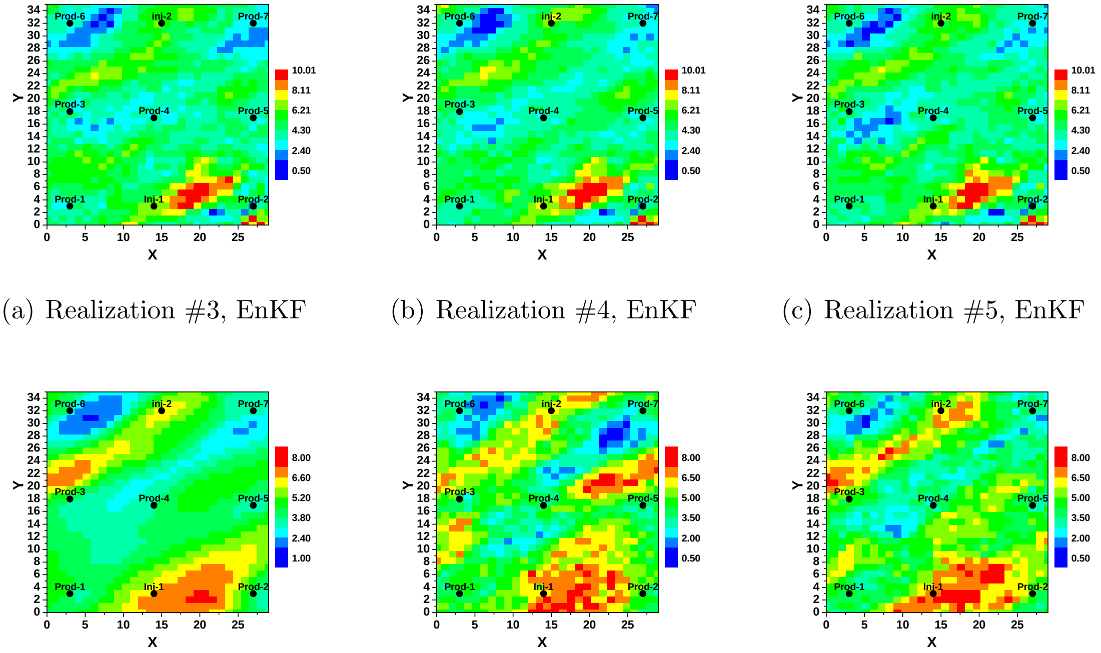
Figs. 3.41, 3.42 and 3.43 show the data matches and predictions with SVD-EnRML and with EnKF during assimilations and rerun. By comparing the results in Figs. 3.42 and 3.43 one can see that during rerun, the predicted data do not match the observed data. This behavior in EnKF is known as inconsistency between data matches during assimilation and rerun. During assimilating observed data, pressures and saturations of gridblocks also get updated. A strong assumption in EnKF is that the updated pressures and saturations (state variables) are consistent with the updated model parameters, which means if the updated model parameters were used to run the simulator from time zero, the pressures and saturations at time t should be statistically consistent with the updated pressures and saturations obtained from the EnKF equation [40]. This issue is not further discussed here, as EnKF is not the main objective of this work.

According to the results of this part, the updated rock property fields from the SVD-EnRML-MI are closer to the truth than those from EnKF. In addition, the data matches and predictions from SVD-EnRML-MI are more reliable.



(a) Realization #3 (b) Realization #4 (c) Realization #5

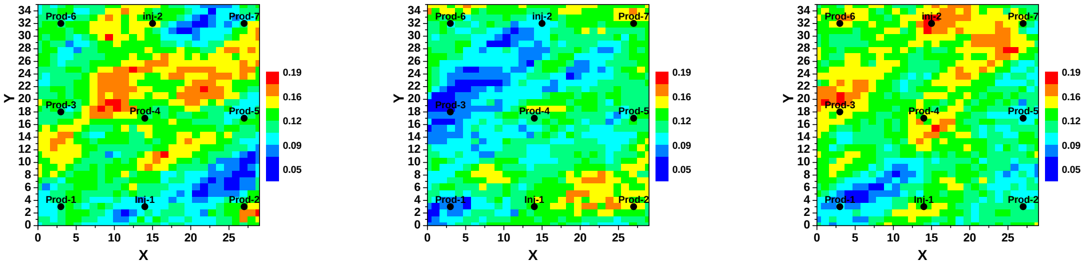
Figure 3.37: 3 prior realizations of log permeability field, Example 3.



(a) Realization #3, EnKF (b) Realization #4, EnKF (c) Realization #5, EnKF

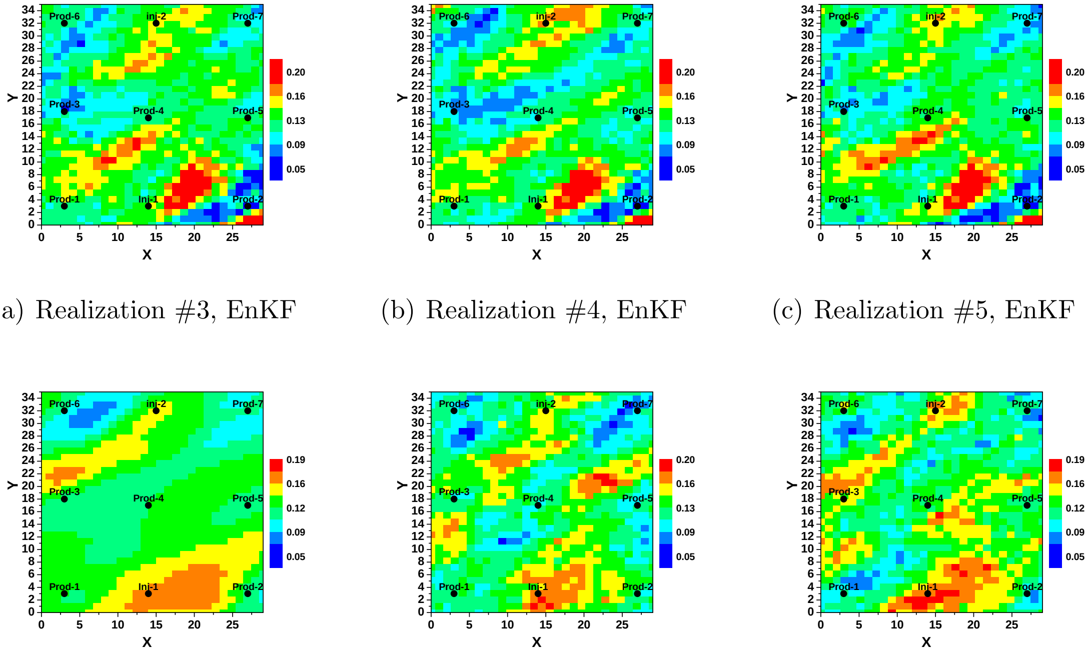
(d) MAP estimate, SVD (e) Realization #3, SVD (f) Realization #5, SVD

Figure 3.38: 3 updated realizations of log permeability field, Top row shows the results from EnKF with localization, the bottom row shows the results from SVD-EnRML-MI, Example 3.



(a) Realization #3 (b) Realization #4 (c) Realization #5

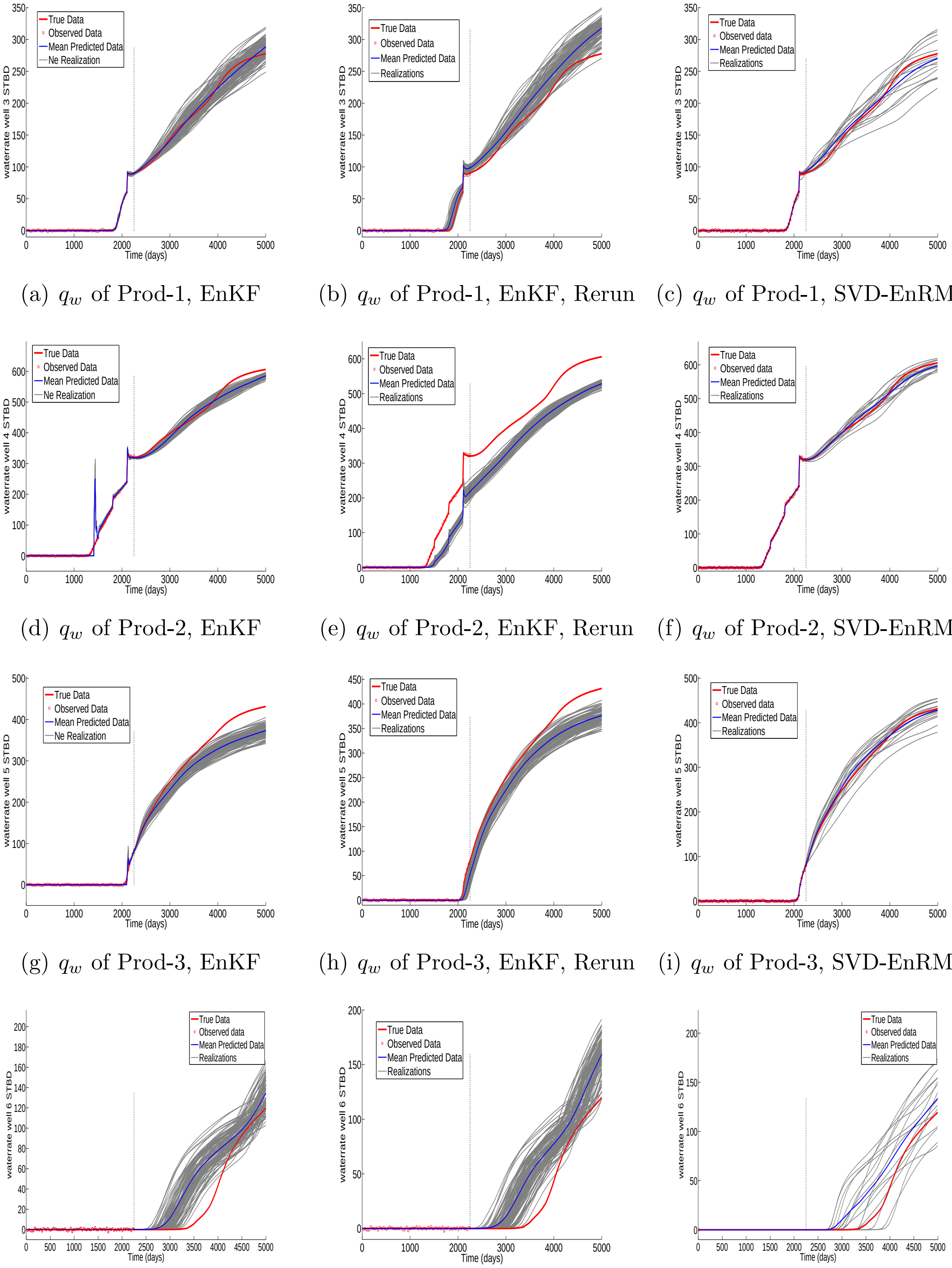
Figure 3.39: 3 prior realizations of porosity field, Example 3.



(a) Realization #3, EnKF (b) Realization #4, EnKF (c) Realization #5, EnKF

(d) MAP estimate, SVD (e) Realization #3, SVD (f) Realization #5, SVD

Figure 3.40: 3 updated realizations of porosity field. Top row shows the results from EnKF with localization, the bottom row shows the results from SVD-EnRML-MI, Example 3.



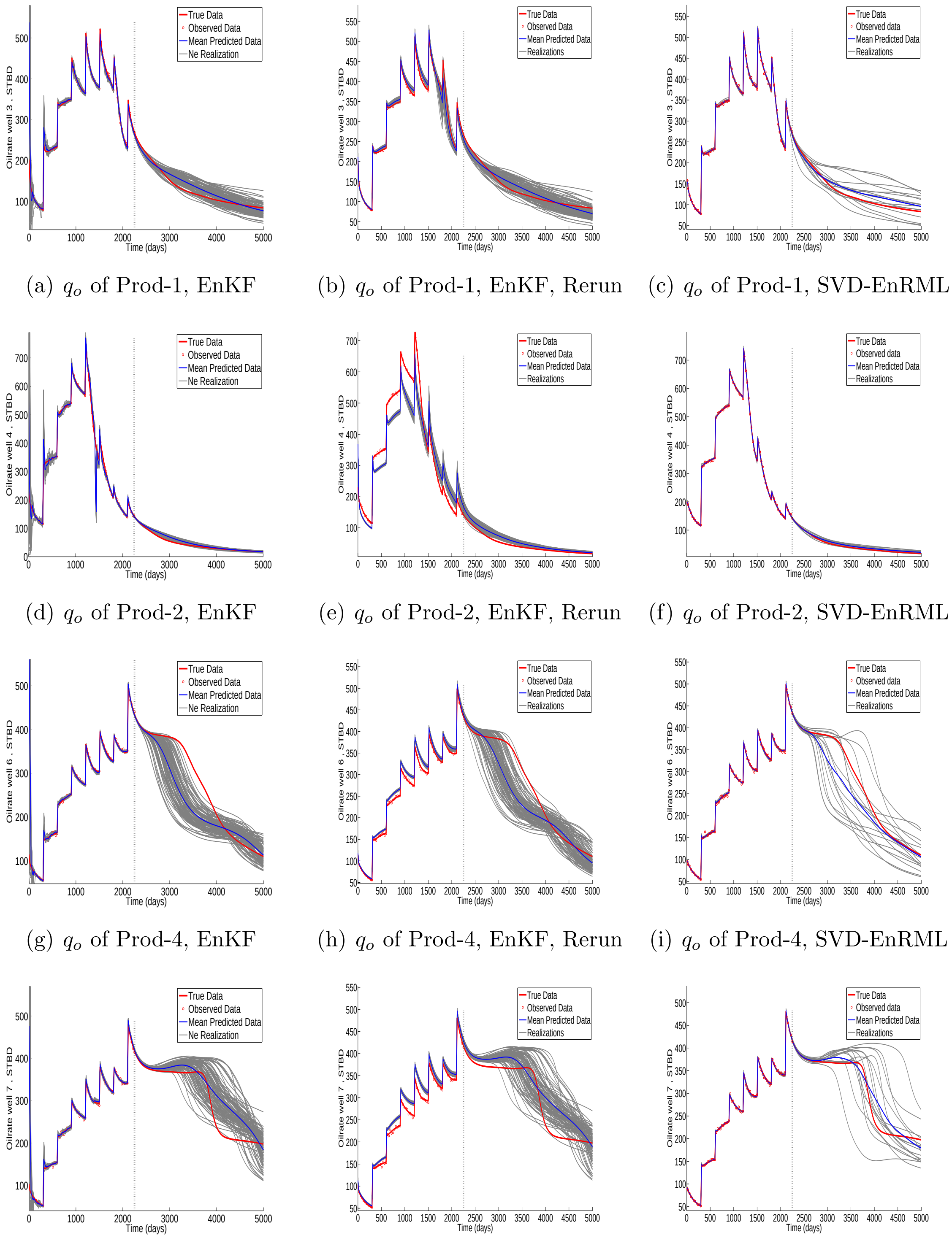
(a) q_w of Prod-1, EnKF (b) q_w of Prod-1, EnKF, Rerun (c) q_w of Prod-1, SVD-EnRML

(d) q_w of Prod-2, EnKF (e) q_w of Prod-2, EnKF, Rerun (f) q_w of Prod-2, SVD-EnRML

(g) q_w of Prod-3, EnKF (h) q_w of Prod-3, EnKF, Rerun (i) q_w of Prod-3, SVD-EnRML

(j) q_w of Prod-4, EnKF (k) q_w of Prod-4, EnKF, Rerun (l) q_w of Prod-4, SVD-EnRML

Figure 3.41: Data matches and predictions of q_w of four of the producers from EnKF and SVD-EnRML-MI. The dashed vertical line shows the end of the history matching period (2250 days), Example 3.



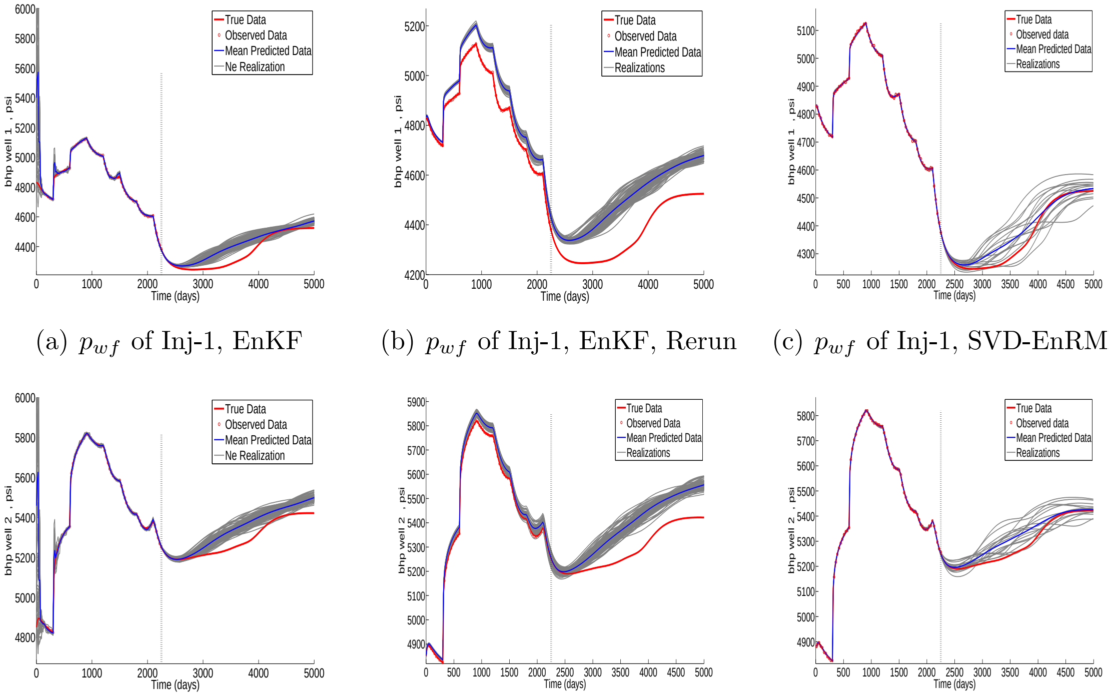
(a) q_o of Prod-1, EnKF (b) q_o of Prod-1, EnKF, Rerun (c) q_o of Prod-1, SVD-EnRML

(d) q_o of Prod-2, EnKF (e) q_o of Prod-2, EnKF, Rerun (f) q_o of Prod-2, SVD-EnRML

(g) q_o of Prod-4, EnKF (h) q_o of Prod-4, EnKF, Rerun (i) q_o of Prod-4, SVD-EnRML

(j) q_o of Prod-5, EnKF (k) q_o of Prod-5, EnKF, Rerun (l) q_o of Prod-5, SVD-EnRML

Figure 3.42: Data matches and predictions of q_o of four of the producers from EnKF and SVD-EnRML-MI. The dashed vertical line shows the end of the history matching period (2250 days), Example 3.



(a) p_{wf} of Inj-1, EnKF

(b) p_{wf} of Inj-1, EnKF, Rerun

(c) p_{wf} of Inj-1, SVD-EnRML

(d) p_{wf} of Inj-2, EnKF

(e) p_{wf} of Inj-2, EnKF, Rerun

(f) p_{wf} of Inj-2, SVD-EnRML

Figure 3.43: Data matches and predictions of p_{wf} of injectors from EnKF and SVD-EnRML-MI. The dashed vertical line shows the end of the history matching period (2250 days), Example 3.

CHAPTER 4

3D EXAMPLES

In the first example of this chapter, the true model is a 3D reservoir where the prior model for the rock property fields is correlated in all three directions. In Examples 2 and 3, the true models are 3D reservoirs with the rock property fields of each layer independent from all other layers, i.e., in the prior model, rock property fields are uncorrelated in the vertical direction.

In tables, ϕ_{mean} and $[\ln(k)]_{\text{mean}}$, denote the prior mean of porosity and log-permeability, respectively. The standard deviations of porosity and log-permeability are denoted by σ_ϕ and $\sigma_{\ln(k)}$, respectively; $\rho_{\phi,\ln(k)}$ denotes the correlation coefficient between porosity and log-permeability; α is the angle measured counterclockwise from the x-axis to the principal correlation direction of the covariance function; r_1 is the correlation range in the principal direction and r_2 is the correlation range in the orthogonal direction.

All examples are for two phase flow (oil and water).

4.1 Example 1

This example pertains to a three-dimensional reservoir model with $28 \times 30 \times 3$ uniform grid. True porosity, horizontal and vertical log permeability fields are generated from a spherical covariance matrix. The prior parameters are listed in Table 4.1. The cross-covariances are calculated by assuming screening criteria, and the correlation coefficient is the same for all three pairs of rock property fields. True rock property fields are shown in Figs. 4.1(a)- 4.1(c), 4.2(a)- 4.2(c) and 4.3(a)- 4.3(c).

Table 4.1: Geostatistical parameters of Example 1.

Parameters	Values
ϕ_{mean}	0.15
$[\ln(k_h)]_{\text{mean}}$	4.50
$[\ln(k_z)]_{\text{mean}}$	3.50
σ_ϕ	0.04
$\sigma_{\ln(k_h)}$	1.4142
$\sigma_{\ln(k_z)}$	1
ρ	0.80
α	30^o
r_1	$22\triangle x$
r_2	$7\triangle x$
r_z	$3\triangle x$

The gridblock dimensions are:

$$\triangle x = \triangle y = 150\,\text{ft},\ \triangle z = 15\,\text{ft}.$$

There are 9 producers and 4 injectors in this reservoir, as shown in the true horizontal log permeability field of Fig. 4.1(a). Producers are perforated only at the top layer while injection wells are perforated at the bottom layer. Table 4.2 shows the well control schedules. At 2250 days, Prod-1 is shut in due to high water cut. The reservoir is initially at irreducible water saturation which is equal to $s_{\text{wc}} = 0.1$. We apply the SVD-EnRML-MI algorithm for 3 history matching periods, 600 days, 870 days and 1080 days. At 600 days, none of the producers had water breakthrough. At 870 days, only Prod-1, Prod-2, Prod-4 and Prod-9 have experienced water breakthrough. At 1080 days, in addition to the 4 producers, Prod-3 has also experienced water breakthrough. Observed data include water rate data of these producers and p_{wf} of all wells at 30 day intervals.

Observed data are generated by adding Gaussian random noise to the true data. The standard deviation of the noise (measurement error) is 2% of rates for rate data and 3 psi for pressure data, i.e., $\sigma_{\text{BHP}} = 3$ psi and $\sigma_q = 0.02q$. The minimum measurement error for rates is specified to 0.5 STB/D while the maximum measurement error is 3

STB/D.

Table 4.2: Well controls of Example 1. Total liquid rate is specified at producers and injection rate is specified at injectors. Rates are in STB/D and time intervals are in days.

Time	inj-1,2	Prod-1	Prod-2, 4, 9	Prod-3	Prod-5	Prod-6, 7, 8
0 − 2250	1250	700	700	600	600	300
2250 − 3000	1060	0	700	600	600	300

4.1.1 Comparison of SVD Parameterization and The Subspace Method

In this part, we use the subspace method and the SVD parameterization algorithm to generate the MAP estimate for the porosity, $\ln(k_h)$ and $\ln(k_z)$ fields. Thus, the model parameters include porosity and horizontal and vertical log permeability of all gridblocks. Fluid properties and the prior mean and covariance are assumed to be known. The number of model parameters and observed data are:

$$N_m = 3N_x \times N_y \times N_z = 7560, \; N_{\mathrm{d}} = 493.$$

The vector of model parameters can be shown as

$$m = \begin{pmatrix} \ln(k_h) \\ \ln(k_z) \\ \phi \end{pmatrix}, \tag{4.1}$$

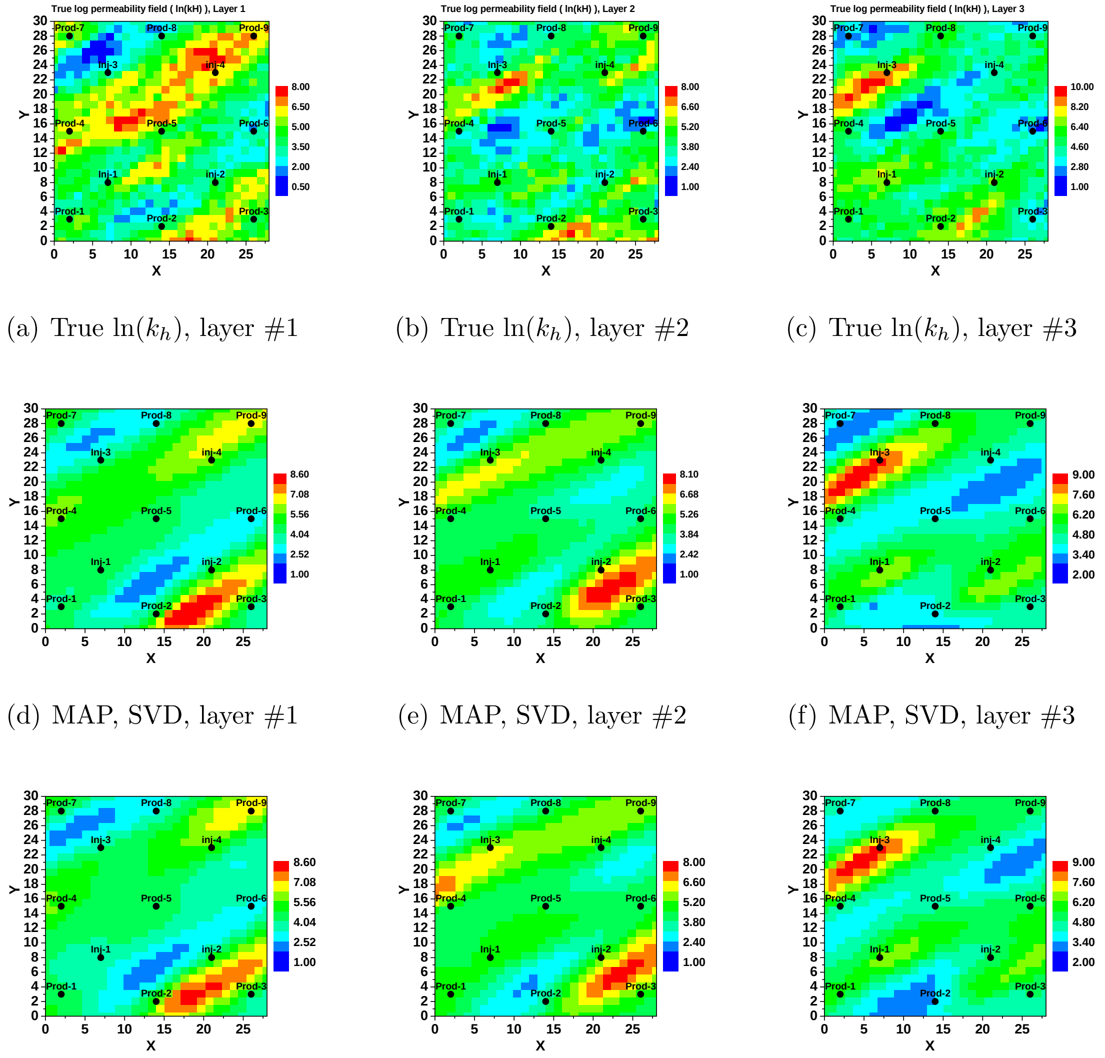
where $\ln(k_h)$ denotes the vector of horizontal log permeability of all gridblocks, $\ln(k_z)$ denotes the vector of vertical log permeability of all gridblocks and ϕ denotes the vector of porosity of all gridblocks.

