UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

# AUTOMATIC HISTORY MATCHING OF GEOLOGIC FACIES

A Dissertation

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

By

**NING LIU**

Norman, Oklahoma
January 2005

UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

# AUTOMATIC HISTORY MATCHING OF GEOLOGIC FACIES

A Dissertation APPROVED FOR THE
MEWBOURNE SCHOOL OF
PETROLEUM AND GEOLOGICAL
ENGINEERING

BY:

Dean S. Oliver, Committee Chair

Chandra Rai

S. Lakshmivarahan
School of Computer Science

Richard Hughes

Samuel Osisanya

# ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my professors, colleagues, friends, and family for their support and wisdom that made my PhD study an incredible achievement through an enjoyable process. I am lucky to have each single one of them in my life.

Foremost, Dr. Dean S. Oliver is the best advisor and mentor one could ever have. He provided me brilliant ideas and deep insights in research, the best environment to work in, many opportunities to present my work to the industry, and a warm heart in my difficult moments. Special thanks extend to Mary Oliver for her hospitality and kindness that always made me feel at home.

My dissertation committee members have offered valuable instructions, critiques and suggestions in the process of my PhD study, which are deeply appreciated. Dr. Chandra Rai is knowledgable both as a professor and as a graduate liaison. The process of my PhD study would not be so smooth without him. Dr. S. Lakshmivarahan brought me into the world of data assimilation using super-computing technology. Dr. Samuel Osisanya taught me to have the industrial applicability in mind when developing a method. Dr. Richard Hughes challenged me to discover the fundamental principles behind experimental findings. I am grateful to all the MPGE faculty members for offering high quality learning experience and making the school a wonderful place to start a career in petroleum and geological engineering. I also thank Dr. Henry Neeman, the director of the OU Supercomputing Center for Education and Research, for providing me access to the super-computers and teaching me skills in operating UNIX and LINUX systems.

Licences for the ECLIPSE$^{TM}$ Black Oil Reservoir Simulator were provided by Schlumberger. Some case studies on multi-phase flow history matching problem used the the Chevron Limited Application Simulation System (CLASS) simulator, which was generously provided by CHEVRON. The adjoint system for the CLASS simulator was developed at the University of Tulsa by the research group TUPREP (University of Tulsa Petroleum Reservoir Exploitation Projects). I thank Dr. Albert Reynolds, the director of TUPREP and Professor of Petroleum Engineering and Mathematics, for his brilliant ideas and rigorous instructions in building and developing some fundamental work I used in this study. I also thank Dr. Ruijian Li and Dr. Fengjun Zhang for their excellent dissertation work under Dr. Reynolds on developing and improving the computation of the adjoint system. I extend thanks to the department of energy for providing sponsorship (award No. DE-FC26-00BC15309) that made this research possible.

I thank Anadarko Petroleum and Schlumberger for providing me internship opportunities from which I gained deep insight to the application of automatic history matching methods. During my internship with Anadarko, Dr. Charles Ohaeri was a wonderful mentor who taught me a great deal from introduction and industrial application of advanced technologies developed in academia to things a green horn engineer needs to know in operations. I also thank the colleagues in engineering and G&G for creating a supportive environment in which I was able to accomplish my projects successfully. I thank Dr. Younes Jalali, manager of the Completions Architecture Department of Schlumberger, for his confidence on me and consistent support throughout my graduate study.

My appreciation extends to all the staff in the MPGE, who have helped me a lot in different ways. I thank Debi Bradley for being a wise career advisor, as well as a fun and caring friend I can fall back on. I thank Chyrl Yerdon for her thoughtful arrangements and timely reminding that helped me meet all the academic challenges. I thank Shalli Young and Mike Shaw for their warm assistance in every day work.

My graduate study would not be as enjoyable without all my wonderful colleagues and dear friends: Siewching Ang, Amina Boughrara, Huanwen Cui, Shi Chen, Yannong Dong, Yaqing Gu, Richard Kho, Bin Liu, Amber Sharples, among many others. They provided encouragement and support throughout my graduate study. I would also like to thank them all for the splendid shared moments.

This work is dedicated to my parents, Fenghua Liu and Yingqiu Liu, for their guidance and unconditional love.

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# ABSTRACT

In this dissertation, two quite different approaches are described to solve the problem of automatic history matching of facies boundaries to hard data and production data. The locations of the geologic facies in a reservoir simulation model are adjusted in order to make predictions from the simulation model to match production data from the field. At the same time, the shapes to facies clusters, and the relations among the facies types must remain consistent with geologic knowledge.

The truncated plurigaussian method for modelling geologic facies is appealing not only for the wide variety of textures and shapes that can be generated, but also because of the internal consistency of the stochastic model. This method has not, however, been widely applied in simulating distributions of reservoir litho-facies or in automatic history matching. One reason seems to be that it is fairly difficult to estimate the parameters of the stochastic model that could be used to generate geological facies maps with the desired properties. The second is that because "facies type" is a discrete variable, it is not straightforward to apply the efficient gradient-based minimization method to generate reservoir facies models that honor production data. Non-gradient methods, however, are too slow for large field-scale simulation and history matching problems.

There are two stages in solving the problem of history matching of litho-facies. The first stage has to do with the specification of a prior geostatistical model, the purpose of which is to ensure plausibility of realizations. This is considerably more complex for the truncated plurigaussian model than for many geostatistical models because it is necessary to specify at least two covariance models (types, ranges, variances, and orientations), as well as the threshold parameters for the truncation. The second stage is adjustment of the facies boundaries for a fixed set of geostatistical model parameters. This requires efficient minimization of an objective function that is not differentiable due to the discontinuity of the model parameters as well as facies types.

For the first stage of estimating the prior geostatistical parameters, a truncated bi-Gaussian method is developed that is characterized by intersecting lines as thresholds. Only six parameters are needed to describe the threshold map, and another six to describe the spacial correlation of the two anisotropic Gaussian random fields. The two Gaussian random fields are generated from two initially identical independent variable fields with size larger than the grids of the facies field. The optimization process minimizes the difference between the simulated facies map and the given training image by perturbing the Gaussian random fields as well as the 12 geostatistical parameters. A gradient minimization method is used to approximate the marginal PDF of the threshold parameters conditional to a stratigraphic cross-section. These geostatistical parameters can then be used in a Bayesian scheme for simulation of facies conditioned to logs or production data.

For the second stage of conditioning the Gaussian fields to the observation data, two approaches were explored: the gradient method and the ensemble Kalman filter (EnKF) method. In the gradient approach, the non-differentiable history-matching problem was replaced with a differentiable problem so that an automatic history matching technique could be applied to the problem of conditional simulation of facies boundaries generated from the truncated plurigaussian method. The resulting realizations are consistent both with the geostatistical model of the observed facies and the historic production. Application of the method requires efficient computation of the gradient of the objective function with respect to model variables. An example five-spot water injection problem is presented with more than 73,000 model variables conditioned to pressure data at wells. The gradient was computed using the adjoint simulator method, and the minimization routine used a quasi-Newton algorithm. The value of the objective function decreased more than 98% in 13 iterations.

In the EnKF approach, the difficulties that prevented the traditional automatic history matching tools from being widely available are minimized. First, the gradient does not need to be computed explicitly, the coding for the EnKF algorithm is easy, and adaptable to any reservoir simulator on a “plug-in” basis. Second, an approximation to differentiability results from the correlation of variables. Third, the ensemble Kalman filter (EnKF) method takes one simulation run per reservoir model realization, and the simulations of the reservoir models in the ensemble are ideal for multiple-processor parallel computation. The truncated plurigaussian model is used to generate random facies realizations. The geostatistical model is fully specified by the threshold truncation map and the covariance models for the two Gaussian random fields. In the first example, the application of the EnKF to the problem of generating facies realizations conditional to observations at 18 wells on a $128 \times 128$ grid is demonstrated. In a second example, realizations of facies on a $50 \times 50$ grid, conditional to facies observations at the wells and to production and injection rates, are generated using the EnKF.

Because of the approximate nature of the EnKF, the realizations from one ensemble tend to underestimate the uncertainty especially for problems that are highly nonlinear. Therefore, the distributions of reservoir model realizations from 20 independent ensembles are compared with the distributions from 20 randomized maximum likelihood (RML) realizations for a 2D water-flood model with one injector and four producers. Despite the nonlinear relationship between data such as production rates and facies observations, and the model variables, the EnKF was effective at history matching the production data. We find that the computational effort to generate 20

independent realizations was similar for the two methods, although the complexity of the code is substantially less for the EnKF.

# CHAPTER I

# INTRODUCTION

Researchers have been building tools for automatic history matching of permeability and porosity distributions to honor production data for many years. The assumption is almost always made that the rock properties are distributed randomly and that the randomness can be adequately described by the mean and the spatial covariance of the property fields. If there is more than one type of rock, region or facies, the assumption is usually made that the location of the boundaries of these regions is known. Recent studies have relaxed this restriction by allowing the boundaries of a three dimensional channel to be adjusted interactively during the history matching process (Bi et al., 2000; Zhang et al., 2002). While the method worked quite well for a single channel in a background low-permeability facies, it became apparent that the extension to a reservoir with geological complexity, such as large numbers of channels, would be impractically difficult. As a result, the truncated plurigaussian model is considered for the description of facies boundaries.

The truncated plurigaussian method is attractive for modelling facies distributions for several reasons.

1. The model is capable of generating a wide variety of facies shapes and neighbor relations.

2. The model is based on Gaussian random fields, which preserves the internal consistency of the stochastic model.

3. The truncation, or threshold maps, can be described by relatively few parameters.

Despite its advantages as a method for simulating facies, application of the truncated plurigaussian method with automatic history matching in characterization of litho-facies distribution is not trivial. There are several challenges that need to be tackled to have this automatic history matching technology practical and efficient. The major challenges are:

1. The truncated plurigaussian method was primarily used as a geostatistical method for producing geological facies maps conditional to facies proportions and hard data. The original rectangular truncation threshold scheme has limited flexibility, and a different truncation scheme has to be designed for more general history matching problems.

2. Facies type is an indicator variable, which made the problem non-differentiable. The non-differentiable history-matching problem must be replaced with a differentiable problem so that an automatic history matching technique can be applied to the problem of conditional simulation of facies boundaries generated from the truncated plurigaussian method. The resulting realizations should be consistent both with the geostatistical model of the observed facies and the historic production.

3. Computation of the gradient of the objective function for search direction in the optimization process requires accurate computation of the sensitivities of the facies at each gridblock to each of the model parameters, as well as sufficient processing resources.

4. The random fields as model parameters for generating the Gaussian fields do not

have physical meanings, which is different with traditional automatic history matching where the random fields are permeability and/or porosity. When hard data of porosity and permeability observations at well locations are to be matched, the simulated facies types have to be right at well locations, but the Gaussian model parameters that determine the facies types are variable. The minimization of the squared mismatch between the simulated and the observed data is more difficult than in the traditional automatic history matching.

In addition to the above problems, the use of the ensemble Kalman filter method to enhance the computational efficiency was investigated. The main idea with the Kalman filter is to update model parameters comparing current observations with predictions from the reservoir simulator. Unlike the traditional history matching problem for reservoir characterization, the model parameters for the Kalman filter include not only the reservoir property fields, but also the pressure and saturation at every grid cell as well as the simulated variables corresponding to data at certain observation time. The vector storing all the parameters is referred as the state vector. The differences between the predicted data from the simulation and the observation data are then used to update the current state vector. It is unnecessary to compute the gradient of the objective function to model parameters in contrast to traditional automatic history matching methods.

The Ensemble Kalman filter (EnKF) has great potential in reservoir characterization for two major reasons. First, any reservoir simulator can be used in the EnKF history matching system without excessive work. This idea becomes clear after the illustration of the EnKF formulation in section 8.2. Second, with the increase in deployment of permanent downhole sensors for monitoring pressure, flow rate and other variables, large amounts of data become available with small time intervals in between. Assimilation of these data is a problem of continuous model updating. The ensemble Kalman filter method by nature is highly suitable for such a problem.

Several problems regarding the use of the ensemble Kalman filter for history matching are investigated in this study. The most important is a comparison of the efficiency with a gradient-based method for a history matching problem with known facies properties but unknown boundary locations. Secondly, the ensemble Kalman filter and a gradient-based method are unlikely to give identical estimates of model variables, so it is also important to know if one method generates better realizations. Finally, since there is often a desire to use the history matched realizations to quantify uncertainty, it is important to determine if one of the methods is more efficient at generating independent realizations.

This work contains ten chapters in total. Chapter 2 introduces the development and previous applications of the methodologies used in this study. Four major topics of this chapter include: the reason why the truncated plurigaussian method was chosen; an overview of previous work on automatic history matching of rock properties; the potential of the sequential continuous model updating using the ensemble Kalman filter method; and the necessity to evaluate the uncertainty quantification performance of the gradient and the EnKF approaches on the problem of history matching on geologic facies. Chapter 3 explains the truncated plurigaussian method in detail. Chapter 4 describes the geostatistical model variables, and the probability distributions of the geostatistical model based on Baysesian framework. Chapter 5 presents the moving average method for generating anisotropic Gaussian fields with a continuously variable covariance. A few examples for generating the Gaussian fields are given for both constant and variable covariance types. In Chapter 6, the gradient approach is used to estimate the geostatistical parameters from a given training image. A transition zone is introduced at the facies boundaries such that the gradients of facies mismatch with respect to the model parameters can be approximated. The derivations of the sensitivities are illustrated in detail. The a posteriori probability density distribution of the geostatistical parameters are analyzed to assist in the selection of the geostatistical parameters for matching hard data and production data. Chapter 7 presents the gradient approach in matching both hard data and historical production data. Chapter 8 presents a study on history matching of geologic facies to hard data and production data using the EnKF approach. Chapter 9 compares the computational efficiency, applicability, and uncertainty quantification properties of the gradient based approach and the EnKF approach. Finally, the conclusions and contributions of the study are summarized in Chapter 10.

# CHAPTER II

# LITERATURE REVIEW

Major improvements in the application of the truncated Gaussian method for lithofacies simulations based on indicators were developed mostly by scholars at the Ecole des Mines de Paris center of geostatistics. By analyzing the limitations and the potential of the truncated Gaussian method, Galli et al. (1994) found a way to apply this method to a 3-D problem with vertical non-stationarity in the proportions of lithofacies. They showed that this method preserved the consistency of the indicator variograms and cross variograms, and allowed more complex neighbor relations than the standard truncated Gaussian model. In the same period, Le Loc'h et al. (1994) showed the flexibility of the truncated plurigaussian method by truncating two Gaussian functions. They pointed out that even if the two underlying Gaussian functions are independent with each other, the resulting facies maps obtained by truncation are correlated in vertical and horizontal directions. The correlation depends on the construction of thresholds of lithotypes. Using uncorrelated Gaussian functions they found that complex theoretical indicator variograms can be produced in combining various anisotropies by choosing different Gaussian functions. They suggested that the choice of a truncation method to the Gaussian functions should be as simple as possible to have easier control over the problem.

Later, Le Loc'h and Galli (1997) presented an insight to implementing the algorithm both for practical structural analysis and conditional simulations. In demonstrating the influence of the thresholds chosen for truncation, the partition of facies was accomplished using rectangles. But even with this relatively simple thresholding method, it is not at all straightforward to choose appropriate thresholds. The difficulty in estimating model parameters that will result in the desired facies distributions has restricted the practical application of this method. An example of a truncated pluri-gaussian simulation conditional to facies data at well locations was presented with a very slow convergence. This problem was attributed to the instability of the Gaussian covariance matrix. Lantuéjoul (2002) discusses the problem of conditioning truncated plurigaussian models to facies observations extensively. Assuming known threshold parameters, the truncated plurigaussian simulation scheme was able to simulate the Gaussian random fields to match given lithofacies observations. As his simulation problem was small, the Markov chain Monte Carlo sampling method was applied to evolve Gaussian random fields. While, once again, the great potential of the truncated plurigaussian method in simulating lithofacies distribution was revealed, two major problems were left unsolved and seem to be limiting the application of this method beyond France. First is the difficulty in estimation of geostatistical parameters, i.e. the geostatiscal quantities such as the range, the variance, the covariance type (Gaussian, Exponential, Spherical, etc.) and the thresholds for discrimination of facies. Second, the application of the truncated plurigausian method in practical conditional simulation problems requires more efficient methods of sampling to deal with reservoir history matching problems.

Conditional simulation of reservoir facies distributions is of great interest of reservoir engineers. Bi et al. (2000) and Zhang et al. (2002) approached the problem of simulating a channel sand by simulating the location of the centerline, the width, and thickness of the channel all along the channel length. In both cases, the Levenberg-Marquardt or Gauss-Newton methods were used for the history matching and the chain rule was used to compute the derivative of the production data mismatch to the values of channel width for example. They were able to do this because many of the intermediate matrices in the computation of the sensitivities were sparse, and because the number of sensitivity coefficients to be computed was relatively small.

In the article by Rahon et al. (1996), they considered two problems in simulating locations of lithofacies conditional to well pressure data. In the first, they attempt to estimate the permeability of each facies whose locations have been fixed. In the second problem, they altered the size of facies whose permeabilities have been fixed. The gradient calculation relating to lithofacies has been successfully implemented in an implicit single-phase fluid flow model. Rahon et al. (1997) applied similar idea in the problem of simulating channel sand locations. This paper parameterizes a channel by triangularization of surface with the nodes of the triangles representing the parameters. The centerline of the channel is assumed known and fixed and the permeability and porosity in both the channel and non-channel facies are assumed to be known. Sensitivities of the well pressure observations with respect to the parameters of the nodes were computed to adjust the size of the channel. Although the idea of using the gradient method to adjust parameters deciding the size of lithofacies was valuable, their work was limited to the kind of problems with known locations of lithofacies. Landa et al. (2000) integrated well test, logging, and geological data to obtain a reservoir description using the gradient method (Anterion et al., 1989). They calculate the sensitivity matrix for permeability by solving the system $n$ times (where $n$ is the number of parameters or gridblocks) and assume that permeability and porosity are perfectly correlated. Their method is computationally unaffordable for problems with large number of model variables.

The method of truncated plurigaussian simulation is very flexible in simulating the distribution of lithofacies, for instance, the location, width and sinuosity of multi-channels. However, to obtain a satisfactory resolution in the lithofacies map, the

Gaussian random fields used need to be large and the method of computing gradients has to be more efficient.

Gradient-based history matching can be performed several ways (e.g. assimilating data in batch or sequentially); a variety of minimization algorithms can be used (e.g. conjugate gradient or quasi-Newton), and several different methods for computing the gradient are available (e.g. adjoint or sensitivity equations). The most efficient of the traditional gradient-based methods seems to be an adjoint method to compute the gradient of the squared data mismatch (Li et al., 2003b) and the limited memory Broyden-Fletcher-Goldfarb-Shanno method (LBFGS Nocedal, 1980) to compute the direction of the change (Fletcher, 1987; Deschamps et al., 1998; Zhang and Reynolds, 2002). Although the adjoint method is popular and widely implemented in many areas (Chen et al. (1974); Chavent et al. (1975); Carter et al. (1974); Li et al. (2003b)), writing an adjoint system for each reservoir simulator is impossible, because the detailed knowledge of the numerical schemes used in each individual simulator has to be known, and the amount of work involved in writing an adjoint system is comparable to developing the corresponding reservoir simulator. The remaining choice is whether to incorporate all data at once or to incorporate the data sequentially.

Simultaneous, or batch, inversion of all data is clearly a well-established history matching procedure. Although data from wells or sensors may arrive nearly continuously, the practice of updating reservoir models as the data arrive is not common. There are several reasons that make sequential assimilation of data difficult for large, nonlinear models: (1) the covariance for all model variables must be updated as new data are assimilated, but the covariance matrix is very large, (2) the covariance may not be a good measure of uncertainty for nonlinear problems, and (3) the sensitivity of a datum to changes in values of model variables is expensive to compute. Bayesian updating in general is described by Woodbury (1989). Modifying a method described by Tarantola (1987), Oliver (1994) evaluated the possibility of using a sequential assimilation approach for transient flow in porous media. He found that the results from sequential assimilation could be almost as good as from batch assimilation if the order of the data was carefully selected. The problem was quite small, however, and an extension to large models was impractical.

The ensemble Kalman filter is a powerful method for sequentially updating estimates of model variables and for assimilating various types of data. One of the problems with the traditional Kalman filter is the difficulty of the computation of the covariance of the model parameters, which is necessary for ensuring that each adjustment to current model parameters does not destroy the match to previous observation. The updated covariance matrix needs to be constructed and stored each time a new set of data are assimilated. A second problem with the traditional Kalman filter is that it is necessary to compute the sensitivity of data to model variables, as in many history matching algorithms (see Bissell et al., 1994; Chu et al., 1995; Omre et al., 1996; He et al., 1997; Gosselin et al., 2001). This computation makes the traditional Kalman filter impractical for even moderate-sized reservoir problems. Eisenmann et al. (1994) and Corser et al. (2000) have attempted to apply the traditional Kalman filter to reservoir characterization problems, but the applications have been restricted to problems with small number of parameters and the relationships between observation and model parameters were nearly linear.

In the first paper on the ensemble Kalman filter, Evensen (1994) described how the evolution of the probability density function for the model variables can be approximated by the motion of “particles” or ensemble members in phase space. Any desired statistical quantities can be estimated from the ensemble of points. When the size of the ensemble is relatively small, however, the approximation of the covariance from the ensemble almost certainly contains substantial errors.

The ensemble Kalman filter has been developed and successfully applied mainly in the fields of physical oceanography and meteorology. Houtekamer and Mitchell (1998) noted the tendency for reduction in variance due to “inbreeding”. When the ensemble estimate is used in a Kalman filter, van Leeuwen (1999) explained how nonlinearity in the covariance update relation causes growth in the error as additional data are assimilated. Houtekamer and Mitchell (2001) have applied the EnKF to assimilate over 100,000 weather observations in a simplified weather model with 409,600 variables. In their study, they introduced a technique for global EnKF analysis and covariance estimation. Anderson (2001) demonstrated the ability of a modified ensemble Kalman filter to assimilate data in a problem with a state vector much larger than the number of ensembles. Evensen (2003) provided a comprehensive review of the progress on the ensemble Kalman filter since its introduction by Evensen (1994).

This method is now beginning to be applied in other fields, including groundwater hydrology (Reichle et al., 2002) and petroleum engineering. Nævdal et al. (2003) applied ensemble Kalman filter techniques for continuous model updating on two 2-D 3-phase reservoir problems. One was a synthetic model with two producers and one injector, the other model was a simplified model of a North Sea oilfield. The measurements in both cases included well pressure, oil rates, GORs, and water cut. The reservoir models were updated by assimilating production data at least once a month and also when a well began production or was shut in. They found that the prediction of future performance from the ensembles improved with more data assimilated; the permeability estimation, however, became worse. Gu and Oliver (2004) applied the EnKF to the PUNQ-S3 reservoir model (Floris et al., 2001). They found that the method was quite efficient compared to the gradient-based methods and gave reasonable estimation of uncertainty.

A secondary objective of history matching is often to assess the uncertainty in the predictions of future reservoir performance or in the estimates of reservoir properties such as permeability, porosity or saturation. In general, uncertainty is estimated from an examination of a moderate number of conditional simulations of the prediction or properties. Unless the realizations are generated fairly carefully and the sample is sufficiently large, however, the estimate of uncertainty can be quite poor. Two large comparative studies of the ability of Monte Carlo methods to quantify uncertainty in history matching have been carried out, one in groundwater (Zimmerman et al., 1998) and one in petroleum (Floris et al., 2001). Neither was conclusive, partly because of the small sample size. Liu and Oliver (2003b) used a smaller reservoir model (fewer variables), but much larger sample size. They found that the method that minimizes an objective function containing a model mismatch part and a data mismatch part, with noise added to observations, created realizations that were distributed nearly the same as realizations from Markov chain Monte Carlo.

In a later study, the comparison was made using history matching on the truncated plurigaussian model for geologic facies. It provided a difficult history matching problem with significant nonlinearities (Liu and Oliver, 2004) that made both the ensemble Kalman filter and the limited memory BFGS method difficult to apply.

# CHAPTER III

# THE TRUNCATED GAUSSIAN SIMULATION METHOD

The distributions of lithofacies in different types of sedimentations exhibit a great variety of features. The truncated Gaussian simulation method simulates the randomness of the lithofacies distributions by truncating a Gaussian random field into separate scale regions, with each region representing a facies type. Fig. 1 is a schematic example showing truncation of a one dimensional random Gaussian field. In this one dimensional reservoir model of 128 gridblocks, each gridblock is assigned a random Gaussian variable. Truncation thresholds are set at 0.5 and $-0.5$. For gridblocks with Gaussian variables greater than 0.5, they are assigned a facies type as sandstone. For those with Gaussian variables between 0.5 and $-0.5$, they are assigned as shale. The rest with Gaussian variables below $-0.5$ are assigned as dolomite. Obviously, a slight perturbation of the thresholds will first change the facies type of the grids at the boundary of facies regions. A more general application of the truncated Gaussian method is to simulate facies distributions in a 2-D plane. An example of the truncation of a 2-D Gaussian random field using the same two thresholds is provided in Fig. 2 to illustrate the resulting facies distributions in a 2-D plane. The figure on the left is the random Gaussian function distributed on the field. The brighter the shade, the greater the Gaussian random numbers are in that location. The simulated facies map of the field is shown on the right. Three shades represent three different lithotypes. Notice that the black and the white areas cannot be directly adjacent to each other. This reflects one of the major drawbacks of the truncated "single" Gaussian method: the lithofacies represented by non-adjacent scales of Gaussian values can never be in direct contact with each other in the facies field map unless the same facies is provided with separate threshold regions. This drawback restricted the method from simulation problems with only two lithotypes, which is unrealistic for most of the reservoir stratigraphic cross-section.

As an extension of the truncated Gaussian simulation method, the plurigaussian simulation allows complex arrangements of several lithofacies to be reproduced. A second Gaussian random field is assigned to each of the gridblocks in the field and the two Gaussian fields together simulate the distribution of the lithofacies in the field. The two random functions can be either correlated or not. For the truncated plurigaussian method, choosing a threshold scheme is no longer straightforward and the efficiency and flexibility of the truncated plurigaussian method is largely ruled by the threshold scheme.

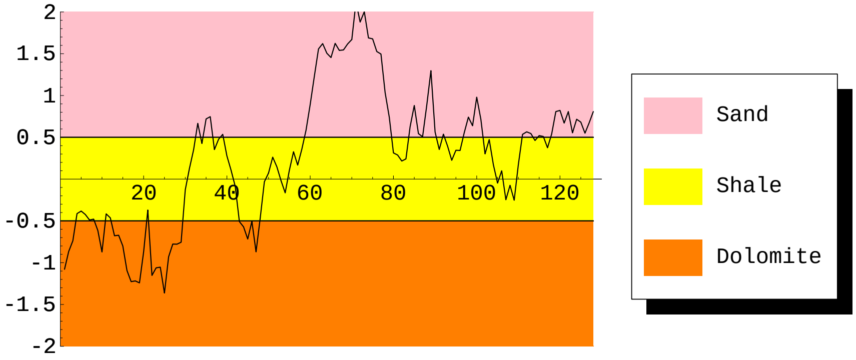


**Figure 1:** A schematic example shows the truncation of a one dimensional random Gaussian field with exponential covariance.

The choice of the truncation method for the Gaussian variables is important in applying truncated Gaussian simulation in automatic history matching to generate

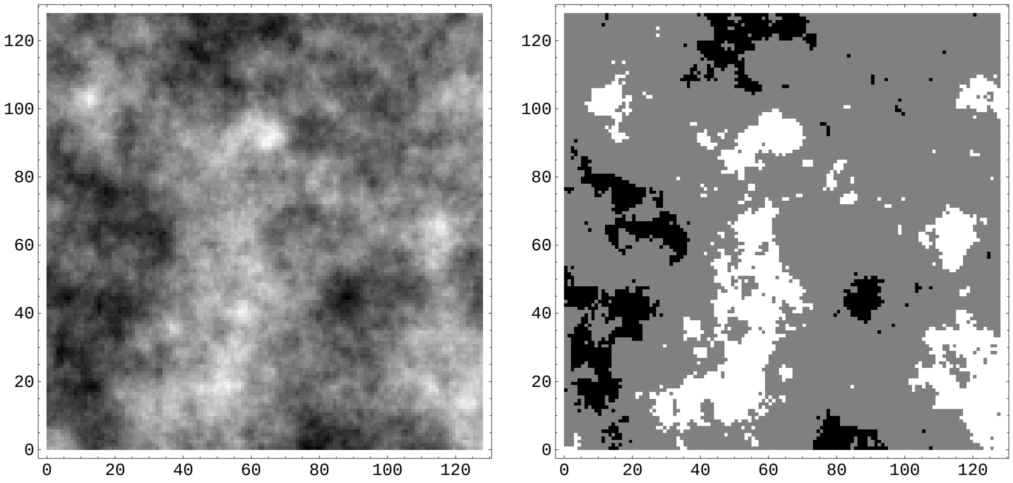


**Figure 2:** Truncation of a 2-D random Gaussian field to simulate the distribution of three lithofacies.

reservoir models satisfying geological requirements. Le Loc'h et al. (1994) have applied rectangular thresholds in truncated Gaussian simulation as shown in Fig. 3. In their approach, lithofacies $F_i$ is modeled by

$$F_i = \{x \in R^3; S^i_{j-1} \leq Y_j(x) < S^i_j, j = 1, \dots, p\}, \tag{1}$$

where $Y_j(x)$, for $j = 1, \dots, p$, are Gaussian random functions which can be independent or dependent. In practice, two Gaussian random functions were used to determine lithotypes, i.e. $p = 2$. The thresholds $S^i_{j-1}$ and $S^i_j$ were decided by the proportion of each lithotypes. Field facies maps with a variety of textures and patterns were generated by truncating groups of Gaussian functions with different combinations of variogram types. Fig. 4 shows some of their results. Although the rectangular thresholds approach is useful for geological simulation problems when conditioning to lithotype proportion data, it does not seem easy to be applicable to the problem of optimization of lithotype grouping in automatic history matching problems.

Alternatively, Voronoi tessellation could be used to create the thresholds. Given a set of points in a plane, a Voronoi diagram can be created by dividing the plane

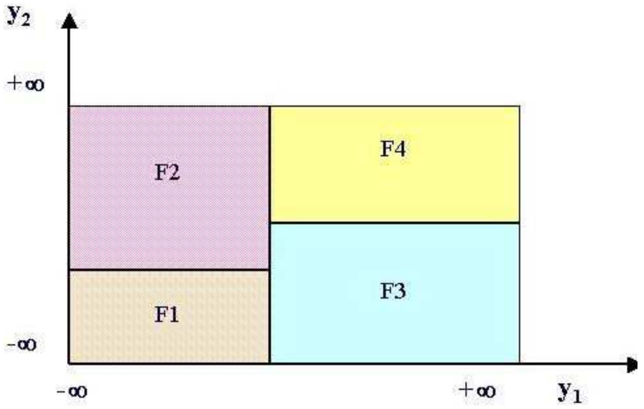
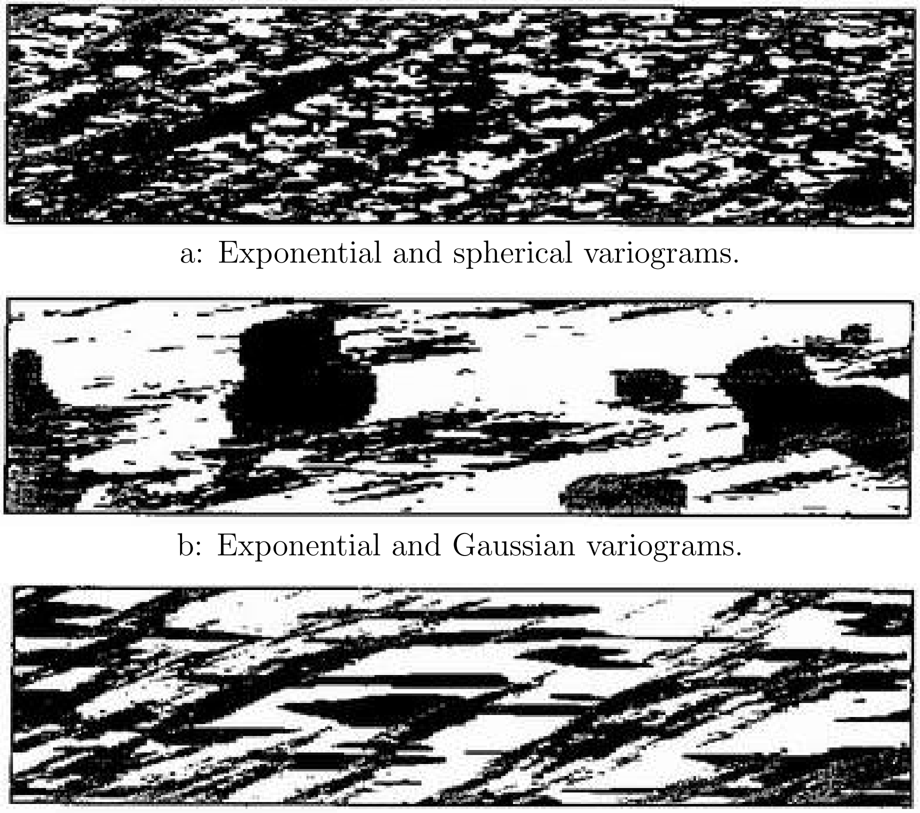


**Figure 3:** Partitions defined by rectangles parallel to the axes from Le Loc'h and Galli (1997).



a: Exponential and spherical variograms.

b: Exponential and Gaussian variograms.

c: Exponential and Gaussian variograms.