We used the lower triangular part, L, of the Cholesky decomposition ($C_M = LL^T$) of the covariance matrix as $C_M^{1/2}$. In the subspace method, the modified LM algorithm is used where we gradually increase the number of subspace vectors. In both algorithms a growth and decay factor of 10 is used for the LM parameter. In SVD parameterization algorithm, parameters for sv-cut, are set to the values mentioned in Section 2.1.

At the first 3 iterations of the subspace method, each data type from each well comprises one subspace vector. In order to increase the number of subspace vectors, we further partitioned the data into time periods, e.g., at iteration 3, p_{wf} data of Prod-2 is partitioned into 2 subobjective functions, one containing data mismatch terms from time 0 until 420 days and the other one containing data mismatch terms from 450 days until 870 days. Water rate data of each of the 4 producers are always used in one subobjective function. This is because there is no water breakthrough before 650 days. From iteration 2 on, the model mismatch term is used in 2 additional subspace vectors, one for porosity and one for log permeabilities.

Both the subspace method and SVD parameterization algorithm converged in 22 iterations. At convergence of both algorithms, the final value of $O_N(m)$ is less than $1 + 5\sqrt{2/N_d} = 1.319$. Figs. 4.1, 4.2 and 4.3 show the rock property fields for the truth and the MAP estimates obtained with the two methods. Note that the rock property fields of the MAP estimates obtained with the two methods are very similar. The rock property fields of the MAP estimate are smooth and some of the main features of the truth are recognizable. However, there are distinct differences between the true fields and the estimates. For example, there is a low vertical permeability and low porosity region in the 2nd layer of the MAP estimate and near to the location of Prod-2 (see Figs. 4.2(e) and 4.3(e)) which does not exist in the truth. This is not unexpected, as we never have enough information contained in production data to resolve all parameters. After matching data, the reservoir description will still be highly uncertain. All we can expect in the Bayesian history matching framework, is that matching production data will reduce the uncertainty in the reservoir description and reservoir performance predictions, while maintaining the essential geological heterogeneity of the prior geological model.

Fig. 4.4(a) shows the normalized objective function of the MAP estimates versus iterations of the two algorithms, and Fig. 4.4(b) shows the model mismatch term ver-



(a) True $\ln(k_h)$, layer #1

(b) True $\ln(k_h)$, layer #2

(c) True $\ln(k_h)$, layer #3

(d) MAP, SVD, layer #1

(e) MAP, SVD, layer #2

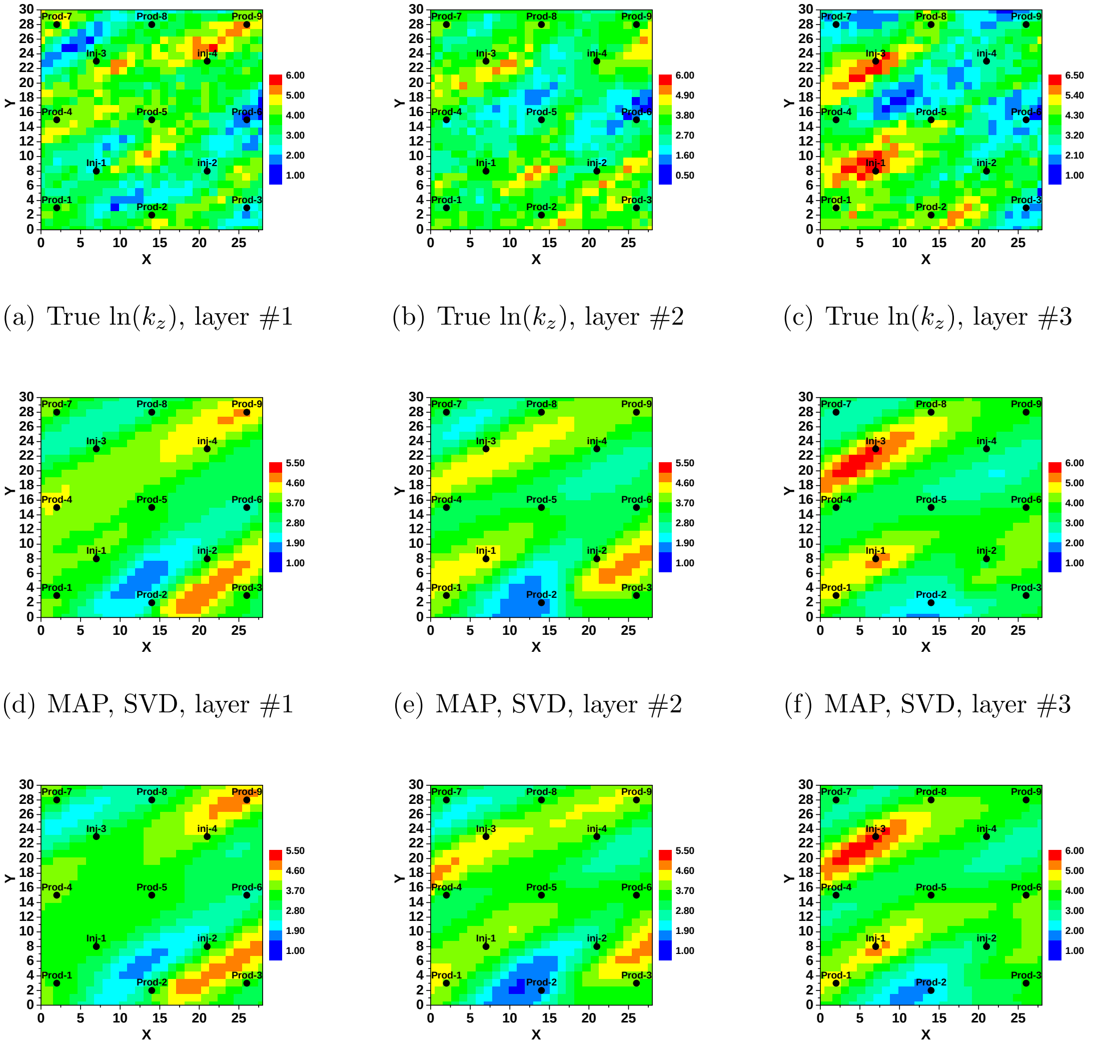
(f) MAP, SVD, layer #3

(g) MAP, subspace, layer #1

(h) MAP, subspace, layer #2

(i) MAP, subspace, layer #3

Figure 4.1: Horizontal log permeability fields. Top row shows the true $\ln(k_h)$ fields, the middle row is the MAP estimate from SVD parameterization, and the bottom row is the MAP estimate from the subspace method. History matching period is 870 days, Example 1.



(a) True $\ln(k_z)$, layer #1

(b) True $\ln(k_z)$, layer #2

(c) True $\ln(k_z)$, layer #3

(d) MAP, SVD, layer #1

(e) MAP, SVD, layer #2

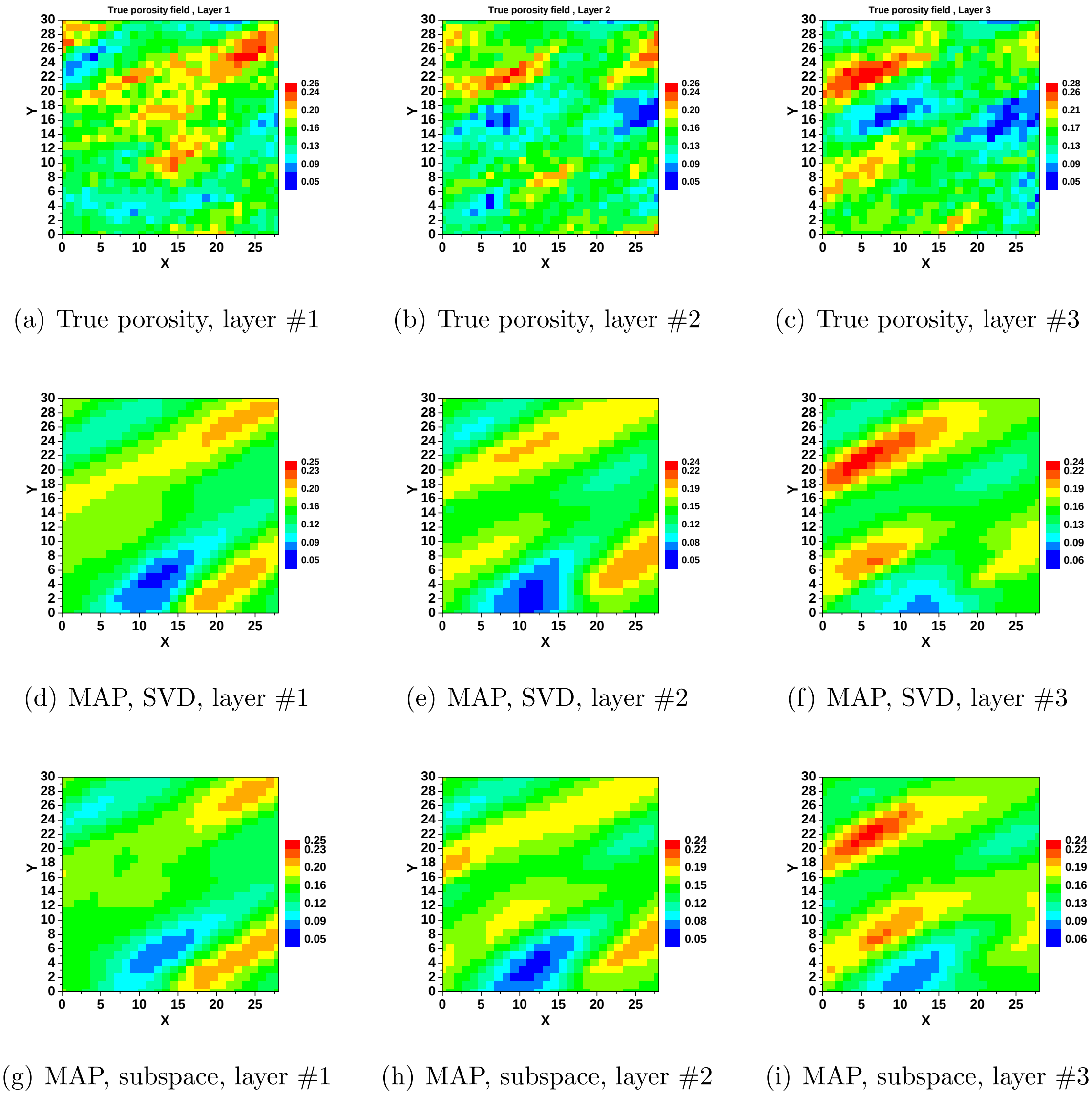
(f) MAP, SVD, layer #3

(g) MAP, subspace, layer #1

(h) MAP, subspace, layer #2

(i) MAP, subspace, layer #3

Figure 4.2: Vertical log permeability fields. Top row shows the true $\ln(k_z)$ fields, the middle row is the MAP estimate from SVD parameterization, and the bottom row is the MAP estimate from the subspace method. History matching period is 870 days, Example 1.



(a) True porosity, layer #1

(b) True porosity, layer #2

(c) True porosity, layer #3

(d) MAP, SVD, layer #1

(e) MAP, SVD, layer #2

(f) MAP, SVD, layer #3

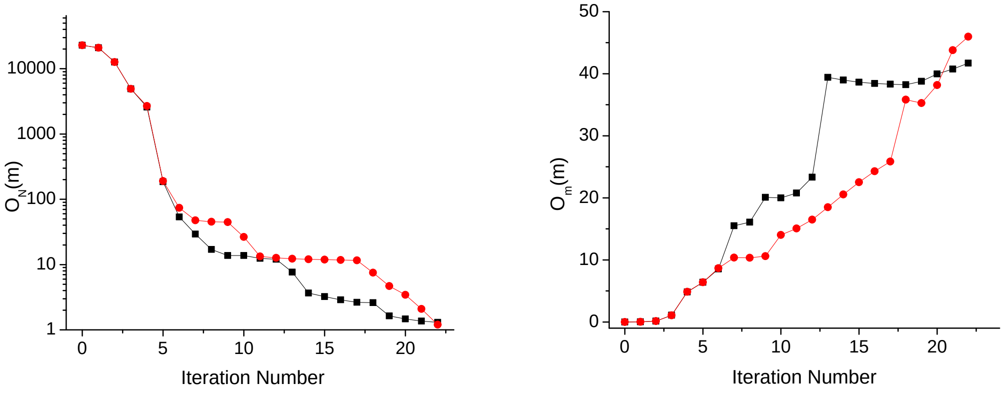
(g) MAP, subspace, layer #1

(h) MAP, subspace, layer #2

(i) MAP, subspace, layer #3

Figure 4.3: Porosity fields. Top row shows the true porosity fields, the middle row is the MAP estimate from SVD parameterization, and the bottom row is the MAP estimate from the subspace method. History matching period is 870 days, Example 1.

sus iterations for the two algorithms. Both algorithms converged in the same number of iterations and the values of model mismatch terms at convergence are very close. Fig. 4.5(a) shows the number of subspace vectors versus iterations, while Fig. 4.5(b) shows the number of retained singular triplets versus iterations of the SVD parameterization algorithm. The number of subspace vectors and the maximum number of SVD parameters were specified as input. Although the number of subspace vectors at each iteration of the subspace method is not exactly equal to the corresponding number of the SVD parameters in SVD parameterization, the two algorithms have similar convergence properties.



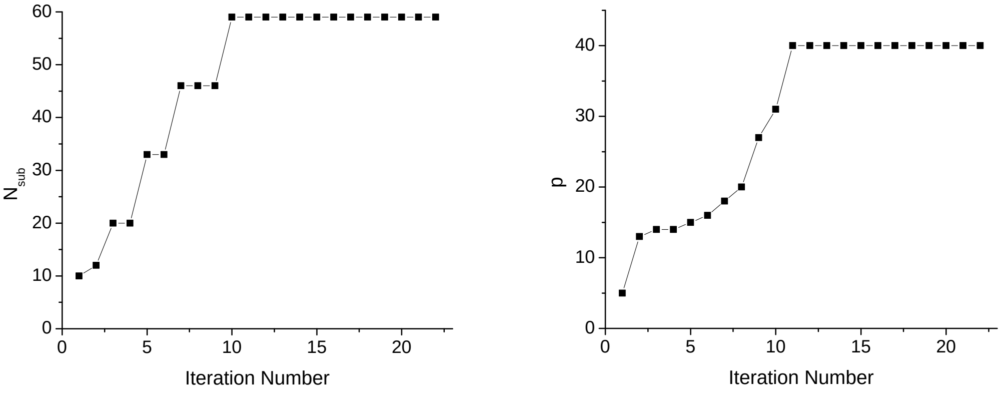
(a) $O_N(m)$

(b) $O_m(m)$

Figure 4.4: $O_N(m)$ and $O_m(m)$ of the MAP estimate with the subspace (black) and SVD parameterization (red) algorithms, Example 1.

Table 4.3 shows the summary of the computational costs of the two algorithms. Note that in calculating equivalent simulation runs, 4 adjoint solutions or 4 applications of the gradient simulator method were roughly assumed to be equivalent with 1 reservoir simulation run. As we show in Subsection 4.1.4, by extending the history matching period to 1080 days and having more water rate data from Prod-2, the MAP estimate converges in fewer iterations.

According to the results of this example, the subspace method and SVD parameterization algorithm have similar convergence properties. However, in performing SVD parameterization, one does not need to consider subobjective functions, which makes the



(a) Number of subspace vectors

(b) Number of retained SVD parameters

Figure 4.5: (a) The number of the subspace vectors in the subpace method, (b) The number of retained SVD parameters in the SVD parameterization algorithm, Example 1.

Table 4.3: Computational cost of the LM algorithm with the subspace and SVD parametrization methods for generating the porosity and horizontal and vertical log permeability fields of the MAP estimate, Example 1.

Algorithm	Simulations	Direct	Adjoint	Equ. Sim. Runs	N_{iter}
Subspace method	36	1032	991	541	22
SVD parameterization	33	874	852	464	22

algorithm a more convenient and general method and easier to implement.

4.1.2 Conditioning Realizations to Pressure Data with SVD-EnRML-MI

In this part, we condition $N_e = 50$ realizations of porosity, horizontal and vertical log permeability fields to only pressure data. In applying the SVD-EnRML-MI algorithm, we used the lower triangular part of the Cholesky decomposition of the covariance matrix as $C_M^{1/2}$. The history matching period is 600 days. At 600 days, none of the producers have experienced water breakthrough; thus the observed data only include pressure data of all producers and injectors. Since the observed data came from the same set which we used for history matching in the previous section, the zero water rates of 4 of the producers (after adding Gaussian random noise) are also included in the observed data vector (80 water rate data). Thus the number of model parameters and observed data are:

$$N_m = 3N_x \times N_y \times N_z = 7560,\ N_{\text{d}} = 340.$$

At convergence of the algorithm, we expect the normalized objective function of a conditional realization be less than $1 + 5\sqrt{2/N_d} = 1.383$. The SVD-EnRML-MI algorithm converged in 23 iterations. At convergence, $O_N(m)$ is less than 1.383 for all $N_e = 50$ realizations. Table 4.4 shows the computational cost of the SVD-EnRML-MI algorithm for conditioning 50 realizations of rock property fields to pressure data. The computational cost is very low, compared to when the true data contain nonzero water rates. Specifically, generating the MAP estimate by itself, when we included nonzero water rate data, required the equivalent of 464 reservoir simulation runs (Table 4.3), but in the current case, generating 50 realizations required only the equivalent of 1392 reservoir simulation runs (Table 4.4). Fig. 4.6 shows $O_N(m)$ values for all realizations versus iterations of the algorithm. Note that the MAP estimate converged in 11 iterations. The convergence rate for all realizations is fast.

Table 4.4: Computational cost of SVD-EnRML-MI method for conditioning 50 realizations to pressure data, Example 1.

Algorithm	Simulations	Direct	Adjoint	Equ. Sim. Runs	N_{iter}
SVD-EnRML-MI	1099	598	575	1392	23

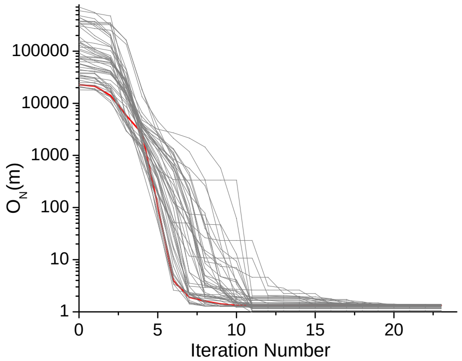
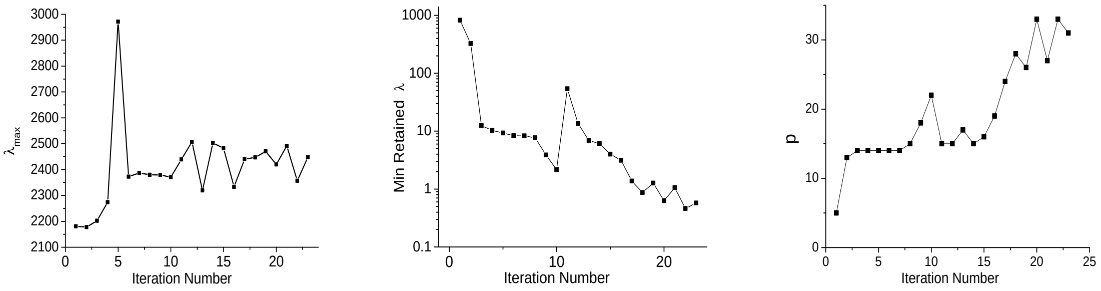


Figure 4.6: $O_N(m)$ of the MAP (red) and 49 RML realizations (gray) versus iterations, l, of SVD-EnRML-MI. The history matching period is 600 days. Example 1.

Fig. 4.7(c) shows the number of retained singular triplets versus iterations of the SVD-EnRML-MI algorithm. Note that only a few singular triplets were calculated, even with a small sv-cut of 0.0003 at a late iteration; this means that the singular values of $G_{D,l}$ decay fast in this case. We also would like to remark that in cases where many of the observed data have similar sensitivity to the model parameters, many singular values of $G_{D,l}$ will be small and only a few singular triplets are required to adequately represent the reservoir model and match data. Fig. 4.7(a) shows the first singular value of $G_{D,l}$ versus iterations and Fig. 4.7(b) shows the smallest retained singular value of $G_{D,l}$ versus iterations. At late iterations of the algorithm, the smallest retained singular value is slightly smaller than 1.



(a) The first λ

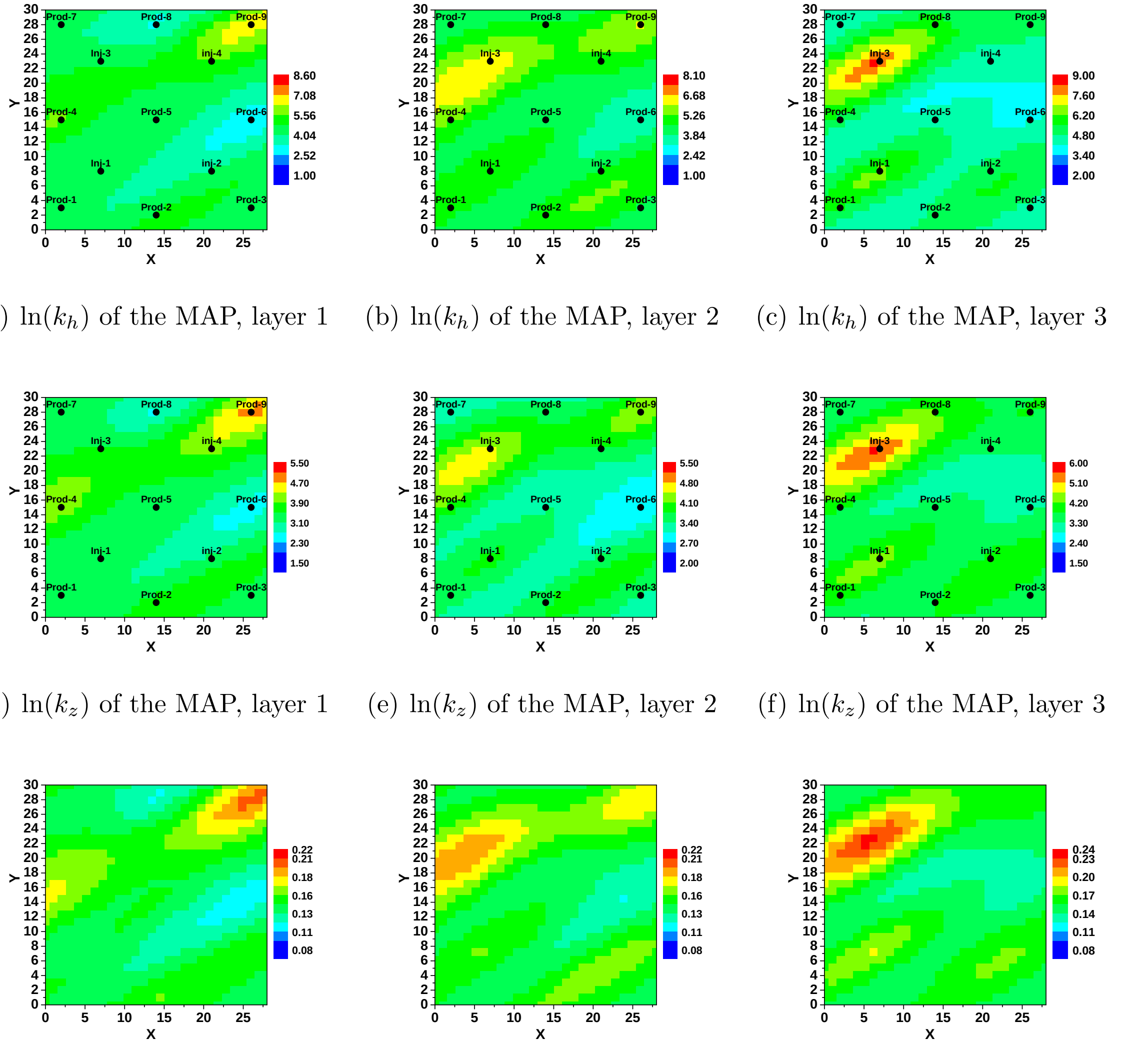
(b) Smallest retained λ

(c) N_{SVD}

Figure 4.7: The largest singular value and the smallest retained singular value of $G_{D,l}$ and the number of retained singular triplets versus iterations of SVD-EnRML-MI; the history matching period is 600 days. Example 1.

Fig. 4.8 shows the rock property fields of the MAP estimate conditioned to 600 days of pressure data. The rock property fields are much smoother than those conditioned to 870 days of pressure and q_w data.

Fig. 4.9 shows the field oil and water production from the posterior and prior realizations. As the total liquid rate was specified at producers, cumulative oil production from the prior matches the truth for $t \leq 600$ days. Fig. 4.10 shows data matches and predictions for some of the wells from both the prior and conditional realizations. By comparing Figs. 4.10(a) and 4.10(d) one can see that data match and predictions of p_{wf} is significantly improved. The uncertainty in water rate and its prediction (mean predic-



(a) $\ln(k_h)$ of the MAP, layer 1 (b) $\ln(k_h)$ of the MAP, layer 2 (c) $\ln(k_h)$ of the MAP, layer 3

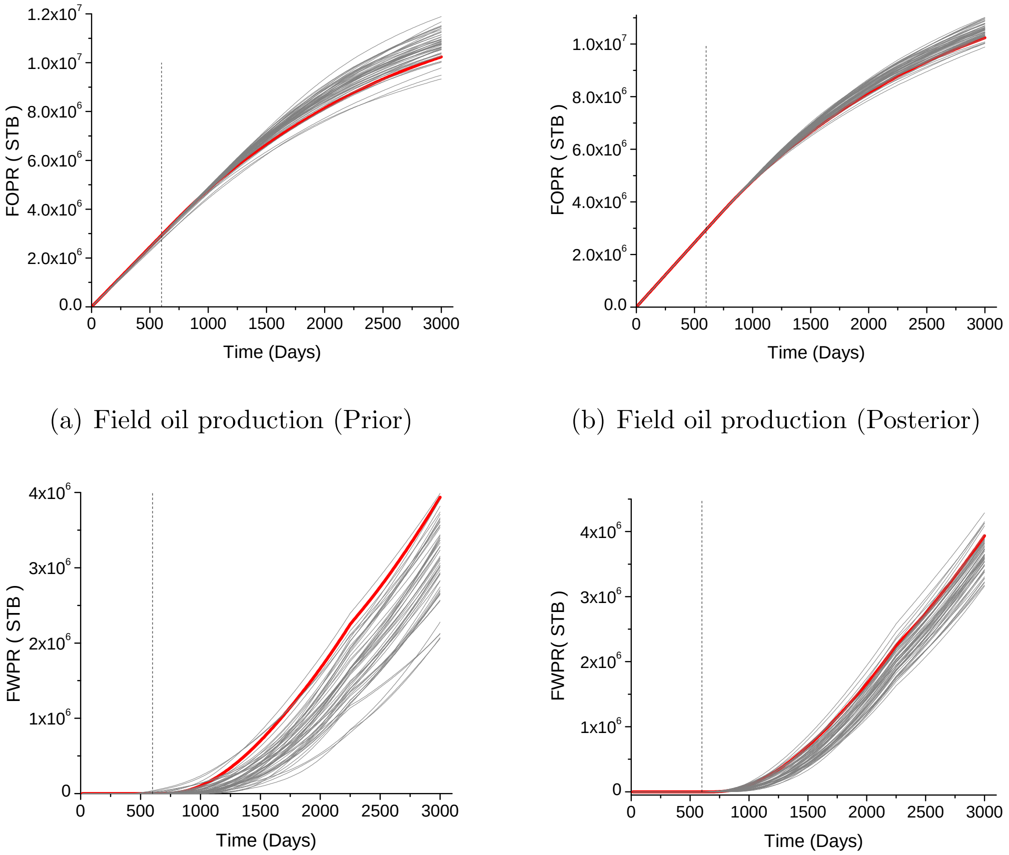
(d) $\ln(k_z)$ of the MAP, layer 1 (e) $\ln(k_z)$ of the MAP, layer 2 (f) $\ln(k_z)$ of the MAP, layer 3

(g) Porosity of the MAP, layer 1 (h) Porosity of the MAP, layer 2 (i) Porosity of the MAP, layer 3

Figure 4.8: Horizontal and vertical log permeability fields, and porosity field of the MAP estimate by history matching 600 days of pressure data, Example 1.

tion) is greatly improved for most of the wells. However, the uncertainty in predictions of q_w data from conditional realizations, is still high; as there is no water breakthrough at 600 days, predictions of water rates are challenging. Data matches and predictions from the prior and conditional realizations for other wells not shown, are similar.

This example shows that with the SVD-EnRML-MI algorithm, one can efficiently condition large number of realizations to pressure data. According to the results of this example, the predictions of water rates from realizations which are conditioned to pressure data before water breakthrough, are not always reliable, and the reduction in uncertainty compared to the prior distribution is not always highly significant.



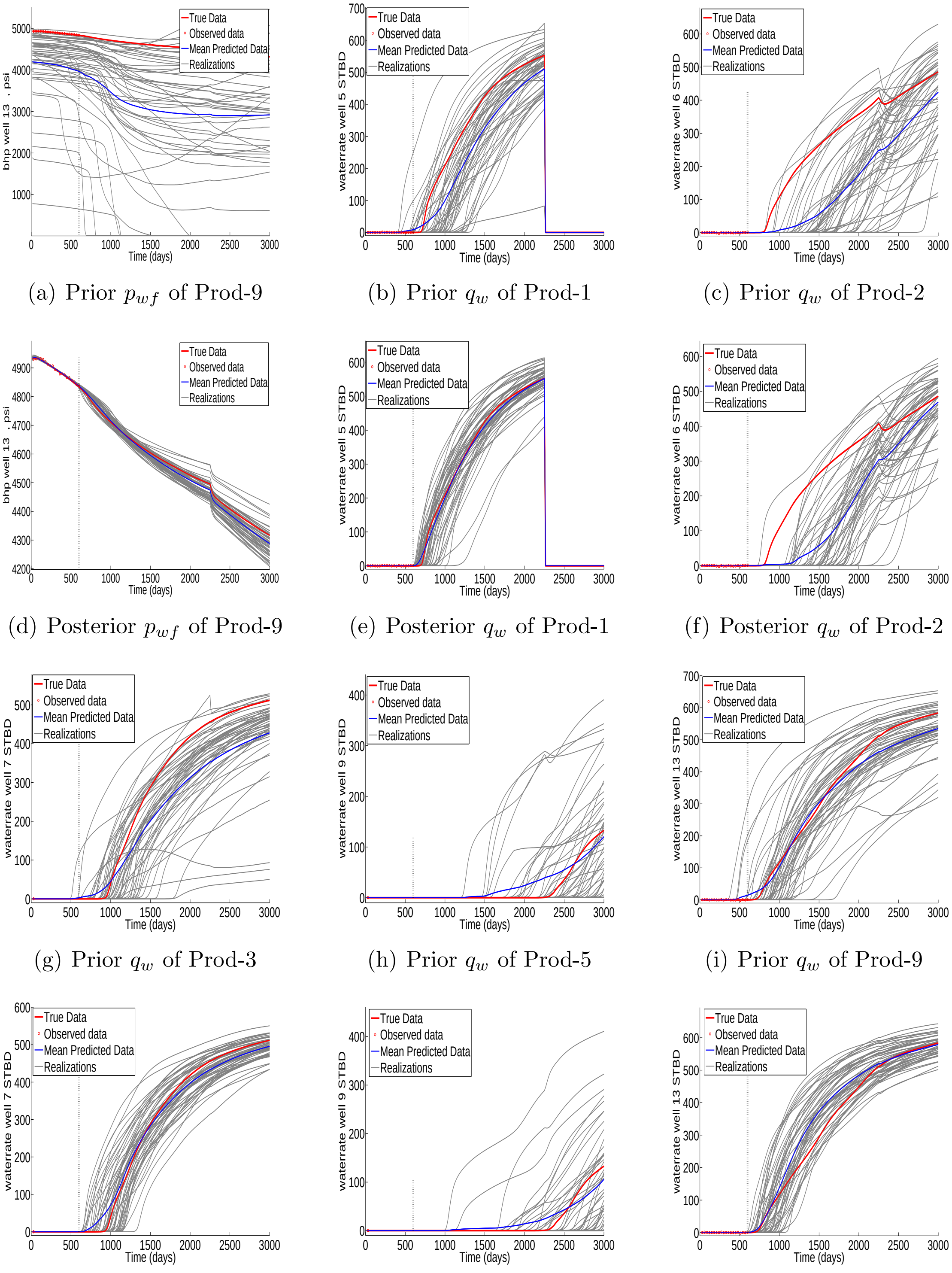
(a) Field oil production (Prior)

(b) Field oil production (Posterior)

(c) Field water production (Prior)

(d) Field water production (Posterior)

Figure 4.9: Field oil and water production from the prior and the posterior samples. Red curve shows the truth, gray curves are from 50 realizations. The dashed vertical line shows the end of history matching (600 days), Example 1.



(a) Prior p_{wf} of Prod-9

(b) Prior q_w of Prod-1

(c) Prior q_w of Prod-2

(d) Posterior p_{wf} of Prod-9

(e) Posterior q_w of Prod-1

(f) Posterior q_w of Prod-2

(g) Prior q_w of Prod-3

(h) Prior q_w of Prod-5

(i) Prior q_w of Prod-9

(j) Posterior q_w of Prod-3

(k) Posterior q_w of Prod-5

(l) Posterior q_w of Prod-9

Figure 4.10: Data matches and predictions for some of the wells from the prior and the posterior samples; the dashed vertical line shows the end of history matching (600 days), Example 1.

4.1.3 Conditioning Realizations to 870 *Days of Data with SVD-EnRML-MI*

At 870 days, four of the producers have experienced water breakthrough. In this history matching period, there are 3 nonzero water rate data from Prod-2. Here, the SVD-EnRML-MI algorithm is used to generate the MAP estimate and $N_e = 10$ RML realizations conditional to 870 days of data. The number of model parameters and observed data are

$$N_m = 3N_x \times N_y \times N_z = 7560,\ N_{\mathrm{d}} = 493.$$

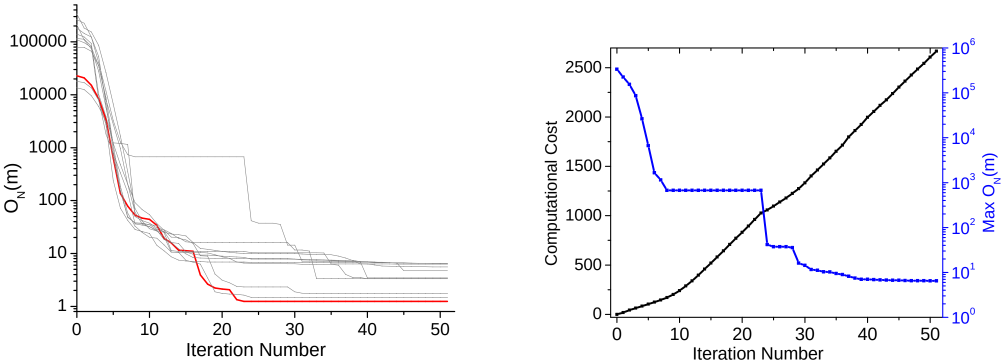
In applying the SVD-EnRML-MI algorithm, we used the lower triangular matrix, L, of the Cholesky decomposition of the covariance matrix, $C_M = LL^T$, for $C_M^{1/2}$. At convergence of the algorithm, we expect the normalized objective function of a conditional realization be less than or equal to $1 + 5\sqrt{2/N_d} = 1.318$. The input parameters are listed in Table 4.5. A maximum of 50 singular triplets was specified as input. The SVD-EnRML-MI algorithm for generating $N_e = 11$ conditional realizations converged in 51 iterations. Fig. 4.11(a) shows $O_N(m)$ of all realizations versus iterations. Note that at late iterations, the rate of decrease of the objective functions is very slow. Fig. 4.11(b) shows the computational cost and maximum $O_N(m)$ versus iterations of SVD-EnRML-MI. It shows that at late iterations, the maximum normalized objective function has a very small decrease. Table. 4.6 shows the summary of the computational costs, and the maximum $O_N(m)$ at convergence.

Fig. 4.12 shows some details of the algorithm. The largest singular value is roughly 2900 at late iterations. The 50th singular value of $G_{D,l}$ is around 1.

Table 4.5: Input Parameters of SVD-EnRML-MI algorithm (870 Days of data), Example 1.

Algorithm	μ_1	μ_2	$N_{\mathrm{SVD,max}}$	$\mu_{\min}$
SVD-EnRML-MI	0.5	0.05	50	0.0003

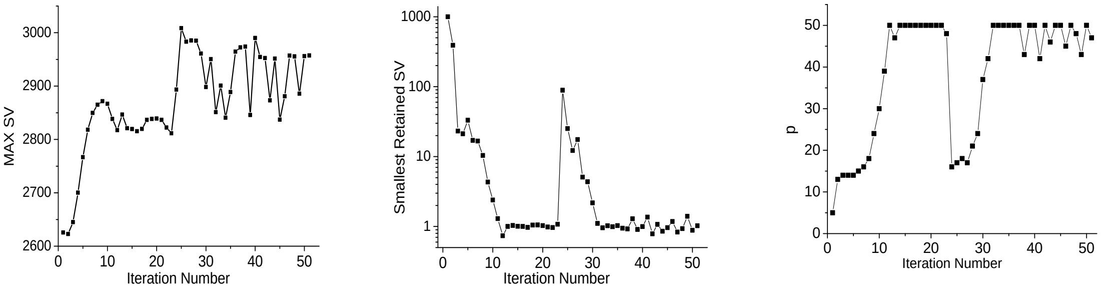
At convergence of the algorithm, the objective functions of realizations are still high. Data matches and predictions for some of the wells from the prior and updated



(a) $O_N(m)$ versus Iterations

(b) Computational cost and the maximum $O_N(m)$

Figure 4.11: (a) $O_N(m)$ of the MAP estimate (red) and 10 RML realizations (gray) versus iterations l of SVD-EnRML-MI. (b) The computational cost in terms of equivalent simulations runs and the maximum $O_N(m)$ versus iterations l of SVD-EnRML-MI. The history matching period is 870 days. Example 1.



(a) First singular value

(b) Smallest retained SV

(c) # of retained SVs

Figure 4.12: The largest singular value and the smallest retained singular value of $G_{D,l}$ and the number of retained singular triplets versus iterations of SVD-EnRML-MI with $N_{\mathrm{SVD,max}} = 50$; the history matching period is 870 days. Example 1.

Table 4.6: Computational cost of SVD-EnRML-MI method for generating the MAP estimate and 10 RML realizations (870 Days of data), and the $\max\{O_N\}$ at convergence, Example 1.

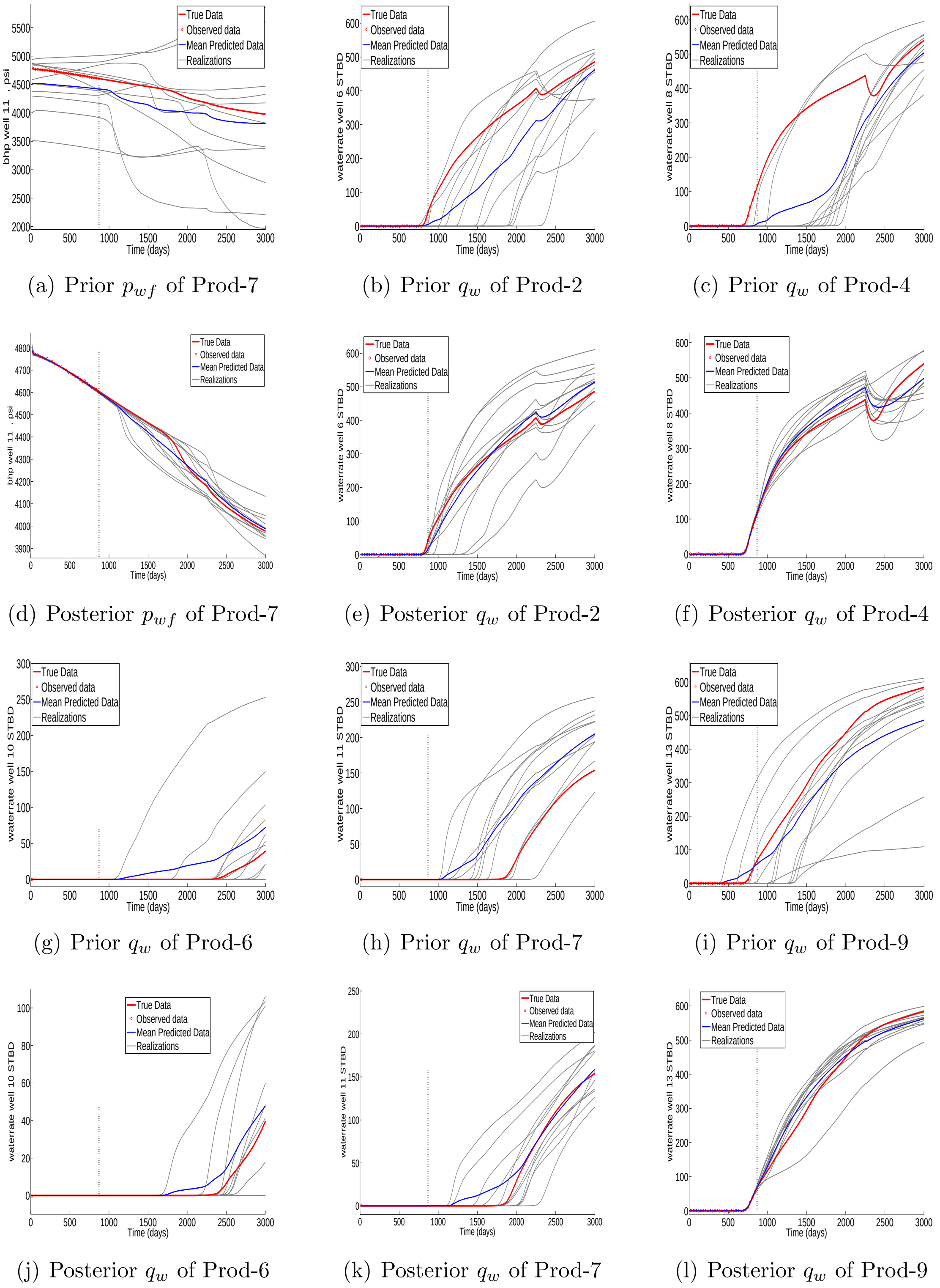
Sim.	Direct	Adjoint	Equ. Sim. Runs	N_{iter}	$\max\{O_N(m)\}$
1397	2566	2515	2667	51	6.482

realizations are shown in Fig. 4.13. In Fig. 4.13(e), some updated realizations do not match the water rate data at late times of history matching period. We noticed that the high objective functions are a result of late breakthrough and few water rate data of Prod-2, as all data matches are good, but water rate of prod-2 for some updated realizations is zero until the end of history matching period. Although not shown, using a smaller or higher number of $N_{\text{SVD,max}}$ did not ameliorate the problem of matching the water rate of Prod-2. According to the results of Fig. 4.13(b), with almost all prior realizations, Prod-2 has water breakthrough later than the truth. In the next subsection, we extend the history matching period to 1080 days. By doing this, the number of nonzero water rate data from Prod-2 increased and at convergence of the algorithm the objective functions of all realizations were small.

4.1.4 Conditioning Realizations to 1080 *Days of Data with SVD-EnRML-MI*

As shown with 2D examples, the SVD-EnRML-MI method is more efficient than SVD-EnRML-SMM. Thus we use the SVD-EnRML-MI algorithm for this example. For the covariance matrix we use a square root approximation of C_M, as discussed later in Section 5.2. For the algorithm not to be affected by a poor approximation of $C_M^{1/2}$, we use a fairly high number of prior realizations, $N_s = 1500$.

The total history matching period is 1080 days. At 1080 days, Prod-1, Prod-2, Prod-3, Prod-4 and Prod-9 have experienced water breakthrough. Observed data include water rate data of these producers and p_{wf} of all wells at 30 day intervals. The number



(a) Prior p_{wf} of Prod-7

(b) Prior q_w of Prod-2

(c) Prior q_w of Prod-4

(d) Posterior p_{wf} of Prod-7

(e) Posterior q_w of Prod-2

(f) Posterior q_w of Prod-4

(g) Prior q_w of Prod-6

(h) Prior q_w of Prod-7

(i) Prior q_w of Prod-9

(j) Posterior q_w of Prod-6

(k) Posterior q_w of Prod-7

(l) Posterior q_w of Prod-9

Figure 4.13: Data matches and predictions for some of the wells from the posterior and the prior realizations. The dashed vertical line shows the end of history matching (870 days), Example 1.

of model parameters and observed data are

$$N_m = 7560,\ N_{\rm d} = 648.$$

The input parameters are listed in Table 4.7. Note that we specified a maximum of 55 singular triplets to be calculated. We chose this number as it has worked well for 2D examples; however we have not done an investigation to see if a smaller or larger number would work better for this case. The SVD-EnRML-MI algorithm for generating $N_e = 11$ conditional realizations converged in 46 iterations. At convergence, the value of $O_N(m)$ of all realizations is less than $1 + 5\sqrt{2/N_d} = 1.278$. Fig. 4.14(a) shows $O_N(m)$ of all realizations versus iterations. Table 4.8 shows the summary of the computational costs.

Table 4.7: Input Parameters of SVD-EnRML-MI algorithm (1080 Days of data), Example 1.

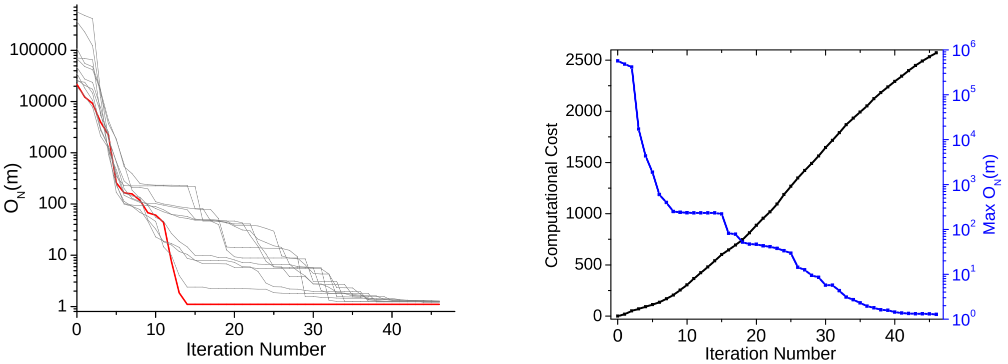
Algorithm	μ_1	μ_2	$N_{\rm SVD,max}$	$\mu_{\min}$
SVD-EnRML-MI	0.5	0.05	55	0.0003

Table 4.8: Computational costs of SVD-EnRML-MI to generate rock property fields of the MAP estimate and 10 RML realizations.

Algorithm	Simulations	Direct	Adjoint	Equ. Sim. Runs	$N_{\rm iter}$
SVD-EnRML-MI	1257	2651	2605	2571	46

Fig. 4.14(b) shows the computational cost and the value of the maximum $O_N(m)$ versus iterations of SVD-EnRML-MI. Note that after 36 iterations, with a computational cost of 2052 equivalent simulation runs, the value of $O_N(m)$ for all 11 realizations is less than 2.

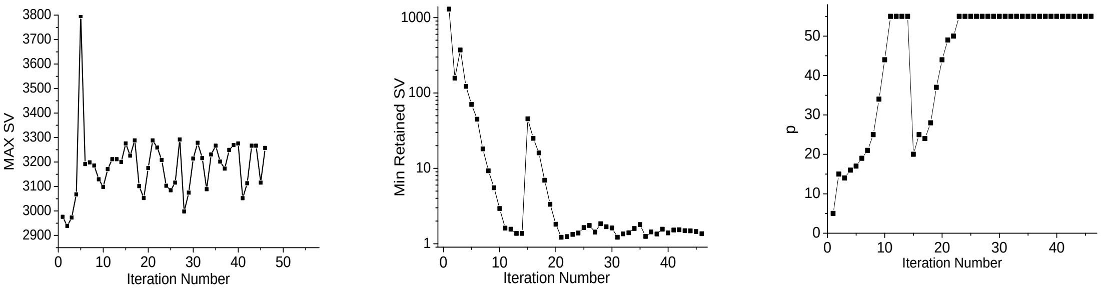
Fig. 4.15 shows some details of the algorithm. The largest singular value is roughly 3200 during iterations. The 55th singular value of $G_{D,l}$ is around 1.4, and it is slightly greater than 1.0 during all iterations.



(a) $O_N(m)$ versus Iterations

(b) Computational cost and maximum $O_N(m)$

Figure 4.14: (a) $O_N(m)$ of the MAP estimate (red) and 10 RML realizations (gray) versus iterations (l). (b) The computational cost in terms of equivalent simulations runs and the maximum $O_N(m)$ versus iterations (l) of SVD-EnRML-MI. The history matching period is 1080 days, Example 1.

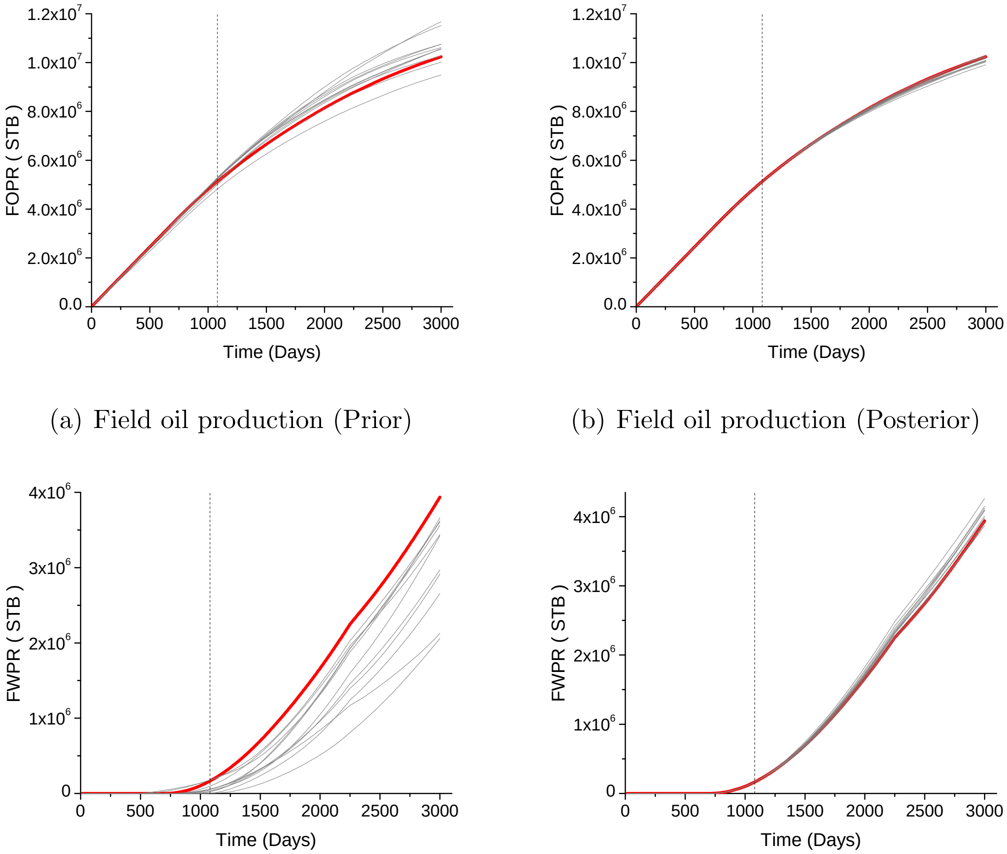


(a) First SV

(b) Smallest retained SV

(c) # of retained SVs

Figure 4.15: The largest singular value and the smallest retained singular value of $G_{D,l}$ and the number of retained singular triplets versus iterations of SVD-EnRML-MI with $N_{\text{SVD,max}} = 55$. The history matching period is 1080 days. Example 1.