**Figure 4:** Simulation using independent Gaussian functions (Le Loc'h et al., 1994).

into regions, so that every location in the region around one point is closer to that point than to any of the other points. Shirriff (1993) described an algorithm for creating Voronoi diagrams on a set of points to generate fractal patterns which resemble the randomness in nature. By attributing each region with a lithotype, a threshold map could be designed from a Voronoi diagram with great flexibility in optimization. Fig. 5 shows Voronoi diagrams generated by Shirriff (1993). However, the difficulty in calculating sensitivities of lithotype at any location in the threshold map to the locations of Voronoi points is the main obstacle in coupling the Voronoi type thresholds with efficient optimization methods which require computation of gradients.

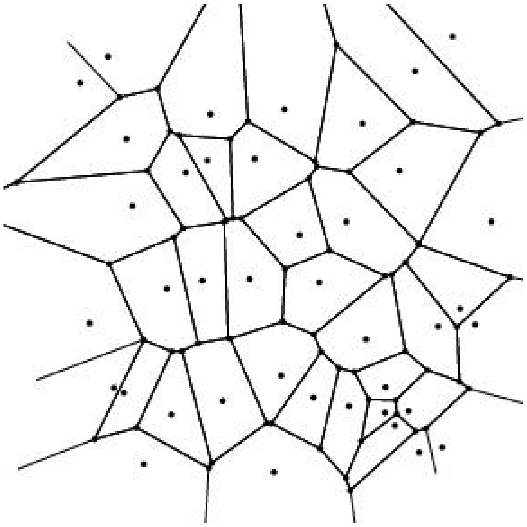


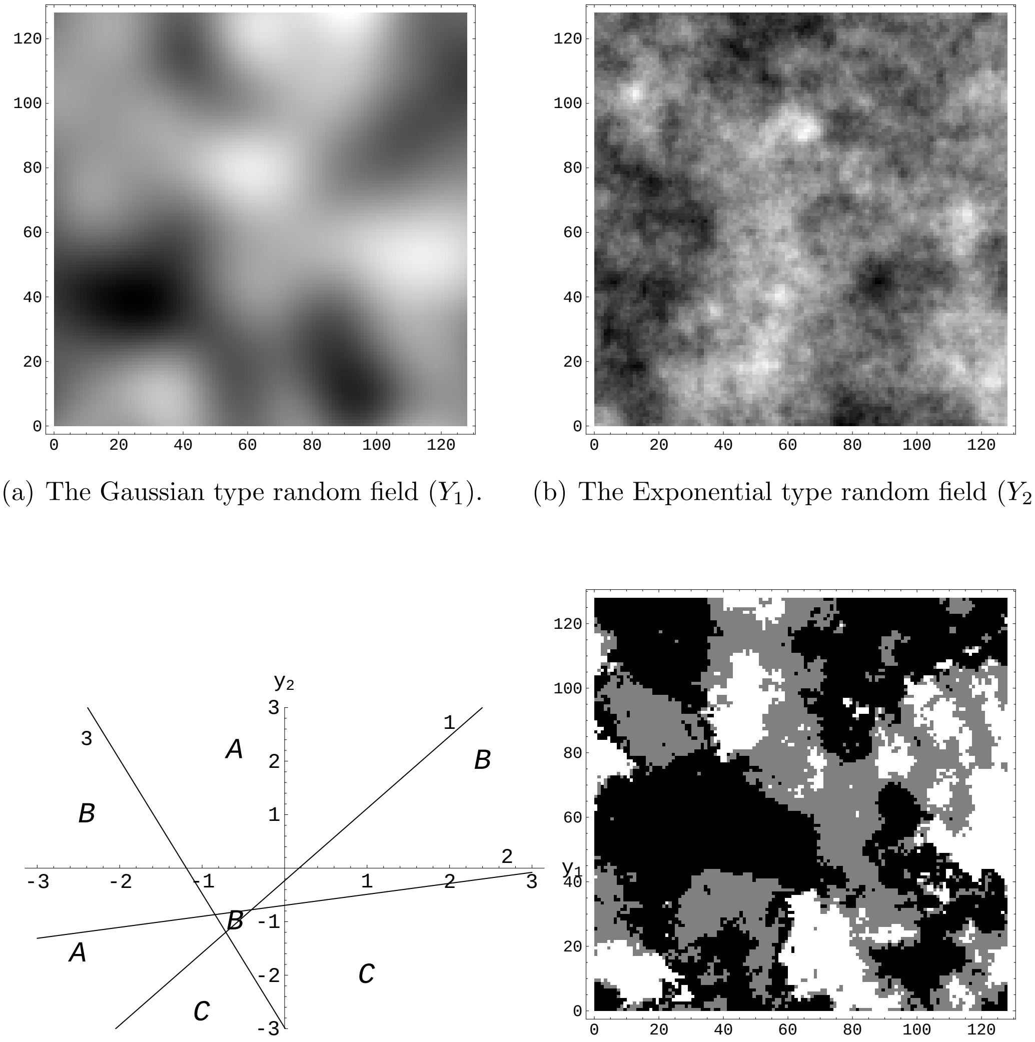
**Figure 5:** Example of Voronoi diagrams by Shirriff (1993).

In this dissertation, truncated Gaussian simulation is performed using three intersecting threshold lines. Three randomly generated lines intersecting each other without all passing through the same point divide the plane into 7 regions. A facies type can be attributed to each region, so up to 7 different facies types can be included in the same plane with appropriate relative percentage. This number of facies is generally enough for geologic maps in petroleum reservoir studies, but if not, another line could be added. The three lines are thresholds for different rock properties. Given a rotation angle $\theta$ and a distance $r$, a threshold line could be described by the following equation:

$$y = \tan(\theta - \frac{\pi}{2})(x - \frac{r}{\cos\theta}), \tag{2}$$

i.e., the threshold line is perpendicular to the line passing through the origin with the slope $\theta$ and intersects the line at a distance $r$.

Fig. 6 is an example illustrating the truncation scheme of intersecting threshold lines. The Gaussian random field $Y_1$ has Gaussian type covariance and $Y_2$ has exponential type covariance. The coordinates of the threshold map (Fig. 6(c)) are $Y_1$ and $Y_2$ respectively. Three kinds of lithotypes, A, B, and C are assigned to the seven regions in the threshold map. Facies type at any gridblock in the field is decided by taking the $Y_1$ and $Y_2$ value of that gridblock to the threshold map. For instance, the gridblock $(20, 40)$ has low values for both its $Y_1$ and $Y_2$. (They both are in areas with dark shade.) So it corresponds to the area in threshold map assigned facies A. We can tell from the facies map (Fig. 6(d)) that the gridblock $(20, 40)$ was assigned facies A. Calculation of the Gaussian fields $Y_1$ and $Y_2$ will be discussed in Chapter 5.



(a) The Gaussian type random field ($Y_1$).

(b) The Exponential type random field ($Y_2$).

(c) Threshold map.

(d) Calculated facies field map.

**Figure 6:** Simulation of lithofacies distribution in the field by truncation of random Gaussian fields $Y_1$ and $Y_2$ using intersecting line thresholds.

# CHAPTER IV

# THE GEOSTATISTICAL MODEL

This chapter discussed the truncated bi-Gaussian method, by which two independent Gaussian random fields, $Y_1$ and $Y_2$ are used to generate a facies map.

## *4.1 Continuous variables*

We begin by considering the continuous variables. Facies are defined by the truncation of two continuous random fields. At the basic level, the variables ($Z_1$ and $Z_2$) on the grid are independent normal deviates with mean 0 and variance 1. These independent deviates must be transformed to correlated random Gaussian fields ($Y_1$ and $Y_2$) for truncation. The parameters of the two covariance functions (such as the ranges of the covariances for the two fields, the principal directions) are variables in the problem of estimation of geostatistical parameters from a given training image.

The locations of the truncation lines that are chosen to partition the Gaussian variables $Y_1$ and $Y_2$, are also continuous variables. Let the number of lines be denoted by $N_l$. The rotation angle and the distance from the origin for each threshold line are $r_i$ and $\theta_i$, for $i = 1, \ldots, N_l$.

| | |
|---|---|
| $\{Z_{11}, Z_{12}, \ldots, Z_{1N_g}\}$ | Independent normal deviates with mean 0 and variance 1. |
| $\{Z_{21}, Z_{22}, \ldots, Z_{2N_g}\}$ | Independent normal deviates with mean 0 and variance 1. |
| $\{a_{11}, a_{12}, \theta_{c1}\}$ | Ranges of covariance and principal direction of anisotropy. |
| $\{a_{21}, a_{22}, \theta_{c2}\}$ | Ranges of covariance and principal direction of anisotropy. |
| $\{(r_1, \theta_1), \ldots, (r_{N_l}, \theta_{N_l})\}$ | Locations of truncation lines. |

**Table 1:** Continuous geostatistical variables.

The variables $Z_1$ and $Z_2$ are assumed to be vectors of standard normal deviates, so their probability distributions are

$$P(Z_1) \propto \exp\bigl(-\frac{1}{2}Z_1^T Z_1\bigr) \tag{3}$$

$$P(Z_2) \propto \exp\bigl(-\frac{1}{2}Z_2^T Z_2\bigr). \tag{4}$$

The prior pdfs for the ranges of the two variograms in the principle directions $(a_{11}, a_{12}, a_{21}, a_{22})$ might realistically be modelled as a $\chi$-square distributions with 0 means and fairly large variances. Even before a training image is given, a positive prior estimate of the spatial correlation ranges can be made with reasonable estimation error. Although the prior estimates of the ranges are always positive, we may want to truncate the distributions so that only positive ranges are allowed. The same is true of $\theta_{c1}$ and $\theta_{c2}$, the orientations for the spatial correlations. Although the prior distributions are uniform on the interval $(0, \pi)$, when the training image is given, the initial estimations are Gaussian with mean at the best estimate of the angle.

The random variable representing distance of the truncation lines from the origin is assumed to be distributed as $N(0, 1)$, so that the unconditional threshold lines are close to the origin. It seems plausible to assume that the prior distributions for the orientation of the partitioning lines $\theta_1, \dots, \theta_{N_l}$ should be uniform on the interval $(0, \pi)$. In this case, the probability density is a constant and can be ignored (or, more accurately, absorbed into the overall constant).

The list of continuous model parameters in the truncated plurigaussian problem is shown in Table 1. The vector of continuous variables, $m_{cG}$, can then be defined whose prior distribution is Gaussian and $m_{cU}$ to be the vector of continuous variables

| | |
|---|---|
| $N_l$ | The number of partitioning lines. |
| $\{m_1, m_2\}$ | Covariance models for fields $Y_1$ and $Y_2$. |
| $\{F_1, F_2, \ldots, F_{N_r}\}$ | Facies assignment for each partitioning region. |

**Table 2:** Discontinuous geostatistical variables.

whose prior distribution is uniform, or

$$m_{cG} = \begin{bmatrix} Z_1 \\ Z_2 \\ a_{11} \\ a_{12} \\ a_{21} \\ a_{22} \\ \theta_{c1} \\ \theta_{c2} \\ r_1 \\ \vdots \\ r_{N_l} \end{bmatrix} \quad \text{and} \quad m_{cU} = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_{N_l} \end{bmatrix}. \tag{5}$$

## 4.2 Discrete variables

There will also be a few parameters or variables that are uncertain but not continuous (and hence not differentiable). The number of partitioning lines is clearly discrete. The covariance model (that is, Gaussian, exponential, spherical, Whittles, etc.) to be used for each of the random fields is not continuous or even numerical. Finally, each region of the truncation map must be assigned a facies. Like the covariance models, the facies do not take continuous values so this is also not continuous or numerical. The group of discontinuous variables is summarized in Table 2.

The existence of discrete variables in this problem makes it harder to apply gradient methods in optimization of model parameters. In this stage of our research, the number of partition lines ($N_l$) is fixed as 3. The covariance models for the Gaussian fields are assumed to be a linear combination of the Gaussian and the exponential type, for which both the Gaussian and the exponential covariance models for a Gaussian field have the same ranges and anisotropy angle. By adding a continuous variable as the weighting factor between the two covariance types, the covariance type becomes differentiable. The reason why the Gaussian and the exponential types are chosen, and how the covariance models are estimated are detailed in Chapter 5.

Chapter 6 will introduce the approach to eliminating the discontinuity of the facies types. As for the probability for assigning a particular facies type to one of the partitioning regions in the threshold map: in the absence of any other information it seems reasonable to assign equal probability to each facies type. As soon as any information on relative abundance of facies is available, the probabilities will not be equal.

## *4.3 Prior probability density*

The prior probability for facies distribution map $F$ on a grid is denoted as $P(F)$. We can write

$$\begin{aligned} P(F) &= P(F|m_{cG}, m_{cU}, m_d)P(m_{cG}, m_{cU}, m_d) \\ &\propto P(m_{cG}, m_{cU}, m_d) \\ &= P(m_{cG})P(m_{cU})P(m_d) \\ &\propto P(m_{cG})P(m_d) \end{aligned} \tag{6}$$

The first term on the right, the conditional probability for the facies map realization $F$ given values of all the model variables, can be ignored as the relationship is deterministic once the variables are given. Independence of the variables in the prior distributions has also been assumed, which explains the third line of Eq. 6. The fourth line is a result of the uniform distribution for some of the variables.

The prior joint probability density for the continuous variables can be written in a compact form as

$$P(m_{cG}) \propto \exp\bigl(-\frac{1}{2} m_{cG}^T C_M^{-1} m_{cG}\bigr), \tag{7}$$

where $C_M$ is the diagonal matrix of variances. For $Z_1$ and $Z_2$ the variances are all equal to 1.

## *4.4 Posterior probability density*

The goal is to generate samples from the posterior distribution, i.e., the distribution of $F$ conditioned to observations, $d_{\text{obs}}$. To do this, we need to be able to characterize the likelihood of the model variables $m$ given the observations, and the prior probability of model variables. Bayes' theorem tells us that

$$P(m_{cG}, m_{cU}, m_d | d_{\text{obs}}) \propto P(d_{\text{obs}} | m_{cG}, m_{cU}, m_d) P(m_{cG}, m_{cU}, m_d). \tag{8}$$

The first term on the right, the likelihood of the model, can be approximated by the following Gaussian expression,

$$P(d_{\text{obs}} | m_{cG}, m_{cU}, m_d) \approx A \exp\Bigl[-\frac{1}{2}(F - F_{\text{obs}})^T C_D^{-1} (F - F_{\text{obs}})\Bigr] \tag{9}$$

where the data or measurement error covariance matrix $C_D$ simply reflects the possibility of error in the identification or modelling of the observed facies. Of course, the vector $F - F_{\text{obs}}$ must be defined in a reasonable way. Facies have no intrinsic numerical value and even if they were assigned numerical values for computation, it might not be reasonable to assume that the difference between Facies 1 and Facies 3 is larger than the difference between Facies 1 and Facies 2. It seems reasonable, for the purpose of conditional simulation and estimation, to assume that the facies are either the same (in which case $F - F_{\mathrm{obs}} = 0$) or they are different (in which case $F - F_{\mathrm{obs}} = 1$).

# CHAPTER V

# GENERATING GAUSSIAN FIELDS

## *5.1 Covariance template construction*

Theoretically, the two Gaussian random fields $Y_1$ and $Y_2$ could be generated from the operation:

$$Y = \mu + LZ, \tag{10}$$

where $\mu$ is the expectation of the Gaussian field $Y$, and $Z$ is a vector of uncorrelated random normal deviates, $Z \in N_m(\mathbf{0}, I)$. The matrix $L$ is the "square root" of the covariance matrix by decomposition, that is,

$$C = LL^T. \tag{11}$$

The dimensions of $L$ are $N_g \times N_g$, where $N_g$ is the total number of grid blocks.