(a) Field oil production (Prior)

(b) Field oil production (Posterior)

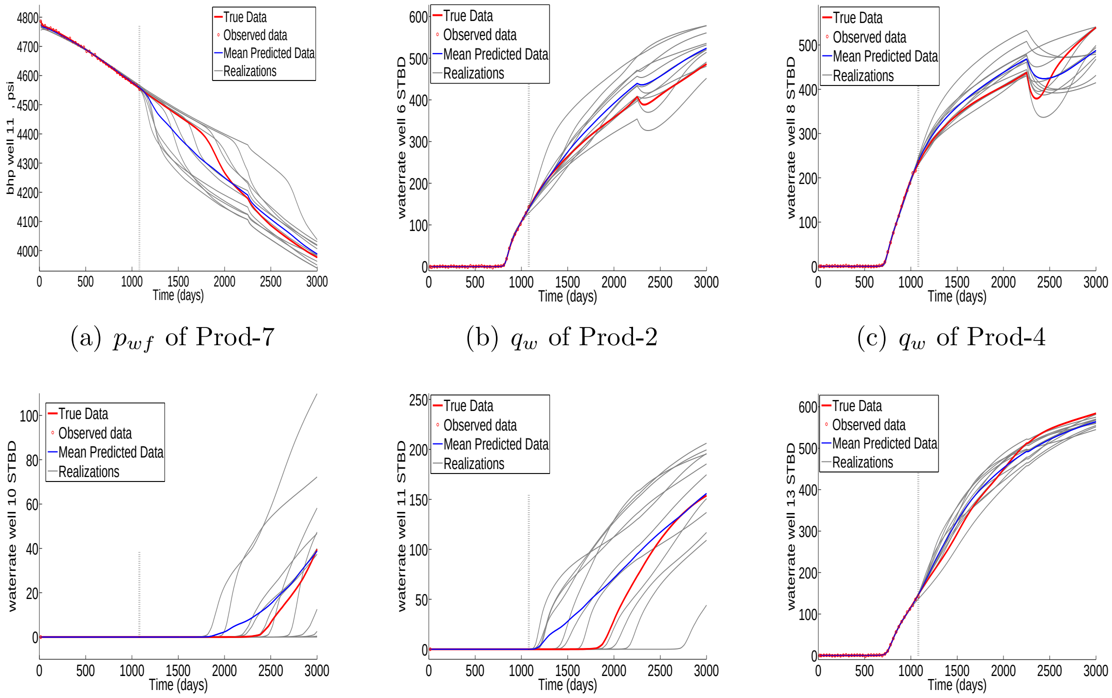
(c) Field water production (Prior)

(d) Field water production (Posterior)

Figure 4.16: Field oil and water production from prior and posterior samples. Red curve shows the truth, Gray curves are from 11 realizations. The dashed vertical line shows the end of history matching (1080 days), Example 1.

Cumulative oil and water production of the reservoir are shown in Fig. 4.16. The uncertainty in the predictions of cumulative oil and water productions is significantly reduced compared to the prior. Data matches and future performance predictions for some of the wells are shown in Fig. 4.17. Data matches and predictions of the wells not shown are similar. Unlike the results of Subsection 4.1.3, the data match and prediction of water rate of Prod-2 are good in this case.

The prior and conditional horizontal and vertical log permeability fields of an RML realization are shown in Fig. 4.18, and the prior and conditional porosity fields of the same realization is shown in Fig. 4.19. The conditional rock property fields, contain some of the main features of the truth.



(a) p_{wf} of Prod-7

(b) q_w of Prod-2

(c) q_w of Prod-4

(d) q_w of Prod-6

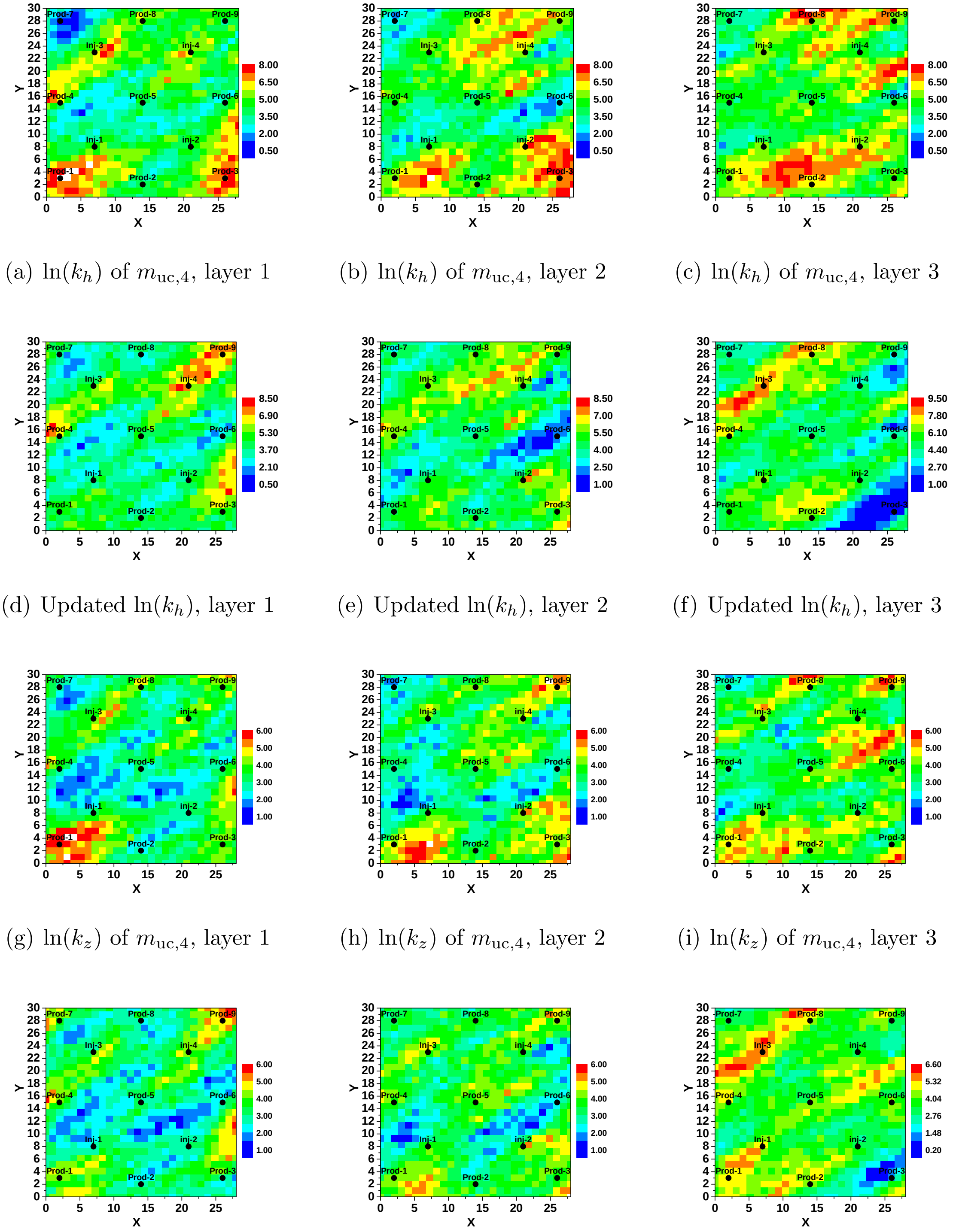
(e) q_w of Prod-7

(f) q_w of Prod-9

Figure 4.17: Data matches and predictions for some of the wells from the posterior sample, the dashed vertical line shows the end of history matching (1080 days), Example 1.

4.2 Example 2

This example pertains to a three-dimensional reservoir model with 30 × 30 × 3



(a) $\ln(k_h)$ of $m_{\mathrm{uc},4}$, layer 1

(b) $\ln(k_h)$ of $m_{\mathrm{uc},4}$, layer 2

(c) $\ln(k_h)$ of $m_{\mathrm{uc},4}$, layer 3

(d) Updated $\ln(k_h)$, layer 1

(e) Updated $\ln(k_h)$, layer 2

(f) Updated $\ln(k_h)$, layer 3

(g) $\ln(k_z)$ of $m_{\mathrm{uc},4}$, layer 1

(h) $\ln(k_z)$ of $m_{\mathrm{uc},4}$, layer 2

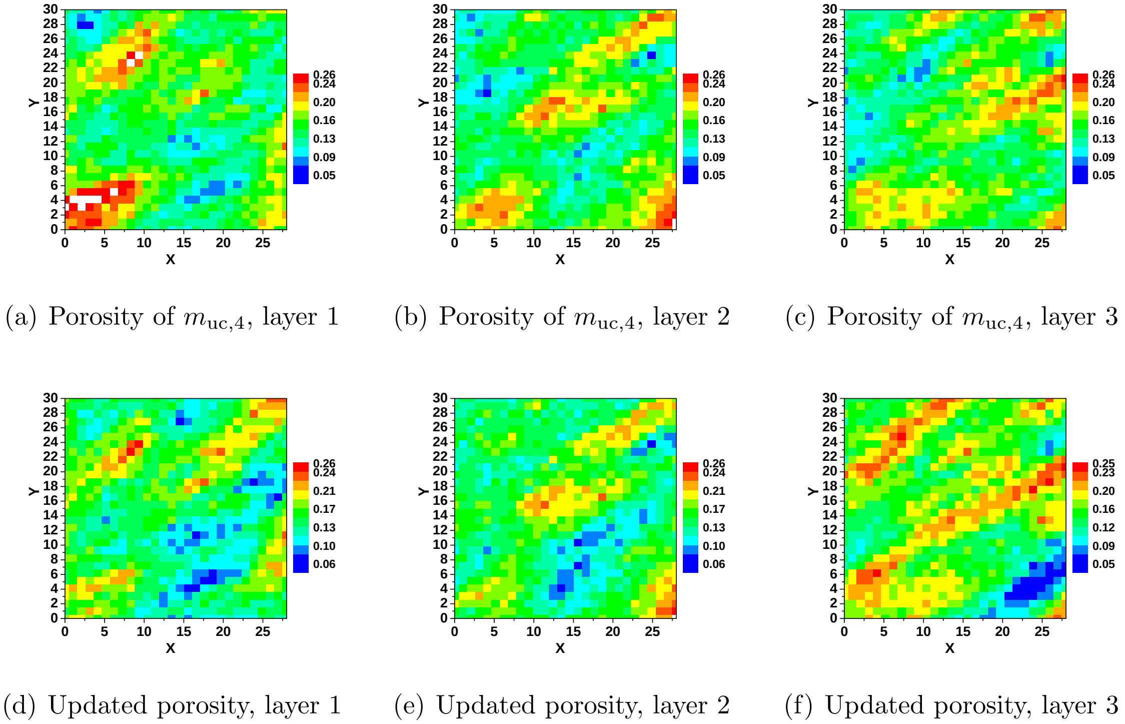
(i) $\ln(k_z)$ of $m_{\mathrm{uc},4}$, layer 3

(j) Updated $\ln(k_z)$, layer 1

(k) Updated $\ln(k_z)$, layer 2

(l) Updated $\ln(k_z)$, layer 3

Figure 4.18: Conditional and prior horizontal and vertical log permeability fields of an RML realization. The history matching period is 1080 days, Example 1.



(a) Porosity of $m_{\mathrm{uc},4}$, layer 1 (b) Porosity of $m_{\mathrm{uc},4}$, layer 2 (c) Porosity of $m_{\mathrm{uc},4}$, layer 3

(d) Updated porosity, layer 1 (e) Updated porosity, layer 2 (f) Updated porosity, layer 3

Figure 4.19: Conditional and prior porosity fields of an RML realization. The history matching period is 1080 days, Example 1.

uniform grid. True porosity, horizontal and vertical log permeability fields are generated from a spherical covariance matrix. The prior parameters are listed in Table 4.9. The cross-covariances between the rock property fields of each layer, are calculated by using the screening hypothesis discussed in Section 1.5. The prior mean of porosity, vertical and horizontal log permeability of all layers are equal, however each layer has a different correlation direction and correlation lengths and there is no correlation in the vertical direction. In Table 4.9 the correlation parameters of each layer are given separately. In this table, l_1 refers to the 1st layer, l_2 refers to the 2nd layer and l_3 refers to the 3rd layer. The correlation coefficients between rock property fields in each layer are the same and equal to 0.65 . True rock property fields are shown in Figs. 4.20(a)- 4.20(c), 4.21(a)- 4.21(c) and 4.22(a)- 4.22(c).

The gridblock dimensions are:

$$\triangle x = \triangle y = 150 \text{ ft}, \ \triangle z = 15 \text{ ft}.$$

Table 4.9: Geostatistical parameters of Example 2.

Parameters	Values
ϕ_{mean}	0.15
$[\ln(k_h)]_{\text{mean}}$	4.50
$[\ln(k_z)]_{\text{mean}}$	3.50
σ_ϕ	0.04
$\sigma_{\ln(k_h)}$	1.4142
$\sigma_{\ln(k_z)}$	1
ρ	0.65
$\alpha(l_1)$	30^o
$r_1(l_1)$	$25\triangle x$
$r_2(l_1)$	$7\triangle x$
$\alpha(l_2)$	60^o
$r_1(l_2)$	$20\triangle x$
$r_2(l_2)$	$7\triangle x$
$\alpha(l_3)$	120^o
$r_1(l_3)$	$25\triangle x$
$r_2(l_3)$	$10\triangle x$

The initial reservoir pressure is 4000 psi. Initially the reservoir is at irreducible water saturation which is $s_{\text{wc}} = 0.1$. There are 9 producers and 4 injectors in this reservoir, located as shown in Fig. 4.20(a). Producers are perforated only in the top layer while injection wells are perforated in the bottom layer. All producers are produced at the same specified p_{wf}, while injectors inject at the same injection rate. Table 4.10 shows the well control schedules. The total history matching period is 1800 days. At 1800 days, only producers P-1, P-3 and P-8 have experienced water breakthrough. Observed data include oil rate and water rate data of producers and p_{wf} of injectors at 30 day intervals.

Table 4.10: Well controls of Example 2. p_{wf} is specified at producers and injection rate is specified at injectors.

Time Period, days	injection rates, STB/D	p_{wf} of producers, psi
$0 - 300$	1100	3800
$300 - 4000$	1000	3500

Observed data are generated by adding Gaussian random noise to the true data.

The standard deviation of the noise (measurement error) is 2% of rates for rate data and 3 psi for pressure data, i.e., $\sigma_{\text{BHP}} = 3$ psi and $\sigma_q = 0.02q$. The minimum measurement error for rates is specified to 0.5 STB/D, and the maximum measurement error is 3 STB/D.

The number of model parameters and observed data are:

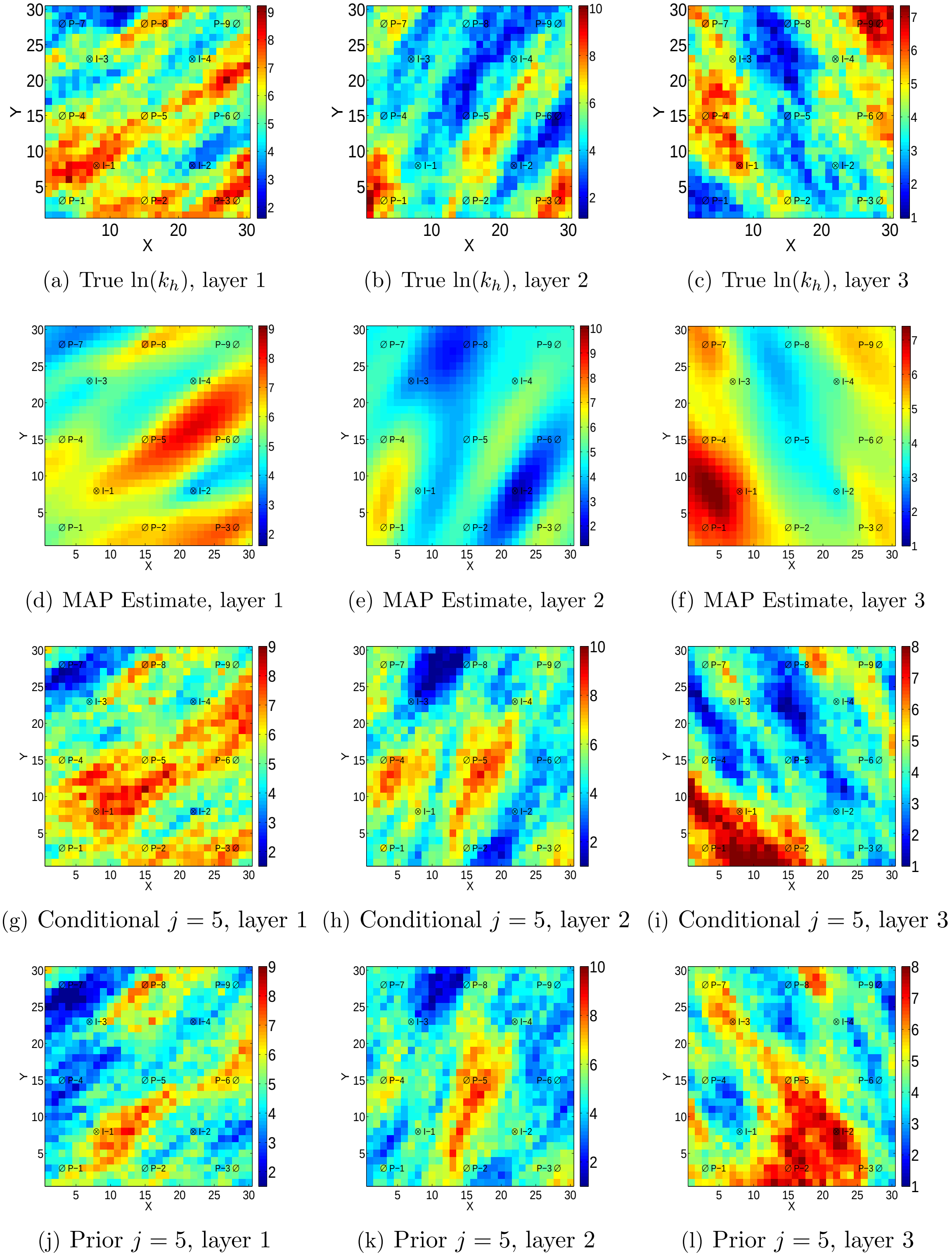
$$N_m = 3N_x \times N_y \times N_z = 8100,\ N_{\text{d}} = 1320.$$

In applying the SVD-EnRML-MI algorithm, we used the reordering of model parameters which is explained in Subsection 2.5. We applied the algorithm with $N_{\text{SVD,max}} = 45$. SVD-EnRML-MI converged in 52 iterations. At convergence only the MAP estimate has $O_N(m)$ less than $1 + 5\sqrt{2/N_d} = 1.194$. However, the maximum value of $O_N(m)$ of realizations is 1.736 which is fairly small. The summary of the computational cost is given in Table 4.11. The rock property fields of the MAP estimate, and a conditional RML realization and its corresponding prior realization are shown in Figs. 4.20, 4.21 and 4.22. The rock property fields of the MAP estimate are smooth and, qualitatively, they display most of the main structural features of the truth. Although the layers are uncorrelated, and none of the wells are perforated in the 2nd layer, many of the main features of this layer are also apparent in the MAP estimate of the rock property fields of layer 2.

Table 4.11: Summary of the computational costs of SVD-EnRML-MI algorithm for generating 11 conditional realizations of rock property fields, and the max(O_N) at convergence, Example 2.

$N_{\text{SVD,max}}$	Sim.	Direct	Adjoint	Equ. Sim. Runs	N_{iter}	$\max\{O_N\}$
45	1588	2466	2412	2807	52	1.736

Fig. 4.23(a) shows the values of $O_N(m)$ of all realizations versus iterations of the algorithm. According to this figure, the objective functions of some of the RML realizations do not decrease or decrease by a negligible amount between iteration $l = 10$



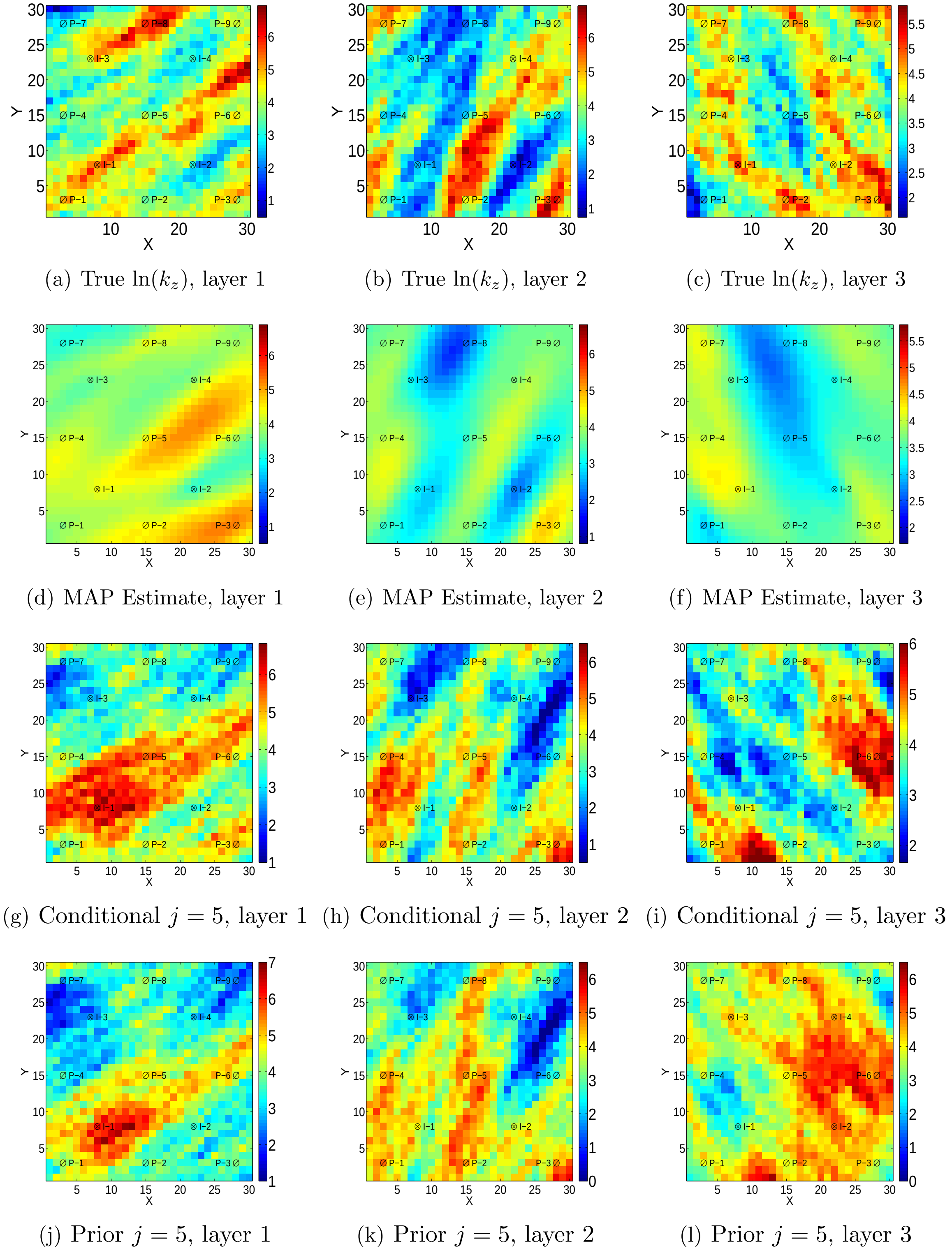
(a) True $\ln(k_h)$, layer 1
(b) True $\ln(k_h)$, layer 2
(c) True $\ln(k_h)$, layer 3

(d) MAP Estimate, layer 1
(e) MAP Estimate, layer 2
(f) MAP Estimate, layer 3

(g) Conditional $j = 5$, layer 1
(h) Conditional $j = 5$, layer 2
(i) Conditional $j = 5$, layer 3

(j) Prior $j = 5$, layer 1
(k) Prior $j = 5$, layer 2
(l) Prior $j = 5$, layer 3

Figure 4.20: Horizontal log permeability fields. Top row shows the true $\ln(k_h)$ fields, the middle rows are the MAP estimate and a conditional realization from SVD-EnRML-MI, the bottom row is the corresponding unconditional realization, Example 2.



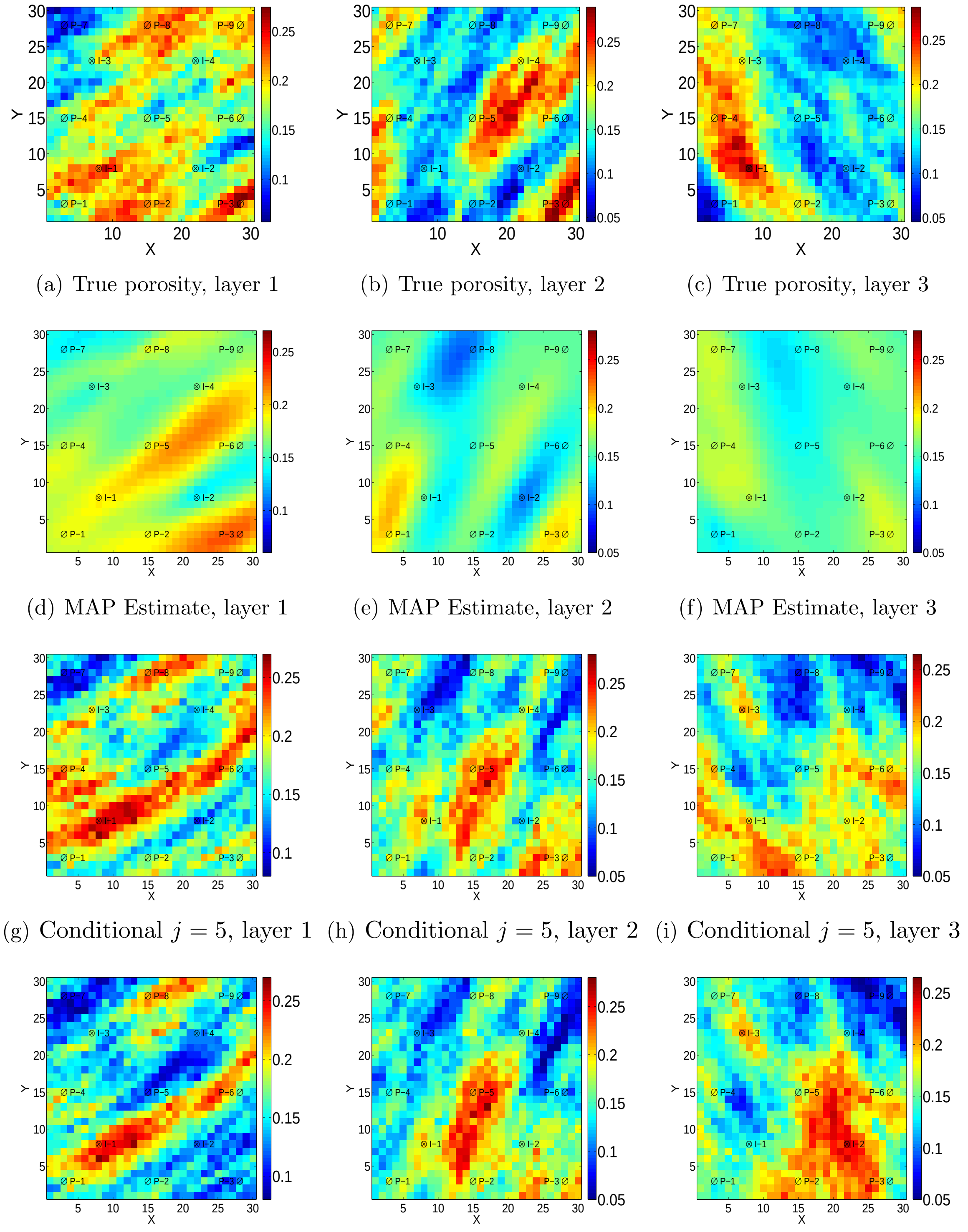
(a) True $\ln(k_z)$, layer 1
(b) True $\ln(k_z)$, layer 2
(c) True $\ln(k_z)$, layer 3

(d) MAP Estimate, layer 1
(e) MAP Estimate, layer 2
(f) MAP Estimate, layer 3

(g) Conditional $j = 5$, layer 1
(h) Conditional $j = 5$, layer 2
(i) Conditional $j = 5$, layer 3

(j) Prior $j = 5$, layer 1
(k) Prior $j = 5$, layer 2
(l) Prior $j = 5$, layer 3

Figure 4.21: Vertical log permeability fields. Top row shows the true $\ln(k_z)$ fields, the middle rows are the MAP estimate and a conditional realization from SVD-EnRML-MI, the bottom row is the corresponding unconditional realization, Example 2.