For problems with large number of grids, decomposition of the covariance matrix become unaffordable. In this research, the moving average method proposed by Oliver (1995) was applied to obtain Gaussian fields by convolution of the uncorrelated random normal deviates with covariance templates, i.e.

$$Y = L_T * Z, \tag{12}$$

where $*$ is the convolution operator, and the expectation of the Gaussian field $Y$ is assumed to be zero for truncated Gaussian simulation of geologic facies. $L_T$ is the covariance template. There are two major advantages with application of moving average method for simulation of Gaussian random fields. First, the template $L_T$ is calculated explicitly from the template operator functions, so it is no longer necessary to construct and store the covariance matrix. Second, the dimensions of $L_T$ is generally determined by the correlation range in the field, and is much smaller in size than the covariance matrix.

The three most popular covariance types are Gaussian, exponential, and spherical. The spatial correlation of the Gaussian field $Y$ can be a weighted combination of all three, i.e.

$$L_T = w_1 L_T^G + w_2 L_T^E + w_3 L_T^S, \tag{13}$$

for $w_1 + w_2 + w_3 = 1$. The weighting terms are all real values between 0 and 1. Each of the three templates has identical principle correlation ranges ($a_1$ and $a_2$), and principle direction ($\theta_c$). By using the combination of the covariance templates, the spatial correlation type becomes a continuous function of the weighting terms. As the features of the exponential covariance and the spherical covariance are similar, in practice the combination of the Gaussian and the exponential templates are used for simplicity. By doing so, the optimization problem for the weighting terms reduces from 2-D to line search. The simplified combination is:

$$L_T = wL_T^G + (1-w)L_T^E. \tag{14}$$

The covariance templates are computed from template operators. For simulating a 2-D Gaussian random field with exponential type covariance,

$$C(r) = \sigma^2 e^{-3r/a},$$

the template operator of its square root was derived from the Fourier transform as:

$$f(r) = \sigma K_{1/4}(27)^{\frac{1}{4}}(\frac{3r}{a})(2\pi^2 a^3 r)^{-\frac{1}{4}}\Gamma(\frac{3}{4})^{-1},$$

where $a$ is the correlation range, and $r$ is the lag distance. The formulation of the

Gaussian type covariance and its kernel in 2-D are as follows:

$$C(r) = \sigma^2 e^{-3r^2/a^2}, \tag{15}$$

$$f(r) = \sigma(12/a^2\pi)^{\frac{1}{2}} \exp(-6r^2/a^2). \tag{16}$$

$f(0)$ corresponds to the center of the template $L_T^E$ or $L_T^G$. The center grid of a template is computed by numerical integration of $f(r)$ over $(-\frac{1}{2}bx, \frac{1}{2}bx)$ and $(-\frac{1}{2}by, \frac{1}{2}by)$, where $bx$ and $by$ are the size of each grid in x and y direction respectively. Both the Gaussian and the exponential templates are central symmetric for isotropic spatial correlation, thus computation of one eighth of the template is sufficient for construction of the whole template.

### 5.1.1 Example

In order to demonstrate the simplicity of the method with fairly large grids, the moving average method is applied to a $256 \times 256$ grid of independent normal deviates $Z$ as shown in Fig. 7. Two-dimensional filtering arrays (with size $25 \times 25$) for the square root of the 2-D Gaussian and the exponential covariance ($L_T^G$ and $L_T^E$) are plotted in Fig. 8.

Assume the expectations for the Gaussian fields to be zero, the random Gaussian fields for truncation are computed as $Y = L_T * Z$. Fig. 9 shows the 2-D random Gaussian field realizations with Gaussian, exponential, spherical, and Whittle covariance models respectively. The random Gaussian fields in Fig. 9 (a) and (b) are results from convolution of the templates in Fig. 8 with the random noise shown in Fig. 7.

The frames of uncorrelated cells surrounding the correlated images are a result of the fact that smoothing is done only for pixels that are surrounded by a large enough region for the filter to be applied. The practical consequence is that one would either have to generate a slightly larger grid than was truly necessary (and then discard the unwanted part after smoothing) or assume a periodic structure to the reservoir.

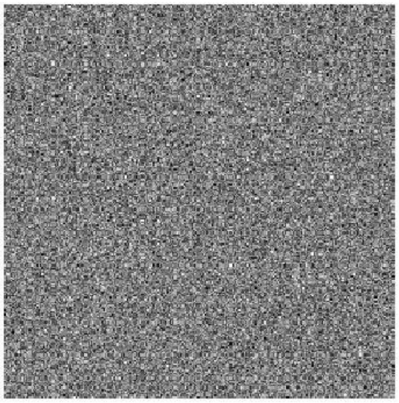
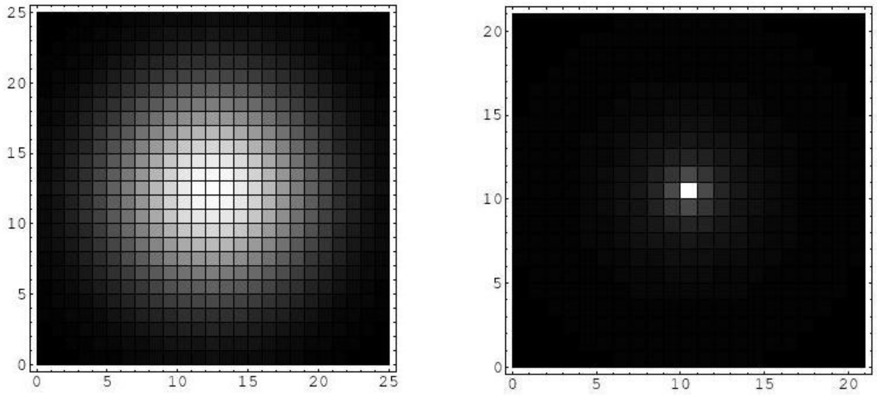


**Figure 7:** The 2-D uncorrelated Gaussian random field $Z \in N(\mathbf{0}, I)$.



(a) 2D Gaussian filter

(b) 2D exponential filter

**Figure 8:** The moving average filters for square root of 2D Gaussian and Exponential covariances.

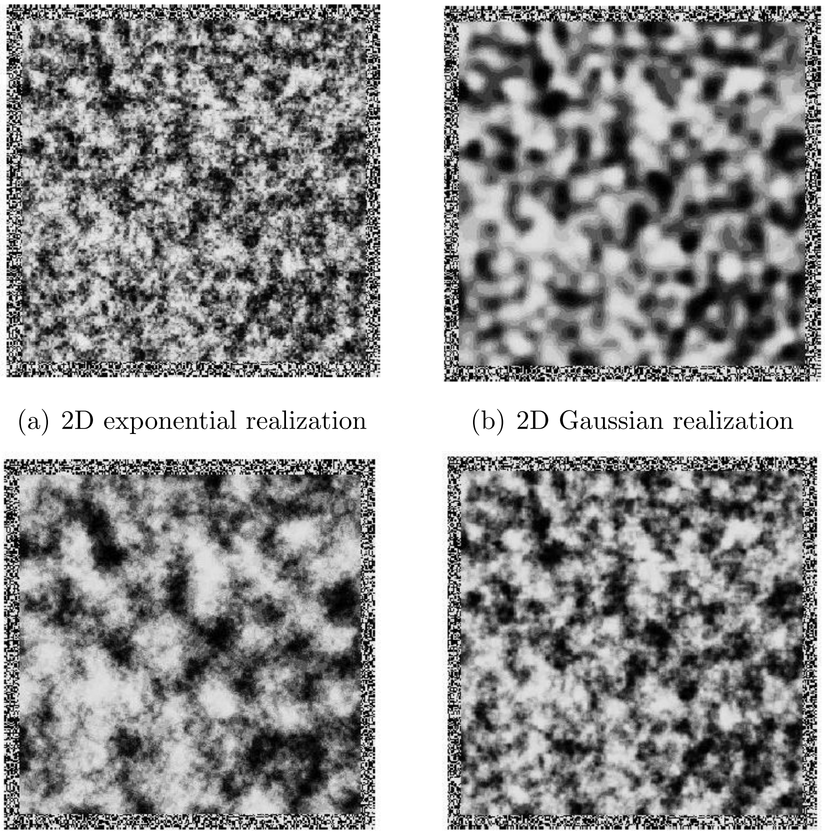
## *5.2 Anisotropic Gaussian covariance*

For an isotropic Gaussian random field, the covariance of a variable at two locations is only a function of the distance between the locations, i.e.

$$\mathrm{cov}(\mathbf{x}, \mathbf{x}') = f(\|\mathbf{x} - \mathbf{x}'\|).$$

But in many cases, the correlation depends on the direction, also. The most obvious example is a channel deposit where the correlation length is much longer in the along-channel direction than in the cross-channel direction.

A property field is called geo-anisotropic if the covariance of the property field



(a) 2D exponential realization

(b) 2D Gaussian realization

(c) 2D spherical realization

(d) 2D Whittle realization

**Figure 9:** The unconditional realizations from the moving average method with different covariance types and $m_{\text{pr}} = \mathbf{0}$.

can an be made isotropic by rotating the coordinate system, then stretching one of the coordinates. Fig. 10 shows an example of a property field that clearly has a longer correlation length in the NNW-SSE direction than in the ENE-WSW direction. Rotation by 30 degree, followed by a factor of 4 stretching in the horizontal direction results in a field that appears to have the same correlation length in all directions.

The rotation from the $x$ - $y$ coordinate system to the $x'$ - $y'$ system is accomplished by

$$x' = x\cos\theta + y\sin\theta$$
$$y' = -x\sin\theta + y\cos\theta$$

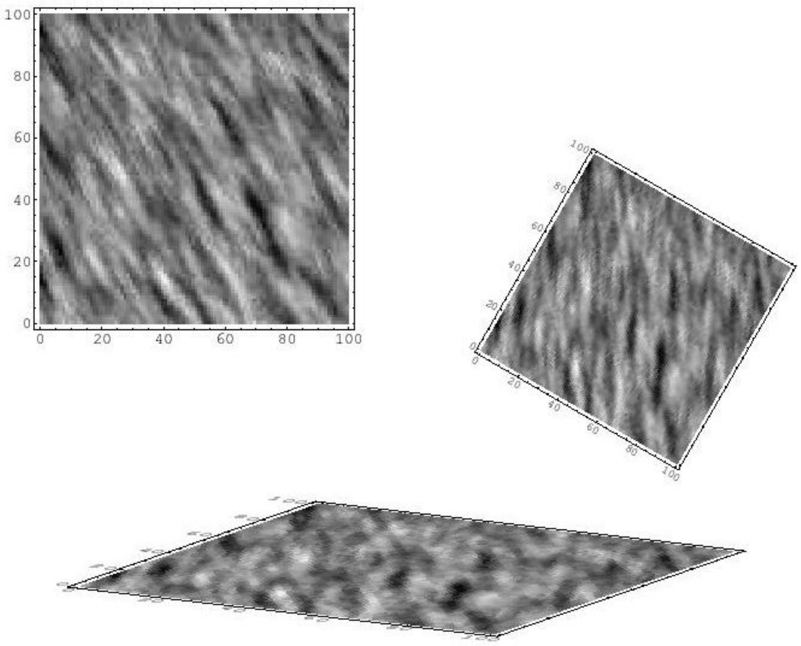


**Figure 10:** Transformation of an anisotropic property field to an isotropic field by rotation and stretching.

while the subsequent stretching is given by

$$
\begin{aligned}
x'' &= x' \\
y'' &= \alpha y'.
\end{aligned}
$$

These transformations can be written as matrix operation form:

$$
\begin{aligned}
\begin{bmatrix} x'' \\ y'' \end{bmatrix} &= \begin{bmatrix} 1 & 0 \\ 0 & \alpha \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \\
&= \mathbf{MT} \begin{bmatrix} x \\ y \end{bmatrix}.
\end{aligned}
$$

In the $x''$ - $y''$ coordinate system, everything is isotropic so we can use regular distance.

The distance between $(x, y)$ and the origin in the isotropic field is:

$$\begin{aligned} r^2 = x''^2 + y''^2 &= \begin{bmatrix} x & y \end{bmatrix} \mathbf{T}^T \mathbf{M}^T \mathbf{M} \mathbf{T} \begin{bmatrix} x \\ y \end{bmatrix} \\ &= \begin{bmatrix} x & y \end{bmatrix} \mathbf{H} \begin{bmatrix} x \\ y \end{bmatrix}. \end{aligned}$$

The matrix $\mathbf{H}$ is the transformation factor between the distance in the true coordinates and the distance after transformation.

The filtering array has the largest value at the center where $r = 0$ and vanishes at the margins. The value of a Gaussian function $f(r) = N(0, 1)$ reduces with $|r|$ increasing. However, only when $|r|$ approaches to infinite does the value of the Gaussian function $f(r)$ approach to zero. Therefore to preserve the efficiency of the moving average method, we truncate the kernel margins with function value less than a cut-off threshold. In practice, we cut-off the margin pixels when the ratio of the probability value at a distance $|r|$ to the center value reaches 0.005. The distance $|r|$ is about 3.25 times the Gaussian correlation range.

### 5.2.1 Example

In this section, an example is presented on generating a Gaussian random field with anisotropic spatial correlation using the moving average method

The porosity field on a uniform lattice of 128 × 128 gridlocks is assumed to be multi-variate normal with mean 0.2 and variance 0.01. The covariance is Gaussian type. The principle correlation is along $-60°$ direction and equivalent to the length of 30 gridblocks. The shortest correlation range in the perpendicular direction has a length of 15 gridblocks. The covariance matrix in this case has size $128^2 \times 128^2$, which is too large for Cholesky decomposition. An unconditional realization of the anisotropic porosity field can be generated by the moving average method following the steps below.

1. Rotate the coordinate system so that the longest correlation direction is along a coordinate axis, then stretch an axis to make the spatial correlation isotropic in the new coordinate system. The coordinate transformation can be written as:
$$\begin{bmatrix} x'' \\ y'' \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} \cos 120^\circ & \sin 120^\circ \\ -\sin 120^\circ & \cos 120^\circ \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$
in which case the distance measure is
$$r^2 = x''^2 + y''^2 = \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} 3.250 & 1.299 \\ 1.299 & 1.750 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}.$$

2. The dimensions of the filtering array in the true coordinate system of $(x, y)$ are decided in the procedures below.

   (a) Let $s$ be a cut-off criterion, $s \ll 1.0$.

   (b) Evaluate the center kernel grid $f(r = 0)$.

   (c) Increase $r$ until $\frac{f(r=r_{\max})}{f(0)} < s$, where $r_{\max}$ is the maximum distance from the center of the filtering array in the coordinate system $(x'', y'')$ within which $f(r)$ should be computed. With the given parameters, the filtering array is computed from
   $$f(r) = 0.1(\frac{12}{30^2\pi})^{\frac{1}{2}} \exp(-6r^2/30^2).$$

   (d) The distance $r$ in the coordinate system $(x'', y'')$ is equivalent to the length along the principle direction in the $(x, y)$ space. For Gaussian templates with the principle direction $\pi/2$ or $-\pi/2$, the cut-off distance $r_{\max}$ along the principle direction corresponds to the dimensions of the template $(x_{\max}, y_{\max})$ in the true coordinate system. In this example, the principle direction is $-\pi/3$, but $x_{\max}, y_{\max}$ are approximated from $r_{\max}$ along principle direction.

$$r_{\max}^2 = \begin{bmatrix} x_{\max} & y_{\max} \end{bmatrix} \begin{bmatrix} 3.250 & 1.299 \\ 1.299 & 1.750 \end{bmatrix} \begin{bmatrix} x_{\max} \\ y_{\max} \end{bmatrix},$$

and

$$y_{\max} = \tan(-60^\circ) x_{\max}.$$

After substitution, $r_{\max}^2 = x_{\max}^2 \times (1 + \sin^2\theta + \tan^2\theta \sin^2\theta)$. Therefore the size of the template corresponds to the cut-off criterion $\frac{f(r=r_{\max})}{f(0)} < s$ can be computed as:

$$\begin{cases} x_{\max} &= r_{\max} \times [1 + \sin^2(-60^\circ) + \tan^2(-60^\circ)\sin^2(-60^\circ)]^{-\frac{1}{2}} \\ y_{\max} &= \tan(-60^\circ) x_{\max} \end{cases}$$

For $s = 0.02$, we obtain $x_{\max} = 13$ and $y_{\max} = 22.5$ for this example. The necessary dimensions of the kernel are (27, 47) after rounding off the coordinates for the number of grids.

3. Compute the filtering template $L_T$ in the necessary dimensions. The pseudo code for computing the Gaussian template in this example is as follows.

```
$L_x = 27;$
$L_y = 47;$
DO $i = 1,\ L_x$
DO $j = 1,\ L_y$

        $x = bx \times [i - (L_x + 1)/2];$
        $y = by \times [j - (L_y + 1)/2];$

        $r^2(x,y) = \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} \cos^2\theta + \alpha^2 \sin^2\theta & \cos\theta\sin\theta - \alpha^2\cos\theta\sin\theta \\ \cos\theta\sin\theta - \alpha^2\cos\theta\sin\theta & \alpha^2\cos^2\theta + \sin^2\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$
        $\quad = (\cos^2\theta + \alpha^2\sin^2\theta)x^2 + 2\cos\theta\sin\theta(1-\alpha^2)xy + (\alpha^2\cos^2\theta + \sin^2\theta)y^2;$

        $r(x,y) = \sqrt{r^2(x,y)};$

        $f(i,j) = \sigma(\frac{12}{a^2\pi})^{\frac{1}{2}} \exp(\frac{-6r^2}{a^2});$

    END DO

    END DO
```

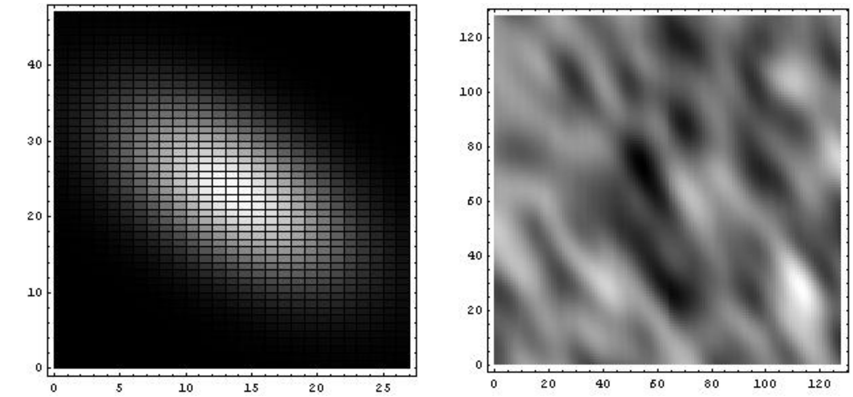
The Gaussian type anisotropic filtering array $L_T$ is plotted in Fig. 11(a).

4. Generate a Gaussian random noise field $Z \in N(\mathbf{0}, I)$. The dimensions of $Z$ are: $(128 + 27 - 1, 128 + 47 - 1) = (154, 174)$, such that the convolution of $L_T$ and $Z$ yields a smoothed Gaussian field with $128 \times 128$ gridblocks.

5. Compute the unconditional realization of the porosity field by:

$$\phi_{uc} = \mu_\phi + L_T * Z.$$

An unconditional realization of the porosity field is shown in Fig. 11(b).



(a) 2D Gaussian kernel (b) Unconditional realization

**Figure 11:** The Gaussian type filtering array and the unconditional realization of the porosity field.

## *5.3 Gaussian fields for truncation*

This section provides the set of procedures used throughout this study in generating Gaussian fields for truncated pluri-gaussian simulation of facies. The two Gaussian fields have independent geostatistical parameters, which is carried by the Gaussian templates $L_1$ and $L_2$. Each covariance template of the Gaussian field is a weighted combination of a Gaussian template and an exponential template.

$$L_1 = w_1 L_1^G + (1 - w_1) L_1^E$$

$$L_2 = w_2 L_2^G + (1 - w_2) L_2^E$$

Denote the dimensions of the grids by $n_x \times n_y$. The principle direction of the Gaussian field $Y_1$ is $\theta_{c1}$ and the range in the principle direction is $a_{11}$. The shortest correlation range in the perpendicular direction has the length of $a_{12}$. The ratio of the short principle range over the long principle range is denoted as $R_1 = a_{12}/a_{11}$, for $0 < R_1 \leq 1$. The weighting term between the Gaussian type and the exponential type for $Y_1$ is $w_1$. So the four parameters $\{a_{11}, a_{12}, \theta_{c1}, w_1\}$ decide the geostatistical model for the Gaussian field $Y_1$. Similarly, $\{a_{21}, a_{12}, \theta_{c2}, w_2\}$ decide the geostatistical features of $Y_2$.

The general procedures for generating a Gaussian field is elaborated below:

1. Decide the dimensions of the templates $L_1^G$ and $L_1^E$. The geostatistical parameters such as the correlation range, the anisotropy rotation angle, and the anisotropy ratio are to be estimated with other reservoir parameters in both the stage of geostatistical model estimation and the stage of history matching to observation data. $Z_1$ and $Z_2$ are reservoir parameters in the truncated pluri-gaussian simulation, and their size is dependent on the dimensions of the templates as their convolution with the covariance templates has to yield the Gaussian fields with the same dimensions as the facies field. To avoid allocation of $Z_1$ and $Z_2$ after each modification to the geostatistical parameters in the optimization process, the dimensions of both templates should be fixed. The principle direction $\theta_{c1}$ is also a model parameter to be optimized, therefore the templates should be square to allow the rotation of the principle direction without resizing the template or sacrificing accuracy.

   Fig. 12 compares the cross-sections of the exponential and the Gaussian templates along the radius. Although the function of the square root of the exponential type covariance has a bigger residue for $r > 1$, the dimensions of the filtering arrays are decided by the Gaussian type function, which is more numerically reliable.

   The procedures for determining the dimensions of the Gaussian template $L_T^G$ are similar to what was introduced in the example of generating an unconditional realization of an anisotropic porosity field.

   (a) Let $s$ be a cut-off criterion, $s \ll 1.0$.

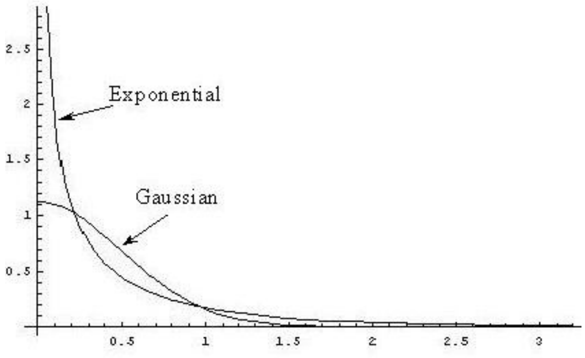


**Figure 12:** The Gaussian and the exponential templates along the radius. Both functions have the same correlation range and unit variance.

(b) Evaluate the central grid of the Gaussian template: $f_{1,G}(r=0)$, where

$$f_{1,G}(r) = (\frac{12}{a_{11}^2\pi})^{\frac{1}{2}} \exp(\frac{-6r^2}{a_{11}^2}).$$

(c) Increase $r$ until $\frac{f_{1,G}(r)}{f_{1,G}(0)} < s$, the distance $r_{\max} = r$ in the coordinate system $(x'', y'')$ is equivalent to the length along the principle direction in the $(x, y)$ space. So the dimensions of the templates are

$$L_{1,x} = L_{1,y} = 2r_{\max} - 1.$$

2. Compute the Gaussian and the exponential type templates, $L_T^G$ and $L_T^E$, with the dimensions $L_{1,x}$ by $L_{1,y}$. The formula for the exponential type template is

$$f_{1,E}(r) = K_{1/4}(\frac{3r}{a_{11}})(27)^{\frac{1}{4}}(2\pi^2 a_{11}^3 r)^{-\frac{1}{4}}\Gamma(\frac{3}{4})^{-1},$$

where $r$ is to be substituted with the pixel indices in the $(x, y)$ space using the relationship:

$$r^2(x,y) = (\cos^2\theta_{c1} + R_1^2 \sin^2\theta_{c1})x^2 + 2\cos\theta_{c1}\sin\theta_{c1}(1-R_1^2)xy + (R_1^2\cos^2\theta_{c1} + \sin^2\theta_{c1})y^2.$$

3. Compute the filtering array for the Gaussian field $Y_1$ using Eq. 14.

4. The other Gaussian field $Y_2$ are computed following the same procedures.

# CHAPTER VI

# ESTIMATION OF THE GEOSTATISTICAL MODEL

Although it is not possible to directly observe the facies distributions in a reservoir formation, the deposition environment mostly decides the geostatistical features of the formation. Traditional history matching adjusts the rock property fields to match the observation data, and the results sometimes are not approved by geologists for being non-realistic. Thus the reservoir realizations for history matching should honor not only the observation data, but also the geostatistical features for that certain deposition type.

Geologists study formation outcrops and sedimentary petrography to develop a training image of a formation, which does not contain any local accuracy but carries geostatistical features about the formation from its deposition environment should have. In this chapter, the geostatistical parameters for a reservoir deposition are estimated by matching to a given training image from a sedimentary cross-section. The estimated geostatistical parameters are then fixed or assumed to have small variances in history matching to observation data.

Five major topics are investigated in this chapter. First, to generate a simulated facies map that matches the training image, an effective algorithm is developed that constructs the gradient of the facies mismatch. The facies mismatch is the difference between the simulated facies map from the estimated geostatistical model parameters and the training image. The original types of facies are not continuous or even numerical. But with a narrow transition zone added to the threshold map, the computation of gradients of the facies mismatch with respect to the geostatistical mode parameters is practical. The second topic deals with the derivation of the gradients, which is illustrated in the section for gradient formulation. The third topic is computation of sensitivities, which are the elements in the arrays of gradients. The fourth topic is a discussion about the problems the optimization process might encounter when the threshold angle takes special values. Finally the work on optimization of threshold lines and analysis of the conditional PDF of the threshold line parameters as well as the covariance parameters is shown.

## *6.1 Continuous gradient of the facies mismatch*

Before history matching to observation data, a prior geostatistical model is required to ensure plausibility of facies realizations. At this stage, homogeneity of rock properties within facies is assumed, i.e. both the permeability and the porosity are constant for the same facies type. The property fields are discontinuous at the facies boundaries. The difference between the simulated facies type and the facies training image at the $i$th gridblock is defined as

$$f_i = \begin{cases} 0 & \text{if } F_i = F_{\text{tr,i}} \\ 1 & \text{if } F_i \neq F_{\text{tr,i}} \ , \end{cases}$$

for $i \in \{1, 2, \ldots, N_g\}$, where $N_g$ is the total number of gridblocks, $F_{\text{tr}}$ is the vector of the facies from the training image, $F_{\text{tr}} = \{F_{\text{tr,i}}, i = 1, 2, \ldots, N_g\}$, and $f$ is the facies mismatch vector, $f = \{f_i, i = 1, 2, \ldots, N_g\}$.

The objective of this stage of the study is to estimate the geostatistical parameters that are appropriate to reproduce the training image. The general objective function for minimization contains the squared facies mismatch part and the squared model mismatch part:

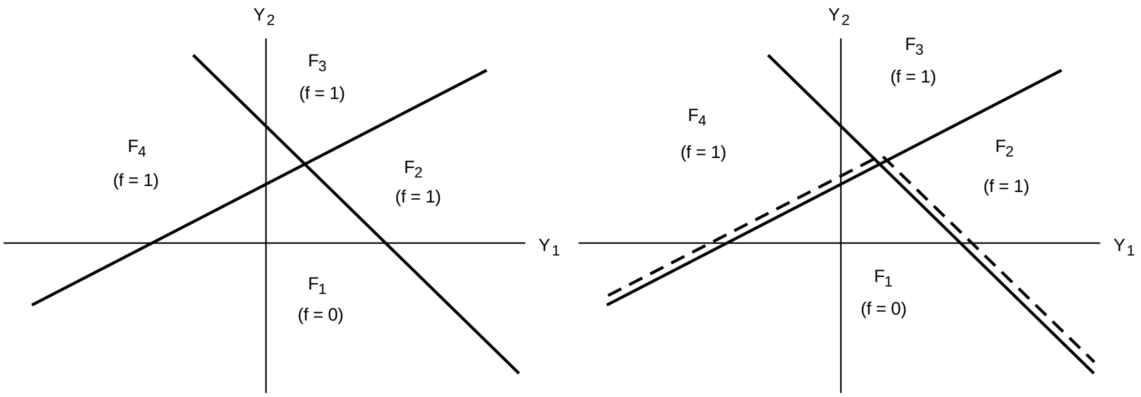
$$O(m) = \frac{1}{2}(g(m) - d_{\mathrm{obs}})^T C_D^{-1}(g(m) - d_{\mathrm{obs}}) + \frac{1}{2}(m - m_{\mathrm{pr}})^T C_M^{-1}(m - m_{\mathrm{pr}}). \tag{17}$$

We need to adjust model parameters to match the training image by minimizing the objective function of the summation of the facies mismatch and the model mismatch:

$$\begin{aligned} O_F(m) &= \frac{1}{2}(F(m) - F_{\mathrm{tr}})^T C_D^{-1}(F(m) - F_{\mathrm{tr}}) + \frac{1}{2}(m - m_{\mathrm{pr}})^T C_M^{-1}(m - m_{\mathrm{pr}}) \\ &= \frac{1}{2} f(m)^T C_{DF}^{-1} f(m) + \frac{1}{2}(m - m_{\mathrm{pr}})^T C_M^{-1}(m - m_{\mathrm{pr}}), \end{aligned} \tag{18}$$

where $m$ is the vector of model parameters to be estimated for generating reservoir models that honor the training image. The $f(m)$ term is the simulated facies mismatch vector with the model $m$. In general, this should be as small as possible since in that case a match is achieved between the training image and the geostatistical simulation. The problem is that this function is not differentiable so we cannot use gradient-based methods to find a minimum. The $C_{DF}$ term is the covariance matrix of the facies observation and is treated as an identity matrix. The variance of the error in facies observation is very small in most cases, therefore the weighting of the model mismatch is much smaller than the data mismatch, and the model mismatch part is negligible in the objective function.

One solution is to redefine the function $f_i$ so that it is differentiable with the locations on the threshold map (only for the purpose of computing the gradient of the squared data mismatch to do minimization). Consider the situation for which the facies observed at some location is $F_1$, and the objective is to estimate the Gaussian variable $x$ by minimizing the difference $f_i^2(x) = \frac{1}{2\sigma_F^2}[F_i(x) - F_1]^2$. In this case, the threshold plot might appear as in Fig. 13. A method for converting the predicted facies to the true facies if the two did not agree would be sought. For now, let us assume that a match will be obtained by moving the truncation lines that separate



(a) Without transition zone (b) With transition zone

**Figure 13:** A schematic example showing adding transition zone on threshold map to make the facies mismatch continuous.

the facies on the threshold plot. In this case, the derivative of the objective function with respect to the location of the threshold lines must be computed. To make the objective function differentiable, a small transition region from $f_i = 0$ to $f_i = 1$ can be added as shown in the plot on the right of Fig. 13. Considering this problem with multi-facies assignment to different regions, when both sides of a threshold line are attributed to facies different from the observed facies, $f_i = 1$ in both regions. There is no transition zone along this section of the threshold line.

This is probably easier to understand if the problem is first considered in 1-D. Let the location of the threshold be at $x = x_0$, that is for $x < x_0$ the facies is the same as the observed facies, so $f_i = 0$. For $x > x_0$ the facies is different from the observed facies, so $f_i = 1$. The function $f_i(x)$ can be made differentiable if a narrow transition zone is added from $x_0$ to $x_0 + \varepsilon$, such that $f(x_0) = 0$ and $f(x_0 + \varepsilon) = 1$. The simplest transition zone is made by a linear interpolation as shown in Fig. 14.