(a) True porosity, layer 1 (b) True porosity, layer 2 (c) True porosity, layer 3

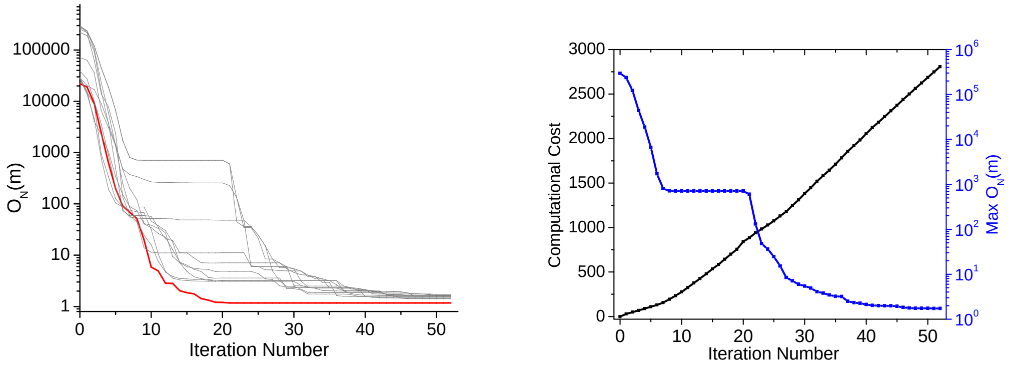
(d) MAP Estimate, layer 1 (e) MAP Estimate, layer 2 (f) MAP Estimate, layer 3

(g) Conditional $j = 5$, layer 1 (h) Conditional $j = 5$, layer 2 (i) Conditional $j = 5$, layer 3

(j) Prior $j = 5$, layer 1 (k) Prior $j = 5$, layer 2 (l) Prior $j = 5$, layer 3

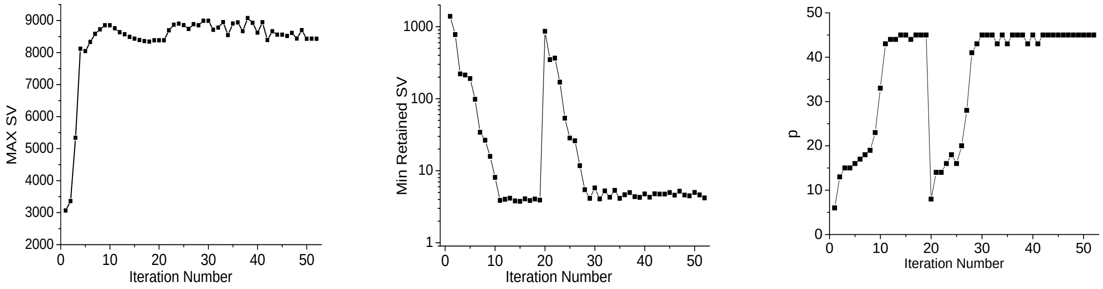
Figure 4.22: Porosity fields. Top row shows the true porosity fields, the middle rows are from the MAP estimate and a conditional realization from SVD-EnRML-MI, the bottom row is the corresponding unconditional realization, Example 2.

and $l = 20$ where the SVD parameters were computed based on the MAP estimate. Fig. 4.23(b) shows the computational cost and the maximum $O_N(m)$ versus iterations of the SVD-EnRML-MI. At early iterations, the rate of decrease in the maximum O_N is fast, while the computational cost has a small rate of increase; but at late iterations, the rate of decrease in the maximum value of $O_N(m)$ is slow.



(a) $O_N(m)$ versus Iterations (b) Computational cost and maximum $O_N(m)$

Figure 4.23: (a) $O_N(m)$ of the MAP estimate (red) and 10 RML realizations (gray) versus iterations, l. (b) The computational cost in terms of equivalent simulation runs and the maximum $O_N(m)$ versus iterations (l) of SVD-EnRML-MI with $N_{\text{SVD,max}} = 45$, Example 2.



(a) First SV (b) Smallest retained SV (c) # of retained SVs

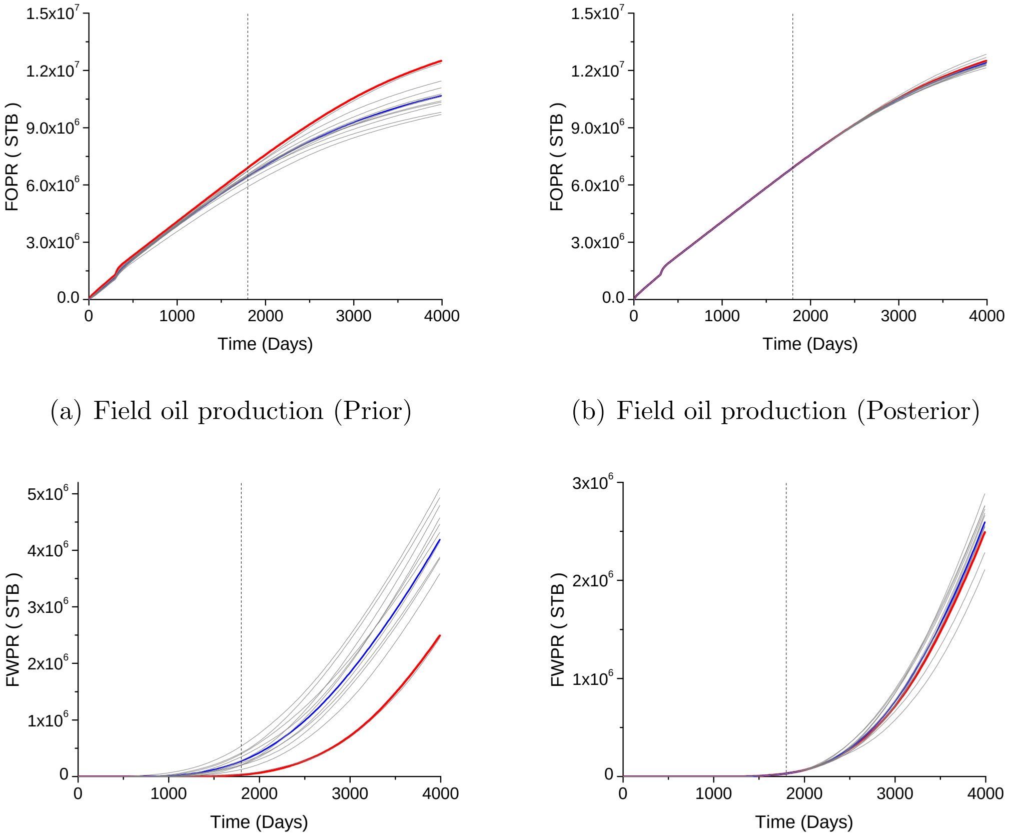
Figure 4.24: The largest singular value and the smallest retained singular value of $G_{D,l}$ and the number of retained singular triplets versus iterations of SVD-EnRML-MI with $N_{\text{SVD-max}} = 45$, Example 2.

Fig. 4.24 shows some details of the algorithm. The largest singular value is between 8000 and 9000 during late iterations. The 45th singular value of $G_{D,l}$ is around 4.5. Thus, it appears that the normalized objective functions are not as small as expected at convergence, is because too few singular triplets are used. Note that at late iterations, the 45th singular value is around 4.5 which is fairly high, and there might be several singular values between 1 and 4.5; in addition, even in 2D examples, we obtained better results by retaining 55 or 65 singular values of $G_{D,l}$ at late iterations of the algorithm, rather than 45. Unfortunately, because of the inefficient iterative solver used in the simulator, the computer run for this result required 250 hours (more than 10 days), therefore, limited computational experiments are possible.

Cumulative oil and water production of the reservoir are shown in Fig. 4.25. The uncertainty in the predictions of cumulative oil and water productions is highly decreased by conditioning the realizations to production data with SVD-EnRML-MI. According to Fig. 4.25(c), the cumulative water production at the end of the history matching period is small, however the future predictions are very good. As expected, the uncertainty bound widens as time proceeds, however the mean of the predictions is close to the truth. Data matches and future performance predictions from the prior and the conditional realizations for some of the wells are shown in Fig. 4.26. The mean of the predictions for q_w of P-3 and P-8 in Figs. 4.26(e) and 4.26(j) are very close to the truth. For q_w of P-5 in Fig. 4.26(f), although there is high uncertainty in the breakthrough time, the mean of the predictions of q_w is very close to the truth. In Fig. 4.26(l), the truth is out of the uncertainty bounds. Note that the mean of the predictions of q_w from the same well, in Fig. 4.26(k), does not match the true prediction, which will affect the prediction of oil rate.

4.3 Example 3

This example pertains to a three-dimensional reservoir model with $30 \times 15 \times 4$ uniform grid. True porosity and horizontal and vertical log permeability fields are generated from a spherical covariance matrix. The prior parameters are presented in Table 4.12. The cross-covariances are calculated by using the screening hypothesis discussed in Section 1.5. The prior mean of each of the porosity, horizontal and vertical log permeability for all layers are equal, however each layer has a different correlation direction and cor-



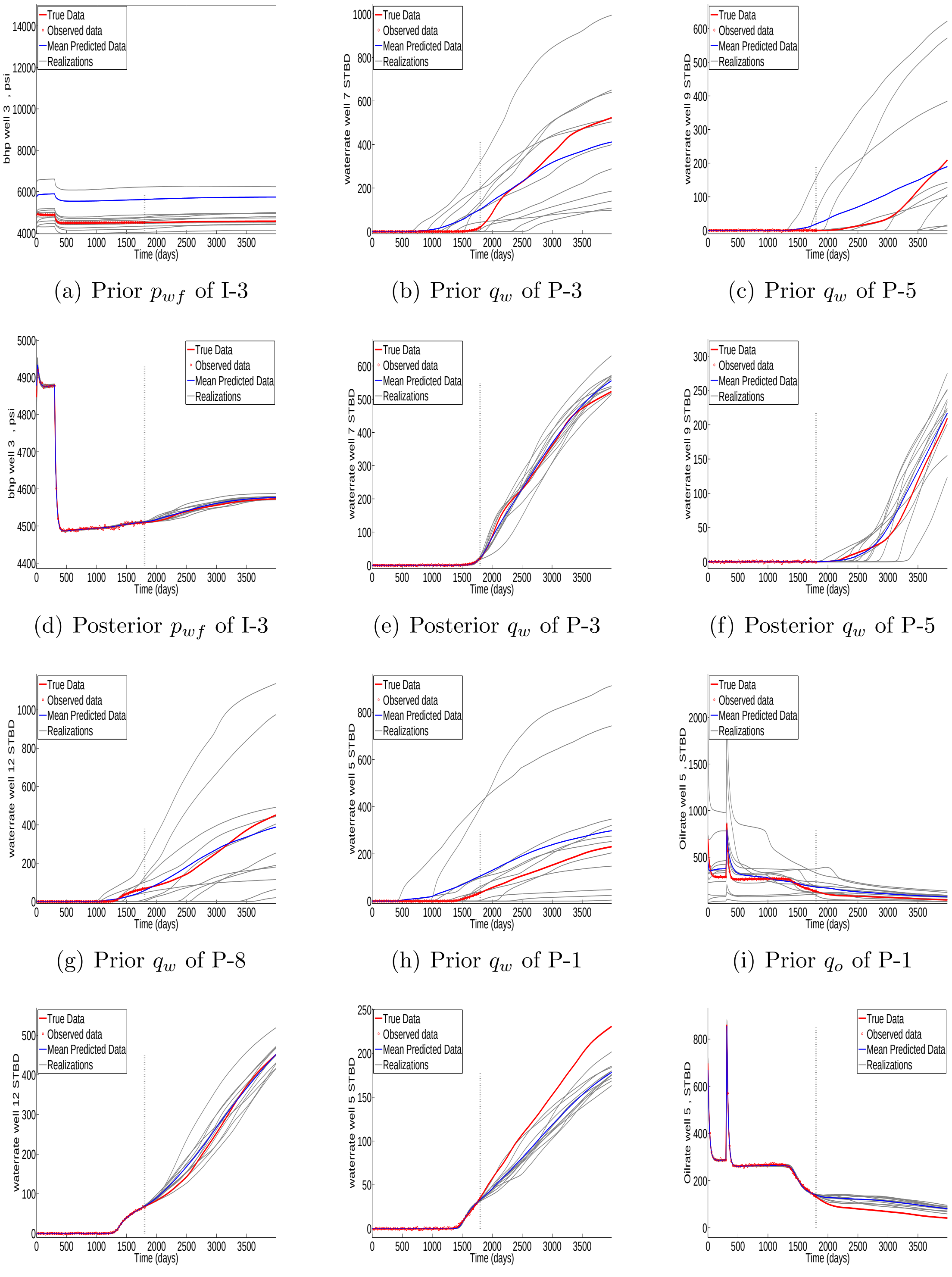
(a) Field oil production (Prior)

(b) Field oil production (Posterior)

(c) Field water production (Prior)

(d) Field water production (Posterior)

Figure 4.25: Field oil and water production from prior and posterior distributions. Red curve shows the truth, gray curves are from 11 realizations, blue curve is the mean of the gray curves. The dashed vertical line shows the end of history matching (1800 days). Example 2.



(a) Prior p_{wf} of I-3

(b) Prior q_w of P-3

(c) Prior q_w of P-5

(d) Posterior p_{wf} of I-3

(e) Posterior q_w of P-3

(f) Posterior q_w of P-5

(g) Prior q_w of P-8

(h) Prior q_w of P-1

(i) Prior q_o of P-1

(j) Posterior q_w of P-8

(k) Posterior q_w of P-1

(l) Posterior q_o of P-1

Figure 4.26: Data matches and predictions for some of the wells from the posterior and prior samples. The dashed vertical line shows the end of history matching (1800 days), Example 2.

relation lengths and there is no correlation in the vertical direction. In Table 4.12 the correlation parameters of each layer are given separately. The correlation coefficients between rock property fields in each layer is the same. True rock property fields are shown in the left columns of Figs. 4.27, 4.28 and 4.29.

Table 4.12: Geostatistical parameters of Example 3.

Parameters	Values
ϕ_{mean}	0.14
$[\ln(k_h)]_{\text{mean}}$	4.50
$[\ln(k_z)]_{\text{mean}}$	3.50
σ_ϕ	0.03
$\sigma_{\ln(k_h)}$	1.4142
$\sigma_{\ln(k_z)}$	1
ρ	0.65
$\alpha(l_1)$	30^o
$r_1(l_1)$	$25\triangle x$
$r_2(l_1)$	$7\triangle x$
$\alpha(l_2)$	60^o
$r_1(l_2)$	$20\triangle x$
$r_2(l_2)$	$7\triangle x$
$\alpha(l_3)$	0^o
$r_1(l_3)$	$18\triangle x$
$r_2(l_3)$	$10\triangle x$
$\alpha(l_4)$	120^o
$r_1(l_4)$	$22\triangle x$
$r_2(l_4)$	$10\triangle x$

The gridblock dimensions are:

$$\triangle x = \triangle y = 150\,\text{ft},\ \triangle z = 15\,\text{ft}.$$

The initial reservoir pressure is 4000 psi. Initially the reservoir is at irreducible water saturation which is $s_{\text{wc}} = 0.1$. There are 6 producers and 2 injectors in this reservoir, as can be seen in Fig. 4.27. Producers are perforated only in the top layer while injection wells are perforated in the two bottom layers (the 3rd and the 4th layers). All producers are produced at the same specified p_{wf}, while injectors inject at the same injection rate. Table 4.13 shows the well control schedules. The total history matching period is 1350 days. At 1350 days, only producers P-2 and P-5 have experienced water breakthrough. Observed data include oil rate and water rate data of all producers and p_{wf} of injectors at 30 day intervals.

Table 4.13: Well controls of Example 3. p_{wf} is specified at producers and injection rate is specified at injectors.

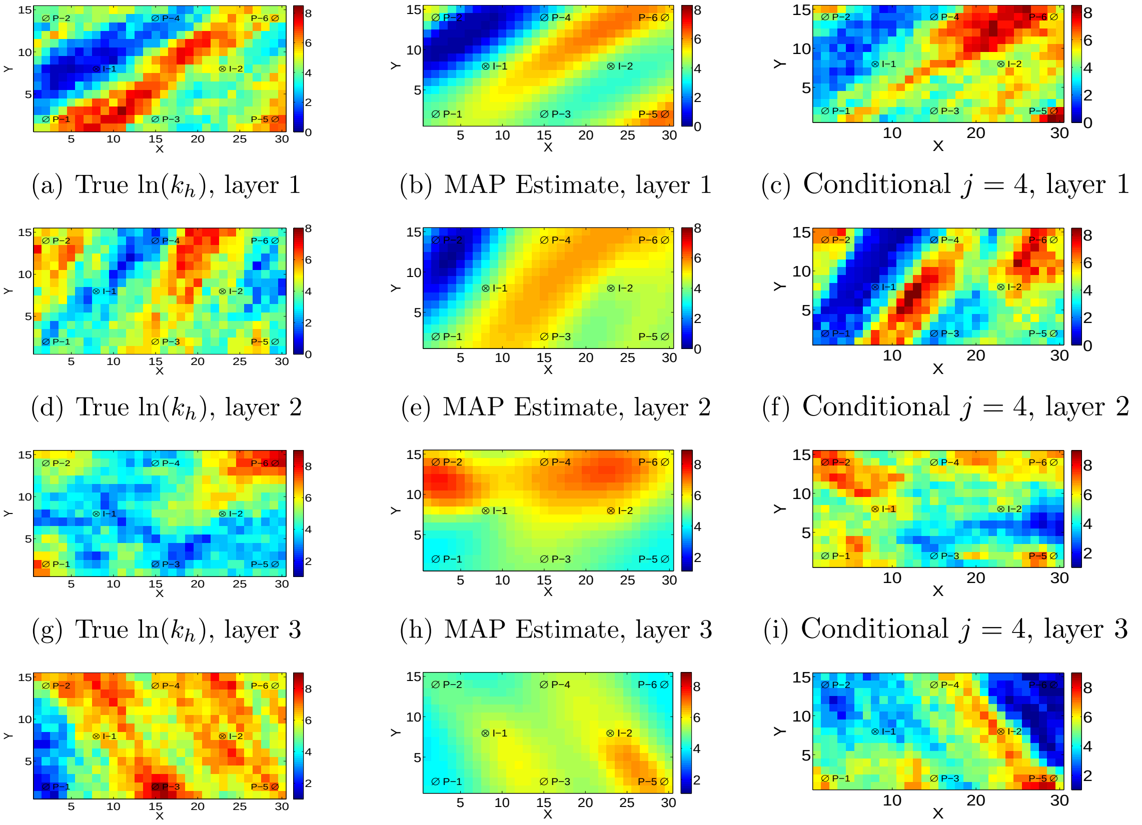
Time Period, days	injection rates, STB/D	p_{wf} of producers, psi
$0 - 300$	1000	3800
$300 - 3000$	1400	3500

Observed data are generated by adding noise to the true data. The standard deviation of the noise (measurement error) is 2% of rates for rate data and 3 psi for pressure data, i.e., $\sigma_{\text{BHP}} = 3$ psi and $\sigma_q = 0.02q$. The minimum measurement error for rates is specified to 0.5 STB/D while the maximum measurement error is 3 STB/D. The number of model parameters and observed data are:

$$N_m = 3N_x \times N_y \times N_z = 5400, \; N_{\text{d}} = 630.$$

In applying the SVD-EnRML-MI algorithm, we used the reordering of model parameters which is explained in Section 2.5. The objective is to generate the MAP estimate and 10 RML realizations. We applied the algorithm with $N_{\text{SVD,max}} = 40$. The values of O_N versus iterations are given in Fig. 4.30(a). As one can see in Fig. 4.30(a), the algorithm converged in 48 iterations. At convergence, some of the realizations have high values of O_N; the maximum O_N is 7.24 and the mean of O_N of the 11 realizations is 3.27. The first singular value, the number of retained singular values and the smallest retained singular value of $G_{D,l}$ versus iterations are shown in Figs. 4.31(a), 4.31(b) and 4.31(c), respectively. Although the 40th singular value of $G_{D,l}$ varies between 1 and 1.4, which is small, the algorithm converged with high values of objective functions.

Next, we applied the SVD-EnRML-MI algorithm with $N_{\text{SVD,max}} = 50$. The al-



(a) True $\ln(k_h)$, layer 1

(b) MAP Estimate, layer 1

(c) Conditional $j = 4$, layer 1

(d) True $\ln(k_h)$, layer 2

(e) MAP Estimate, layer 2

(f) Conditional $j = 4$, layer 2

(g) True $\ln(k_h)$, layer 3

(h) MAP Estimate, layer 3

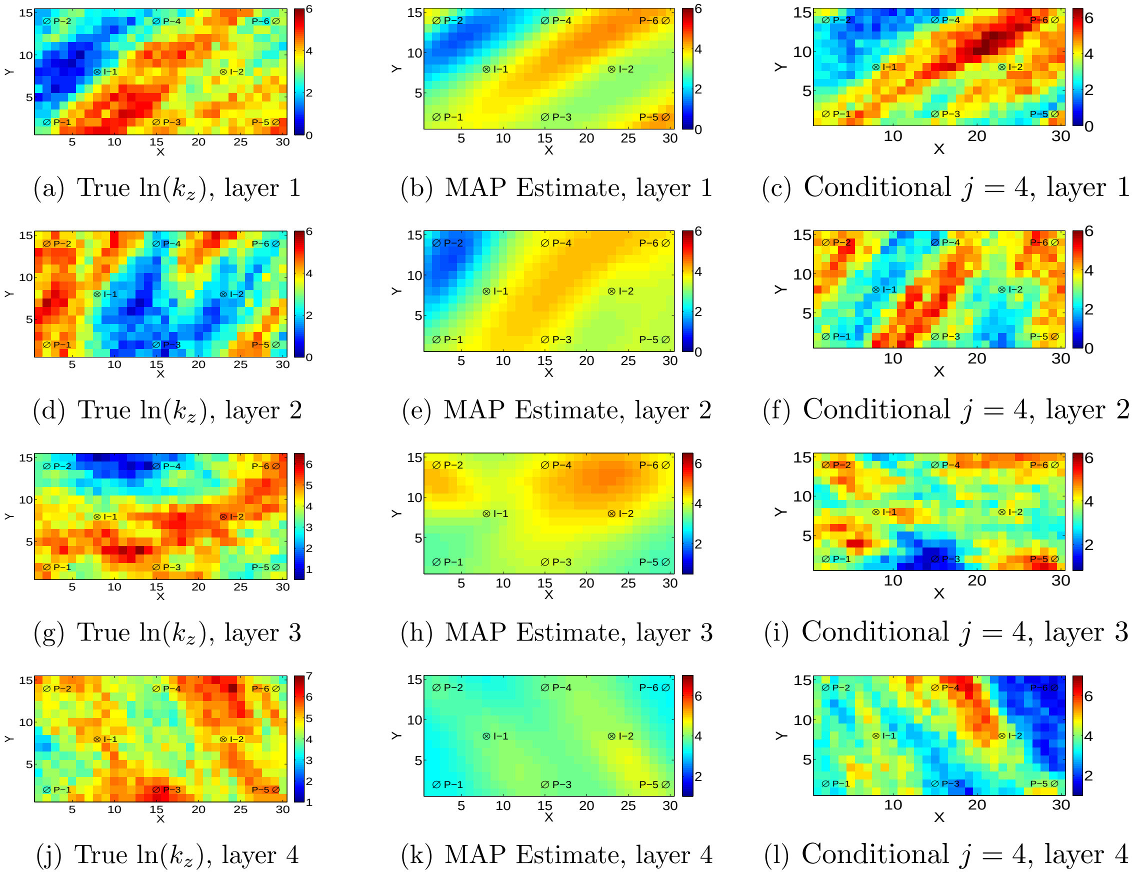
(i) Conditional $j = 4$, layer 3

(j) True $\ln(k_h)$, layer 4

(k) MAP Estimate, layer 4

(l) Conditional $j = 4$, layer 4

Figure 4.27: Horizontal log permeability fields. Left column shows the true $\ln(k_h)$ fields, the middle column is from the MAP estimate and the right column is a conditional realization from SVD-EnRML-MI, Example 3.