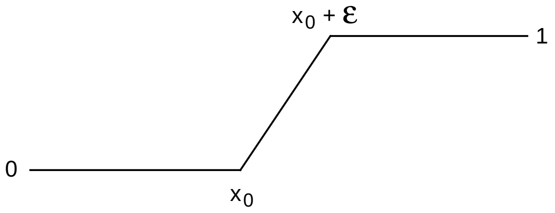


**Figure 14:** A schematic 1-D example demonstrating a linear interpolation for facies mismatch.

Mathematically, this is written as

$$f_i(x) = \begin{cases} 0 & \text{for } x < x_0 \\ \frac{1}{\epsilon}(x - x_0) & \text{for } x_0 \le x < x_0 + \epsilon \\ 1 & \text{for } x \ge x_0 + \epsilon. \end{cases} \tag{19}$$

The derivative of the squared facies mismatch at the gridblock $i$, $O_{F_i}$ with respect to $x$ is

$$\frac{dO_{F_i}}{dx} = \frac{df_i}{\sigma_F^2 dx} f_i = \begin{cases} 0 & \text{for } x < x_0 \\ \frac{x - x_0}{\sigma_F^2 \epsilon^2} & \text{for } x_0 \le x < x_0 + \epsilon \\ 0 & \text{for } x \ge x_0 + \epsilon \ , \end{cases} \tag{20}$$

which has the disadvantage that the derivative grows larger with increasing distance from the actual boundary of the facies. Instead, $f_i(x)$ can be defined as

$$f_i(x) = \begin{cases} 0 & \text{for } x < x_0 \\ \sqrt{\frac{x - x_0}{\epsilon}} & \text{for } x_0 \le x < x_0 + \epsilon \\ 1 & \text{for } x \ge x_0 + \epsilon \ , \end{cases} \tag{21}$$

for which the derivative of the squared facies mismatch with respect to $x$ is

$$\frac{dO_{F_i}}{dx} = \frac{df_i}{\sigma_F^2 dx} f_i = \begin{cases} 0 & \text{for } x < x_0 \\ 1/(2\sigma_F^2 \epsilon) & \text{for } x_0 \le x < x_0 + \epsilon \\ 0 & \text{for } x \ge x_0 + \epsilon \ . \end{cases}$$

When a truncated plurigaussian method with two Gaussian fields is used, each grid is assigned two Gaussian variables to decide the facies type. The truncation map needs to be 2-D instead of 1-D. The definition of a differentiable objective function becomes more complicated. The distance from the line that separates the two facies regions needs to be computed. Consider one of the truncation lines characterized by distance from the origin $r$ and angle $\theta$. Let the shortest vector from the origin to the truncation line be denoted by $\vec{u}$, where

$$\vec{u} = (r\cos\theta, r\sin\theta). \tag{22}$$

For a given point $\vec{v} = (x, y)$, the distance from the point $\vec{v}$ to the line $\vec{u}$ is clearly given by

$$\begin{aligned} d(\vec{v},\vec{u}) &= |\vec{u}| - \frac{\vec{u}\cdot\vec{v}}{|\vec{u}|} \\ &= \frac{\tan(\theta-\frac{\pi}{2})x - y - \tan(\theta-\frac{\pi}{2})\frac{r}{\cos\theta}}{\sqrt{1+\tan^2(\theta-\frac{\pi}{2})}} \\ &= \sin(\theta-\frac{\pi}{2})x - \cos(\theta-\frac{\pi}{2})y + r. \end{aligned} \tag{23}$$

Still assuming the facies observation at point $\vec{v} = (x, y)$ is $F_1$ and the point $(x, y)$ is the $i$th node, the facies mismatch $f_i(x, y)$ is determined by:

$$f_i = \begin{cases} 0 & \text{for } F(x,y) = F_1 \\ \sqrt{\frac{|d|}{\epsilon}} & \text{for } F(x,y) \neq F_1 \text{ and } |d| \leq \epsilon \\ 1 & \text{for } F(x,y) \neq F_1 \text{ and } |d| > \epsilon \ . \end{cases} \tag{24}$$

The gradient of the squared facies mismatch at $(x, y)$ in the two axis directions are derived as:

$$\frac{dO_{F_i}}{dx} = f_i \frac{df_i}{\sigma_F^2 dd(x,y)} \frac{\partial d(x,y)}{\partial x} = \begin{cases} 0 & \text{for } F(x,y) = F_1 \\ \frac{1}{2\sigma_F^2 \epsilon} \sin(\theta - \frac{\pi}{2}) & \text{for } F(x,y) \neq F_1 \text{ and } |d| \leq \epsilon \\ 0 & \text{for } F(x,y) \neq F_1 \text{ and } |d| > \epsilon \end{cases} \tag{25}$$

and

$$\frac{dO_{F_i}}{dy} = f_i \frac{df_i}{\sigma_F^2 dd(x,y)} \frac{\partial d(x,y)}{\partial y} = \begin{cases} 0 & \text{for } F(x,y) = F_1 \\ -\frac{1}{2\sigma_F^2 \epsilon} \cos(\theta - \frac{\pi}{2}) & \text{for } F(x,y) \neq F_1 \text{ and } |d| \leq \epsilon \\ 0 & \text{for } F(x,y) \neq F_1 \text{ and } |d| > \epsilon \ . \end{cases} \tag{26}$$

## 6.2 Gradient formulation

The gradient of the field-wide squared facies mismatch about the model variables is $\nabla_m O_F(m) = G(m)^T f(m)$, where the model variables are in three categories: continuous Gaussian distributed, continuous uniform distributed, and discrete, $m = (m_{cG}, m_{cU}, m_d)^T$. The array $G(m)$ is called the sensitivity coefficient matrix, which is the sensitivity of the data to perturbations of model parameters. The relation between the facies type at a grid node and its relative location to the facies transition zone is nonlinear, so $G(m)$ varies with model parameters.

$$G(m) = \{\nabla_{m_{cG}} f(m)^T, \nabla_{m_{cU}} f(m)^T, \nabla_{m_d} f(m)^T\}$$

Both $\nabla_{m_{cG}} f(m)^T$ and $\nabla_{m_{cU}} f(m)^T$ are relatively straightforward to derive. The $m_d$ terms include the number of threshold lines and the facies assignment to the regions in the threshold map. The computation of $\nabla_{m_d} f^T$ is discussed in this section.

The gradient of the facies mismatch $f(m)$ about the continuous Gaussian type model variables is:

$$\nabla_{m_{cG}} f^T = \begin{bmatrix} \nabla_{Z_{1,1}} f^T \\ \vdots \\ \nabla_{Z_{1,N_g}} f^T \\ \nabla_{Z_{2,1}} f^T \\ \vdots \\ \nabla_{Z_{2,N_g}} f^T \\ \nabla_{a_{11}} f^T \\ \nabla_{a_{12}} f^T \\ \nabla_{\theta_{c1}} f^T \\ \nabla_{a_{21}} f^T \\ \nabla_{a_{22}} f^T \\ \nabla_{\theta_{c2}} f^T \\ \nabla_{r_1} f^T \\ \vdots \\ \nabla_{r_{N_l}} f^T \end{bmatrix}. \tag{27}$$

The elements in the sensitivity coefficient matrix can be derived by chain rule. For instance, let $f_1$ be the facies difference $f$ at gridblock 1, $Y_{1,i}$ and $Y_{2,i}$ be the Gaussian field $Y_1$ and $Y_2$ at gridblock $i$, for $i = 1, \dots, N_g$, and $Z_{1,1}$ be the model random field $Z_1$ at gridblock 1, the sensitivity of $f_1$ with respect to $Z_{1,1}$ is

$$\begin{aligned} \frac{\partial f_1}{\partial Z_{1,1}} &= \frac{\partial f_1}{\partial Y_{1,1}} \frac{\partial Y_{1,1}}{\partial Z_{1,1}} + \frac{\partial f_1}{\partial Y_{1,2}} \frac{\partial Y_{1,2}}{\partial Z_{1,1}} + \dots + \frac{\partial f_1}{\partial Y_{1,N_g}} \frac{\partial Y_{1,N_g}}{\partial Z_{1,1}} \\ &= \nabla_{Y_1} f_1 \cdot \frac{\partial Y_1}{\partial Z_{1,1}}. \end{aligned} \tag{28}$$

Applying chain rule to each of the sensitivity elements, the gradient of $f$ about $Z_1$

can be expressed as:

$$\begin{aligned}
\begin{bmatrix} \nabla_{Z_{1,1}} f^T \\ \nabla_{Z_{1,2}} f^T \\ \vdots \\ \nabla_{Z_{1,N_g}} f^T \end{bmatrix} &= \begin{bmatrix} \frac{\partial Y_{1,1}}{\partial Z_{1,1}} & \frac{\partial Y_{1,2}}{\partial Z_{1,1}} & \cdots & \frac{\partial Y_{1,N_g}}{\partial Z_{1,1}} \\ \frac{\partial Y_{1,1}}{\partial Z_{1,2}} & \frac{\partial Y_{1,2}}{\partial Z_{1,2}} & \cdots & \frac{\partial Y_{1,N_g}}{\partial Z_{1,2}} \\ \vdots & & \ddots & \\ \frac{\partial Y_{1,1}}{\partial Z_{1,N_g}} & \frac{\partial Y_{1,2}}{\partial Z_{1,N_g}} & \cdots & \frac{\partial Y_{1,N_g}}{\partial Z_{1,N_g}} \end{bmatrix} \begin{bmatrix} \nabla_{Y_{1,1}} f^T \\ \nabla_{Y_{1,2}} f^T \\ \vdots \\ \nabla_{Y_{1,N_g}} f^T \end{bmatrix} \\
&= \nabla_{Z_1} Y_1^T \cdot \nabla_{Y_1} f^T \\
&= L_1 \cdot \nabla_{Y_1} f^T. \qquad (29)
\end{aligned}$$

Similarly the gradient of $f$ about $Z_2$ is:

$$\begin{bmatrix} \nabla_{Z_{2,1}} f^T \\ \nabla_{Z_{2,2}} f^T \\ \vdots \\ \nabla_{Z_{2,N_g}} f^T \end{bmatrix} = L_2 \cdot \nabla_{Y_2} f^T. \qquad (30)$$

$L_1$ and $L_2$ are the square root of the covariance matrices for the Gaussian fields $Y_1$ and $Y_2$. In practice, because the dimensions of the model are so large, a convolution of a single row is used as described in the section about moving average. Notice that $f$ is a vector of $N_g$ elements, so $\nabla_{Y_1} f$ and $\nabla_{Y_2} f$ are both $N_g \times N_g$ matrices.

To derive the term for the sensitivity of $f$ to $Y_1$ in Eq. 29, we first consider the gradient of the facies mismatch vector to $Y_{1,1}$:

$$\nabla_{Y_{1,1}} f = \begin{bmatrix} \frac{\partial f_1}{\partial Y_{1,1}} \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \qquad (31)$$

The facies mismatch at a gridblock is only sensitive to the perturbation to $Y_1$ at that gridblock. So the gradient of the array $f$ about the Gaussian field $Y_i$ is a diagonal matrix:

$$\nabla_{Y_i} f^T = \begin{bmatrix} \frac{\partial f_1}{\partial Y_{i,1}} & 0 & \dots & 0 \\ 0 & \frac{\partial f_2}{\partial Y_{i,2}} & \dots & 0 \\ \vdots & & \ddots & \\ 0 & 0 & \dots & \frac{\partial f_{N_g}}{\partial Y_{i,N_g}} \end{bmatrix}, \tag{32}$$

for $i = 1, 2$.

The geostatistical parameters $\{a_{11}, a_{12}, \theta_{c1}, a_{21}, a_{22}, \theta_{c2}\}$ impact the facies mismatch vector through the covariance templates $L_1$ and $L_2$. The gradients of $f$ about this group of parameters are derived by chain rule:

$$\begin{aligned} \nabla_{a_{1i}} f^T &= \nabla_{Y_1} f^T \nabla_{a_{1i}} Y_1 \\ &= \nabla_{Y_1} f^T \nabla_{a_{1i}} (L_1 Z_1) \\ &= \nabla_{Y_1} f^T (\nabla_{a_{1i}} L_1) Z_1 \end{aligned} \tag{33}$$

for $i = 1, 2$,

$$\nabla_{\theta_{c1}} f^T = \nabla_{Y_1} f^T (\nabla_{\theta_{c1}} L_1) Z_1, \tag{34}$$

$$\nabla_{a_{2i}} f^T = \nabla_{Y_2} f^T (\nabla_{a_{2i}} L_2) Z_2 \tag{35}$$

for $i = 1, 2$, and

$$\nabla_{\theta_{c2}} f^T = \nabla_{Y_2} f^T (\nabla_{\theta_{c2}} L_2) Z_2. \tag{36}$$

The square roots of the two covariance matrices $L_1$ and $L_2$ are too large in size for Cholesky decomposition, therefore the moving average algorithm is used and the multiplication operations of $L_i Z_i$ become convolutions of $Z_i$ with a template $L_{iT}$, for $i = 1, 2$. The template is a linear combination of a Gaussian template and an exponential template:

$$L_{iT} = w_i L_{iT}^G + (1 - w_i) L_{iT}^E.$$

The computation of the gradients of the template $L_i$ with respect to the continuous Gaussian type geostatistical parameters, $\{a_{i1}, a_{i2}, \theta_{ci}\}$, for $i = 1, 2$, is discussed below. First the derivation of $\nabla_{a_{i1}} L_{iT}$ will be shown in detail. As $L_{iT}$ is a weighted combination of two terms, both of which are functions about $a_{i1}$,

$$\nabla_{a_{i1}} L_{iT} = w_i \nabla_{a_{i1}} L_{iT}^G + (1 - w_i) \nabla_{a_{i1}} L_{iT}^E,$$

we need to derive the gradients of the two templates with respect to the parameter of the principle range: $\nabla_{a_{i1}} L_{iT}^G$ and $\nabla_{a_{i1}} L_{iT}^E$.

$$\nabla_{a_{i1}} L_{iT}^G = \begin{bmatrix} \frac{dL_{iT}^G(1,1)}{da_{i1}} & \frac{dL_{iT}^G(1,2)}{da_{i1}} & \cdots & \frac{dL_{iT}^G(1,l_{i,x})}{da_{i1}} \\ \frac{dL_{iT}^G(2,1)}{da_{i1}} & \frac{dL_{iT}^G(2,2)}{da_{i1}} & \cdots & \frac{dL_{iT}^G(2,l_{i,x})}{da_{i1}} \\ \vdots & & \ddots & \\ \frac{dL_{iT}^G(L_{i,y},1)}{da_{i1}} & \frac{dL_{iT}^G(L_{i,y},2)}{da_{i1}} & \cdots & \frac{dL_{iT}^G(l_{i,y},l_{i,x})}{da_{i1}} \end{bmatrix}$$

$$\nabla_{a_{i1}} L_{iT}^E = \begin{bmatrix} \frac{dL_{iT}^E(1,1)}{da_{i1}} & \frac{dL_{iT}^E(1,2)}{da_{i1}} & \cdots & \frac{dL_{iT}^E(1,l_{i,x})}{da_{i1}} \\ \frac{dL_{iT}^E(2,1)}{da_{i1}} & \frac{dL_{iT}^E(2,2)}{da_{i1}} & \cdots & \frac{dL_{iT}^E(2,l_{i,x})}{da_{i1}} \\ \vdots & & \ddots & \\ \frac{dL_{iT}^E(L_{i,y},1)}{da_{i1}} & \frac{dL_{iT}^E(L_{i,y},2)}{da_{i1}} & \cdots & \frac{dL_{iT}^E(l_{i,y},l_{i,x})}{da_{i1}} \end{bmatrix}$$

The gradients of the templates to $a_{12}$, $a_{22}$, $\theta_{c1}$, and $\theta_{c2}$ take similar form. The elements in the matrices are discussed in the next section on the sensitivity computation.

The gradients of $f$ about the threshold lines $\{r_1, r_2, \ldots, r_{N_l}\}$ are expanded into matrix form as:

$$\begin{bmatrix} \nabla_{r_1} f^T \\ \nabla_{r_2} f^T \\ \vdots \\ \nabla_{r_{N_l}} f^T \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial r_1} & \frac{\partial f_2}{\partial r_1} & \cdots & \frac{\partial f_{N_g}}{\partial r_1} \\ \frac{\partial f_1}{\partial r_2} & \frac{\partial f_2}{\partial r_2} & \cdots & \frac{\partial f_{N_g}}{\partial r_2} \\ \vdots & & \ddots & \\ \frac{\partial f_1}{\partial r_{N_l}} & \frac{\partial f_2}{\partial r_{N_l}} & \cdots & \frac{\partial f_{N_g}}{\partial r_{N_l}} \end{bmatrix} \tag{37}$$

The gradient of the facies mismatch vector about the uniformly distributed model variables $m_{cU} = \{\theta_1, \theta_2, \ldots, \theta_{N_l}\}$ is:

$$\begin{aligned} \nabla_{m_{cU}} f^T &= \begin{bmatrix} \nabla_{\theta_1} f^T \\ \nabla_{\theta_2} f^T \\ \vdots \\ \nabla_{\theta_{N_l}} f^T \end{bmatrix} \\ &= \begin{bmatrix} \frac{\partial f_1}{\partial \theta_1} & \frac{\partial f_2}{\partial \theta_1} & \ldots & \frac{\partial f_{N_g}}{\partial \theta_1} \\ \frac{\partial f_1}{\partial \theta_2} & \frac{\partial f_2}{\partial \theta_2} & \ldots & \frac{\partial f_{N_g}}{\partial \theta_2} \\ \vdots & & \ddots & \\ \frac{\partial f_1}{\partial \theta_{N_l}} & \frac{\partial f_2}{\partial \theta_{N_l}} & \ldots & \frac{\partial f_{N_g}}{\partial \theta_{N_l}} \end{bmatrix}. \end{aligned} \tag{38}$$

The discontinuous model parameters $m_d$ include the number of threshold lines $N_l$, the covariance types of the two Gaussian fields, and the facies arrangement in the threshold map. The number of threshold lines is set to 3 throughout this study. If there are too many facies to be simulated in a reservoir formation, more threshold lines can be added on a case to case basis. The covariance types have been converted to continuous variables as weighting terms $w_1$ and $w_2$. The gradient of the facies mismatch vector $f$ with respect to $w_i$, for $i = 1, 2$, is:

$$\begin{aligned} \nabla_{w_i} f^T &= \nabla_{Y_i} f^T (\nabla_{w_i} L_i) Z_i \\ &= \nabla_{Y_i} f^T (\nabla_{w_i} L_{iT}) * Z_i \\ &= \nabla_{Y_i} f^T (L_{iT}^G - L_{iT}^E) * Z_i. \end{aligned} \tag{39}$$

If there are up to three facies to be simulated in a reservoir model, the total number of probable facies assignments to a threshold map with seven regions is $3^7 = 2187$. A method to make the facies assignments differentiable has not been determined. The selection of the facies assignment on the threshold map is based on the Monte Carlo method. If the chosen facies assignment does not allow the adjustments of other model parameters to generate facies realizations that match the training image, the facies assignment is abandoned and another will be randomly chosen for the optimization.

## 6.3 Sensitivity calculation

The model variables for simulating the facies distributions fall into two categories: those that decide the relative locations of the Gaussian variables at a grid to the threshold lines, and those that impact the templates for generating the Gaussian fields. The calculation of the sensitivities of the facies type at one location to a slight perturbation to any of the parameters characterizing the threshold lines will be discussed.

Calculations of $\frac{\partial f_i}{\partial Y_{1,i}}$, $\frac{\partial f_i}{\partial Y_{2,i}}$, $\frac{\partial f_i}{\partial r_m}$ and $\frac{\partial f_i}{\partial \theta_m}$, for $i = 1, 2, \ldots, N_g$ and $m = 1, 2, \ldots, N_l$, are illustrated under different conditions.

1. $F_i = F_{\mathrm{obs,i}}$.

   In this case, $f_i = 0$, and the sensitivities should all be zero so that the facies type does not change by perturbing threshold lines, i.e.

   $$\begin{aligned} \frac{\partial f_i}{\partial Y_{1,i}} &= 0; \\ \frac{\partial f_i}{\partial Y_{2,i}} &= 0; \\ \frac{\partial f_i}{\partial r_m} &= 0; \\ \frac{\partial f_i}{\partial \theta_m} &= 0. \end{aligned}$$

2. $F_i \neq F_{\mathrm{obs,i}}$.

   For $f_i = 1$, first we need to find out the distance of the $i$th location to each of the threshold lines. For points not in the transition zone of a threshold line, the sensitivities of facies to the parameters describing this line are all zero. Given an angle $\theta$ and a distance $r$, a line is described by the following equation:

$$y = \tan(\theta - \frac{\pi}{2})(x - \frac{r}{\cos\theta}), \tag{40}$$

i.e. the threshold line is perpendicular to the line passing through the origin with the slope $\theta$ and intersects the line at a distance $r$. The distance of point $(Y_{1,i}, Y_{2,i})$ to the $m$th threshold line is calculated as:

$$\begin{aligned} d_m &= \frac{\tan(\theta_m - \frac{\pi}{2})Y_{1,i} - Y_{2,i} - \tan(\theta_m - \frac{\pi}{2})\frac{r_m}{\cos\theta_m}}{\sqrt{1 + \tan^2(\theta_m - \frac{\pi}{2})}} \\ &= \sin(\theta_m - \frac{\pi}{2})Y_{1,i} - \cos(\theta_m - \frac{\pi}{2})Y_{2,i} + r_m \end{aligned} \tag{41}$$

where $m = 1, 2, \ldots, N_l$. As $\theta_m \in (0, \pi)$, $\cos(\theta_m - \frac{\pi}{2})$ is always greater than zero. The distance can be positive or negative depending on the location of $(Y_{1,i}, Y_{2,i})$ relative to the threshold line. Let $y_i$ be the $y$ coordinate of the point on the threshold line with $x$ coordinate $Y_{1,i}$:

$$y_i = \tan(\theta_m - \frac{\pi}{2})(Y_{1,i} - \frac{r_m}{\cos\theta_m}).$$

When $Y_{2,i} < y_i$, the distance of point $(Y_{1,i}, Y_{2,i})$ to the threshold line $d_m$ is greater than zero, otherwise, it is less than zero. Fig. 15 is a schematic diagram showing the distance of a point to one of the threshold lines. In this figure, $d_m$ is negative and $d'_m$ is positive.

Assume the width of the transition region stated before is $\epsilon$, which may also vary with the given facies data. There are two possibilities: $(Y_{1,i}, Y_{2,i})$ is within a distance of $\epsilon$ to threshold line $m$, or $(Y_{1,i}, Y_{2,i})$ is outside of its transition region.

(a) $\min(|d_1|, |d_2|, \ldots, |d_{N_l}|) \geq \epsilon$:

The point $(Y_{1,i}, Y_{2,i})$ is more than $\epsilon$ away from the threshold line $m$, so the sensitivities of the facies type at this point are zero to all the parameters

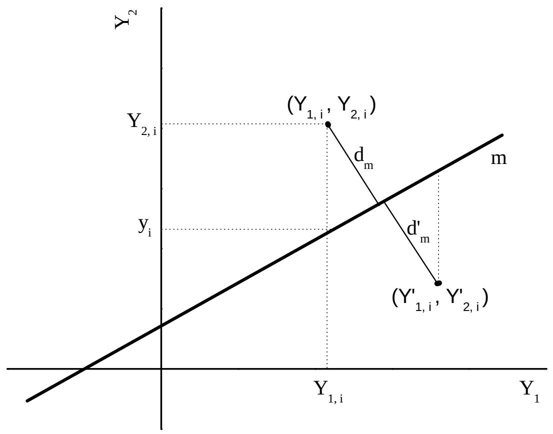


**Figure 15:** The distance $d_m$ is negative and $d'_m$ is positive.

of line $m$.

$$\frac{\partial f_i}{\partial Y_{1,i}} = 0;$$
$$\frac{\partial f_i}{\partial Y_{2,i}} = 0;$$
$$\frac{\partial f_i}{\partial r_m} = 0;$$
$$\frac{\partial f_i}{\partial \theta_m} = 0.$$

(b) $|d_m| < \epsilon$:

The distance to one or more threshold lines is less than $\epsilon$. In this case, we must check whether both sides of the threshold lines are the wrong facies types. If both sides are incorrect, there is no transition area along this section of the threshold line, so the gradient should be zero. The method of checking facies types on both sides is illustrated as follows. Assume a point $(a, b)$ is on a line perpendicular to $l_m$ through the point $(Y_{1,i}, Y_{2,i})$, and the squared distance between the point $(a, b)$ and the threshold line

$l_m$ is:

$$\begin{aligned} d &= \frac{\tan(\theta_m - \frac{\pi}{2})a - b - \tan(\theta_m - \frac{\pi}{2})\frac{r_m}{\cos\theta_m}}{\sqrt{1+\tan^2(\theta_m - \frac{\pi}{2})}} \\ &= \sin(\theta_m - \frac{\pi}{2})a - \cos(\theta_m - \frac{\pi}{2})b + r_m \\ &= \pm\delta \end{aligned} \tag{42}$$

where $\delta$ is a very small positive number so that $(a, b)$ is very close to $l_m$ and right on the other side of the line to point $(Y_{1,i}, Y_{2,i})$. $d$ takes the opposite sign with $d_m$, i.e. in Eq. 42, the result takes a positive sign when $d_m$ is less than zero and otherwise takes a negative sign. As for two lines perpendicular to each other, the product of their slopes is $-1$, the other necessary equation for determining $a$ and $b$ is:

$$\frac{(b - Y_{2,i})}{(a - Y_{1,i})}\tan(\theta - \frac{\pi}{2}) = -1, \tag{43}$$

The coordinate of the probe point on the other side of the threshold line can be obtained from Eq. 42 and 43. Thus the facies type corresponding to the probe point is decided by the facies distribution in the threshold map.

i. If the other side of the threshold line $l_m$ does not have the correct facies type, then again, all the gradients are zero.

$$\begin{aligned} \frac{\partial f_i}{\partial Y_{1,i}} &= 0; \\ \frac{\partial f_i}{\partial Y_{2,i}} &= 0; \\ \frac{\partial f_i}{\partial r_m} &= 0; \\ \frac{\partial f_i}{\partial \theta_m} &= 0. \end{aligned}$$

ii. Otherwise, the sensitivities of facies type at each gridblock to all the parameters deciding the facies type can be calculated as follows. Here the chain rule was applied in the derivation.

$$\begin{aligned}\frac{\partial f_i}{\partial Y_{1,i}} &= \frac{df_i}{d(d_m)}\frac{\partial d_m}{\partial Y_{1,i}} \\ &= \frac{d_m}{|d_m|}\frac{1}{2\sqrt{\epsilon d_m}}\sin(\theta_m - \frac{\pi}{2}) \qquad (44)\end{aligned}$$

$$\begin{aligned}\frac{\partial f_i}{\partial Y_{2,i}} &= \frac{df_i}{d(d_m)}\frac{\partial d_m}{\partial Y_{2,i}} \\ &= -\frac{d_m}{|d_m|}\frac{1}{2\sqrt{\epsilon d_m}}\cos(\theta_m - \frac{\pi}{2}) \qquad (45)\end{aligned}$$

$$\begin{aligned}\frac{\partial f_i}{\partial r_m} &= \frac{df_i}{d(d_m)}\frac{\partial d_m}{\partial r_m} \\ &= \frac{d_m}{|d_m|}\frac{1}{2\sqrt{\epsilon d_m}} \qquad (46)\end{aligned}$$

$$\begin{aligned}\frac{\partial f_i}{\partial \theta_m} &= \frac{df_i}{d(d_m)}\frac{d(d_m)}{d\theta_m} \\ &= \frac{d_m}{|d_m|}\frac{1}{2\sqrt{\epsilon d_m}}[\cos(\theta_m - \frac{\pi}{2})Y_{1,i} + \sin(\theta_m - \frac{\pi}{2})Y_{2,i}] \qquad (47)\end{aligned}$$

The second category of variables are those that characterize the templates $L_{1T}$ and $L_{2T}$. Each of the templates is a weighted combination of a Gaussian and an exponential type template. The sensitivities are first derived by pieces, then are put together to obtain the elements for the gradients. The sensitivity of the Gaussian template value at the pixel $(i, j)$ is denoted as $L_{kT}^{G}$, for $k = 1, 2$, and its gradient about the range in the principle direction is:

$$\begin{aligned}\frac{dL_{kT}^{G}(i,j)}{da_{k1}} &= \frac{d}{da_{k1}}[\sigma(\frac{12}{a_{k1}^2\pi})^{0.5}\exp(-\frac{6r_{ij}^2}{a_{k1}^2})] \\ &= \sigma\exp(-\frac{6r_{ij}^2}{a_{k1}^2})(\frac{24\sqrt{3}r_{ij}^2}{a_{k1}^4\pi^{0.5}} - \frac{2\sqrt{3}}{a_{k1}^2\pi^{0.5}}) \\ &= \frac{2\sqrt{3}\sigma}{a_{k1}^2\pi^{0.5}}\exp(-\frac{6r_{ij}^2}{a_{k1}^2})(\frac{12r_{ij}^2}{a_{k1}^2} - 1),\end{aligned}$$

where $r_{ij}$ is the distance of the pixel $(i,j)$ to the center of the template in an isotropic coordinate space. The gradient of the exponential template about the principle range $a_{k1}$ is

$$\begin{aligned}\frac{dL_{kT}^{E}(i,j)}{da_{k1}} &= \frac{d}{da_{k1}}[\sigma K_{\frac{1}{4}}(\frac{3r_{ij}}{a_{k1}})(\frac{2}{27}\pi^2 a_{k1}^3 r_{ij})^{-0.25}\Gamma(\frac{3}{4})^{-1}] \\ &= \sigma(\frac{2}{27}\pi^2 r_{ij})^{-0.25}\Gamma(\frac{3}{4})^{-1}\frac{d}{da_{k1}}[K_{\frac{1}{4}}(\frac{3r_{ij}}{a_{k1}})a_{k1}^{-\frac{3}{4}}],\end{aligned}$$

where the gradient

$$\frac{dK_{\frac{1}{4}}(x)}{dx} = -\frac{1}{2}(K_{-\frac{3}{4}}(x) + K_{\frac{5}{4}}(x)).$$

So the gradient $\frac{dL_{kT}^{E}(i,j)}{da_{k1}}$ is:

$$\frac{dL_{kT}^{E}(i,j)}{da_{k1}} = \sigma(\frac{2}{27}\pi^2 r_{ij})^{-0.25}\Gamma(\frac{3}{4})^{-1}[\frac{3}{2}a_{k1}^{-\frac{11}{4}}r_{ij}(K_{-\frac{3}{4}}(\frac{3r_{ij}}{a_{k1}})+K_{\frac{5}{4}}(\frac{3r_{ij}}{a_{k1}}))-\frac{3}{4}a_{k1}^{-\frac{7}{4}}K_{\frac{1}{4}}(\frac{3r_{ij}}{a_{k1}})].$$

The secondary ranges $a_{k2}$ and the anisotropy angles $\theta_{ck}$, for $k = 1, 2$, relate with the templates through the distance $r$. Using the chain rule,

$$\begin{aligned}\frac{dL_{kT}^{G}(i,j)}{da_{k2}} &= \frac{dL_{kT}^{G}(i,j)}{da_{k1}R_k} \\ &= \frac{1}{a_{k1}}\frac{dL_{kT}^{G}(i,j)}{dr_{ij}}\frac{dr_{ij}}{dR_k} \\ &= \frac{1}{a_{k1}}\frac{d}{dr_{ij}}[\sigma(\frac{12}{a_{k1}^2\pi})^{0.5}\exp(-\frac{6r_{ij}^2}{a_{k1}^2})]\frac{1}{2r_{ij}}\frac{dr_{ij}^2}{dR_k} \\ &= -\sigma(\frac{12r_{ij}}{a_{k1}^3})(\frac{12}{a_{k1}^2\pi})^{\frac{1}{2}}\exp(-\frac{6r_{ij}^2}{a_{k1}^2})\frac{R_k}{r_{ij}}(\sin\theta_{c1}i - \cos\theta_{c1}j)^2 \\ &= -\sigma(\frac{24\sqrt{3}R_k}{a_{k1}^4\sqrt{\pi}})\exp(-\frac{6r_{ij}^2}{a_{k1}^2})(\sin\theta_{c1}i - \cos\theta_{c1}j)^2.\end{aligned}$$