(a) True $\ln(k_z)$, layer 1

(b) MAP Estimate, layer 1

(c) Conditional $j = 4$, layer 1

(d) True $\ln(k_z)$, layer 2

(e) MAP Estimate, layer 2

(f) Conditional $j = 4$, layer 2

(g) True $\ln(k_z)$, layer 3

(h) MAP Estimate, layer 3

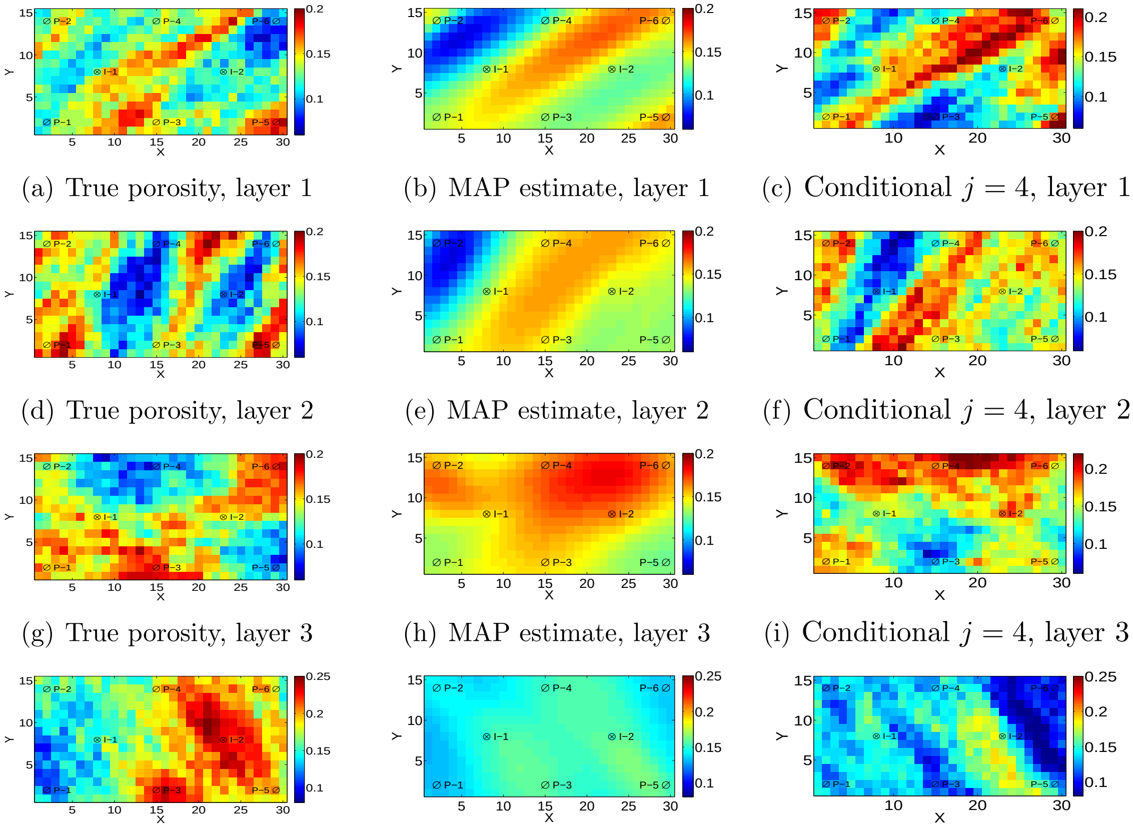
(i) Conditional $j = 4$, layer 3

(j) True $\ln(k_z)$, layer 4

(k) MAP Estimate, layer 4

(l) Conditional $j = 4$, layer 4

Figure 4.28: Vertical log permeability fields. Left column shows the true $\ln(k_z)$ fields, the middle column is from the MAP estimate and the right column is a conditional realization from SVD-EnRML-MI, Example 3.



(a) True porosity, layer 1

(b) MAP estimate, layer 1

(c) Conditional $j = 4$, layer 1

(d) True porosity, layer 2

(e) MAP estimate, layer 2

(f) Conditional $j = 4$, layer 2

(g) True porosity, layer 3

(h) MAP estimate, layer 3

(i) Conditional $j = 4$, layer 3

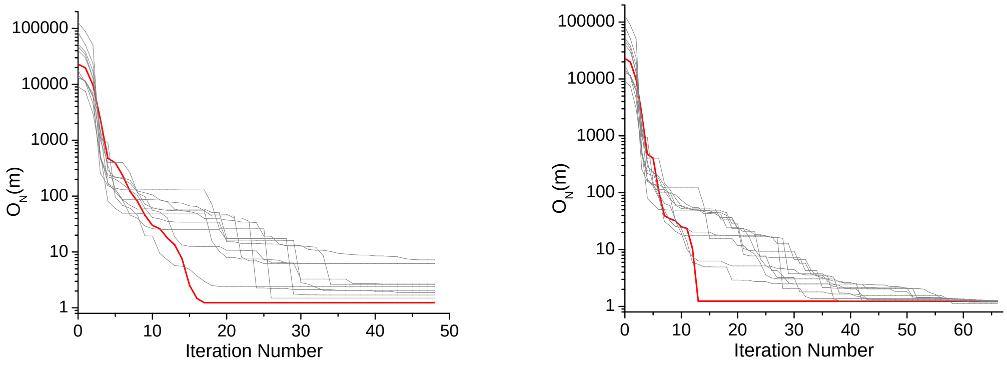
(j) True porosity, layer 4

(k) MAP estimate, layer 4

(l) Conditional $j = 4$, layer 4

Figure 4.29: Porosity fields. Left column shows the true porosity fields, the middle column is from the MAP estimate and the right column is a conditional realization from SVD-EnRML-MI, Example 3.

gorithm converged in 66 iterations. The values of O_N versus iterations are given in Fig. 4.30(b). At convergence, the values of $O_N(m)$ are less than $1 + 5\sqrt{2/N_d} = 1.28$ for all realizations. The first singular value of $G_{D,l}$ versus iterations is shown in Fig. 4.33(a). After the first few iterations, the first singular value fluctuates around 1500. The number of singular values and the smallest retained singular values of $G_{D,l}$ versus iterations are shown in Figs. 4.33(b) and 4.33(c), respectively. The 50th singular value of $G_{D,l}$ changes from 0.6 to 1.27. Fig. 4.32 shows the distribution of the singular values at a late iteration of the SVD-EnRML-MI algorithm. Note that only the first 18 singular values are greater than 10, in addition, many of the retained singular values are around 1. The 40th and the 50th singular values are very close.



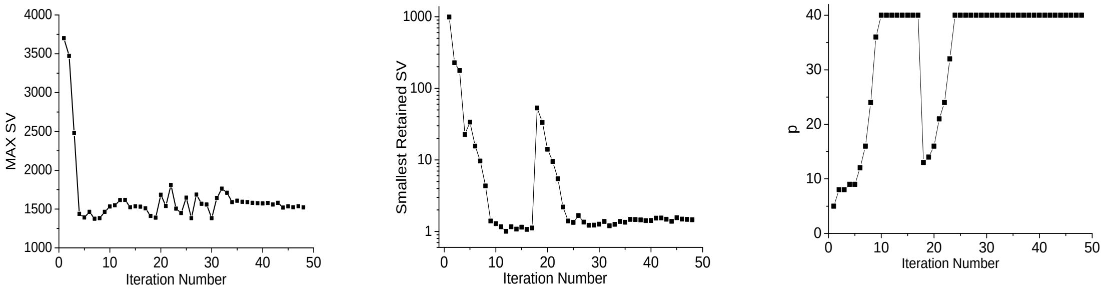
(a) $O_N(m)$, $N_{\text{SVD,max}} = 40$

(b) $O_N(m)$, $N_{\text{SVD,max}} = 50$

Figure 4.30: $O_N(m)$ of the MAP estimate (red) and 10 RML realizations (gray) versus iterations, l, of SVD-EnRML-MI with $N_{\text{SVD,max}}$ of 50 and 40, Example 3.

By comparing Figs. 4.30(a) and 4.30(b), one can see that with $N_{\text{SVD,max}} = 40$, the MAP estimate converged in 17 iterations, while with $N_{\text{SVD,max}} = 50$ the MAP estimate converged in 13 iterations. If we were to generate only the MAP estimate, the computational cost with $N_{\text{SVD,max}} = 50$, would be less than the one with $N_{\text{SVD,max}} = 40$.

The computational cost of the SVD-EnRML-MI algorithm with $N_{\text{SVD,max}} = 40$ and $N_{\text{SVD,max}} = 50$ are given in the 1st and 2nd rows of Table. 4.14, respectively. By increasing $N_{\text{SVD,max}}$ from 40 to 50, the computational cost increased, however, at convergence, all normalized objective functions are less than $1 + 5\sqrt{2/N_d} = 1.28$. Since



(a) First SV (b) Smallest retained SV (c) # of retained SVs

Figure 4.31: The largest singular value (SV) and the smallest retained singular value of $G_{D,l}$ and the number of retained singular triplets versus iterations of SVD-EnRML-MI with $N_{\mathrm{SVD,max}} = 40$, Example 3.

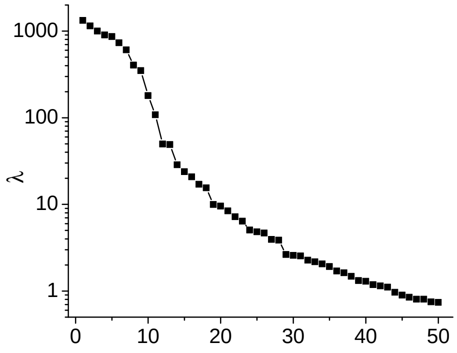
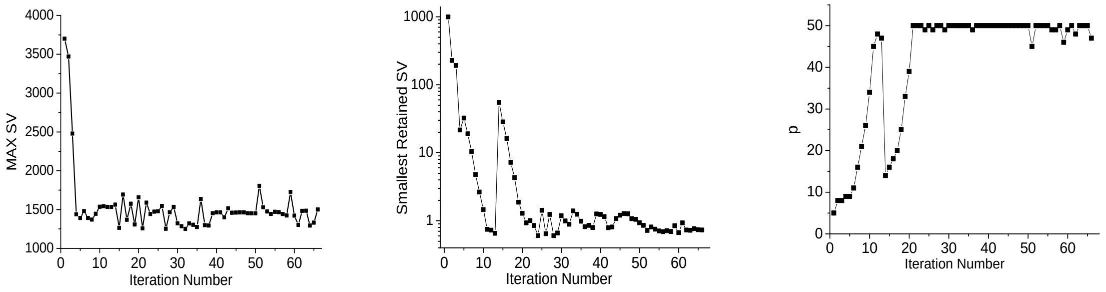


Figure 4.32: 50 singular values of $G_{D,l}$ at a late iteration of the SVD-EnRML-MI algorithm, Example 3.



(a) First SV (b) Smallest retained SV (c) # of retained SVs

Figure 4.33: The largest singular value (SV) and the smallest retained singular value of $G_{D,l}$ and the number of retained singular triplets versus iterations of SVD-EnRML-MI with $N_{\mathrm{SVD,max}} = 50$, Example 3.

the history matching is based on Bayesian framework, to have a more probable model, the corresponding objective function should be small. A realization with a normalized objective function of 7.24, compared to the one with $O_N(m) = 1.27$, is far less probable.

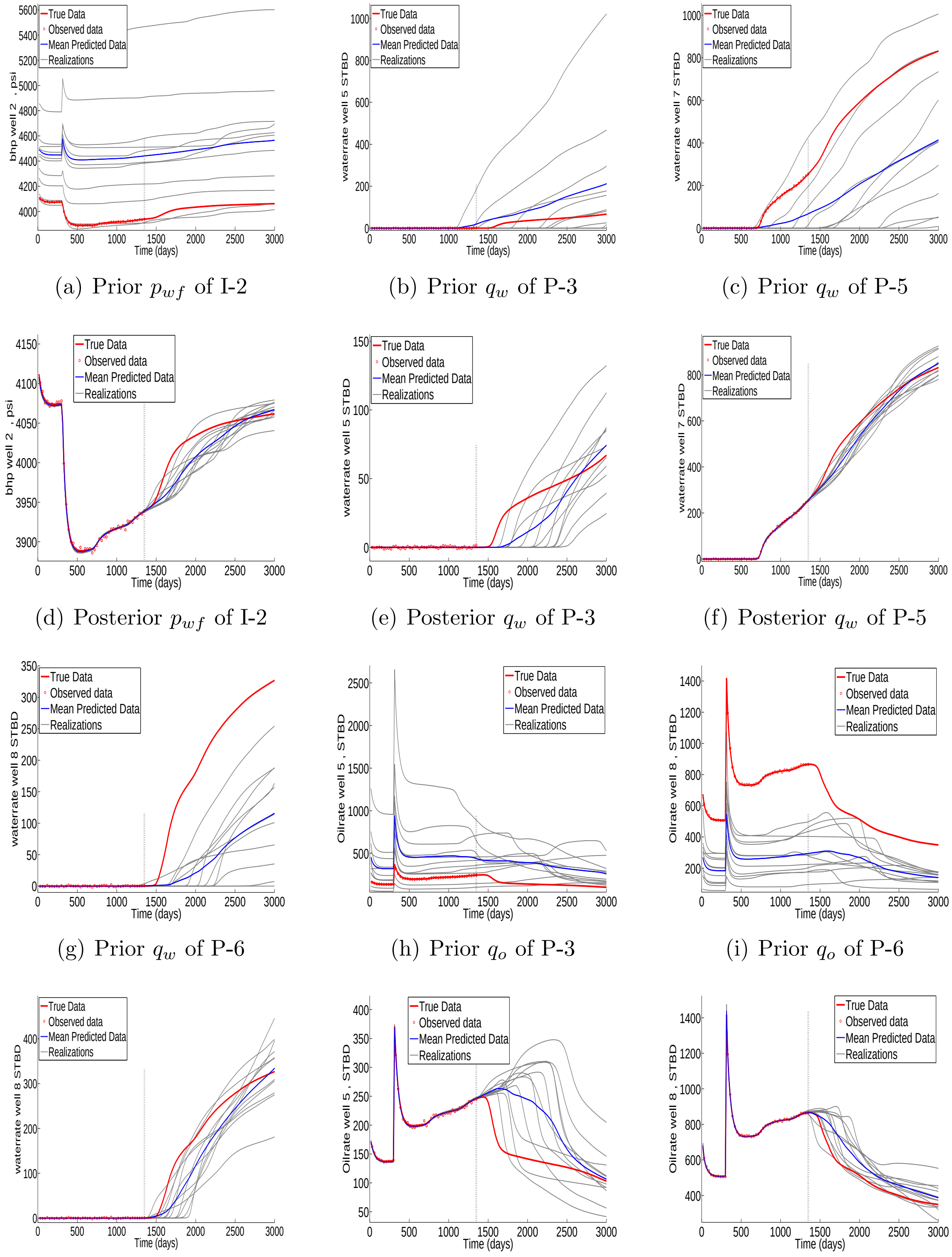
Table 4.14: Summary of the computational costs of SVD-EnRML-MI algorithm for generating 11 conditional realizations of rock property fields, and the $\max\{O_N(m)\}$ at convergence, Example 3.

$N_{\text{SVD,max}}$	Sim.	Direct	Adjoint	Equ. Sim. Runs	N_{iter}	$\max\{O_N\}$
40	1462	2145	2097	2522	48	7.24
50	1752	3553	3487	3512	66	1.276

The rock property fields of the true model, the MAP estimate and a conditional realization, are shown at Figs. 4.27, 4.28 and 4.29. The rock property fields of the MAP estimate are smooth and have, at best, some general features of the truth. In particular, the estimated properties of layer 2 (no well is completed in layer 2) are distinctly qualitatively different from the truth. The conditional realization also has many features different from the truth, however, the conditional realization is plausible based on the prior geological model and is consistent with the production. This is all we can expect from a history matched model. Moreover, the producers are completed only in the top layer, no well is completed in layer 2, and the only completions in layers 3 and 4 are those pertaining to the two injection wells. Thus, it is not surprising that data is insufficient to provide a realization that has all of the main features of the truth.

Data matches and predictions from the prior and conditional realizations for some of the wells are given in Fig. 4.34. The uncertainty in predictions of q_w of the wells is highly decreased in Figs. 4.34(f) and 4.34(j). The high uncertainty in the oil rate of P-3 in Fig. 4.34(k), is because the water breakthrough at this well happens after the end of history matching; as can be seen in Fig. 4.34(e), there is high uncertainty in the breakthrough time at P-3.

The cumulative oil and water production from the prior and conditional realizations are shown in Fig. 4.35. The uncertainty in the cumulative oil and water production



(a) Prior p_{wf} of I-2

(b) Prior q_w of P-3

(c) Prior q_w of P-5

(d) Posterior p_{wf} of I-2

(e) Posterior q_w of P-3

(f) Posterior q_w of P-5

(g) Prior q_w of P-6

(h) Prior q_o of P-3

(i) Prior q_o of P-6

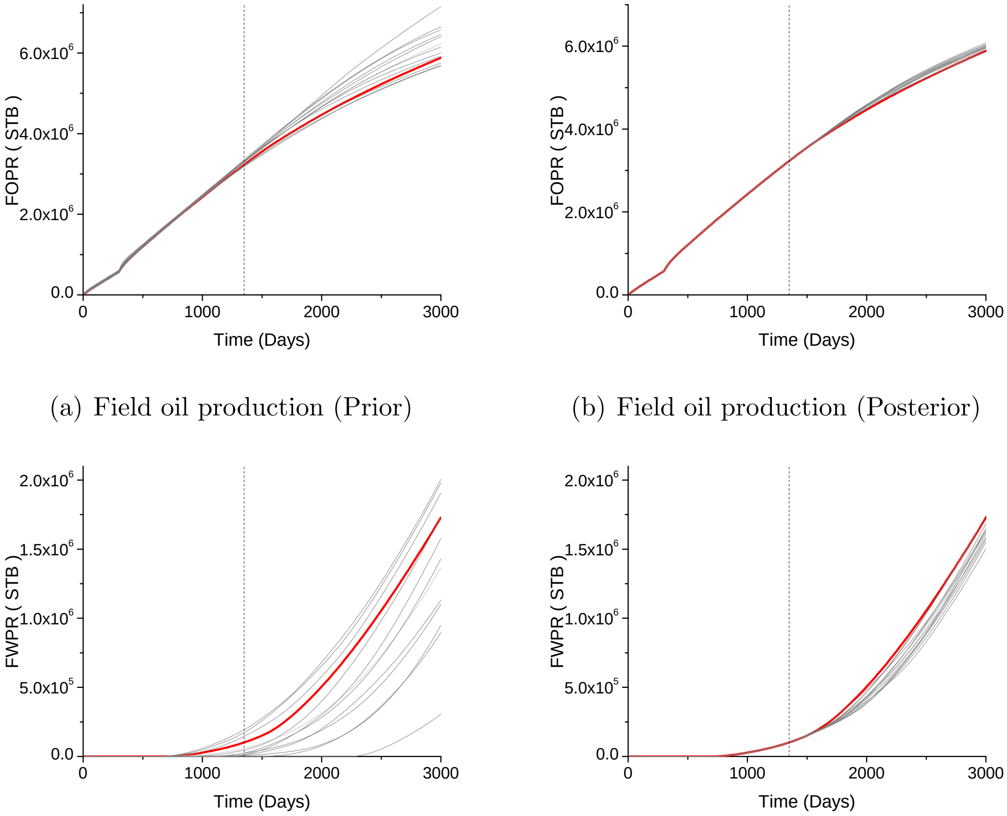
(j) Posterior q_w of P-6

(k) Posterior q_o of P-3

(l) Posterior q_o of P-6

Figure 4.34: Data matches and predictions for some of the wells from the posterior and prior samples. The dashed vertical line shows the end of history matching (1350 days), Example 3.

is highly decreased, however the true prediction for water production is very close to the upper bound and for oil production is almost at lower bound. This may be due the fact that the number of realizations generated, is not enough to perform reliable predictions. In addition, only 2 of the 6 producers had water breakthrough before the end of history matching period. Because of high uncertainty in the breakthrough time of the other 4 producers, accurate predictions of water rate is difficult.



(a) Field oil production (Prior)

(b) Field oil production (Posterior)

(c) Field water production (Prior)

(d) Field water production (Posterior)

Figure 4.35: Field oil and water production from the prior and the posterior samples. Red curve shows the truth, gray curves are from 11 realizations. The dashed vertical line shows the end of history matching (1350 days), Example 3.

We also applied the SVD-EnRML-MI algorithm with a square root approximation of C_M, as explained in Section 5.2. We use a high number of prior realizations, $N_s = 2500$, in the approximation of $C_M^{1/2}$. We applied the algorithm with $N_{\text{SVD,max}} = 50$. The values of O_N versus iterations are given in Fig. 4.36. As one can see in this figure, the algorithm converged in 32 iterations and at convergence, all of the realizations have high values of O_N. Since we used a high number of prior realizations in the approximation of $C_M^{1/2}$, and a high number of singular triplets ($N_{\text{SVD,max}} = 50$), we do not expect this behavior. We noticed that the slow rate of convergence and the high values of normalized objective functions are due to introduction of correlation between layers. By grouping the realizations of rock property fields of different layers, into a single column vector, spurious cross-correlations between layers is introduced; as a result the rock property fields of each layer is forced to have correlations with those of the neighboring layers.

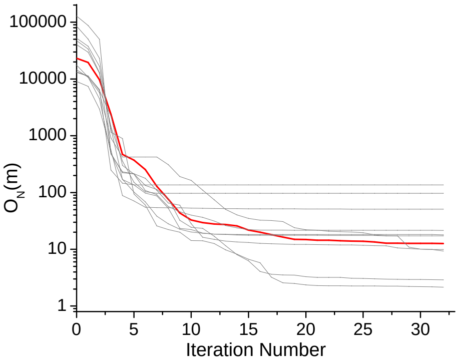
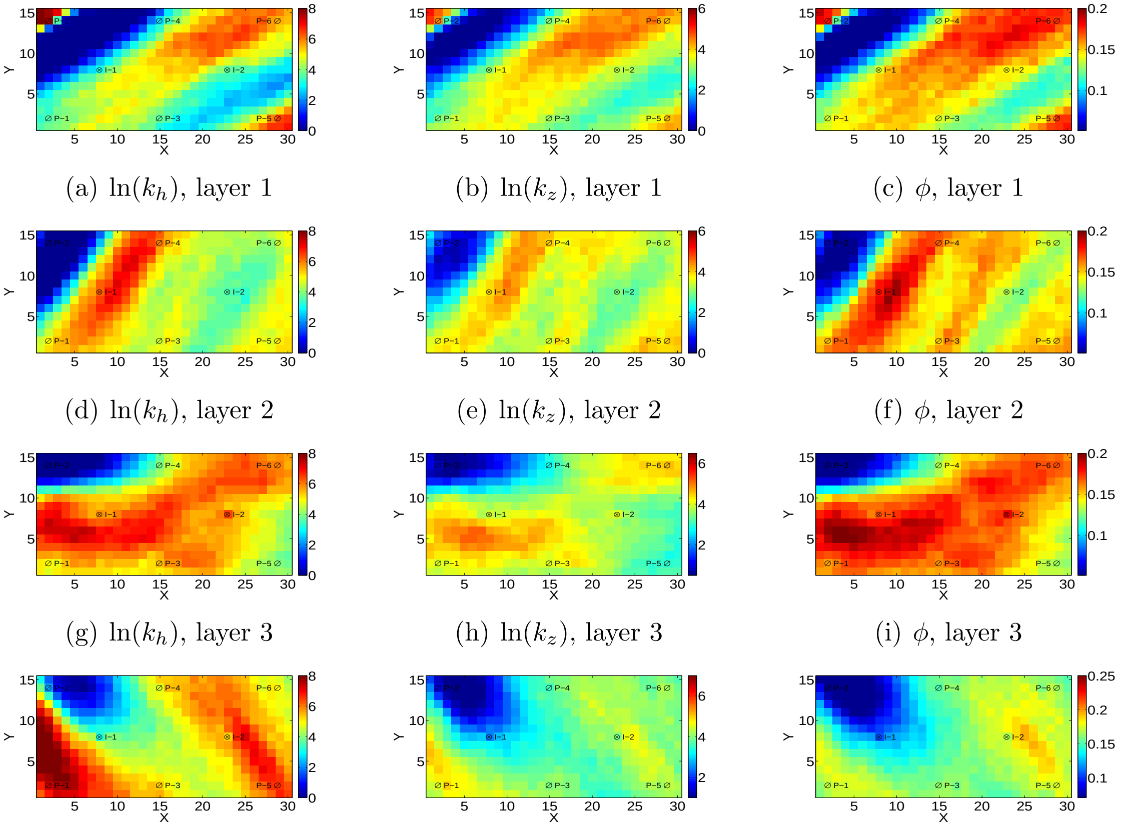


Figure 4.36: $O_N(m)$ of the MAP estimate (red) and 10 RML realizations (gray) versus iterations, l, of SVD-EnRML-MI with $N_{\text{SVD,max}} = 50$ and square root approximation of C_M, Example 3.

Fig. 4.37 shows the rock property fields of the MAP estimate generated in this part. The normalized objective function of the MAP estimate is 12.6, which is high. By comparing each of the $\ln(k_h)$, $\ln(k_z)$ and porosity fields of different layers in Fig. 4.37, one can see the introduction of cross-correlation between layers. The true rock property fields of the first layer in Figs. 4.27(a), 4.28(a) and 4.29(a) have a low permeability and low porosity region on the North-West of the field. However, in the North-West of the other layers of the truth, this region is different, with higher values of permeability and porosity. As one can see in Fig. 4.37, the North-West corner of all layers have a low porosity and low permeability, which is a result of introducing spurious cross-correlations between layers.

According to the results of this example, with retaining 50 singular triplets of $G_{D,l}$ at late iterations of the SVD-EnRML-MI algorithm, we were able to decrease the normalized objective functions of all realizations to small values; however with 40 singular triplets, the values of the normalized objective functions of realizations were high at convergence. The 40th and 50th singular values of $G_{D,l}$, were both close to 1.

The results of this example also show that using the square root approximation of C_M, in case that the layers are uncorrelated, can introduce spurious cross-correlation between layers. In order to avoid this issue, one should use the reordering of model parameters, so that the correlated parameters are grouped together, and then use a square root approximation of C_M for each group of the correlated parameters, separately.



(a) $\ln(k_h)$, layer 1

(b) $\ln(k_z)$, layer 1

(c) ϕ, layer 1

(d) $\ln(k_h)$, layer 2

(e) $\ln(k_z)$, layer 2

(f) ϕ, layer 2

(g) $\ln(k_h)$, layer 3

(h) $\ln(k_z)$, layer 3

(i) ϕ, layer 3

(j) $\ln(k_h)$, layer 4

(k) $\ln(k_z)$, layer 4

(l) ϕ, layer 4

Figure 4.37: Rock property fields of the MAP estimate from SVD-EnRML-MI with a square root approximation of C_M. Left column shows the $\ln(k_h)$ field, the middle column is the $\ln(k_z)$ field and the right column is the porosity field, Example 3.

CHAPTER 5

USING SVD PARAMETRIZATION TO ESTIMATE NON-GAUSSIAN MODELS

In inverse problems, the objective function that is minimized to either find an estimate of the model or to sample the a posteriori pdf, can have two main parts, the data mismatch part and a regularization term. The regularization term usually contains prior information about the model, or it imposes smoothness conditions.

In history matching problems, the objective function is usually written as

$$O(m) = O_m(m) + O_d(m), \tag{5.1}$$

where $O_m(m)$ is called the model mismatch term, and $O_d(m)$ is the data mismatch term. Minimizing only $O_d(m)$, will provide an estimate which is called the maximum likelihood estimate. Minimizing $O(m)$ in Eq. 5.1 provides an estimate which is the most probable model and is referred to as the maximum a posteriori (MAP) estimate.

In Bayesian estimation, $O_m(m)$ is included in the objective function, and one tends to preserve the prior geologic information that was used to build the prior model. If the prior model is Gaussian and the mean and the covariance matrix are available, $O_m(m)$ can be written as

$$O_m(m) = \frac{1}{2}(m - m_{\text{prior}})^T C_M^{-1} (m - m_{\text{prior}}). \tag{5.2}$$

However, this is not always the case, since the prior pdf is not always Gaussian. An important example is when the true model has features in different directions. The other case, is when there is uncertainty in prior information, e.g., the model may have channels in an angle between 20^o and 60^o with the x axis, or the correlation lengths are uncertain.

5.1 First Order and Second Order Regularization Terms

In the discrete inverse problems, difference operators replace derivatives. Consider $(N_m - 1) \times N_m$ matrix D and $(N_m - 2) \times N_m$ matrix W for the first order and second order regularization, respectively. For a 1-dimensional problem, the matrix D has the following structure:

$$D = \begin{bmatrix} -1 & 1 & 0 & 0 & \dots & 0 \\ 0 & -1 & 1 & 0 & \dots & 0 \\ 0 & 0 & -1 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & -1 & 1 \end{bmatrix}. \tag{5.3}$$

Letting $d_{i,j}$ denote the element in the ith row and jth column of D, $d_{i,i} = -1$ and $d_{i,i+1} = 1$ for $i = 1, 2, \cdots, N_m - 1$, with all other entries of D equal to zero.

For a 1-dimensional problem, the matrix W for the second order regularization, has the following form:

$$W = \begin{bmatrix} -1 & 2 & -1 & 0 & \dots & 0 \\ 0 & -1 & 2 & -1 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & -1 & 2 & -1 \end{bmatrix}. \tag{5.4}$$

With the first order regularization, the model misfit term is defined as

$$O_m(m) = \frac{1}{2}(m - m_{\text{prior}})^T D^T D(m - m_{\text{prior}}). \tag{5.5}$$

Although $D^T D$ or $W^T W$ are positive semi-definite, they may not be positive definite; since their inverse is required, an identity matrix is also added to them. In this case, the model mismatch term in Eq. 5.5 changes to the following equation:

$$\begin{aligned} O_m(m) &= \frac{1}{2}(m - m_{\text{prior}})^T D^T D (m - m_{\text{prior}}) + \frac{1}{2}(m - m_{\text{prior}})^T (m - m_{\text{prior}}) \\ &= \frac{1}{2}(m - m_{\text{prior}})^T (D^T D + I)(m - m_{\text{prior}}), \end{aligned} \quad (5.6)$$

where I is the identity matrix of dimension N_m. Comparing Eqs. 5.2 and 5.6, one can obtain

$$C_M^{-1} = D^T D + I. \quad (5.7)$$

In order to calculate C_M and $C_M^{1/2}$, one can take the SVD of $D^T D + I$. As the matrix $D^T D + I$ is real symmetric positive definite, the right singular vectors are the same as the left singular vectors, i.e.,

$$C_M^{-1} = D^T D + I = U \Lambda U^T, \quad (5.8)$$

$$C_M = U \Lambda^{-1} U^T, \quad (5.9)$$

$$C_M^{1/2} = U \Lambda^{-1/2} U^T. \quad (5.10)$$

One may also use the Cholseky decomposition instead of the square root obtained by SVD. In this case, after calculating C_M^{-1} with Eq. 5.7, C_M is obtained by Eq. 5.9, and then Cholesky decomposition of C_M is computed as $C_M = LL^T$. In the example presented here, for the first and second order regularization, the Cholesky decomposition is used.

Similar to Eq. 5.6, the second order regularization term can be written as

$$O_m(m) = \frac{1}{2}(m - m_{\text{prior}})^T (W^T W + I)(m - m_{\text{prior}}), \quad (5.11)$$

where I is the identity matrix of dimension N_m. Similar to Eq. 5.9,

$$C_M^{-1} = W^T W + I = U \Lambda U^T. \quad (5.12)$$

By using Eq. 5.9, C_M is obtained and its Cholesky decomposition can be computed.

5.2 SVD Parameterization Using a Square Root Approximation of C_M

In order to use SVD parameterization to generate an estimate of the model, one needs to have a prior mean and covariance matrix; a covariance matrix is needed because a truncated SVD of the dimensionless sensitivity matrix, $G_{D,l}$, is computed at each iteration of algorithm which requires $C_M^{1/2}$. In addition, a transformed vector of model parameters is used, which requires having $C_M^{1/2}$.

Suppose there is uncertainty in the prior model of log permeability field and instead of having a specified C_M, there exists N_s realizations all of which may not be from a single Gaussian distribution. In order to use the SVD parameterization method, we still need to have an expression for $C_M^{1/2}$.

Here, the following method is used to find an appropriate square root of the covariance matrix which contains the information of all available prior realizations. Consider a $N_m \times N_s$ matrix M where N_m is the number of model parameters and N_s is the number of prior realizations,

$$M = \begin{pmatrix} m_1 & m_2 & ... & m_{N_s-1} & m_{N_s} \end{pmatrix}, \tag{5.13}$$

where each m_j, $j = 1, 2, ...N_s$ is an N_m-dimensional column vector which is a prior realization of log permeability field. The covariance matrix, C_M, can be approximated with the following expression:

$$C_M \cong \frac{1}{N_s - 1} \sum_{j=1}^{j=N_s} (m_j - \bar{m})(m_j - \bar{m})^T = \frac{1}{N_s - 1}(M - \bar{M})(M - \bar{M})^T, \tag{5.14}$$

where $\bar{M}$ is a matrix with the same dimension as M and all columns of $\bar{M}$ are equal to $\bar{m}$; $\bar{m}$ is the average of N_s prior realizations, i.e,

$$\bar{m} = \frac{1}{N_s} \sum_{j=1}^{j=N_s} m_j. \tag{5.15}$$

Thus, C_M contains information from all prior models. The SVD of $M - \bar{M}$ is written as

$$M - \bar{M} = U\Lambda V^T, \tag{5.16}$$

where U is the $N_m \times N_m$ matrix of left singular vectors, Λ is the $N_m \times N_s$ matrix of singular values and V is the $N_s \times N_s$ matrix of right singular vectors.

Since some of the singular values may be very small or zero, we use a truncated SVD of $M - \bar{M}$, by preserving a portion of the total energy. In our implementation, we use a truncated SVD based on 0.9999 of the total energy, i.e., we kept p singular triplets, where p is the smallest number such that

$$\sum_{j=1}^{p} \lambda_j \geq 0.9999 \sum_{j=1}^{N_s} \lambda_j. \tag{5.17}$$

The truncated SVD of $M - \bar{M}$ can be written as

$$M - \bar{M} \approx U_p \Lambda_p V_p^T, \tag{5.18}$$

where U_p is a $N_m \times p$ matrix of left singular vectors, Λ_p is the $p \times p$ diagonal matrix of p largest singular values retained and V is the corresponding $p \times N_s$ matrix of right singular vectors.

Thus C_M can be approximated by:

$$\begin{aligned} C_M &\approx \frac{1}{N_s - 1}(M - \bar{M})(M - \bar{M})^T \approx \frac{1}{N_s - 1}(U_p\Lambda_p V_p^T)(U_p\Lambda_p V_p^T)^T \\ &= \frac{1}{N_s - 1}U_p\Lambda_p V_p^T V_p \Lambda_p U_p^T = \frac{1}{N_s - 1}U_p\Lambda_p\Lambda_p U_p^T = \frac{1}{N_s - 1}U_p\Lambda_p U_p^T U_p \Lambda_p U_p^T \\ &= \frac{1}{N_s - 1}(U_p\Lambda_p U_p^T)(U_p\Lambda_p U_p^T) = (\frac{1}{\sqrt{N_s - 1}}U_p\Lambda_p U_p^T)^2 \end{aligned} \tag{5.19}$$

Consequently, $C_M^{1/2}$ can be approximated by the following equation:

$$C_M^{1/2} = \frac{1}{\sqrt{N_s - 1}}U_p\Lambda_p U_p^T, \tag{5.20}$$

with its pseudo inverse, $C_M^{-1/2}$, is given by

$$C_M^{-1/2} = \sqrt{N_s - 1} U_p \Lambda_p^{-1} U_p^T. \tag{5.21}$$

5.2.1 Calculating the Value of Model Mismatch Term

By using the square root formula, the model mismatch term can simply be approximated by using the following equation:

$$\begin{aligned} O_m(m) &= \frac{N_s - 1}{2}(m - m_{\text{prior}})^T U_p \Lambda_p^{-1} \Lambda_p^{-1} U_p^T (m - m_{\text{prior}}) \\ &= \frac{N_s - 1}{2}((m - m_{\text{prior}})^T U_p \Lambda_p^{-1})((m - m_{\text{prior}})^T U_p \Lambda_p^{-1})^T \\ &= \frac{N_s - 1}{2} \sum_{j=1}^{p} ((m - m_{\text{prior}})^T u_j / \lambda_j)^2 \end{aligned} \tag{5.22}$$

5.2.2 G_D Times a Vector

To calculate G_D times a vector, the gradient simulator method is used. As $G_D = C_D^{-1/2} G C_M^{1/2}$, to calculate $G_D \times v$, at first the product of $w = C_M^{1/2} \times v$ is calculated and then Gw is computed with the gradient simulator method; then the result is left multiplied by $C_D^{-1/2}$.

If the square root formula in Eq. 5.20 is used for $C_M^{1/2}$, this matrix is not required to be explicitly computed or stored. The product of $w = C_M^{1/2} \times v$ is computed as follows:

$$C_M^{1/2} \times v = \frac{1}{\sqrt{N_s - 1}} U_p \Lambda_p U_p^T \times v = \frac{1}{\sqrt{N_s - 1}} U_p \Lambda_p (U_p^T \times v), \tag{5.23}$$

$$C_M^{1/2} \times v = \frac{1}{\sqrt{N_s - 1}} \sum_{j=1}^{p} (\lambda_j u_j^T v) u_j \equiv \frac{1}{\sqrt{N_s - 1}} U_p \beta = \frac{1}{\sqrt{N_s - 1}} \sum_{j=1}^{p} b_j u_j, \tag{5.24}$$

where u_j is an N_m-dimensional vector which is the j^{th} column of U and β is a p-dimensional column vector:

$$\beta = [\lambda_1 u_1^T v, \lambda_2 u_2^T v, ..., \lambda_p u_p^T v]^T. \tag{5.25}$$

5.2.3 G_D^T Times a Vector

To calculate G_D^T times a vector, the adjoint method is used. As $G_D^T = C_M^{T/2} G^T C_D^{-T/2}$, to calculate $G_D^T \times y$, at first the product of G^T times $C_D^{-T/2} y$ is calculated with the adjoint method; then the result is left multiplied by $C_M^{T/2}$.

If the square root formula in Eq. 5.20 is used for $C_M^{1/2}$, this matrix is not required to be explicitly computed or stored. If $x = G^T C_D^{-T/2} y$, the product of $C_M^{T/2} x$ is computed as follows:

$$C_M^{T/2} \times x = \frac{1}{\sqrt{N_s - 1}} U_p \Lambda_p U_p^T \times x = \frac{1}{\sqrt{N_s - 1}} U_p \Lambda_p (U_p^T \times x), \tag{5.26}$$

$$C_M^{T/2} \times x = \frac{1}{\sqrt{N_s - 1}} U_p \beta = \frac{1}{\sqrt{N_s - 1}} \sum_{j=1}^{p} (\lambda_j u_j^T x) u_j = \frac{1}{\sqrt{N_s - 1}} \sum_{j=1}^{p} b_j u_j, \tag{5.27}$$

where u_j is an N_m dimensional vector which is the j^{th} column of U; β is a N dimensional column vector:

$$\beta = \Lambda_p U_p^T \times x = [\lambda_1 u_1^T x, \lambda_2 u_2^T x, ..., \lambda_p u_p^T x]^T. \tag{5.28}$$

5.2.4 Calculation of δm from $\delta \tilde{m}$

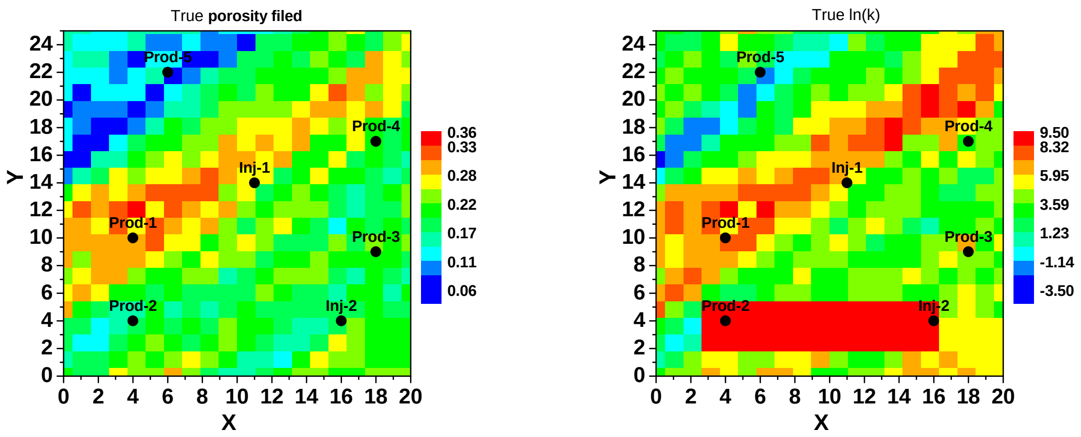
In SVD parameterization algorithm, the minimization is performed in a transformed space; δm is the search direction vector is the original space, and $\delta \tilde{m}$ is the search direction vector in the transformed space. If the square root formula in Eq. 5.20 is used for $C_M^{1/2}$, δm is computed using the following equation:

$$\delta m = C_M^{1/2} \delta \tilde{m} = \frac{1}{\sqrt{N_s - 1}} U_p \Lambda_p U_p^T \times \delta \tilde{m} = \frac{1}{\sqrt{N_s - 1}} U_p \Lambda_p (U_p^T \times \delta \tilde{m}), \tag{5.29}$$

where the product of $U_p^T \times \delta \tilde{m}$ only requires calculating inner products of $\delta \tilde{m}$ and columns of U_p.

5.3 Example 1

This example pertains to a two-dimensional horizontal reservoir model with 20 × 25 uniform grid. True porosity and log permeability fields are shown in Fig. 5.1. Note



(a) True porosity

(b) True log $\ln(k)$

Figure 5.1: True porosity and log permeability fields.

that the horizontal channel was simply added by changing the log permeability of the blocks to 9. In this example, the objective is to generate an estimate of the log permeability field, while the porosity field is assumed to be known.

The key geostatistical parameters used to generate the true model without the horizontal channel are listed in Table 5.1. The prior mean of the vector of model parameters which contains horizontal log-permeability of gridblocks is uniform and equal to 4.5.

Table 5.1: Geostatistical parameters of Example 1.

Parameters	Values
φ_{mean}	0.2
$[\ln(k)]_{\text{mean}}$	4.50
σ_{φ}	0.05
$\sigma_{\ln(k)}$	2.0
$\rho_{\varphi,\ln(\text{k})}$	0.80
α	40^o
r_1	$28\triangle x$
r_2	$5\triangle x$

The gridblock dimensions are:

$$\triangle x = \triangle y = 300\,\text{ft},\ \triangle z = 10\,\text{ft}.$$

The initial reservoir pressure is 4800 psi. Initially the reservoir is at irreducible water saturation. There are 5 producers and 2 injectors in this reservoir. The total history matching period is 1650 days. At 1650 days, Prod-2, Prod-3 and Prod-4 have experienced water breakthrough. Observed data include the flowing bottom hole pressure (BHP) of all wells and water rates of all producers at 30 day intervals.

Synthetic observed data are generated by adding Gaussian random noise to the true data, where the true data are the simulator output when it is run with the true model. The standard deviation of noise (measurement error) is 2% of rates for water rate data and 1.4 psi for pressure data, i.e., $\sigma_{\text{BHP}} = 1.4$ psi and $\sigma_{q_w} = 0.02q_w$. The minimum measurement error for water rate is specified to 0.5 STB/D while the maximum measurement error is 3 STB/D.

Table 5.2 shows the summary of well controls. At 2400 days, Prod-2 and Prod-4 are shut in due to high water cut.

Table 5.2: Well controls of Example 1. Total liquid rate is specified at producers and injection rate is specified at injectors. Time period is in days and the rates are in STB/D.

Time Period	Inj-1	Inj-2	Prod-1	Prod-2	Prod-3	Prod-4	Prod-5
0 – 600	500	500	200	200	200	200	200
600 – 990	750	750	300	350	250	350	250
990 – 2400	700	700	350	200	300	250	300
2400 – 3300	500	500	300	0	400	0	300

Model parameters include log permeability of all gridblocks. The porosity field and fluid properties are assumed to be known. The number of model parameters and observed data are:

$$N_m = N_x \times N_y = 500,\ N_{\text{d}} = 660.$$

We use the SVD parameterization method to estimate the log permeability field by generating the MAP estimate. Suppose it is known that the log permeability field has 2 possible features, which are horizontal channels and features in a direction which makes a 40^o angle with x axis (corresponding to Table 5.1). There is no information about the location of channels. What exists as prior information is 400 realizations from two different Gaussian distributions, one corresponding to Table 5.1 and the other one with features in horizontal direction. In order to find an appropriate $C_M^{1/2}$ and $C_M^{-1/2}$ to use in the SVD parameterization algorithm, the square root method is used. The covariance matrix from which the true model (without the horizontal channel) was generated is referred to as C_{M_1}. A spherical covariance matrix is generated with a correlation length of $16\triangle(x)$ in principal direction which is aligned with the x axis, and a correlation length of $4\triangle(x)$ in the perpendicular direction and with $\sigma_{\ln(k)} = 2.0$. This covariance matrix is referred to as C_{M_2}. Table 5.3 shows the summary of geostatistical parameters used in generating realizations from $N(4.5, C_{M_2})$.

Table 5.3: Geostatistical parameters of $N(\bar{m}, C_{M_2})$

Parameters	Values
$[\ln(k)]_{\text{mean}}$	4.50
$\sigma_{\ln(k)}$	2.0
α	0^o
r_1	$16\triangle x$
r_2	$4\triangle x$

Two hundred realizations are generated from $N(\bar{m}, C_{M_1})$ and another two hundred are generated from $N(\bar{m}, C_{M_2})$, where $\bar{m}$ is a uniform vector with all entries equal to 4.5. Thus the matrix M, given by Eq. 5.13, contains 200 realizations from $N(4.5, C_{M_1})$ and 200 realizations from $N(4.5, C_{M_2})$. By using Eqs. 5.20 and 5.21, $C_M^{1/2}$ and $C_M^{-1/2}$ are calculated. Note that in this case $N_s = 400$.

We applied the SVD parameterization algorithm with $N_{\text{SVD,max}} = 45$, $\mu_1 = 0.5$, $\mu_2 = 0.05$, $\mu_{\min} = 0.0003$. The algorithm converged in 28 iterations with $O_N(m) = 1.265$ which is less than $1 + 5\sqrt{2/N_D} = 1.275$. The MAP estimate is shown in Fig. 5.3(b).

We also used the SVD parameterization with other regularization terms. We applied the algorithm for 3 cases, using first order regularization term, second order regularization term and ‘first and second regularization term’. For the first order regularization term, the model mismatch term given by Eq. 5.6 is used. This regularization term is simply referred to as $C_M^{-1} = I - \triangle$. For the second order regularization term, the model mismatch term given by Eq. 5.11 is used. In the third case (‘first and second regularization term’), both regularization terms were used with the following model mismatch term:

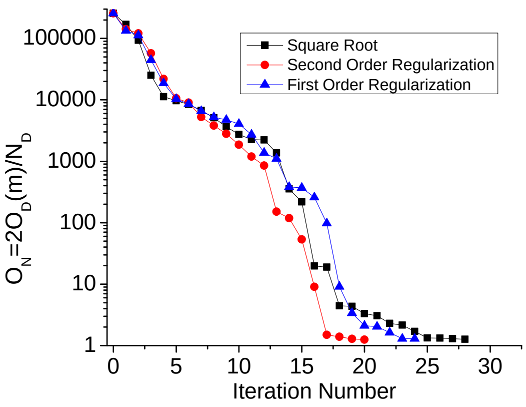
$$O_m(m) = \frac{1}{2}(m - m_{\text{prior}})^T (D^T D + W^T W + I)(m - m_{\text{prior}}). \tag{5.30}$$

In all three cases, the Cholesky decomposition of C_M was used for SVD parameterization. We applied the algorithm with $N_{\text{SVD,max}} = 45$. Sv-cut was used to control the number of parameters, beginning with 0.5 at the first iteration and dividing this number by 2 at each subsequent iteration. The computational costs and final values of normalized objective function based on data mismatch term, are shown in Table 5.4. The MAP estimates obtained with different regularization terms are shown in Fig. 5.3.

Table 5.4: Computational costs of SVD-EnRML algorithm with different regularization terms to generate an estimate of the log permeability field.

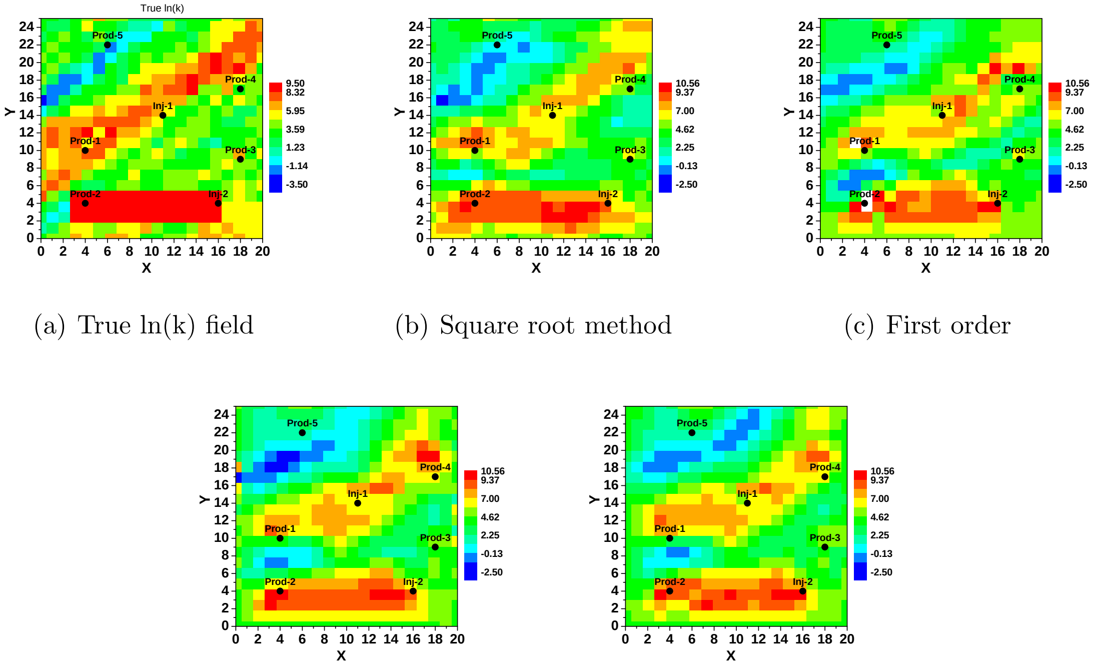
Regu. Term	Sim.	Direct	Adjoint	Equ. Sim. Runs	N_{iter}	$2O_d(m)/N_d$
Square Root	78	1238	1210	690	28	1.263
First Order	57	1003	1006	566	24	1.28
Second Order	49	788	768	438	20	1.245
First+Second	70	1326	1297	725	29	1.852

With the first order regularization term, a couple of gridblocks have unreasonably high values of log permeability. The gridblock of Prod-2 has a ln(k) of 12.5, which is relatively high. Although not shown, the model mismatch term at convergence in this case is noticeably high (1991), compared to data mismatch term which is 422.5. This is not a problem, since a true covariance matrix is not used to calculate the model



(a)

Figure 5.2: The normalized data mismatch term versus iterations of SVD parameterization algorithm with different regularization terms.



(a) True ln(k) field

(b) Square root method

(c) First order

(d) Second order

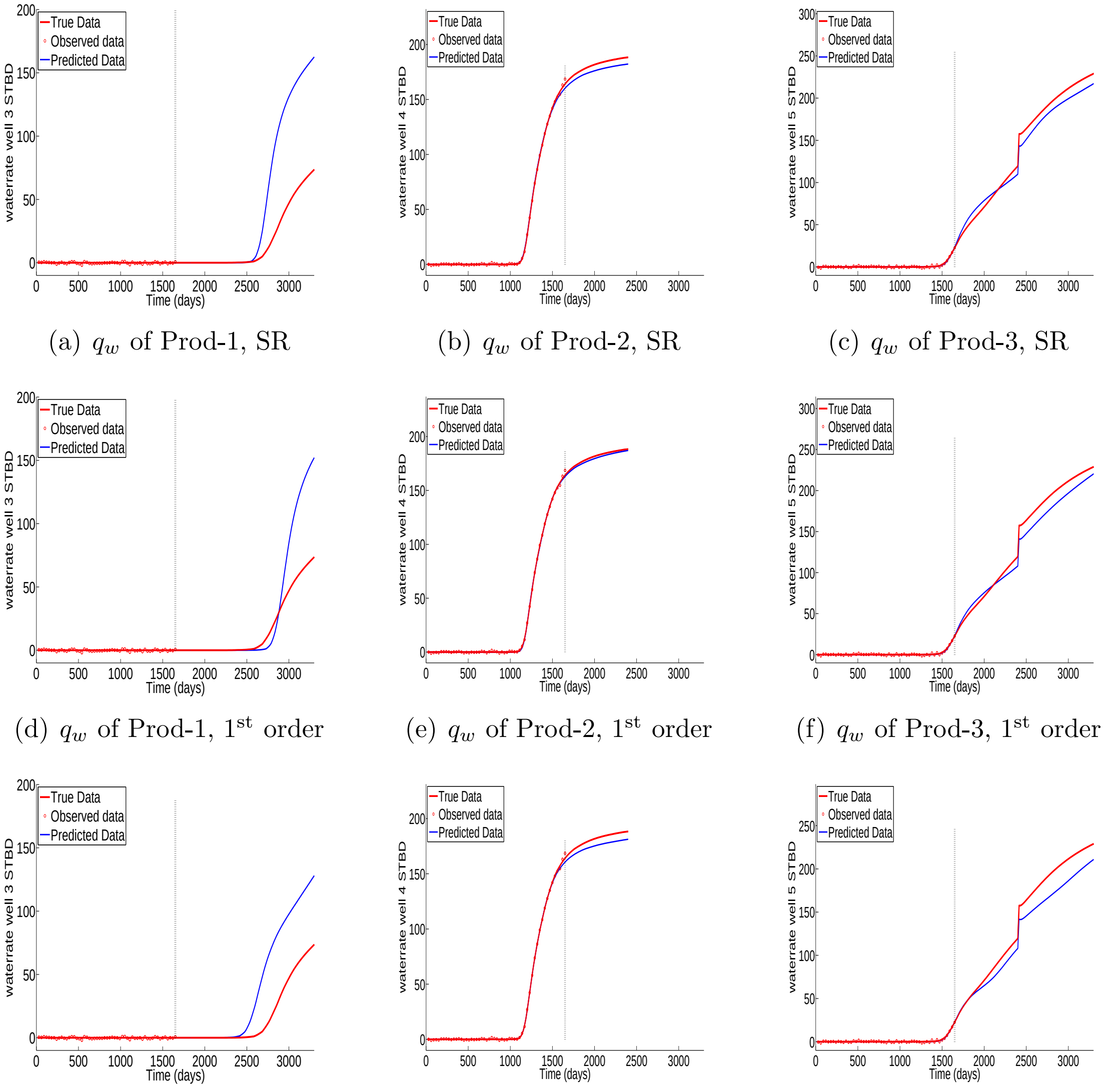
(e) First order and second order

Figure 5.3: ln(k) field of the MAP estimate, with SVD parameterization and different regularization terms.

mismatch term; in addition, we should compute the normalized objective function based on data mismatch term, as the model mismatch term is not realistic. The value of model mismatch term at convergence when using Eq. 5.30 as the regularization term, is 2833.

In Fig. 5.3, by comparing the MAP estimate obtained with different regularization term and the true model, one can see that the MAP estimate obtained with all regularization terms has captured some main features of the truth.

Data matches and predictions for the 3 cases are shown in Figs. 5.4, 5.5, 5.6 and 5.7. The data matches and predictions of p_{wf} with the second order regularization is better than other cases, however the predictions of water rates in all 3 cases show discrepancies with the true predictions.



(a) q_w of Prod-1, SR

(b) q_w of Prod-2, SR

(c) q_w of Prod-3, SR

(d) q_w of Prod-1, 1^{st} order

(e) q_w of Prod-2, 1^{st} order

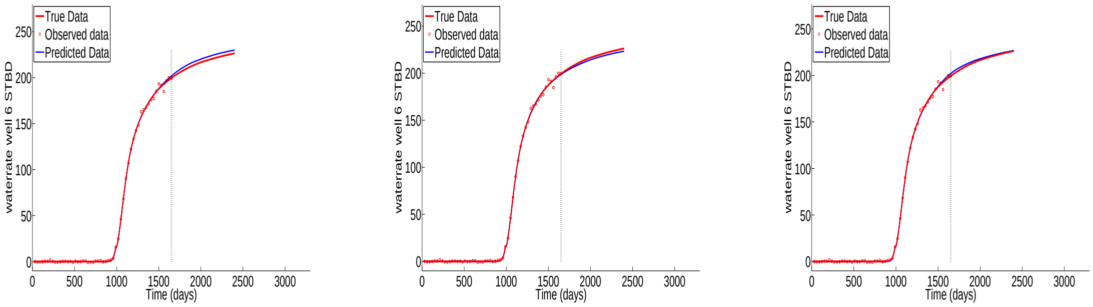
(f) q_w of Prod-3, 1^{st} order

(g) q_w of Prod-1, 2^{nd} order

(h) q_w of Prod-2, 2^{nd} order

(i) q_w of Prod-3, 2^{nd} order

Figure 5.4: Data matches and predictions of water rates of 3 of the producers, first row is from the square root method, second row is from the first order regularization and third row is from the second order regularization. The dashed vertical line shows the end of history matching (1650 days).

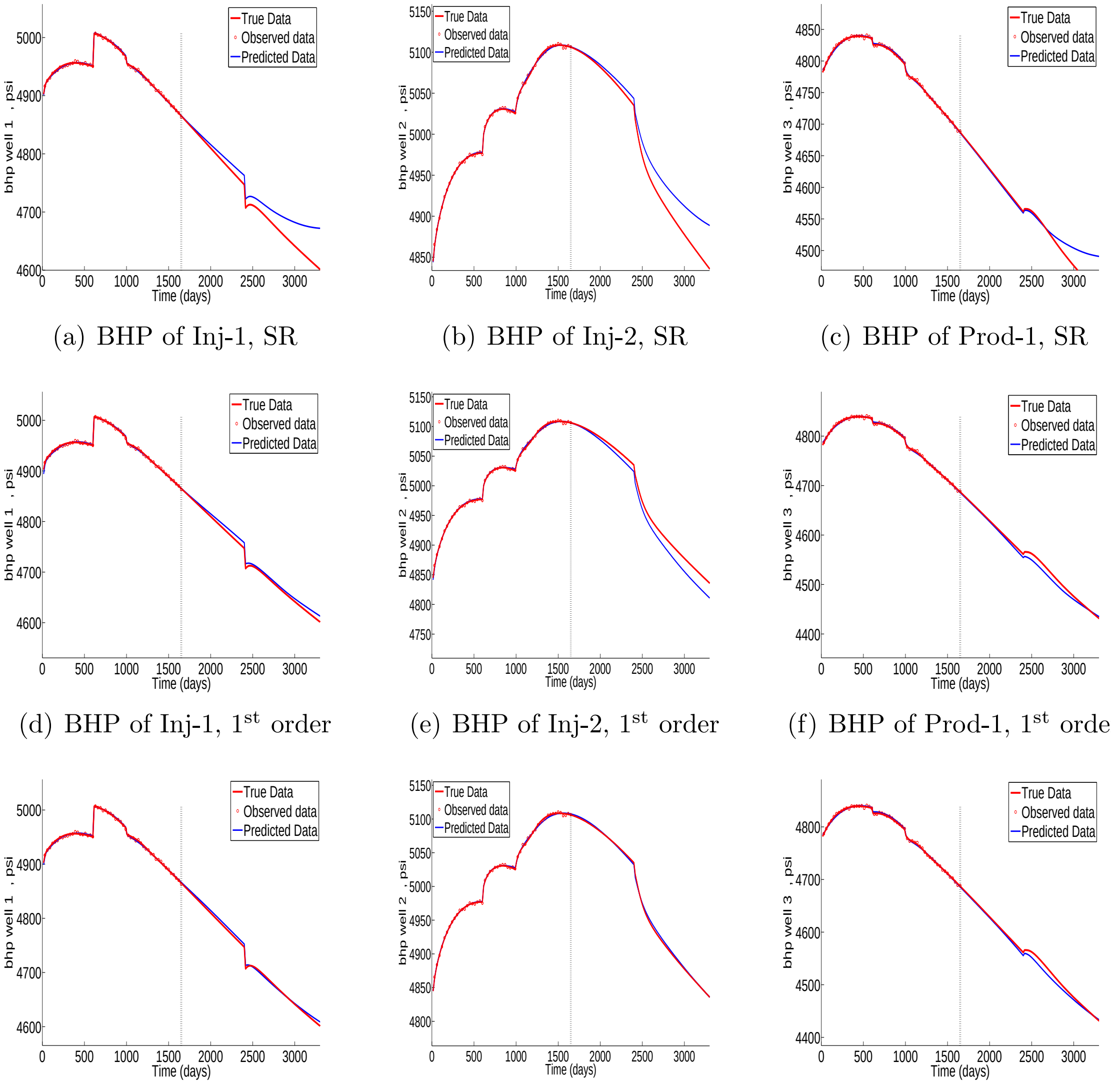


(a) q_w of Prod-4, SR

(b) q_w of Prod-4, 1^{st} order

(c) q_w of Prod-4, 2^{nd} order

Figure 5.5: Data matches and predictions of water rates of Prod-4, the left one is from the square root method, the middle is from the first order regularization and the right one is from the second order regularization. The dashed vertical line shows the end of history matching (1650 days).



(a) BHP of Inj-1, SR

(b) BHP of Inj-2, SR

(c) BHP of Prod-1, SR

(d) BHP of Inj-1, 1^{st} order

(e) BHP of Inj-2, 1^{st} order

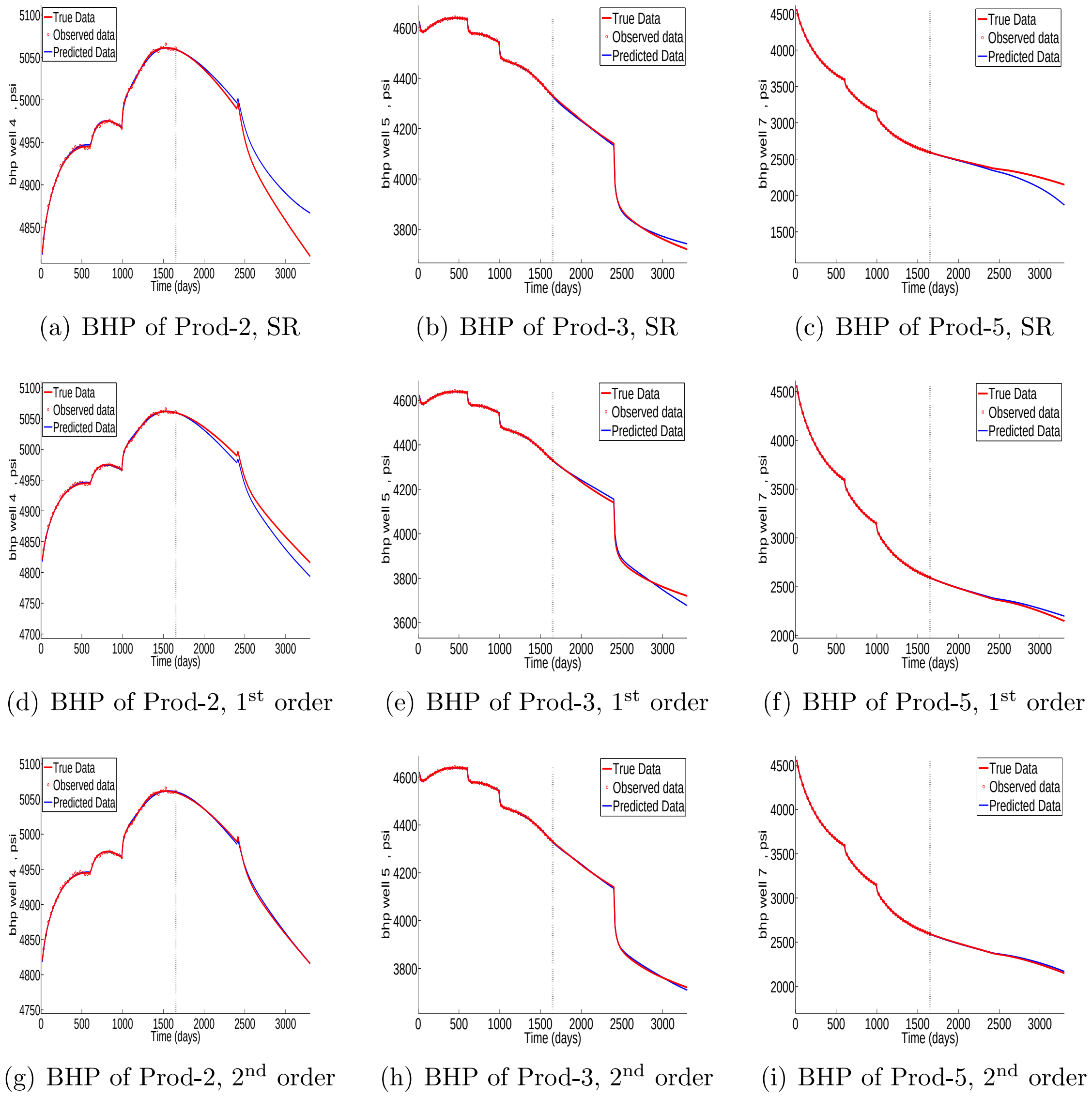
(f) BHP of Prod-1, 1^{st} order

(g) BHP of Inj-1, 2^{nd} order

(h) BHP of Inj-2, 2^{nd} order

(i) BHP of Prod-1, 2^{nd} order

Figure 5.6: Data matches and predictions of p_{wf} of 3 of the wells, first row is from the square root method, second row is from the first order regularization and third row is from the second order regularization. The dashed vertical line shows the end of history matching (1650 days).



(a) BHP of Prod-2, SR

(b) BHP of Prod-3, SR

(c) BHP of Prod-5, SR

(d) BHP of Prod-2, 1st order

(e) BHP of Prod-3, 1st order

(f) BHP of Prod-5, 1st order

(g) BHP of Prod-2, 2nd order

(h) BHP of Prod-3, 2nd order

(i) BHP of Prod-5, 2nd order

Figure 5.7: Data matches and predictions of p_{wf} of 3 of the producers, first row is from the square root method, second row is from the first order regularization and third row is from the second order regularization. The dashed vertical line shows the end of history matching (1650 days).

CHAPTER 6

COMMENTS AND CONCLUSIONS

A direct application of gradient based history matching algorithms, i.e., the Gauss-Newton (GN) and Levenberg-Marquardt (LM) algorithms, requires forming the whole sensitivity matrix at each iteration, which is computationally expensive. In order to avoid forming the sensitivity matrix, one can apply the GN or LM algorithm after parameterizing the vector of the change in model parameters in terms of the right singular vectors of a dimensionless sensitivity matrix, $G_{D,l}$. With this parameterization, at each iteration of the GN or LM algorithm, only a truncated SVD of $G_{D,l}$ should be computed. The Lanczos method is used to compute a truncated SVD of the dimensionless sensitivity matrix. For computing a truncated SVD of this matrix, the Lanczos method only requires the products of G times a vectors and G^T times a vector; these products are calculated by the gradient simulator method and adjoint method, respectively.

The SVD parameterization algorithm for generating an estimate of the reservoir model parameters is a computationally efficient method. To characterize the uncertainty in performance predictions, one should generate multiple realizations from the a posteriori pdf. Generating multiple realizations, can be performed by using the RML method. One can generate N_e RML realizations, through N_e applications of the SVD parameterization algorithm. However, as we showed in Example 1-4 of Chapter 2, generating realizations one by one, is computationally expensive. With the SVD-EnRML algorithms proposed by Tavakoli and Reynolds [39], one can simultaneously generate the MAP estimate and N_e RML realizations, with a low computational cost. In this work, we proposed some modifications to the the SVD-EnRML algorithm; the modified algorithm is computationally more efficient than the previous SVD-EnRML algorithms. To further improve the computational efficiency, we added an inner loop to the SVD-EnRML algorithm, where in the inner loop iterations, the truncated SVD parameterization is not updated. This modified SVD-EnRML algorithm, is called SVD-EnRML-MI, where MI refers to multiple iterations. We applied this modified algorithm to some 2D and 3D synthetic reservoirs, and in all examples, we were able to obtain conditional rock property fields which resulted in appropriately small values of the relevant objective function and good future predictions.

The main contribution of this work was to investigate the applicability of the SVD parameterization algorithms for history matching production data for 3D reservoirs and to generate conditional realizations of rock property fields. For 3D reservoirs, the size of the covariance matrix can be very large. For applying the SVD parameterization algorithm to 3D reservoirs, we used a square root approximation of the covariance matrix which decreases the size of the matrix. We showed that this approximation can be used only for correlated model parameters. We also used a reordering of model parameters, for the case that the rock properties of some layers of the reservoir are uncorrelated with those of other layers. With reordering of model parameters, the size of the covariance matrix that is saved during a computer run is reduced.

In addition to the modified SVD-EnRML algorithms which are based on the LM method, we also provided a step by step description of SVD-EnRML with the GN algorithm. We applied this algorithm in Example 1 of Chapter 2, and the results showed that by applying the GN algorithm with SVD paramererization, one can efficiently generate the MAP estimate and RML realizations, if the number of SVD parameters are gradually increased.

In SVD-EnRML algorithms, the number of singular triplets or truncation level, p, is gradually increased. The number of retained singular triplets are determined based on the ratio of the smallest retained singular value to the largest singular value. Thus, the algorithm computes p singular triplets, where p is the smallest number such that $\lambda_p/\lambda_1 \leq$ sv-cut. The singular cut off, which is denoted by sv-cut, is an input for the Lanczos algorithm. To gradually increase the number of retained singular triplets, as iteration proceeds, sv-cut is decreased. As shown in the examples of this work, after the first few iterations, the largest singular value of the dimensionless sensitivity matrix, $G_{D,l}$, does not change significantly. The value of the largest singular value at late iterations differs from example to example. We noticed that for a small value of sv-cut (at late iterations), the number of retained singular values can be very sensitive to a small change of sv-cut, i.e., with a small change of sv-cut, the number of singular values may change significantly. In addition, the distribution of the singular values of G_D for each example can be different. Thus, we modified the algorithm, so that the maximum number of SVD parameters, $N_{\text{SVD,max}}$, that the Lanczos method computes, is also specified as input, i.e., the Lanczos method is terminated at either $p = N_{\text{SVD,max}}$ or the smallest value of p such that $\lambda_p/\lambda_1 \leq$ sv-cut, whichever is reached first. We usually tend to set a small number for the final value of sv-cut, e.g., 0.0002, so that at late iterations of the algorithm, the Lanczos method would usually compute the specified number of singular triplets, $N_{\text{SVD,max}}$. We also noticed that to obtain computational efficiency, fast convergence and small values of the objective functions at convergence of the SVD parameterization algorithms, the number of retained singular triplets should not be small. In addition, the smallest retained singular value of the dimensionless sensitivity matrix should not be significantly greater than 1. In the examples, we usually obtained the small values of normalized objective functions that we expected, by retaining 50 to 65 singular triplets at late iterations of the SVD-EnRML-MI algorithm. The smallest retained singular values were close to 1, and in most cases, slightly greater than 1, e.g., 2.

In Chapter 2, we compared the Gauss-Newton and Levenberg-Marquardt algorithms with SVD parameterization. Previously, Tavakoli and Reynolds [38] observed that in the Levenberg-Marquardt algorithm with SVD parameterization, the initial rate of reduction in the objective function is almost independent of the number of retained singular triplets, however they did not provided a reason for this observation. We showed that with a high LM parameter at early iterations, the main components of the search direction are in the direction determined by a few right singular vectors corresponding to the largest singular values, while the right singular vectors corresponding to small singular values have a negligible influence on the search direction. This result is important, because it shows that a direct application of the LM algorithm based on forming the whole sensitivity matrix has no computational merit at least at early iterations. In other words, the search direction at early iterations of the LM algorithm, based on computing only a few singular triplets corresponding to the largest singular values of $G_{D,l}$, is almost the same as the search direction obtained by computing the complete SVD of $G_{D,l}$.

In Chapter 2, we showed that the LM algorithm generates a good estimate of the model, because the algorithm gradually resolves the important features of the model; in particular, at early iterations of the LM algorithm, the search direction is not corrupted by the low frequency and noisy right singular vectors. On the other side, in the Gauss-Newton algorithm with SVD parameterization, the search direction vector at early iterations may have large components in the direction of right singular vectors associated with small singular values. Thus, the GN search direction based on computing many singular triplets at early iteration, changes many parts of the model and roughness is added to the model. Early iterations of a history matching algorithm, are very important because the large data mismatches and the natural ill-posedness of history matching problems may result in convergence to a model that is a poor representation of the reservoir.

We have also compared the convergence properties of the subspace method with SVD parameterization algorithm for generating the MAP estimate. Our results showed that the convergence properties of the two algorithms are very similar, however the SVD parameterization algorithm is computationally more efficient than the subspace method.

BIBLIOGRAPHY

[1] Sigurd I. Aanonsen, Geir Nævdal, Dean S. Oliver, Albert C. Reynolds, and Brice Vallés. Review of ensemble Kalman filter in petroleum engineering. *SPE Journal*, 14(3):393–412, 2009.

[2] Y. Abacioglu, D. S. Oliver, and A. C. Reynolds. Efficient reservoir history matching using subspace vectors. *Computational Geosciences*, 5(2):151–172, 2001.

[3] Yafes Abacioglu. *The Use of Subspace Methods for Efficient Conditioning of Reservoir Models to Production Data*. Ph.D. thesis, The University of Tulsa, Tulsa, Oklahoma, 2001.

[4] F. Anterion, B. Karcher, and R. Eymard. Use of parameter gradients for reservoir history matching. In *Proceedings of the 10th SPE Reservoir Simulation Symposium*, number SPE 18433, pages 339–354, 1989.

[5] Yan Chen, Dean S. Oliver, and Dongxiao Zhang. Data assimilation for nonlinear problems by ensemble Kalman filter with reparameterization. *Journal of Petroleum Science and Engineering*, 66:1–14, 2007.

[6] Lifu Chu, Albert C. Reynolds, and Dean S. Oliver. Reservoir description from static and well-test data using efficient gradient methods. In *Proceedings of the International Meeting on Petroleum Engineering, 14-17 November 1995, Beijing, China*, number SPE 29999, page 16 pages, 1995.

[7] Lifu Chu, M. Komara, and R. A. Schatzinger. An efficient technique for inversion of reservoir properties using iteration method. *SPE Journal*, 5(1):71–81, 2000.

[8] F. Dickstein, P. Goldfeld, G.T. Pfeiffer, E.P.S. Amorim, R.W. dos Santos, and S.G. Gmez. A study of the impact of 4d-seismic data on tsvd-based schemes for

history matching. In *SPE Latin American and Caribbean Petroleum Engineering Conference*, number SPE 138864, 2010.

[9] Alexandre A. Emerick and Albert C. Reynolds. Combining sensitivities and prior information for covariance localization in the ensemble Kalman filter for petroleum reservoir applications. *Computational Geosciences*, Online First, 2010.

[10] Alexandre A. Emerick and Albert 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*, number SPE 141216, 2011.

[11] Alexandre A. Emerick and Albert C. Reynolds. Combining the ensemble kalman filter with markov chain monte carlo for improved history matching and uncertainty characterization. In *Proccedings of the SPE Reservoir Simulation Symposium, The Woodlands, Texas, USA, 21-23 February*, number SPE 141336, 2011.

[12] Geir 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.

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

[14] Guohua Gao, Mohammad Zafari, and Albert C. Reynolds. Quantifying uncertainty for the PUNQ-S3 problem in a Bayesian setting with RML and EnKF. *SPE Journal*, 11(4):506–515, 2006.

[15] G. Gaspari and S. E. Cohn. Construction of correlation functions in two and three dimensions. *Quarterly Journal of the Royal Meteorological Society*, 125(554):723–757, 1999.

[16] G. R. Gavalas, P. C. Shah, and John H. Seinfeld. Reservoir history matching by Bayesian estimation. *SPE Journal*, 16(6):337–350, 1976.

[17] Gene H. Golub and Charles F. van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, second edition, 1989.

[18] P. Jacquard and C. Jain. Permeability distribution from field pressure data. *SPE Journal*, 5(4):281–294, 1965.

[19] Hans O. Jahns. A rapid method for obtaining a two-dimensional reservoir description from well pressure response data. *SPE Journal*, 6(12):315–327, 1966.

[20] Rintu Kalita. *Conditioning a Three Dimensional Reservoir Model to Gas Production Data*. M.S. thesis, The University of Tulsa, Tulsa, Oklahoma, 2000.

[21] Kenneth Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics*, 2:164–168, 1944.

[22] Ruijian Li, A. C. Reynolds, and D. S. Oliver. History matching of three-phase flow production data. *SPE Journal*, 8(4):328–340, 2003.

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

[24] Eliana M. Makhlouf, Wen H. Chen, Mel L. Wasserman, and John H. Seinfeld. A general history matching algorithm for three-phase, three-dimensional petroleum reservoirs. *SPE Advanced Technology Series*, 1(2):83–91, 1993.

[25] Donald W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *J. Soc. Indust. Appl. Math.*, 11(2):431–441, 1963.

[26] Dean S. Oliver. Multiple realizations of the permeability field from well-test data. *SPE Journal*, 1(2):145–154, 1996.

[27] Dean S. Oliver, Nanqun He, and Albert C. Reynolds. Conditioning permeability fields to pressure data. In *Proccedings of the European Conference for the Mathematics of Oil Recovery*, 1996.

[28] Dean S. Oliver, Albert C. Reynolds, and Ning Liu. *Inverse Theory for Petroleum Reservoir Characterization and History Matching.* Cambridge University Press, Cambridge, UK, 2008.

[29] A. C. Reynolds, Mohammad Zafari, and Gaoming Li. Iterative forms of the ensemble Kalman filter. *Proceedings of 10th European Conference on the Mathematics of Oil Recovery*, 2006.

[30] Albert C. Reynolds, Nanqun He, Lifu Chu, and Dean S. Oliver. Reparameterization techniques for generating reservoir descriptions conditioned to variograms and well-test pressure data. *SPE Journal*, 1(4):413–426, 1996.

[31] Albert C. Reynolds, Nanqun He, and Dean S. Oliver. Reducing uncertainty in geostatistical description with well testing pressure data. In Richard A. Schatzinger and John F. Jordan, editors, *Reservoir Characterization–Recent Advances*, pages 149–162. American Association of Petroleum Geologists, 1999.

[32] José R. P. Rodriques. Calculating derivatives for history matching in reservoir simulators. In *Proceedings of the SPE Reservoir Simulation Symposium*, number SPE 93445, page 9, 2005.

[33] José R. P. Rodriques. Calculating derivatives for automatic history matching. *Computational Geosciences*, 10:119–136, 2006.

[34] P. C. Shah, G. R. Gavalas, and J. H. Seinfeld. Error analysis in history matching: The optimum level of parameterization. *SPE Journal*, 18(6):219–228, 1978.

[35] Albert Tarantola. *Inverse Problem Theory: Methods for Data Fitting and Model Parameter Estimation.* Elsevier, Amsterdam, The Netherlands, 1987.

[36] Reza Tavakoli. *History Matching With Parameterization Based on the SVD of a Dimensionless Sensitivity Matrix.* PhD thesis, The University of Tulsa, 2010.

[37] Reza Tavakoli and Albert C. Reynolds. History matching with parameterization based on the SVD of a dimensionless sensitivity matrix. In *Proceedings of the SPE Reservoir Simulation Symposium, The Woodlands, Texas, USA, 2–4 February*, number SPE 118952, 2009.

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

[39] Reza Tavakoli and Albert C. Reynolds. Monte carlo simulation of permeability fields and reservoir performance predictions with SVD parameterization in RML compared with EnKF. *Computational Geociences*, Online First, 2010.

[40] Kristian Thulin, Gaoming Li, Sigurd Ivar Aanonsen, and Albert C. Reynolds. Estimation of initial fluid contacts by assimilation of production data with EnKF. In *Proceedings of the SPE Annual Technical Conference and Exhibition*, number SPE 109975, 2007.

[41] W. Xu, T. T. Tran, R. M. Srivastava, and A. G. Journel. Integrating seismic data in reservoir modeling: the collocated cokriging approach. In *Proceedings of the SPE Annual Technical Conference and Exhibition*, number SPE 24742, 1992.

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

[43] F. Zhang and A. C. Reynolds. Optimization algorithms for automatic history matching of production data. In *Proceedings of 8th European Conference on the Mathematics of Oil Recovery*, 2002.

[44] 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.