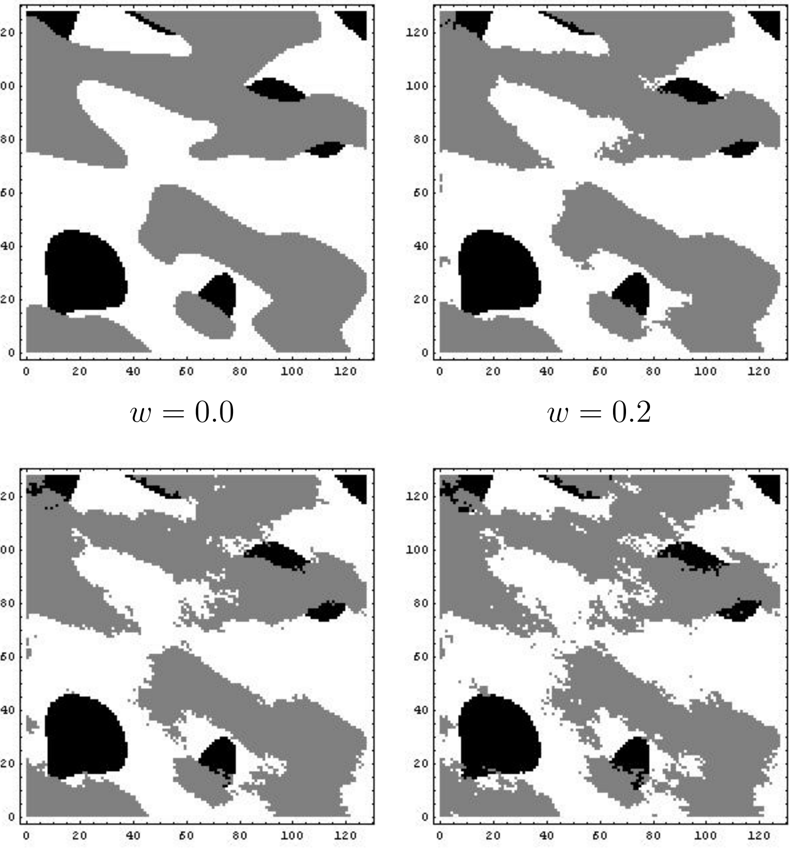
$$\begin{aligned}
\frac{dL^E_{kT}(i,j)}{da_{k2}} &= \frac{dL^E_{kT}(i,j)}{da_{k1}R_k} \\
&= \frac{1}{a_{k1}}\frac{dL^E_{kT}(i,j)}{dr_{ij}}\frac{dr_{ij}}{dR_k} \\
&= \frac{1}{a_{k1}}\frac{d}{dr_{ij}}[\sigma K_{\frac{1}{4}}(\frac{3r_{ij}}{a_{k1}})(\frac{2}{27}\pi^2 a^3_{k1} r_{ij})^{-0.25}\Gamma(\frac{3}{4})^{-1}]\frac{1}{2r_{ij}}\frac{dr^2_{ij}}{dR_k} \\
&= -\sigma(\frac{2}{27}\pi^2 a^3_{k1})^{-0.25}\Gamma(\frac{3}{4})^{-1} \\
&\quad [\frac{3}{2a_{k1}} r_{ij}^{-\frac{1}{4}}(K_{-\frac{3}{4}}(\frac{3r_{ij}}{a_{k1}}) + K_{\frac{5}{4}}(\frac{3r_{ij}}{a_{k1}})) + \frac{1}{4} r_{ij}^{-\frac{5}{4}} K_{\frac{1}{4}}(\frac{3r_{ij}}{a_{k1}})]\frac{R_k}{r_{ij}}(\sin\theta_{c1} i - \cos\theta_{c1} j)^2.
\end{aligned}$$

$$\begin{aligned}
\frac{dL^G_{kT}(i,j)}{d\theta_{ck}} &= \frac{dL^G_{kT}(i,j)}{dr_{ij}}\frac{dr_{ij}}{d\theta_{ck}} \\
&= -\sigma(\frac{12r_{ij}}{a^2_{k1}})(\frac{12}{a^2_{k1}\pi})^{0.5}\exp(-\frac{6r^2}{a^2_{k1}})\frac{1}{r_{ij}}(1-R_k)^2(\cos\theta_{ck} i + \sin\theta_{ck} j)(\cos\theta_{ck} j - \sin\theta_{ck} i) \\
&= -\sigma\frac{24\sqrt{3}}{a^3_{k1}\sqrt{\pi}}\exp(-\frac{6r^2_{ij}}{a^2_{k1}})(1-R_k)^2(\cos\theta_{ck} i + \sin\theta_{ck} j)(\cos\theta_{ck} j - \sin\theta_{ck} i).
\end{aligned}$$

$$\begin{aligned}
\frac{dL^E_{kT}(i,j)}{d\theta_{ck}} &= \frac{dL^E_{kT}(i,j)}{dr_{ij}}\frac{dr_{ij}}{d\theta_{ck}} \\
&= -\sigma(\frac{2}{27}\pi^2 a^3_{k1})^{-0.25}\Gamma(\frac{3}{4})^{-1}[\frac{3}{2a_{k1}} r_{ij}^{-\frac{1}{4}}(K_{-\frac{3}{4}}(\frac{3r_{ij}}{a_{k1}}) + K_{\frac{5}{4}}(\frac{3r_{ij}}{a_{k1}})) + \frac{1}{4} r_{ij}^{-\frac{5}{4}} K_{\frac{1}{4}}(\frac{3r_{ij}}{a_{k1}})] \\
&\quad \frac{(1-R_k)^2}{r_{ij}}(\cos\theta_{ck} i + \sin\theta_{ck} j)(\cos\theta_{ck} j - \sin\theta_{ck} i).
\end{aligned}$$

for $k = 1, 2$.

An experiment on the impact of the weighting term $w$ to the features of the facies realizations is shown in Fig. 16. The $w$ term adjusts the roughness of the facies boundaries and has no obvious impact on the distribution of the facies clusters on the large scale. Some geological features may require larger $w$, but this makes the matching process difficult. As the facies shapes on the large scale do not seem affected by $w$, $w = 0$ is first assumed and a match to the training image is performed using



$w = 0.0$ $w = 0.2$

$w = 0.4$ $w = 0.6$

**Figure 16:** The facies boundaries are getting more jagged with increasing of the weighting term $w$ of the exponential type covariance.

a smooth model. After all other parameters have been decided by minimizing the facies mismatch, a $w$ value is estimated based on the roughness of facies boundaries in the training image.

## *6.4 Exploration on optimization of threshold lines*

The gradient of the facies mismatch to model parameters has shown that the objective function for facies mismatch is far more sensitive to the parameters deciding threshold lines than to those deciding random Gaussian fields, which indicates that adjustment of $\theta$s and $r$s mostly controls the optimization process. So the first step towards a complete optimization of all parameters can be taken by fixing the two Gaussian random fields as the "true" and leaving the threshold parameters as the only set of variables to be optimized. Another purpose for leaving the Gaussian random field fixed is to be able to test the validity of the gradient of the objective function about threshold parameters. By fixing the Gaussian random fields, the optimization problem becomes fairly small (with only 6 variable parameters) and it is affordable to calculate the inverse of the Hessian directly, such that the Levenberg-Marquardt(LM) algorithm could be used instead of L-BFGS.

### 6.4.1 Case 1: using true $Z_1$ and $Z_2$

The optimization problem was applied on a fine grid field of $128 \times 128$. Prior experience with this research has proven that a coarse grid, such as $10 \times 10$, would make the optimization rather difficult. The objective function is the squared difference between the facies realization map and a given training image with facies observations at each grid. A set of threshold lines was generated randomly as a prior model, and thus contains no prior knowledge about the truth. By truncating the "true" Gaussian random fields with this prior threshold model, the objective function and a search direction were calculated. After 10 steps of LM iterations, the objective function was reduced from the prior 7383 to 68. The total number of gridblocks is $128 \times 128 = 16384$, and it is reasonable to accept the convergence when less than 1% of the total gridblocks have facies type different from the truth.

The reduction of the objective function with LM iterations is shown in Fig. 17. In Fig. 18, the threshold model after the tenth step and its corresponding facies distribution were compared with the truth case. The data mismatch of 68 is small enough that the difference between the calculated and the truth can hardly be recognized. Three sets of threshold lines, the random prior, the posterior and the true are shown

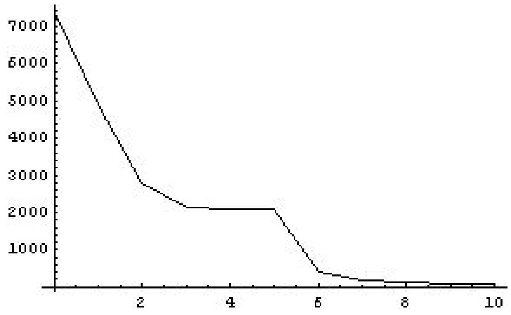
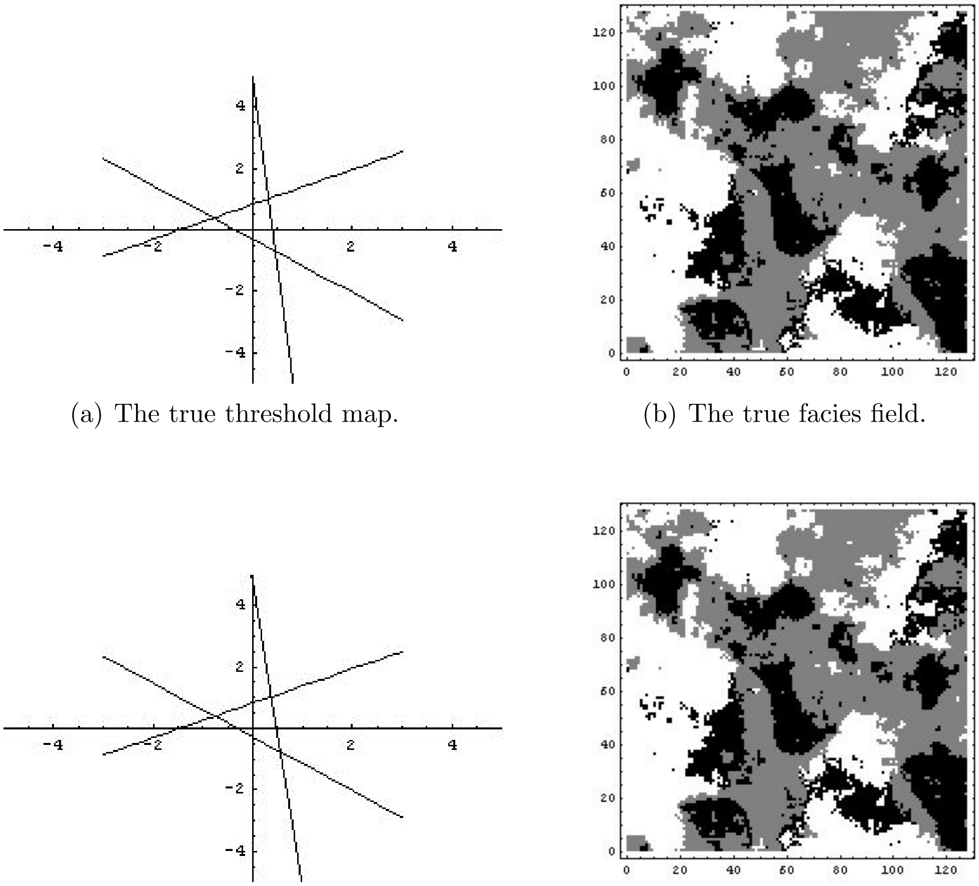


**Figure 17:** The reduction of the objective function with LM iterations.

in Fig. 19. A slight mismatch of the posterior threshold and the truth could be observed and the lines in the posterior model have been shifted far from their prior location.

### 6.4.2 Case 2: unknown true $Z_1$ and $Z_2$

Given a map of lithofacies distribution, either from a geologist's experience, or from an outcrop, a set of geosatistical parameters which would generate a lithofacies realization with similar features and appearance can be estimated by the LBFGS optimization iterations. Unfortunately, a single estimate of the parameters is unlikely to provide a reasonable characterization of the uncertainty. Simple reasoning, for example, tells us that there are some symmetries to the threshold map. Because the variability in the geostatistical parameters is expected to be large (and non-Gaussian), the pdf of the threshold line parameters can be characterized using the method of randomized maximum likelihood. In this method, unconditional realizations of model parameters $Z_1, Z_2, R, \Theta$ are generated, and the model parameters are adjusted to minimize the difference between the true (or training) facies map and the predicted facies map from the current model. If this is done a large number of times, an empirical estimate of the pdf for threshold line parameters conditional to the training image can be developed.



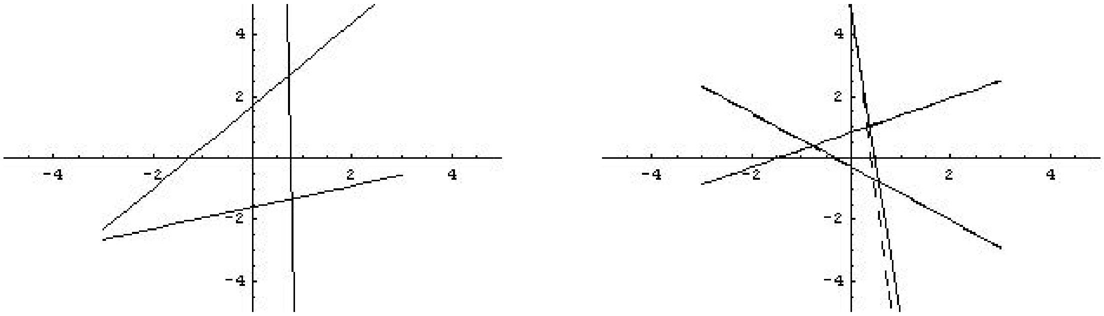
(a) The true threshold map.

(b) The true facies field.

(c) Calculated threshold map.

(d) Calculated facies field.

**Figure 18:** The comparison of an optimized estimation after 10 LM iterations with the "true" threshold map and facies field.



(a) The prior threshold model.

(b) Comparison between calculated and the truth.

**Figure 19:** Comparison of the prior, the posterior and the true threshold map. Solid lines are stochastic model estimations and dashed lines are the truth.

The objective function for this problem contains both the squared data mismatch term and the squared model mismatch term, as shown in Eq. 17. The model mismatch part was used to prevent singularity of the inverse problem. As the mean or any prior knowledge of the model parameters are unavailable, the initial guess of model parameters was used as a constraint, consistent with application of Randomized Maximum Likelihood,

$$O(m_k) = \frac{1}{2}(m_k - m_{\text{pr}})^T C_M^{-1}(m_k - m_{\text{pr}}) + \frac{1}{2}(F(m_k) - F_{\text{obs}})^T C_D^{-1}(F(m_k) - F_{\text{obs}}). \quad (48)$$

There is little information of a reasonable magnitude of the error in a facies observation, especially for a map created by the experience of a geologist. Therefore, we simply assumed the covariance matrix of the facies observations $C_{\text{D}} = 1$ and seemed to obtain reasonable answers.

As an example synthetic problem, the reservoir model is two dimensional and heterogeneous, with the size 300 ft ×300 ft. It is discretized into $128 \times 128$ gridblocks. Three threshold lines are used for truncation and three lithotypes were assigned to the threshold map. Both $y_1$ and $y_2$ have isotropic Gaussian type covariance with same range of 90 ft. The model variables were defined on an augmented grid whose dimensions were $192 \times 192$. Because two of these grids are need for truncation there are approximately 73,000 variables and 16,000 data in the problem.

Three threshold lines were used for truncation and three lithotypes were assigned to the threshold map. Fig. 20 shows the facies distribution from the initial guess of model parameters. The initial facies mismatch is 12525, i.e. 76% of the gridblocks have the wrong facies types compared to the training image [Fig. 23 (left)].

This problem is far too large for Levenberg-Marquardt, so the limited memory version of the BFGS (L-BFGS) algorithm (Nocedal, 1980) was used for the minimization of the objective function because it requires storage of only a small set of

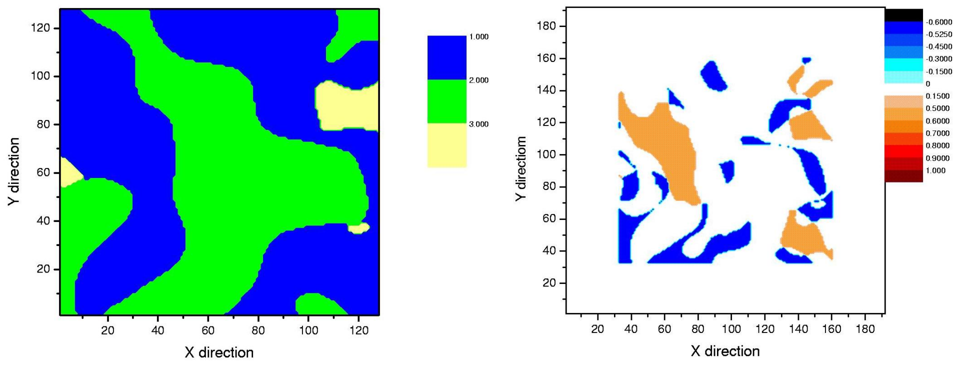


**Figure 20:** The initial facies map (left) and the gradient of the objective function with respect to the field evaluated at the initial map (right).

vectors instead of the whole inverse Hessian matrix. The inverse Hessian is iteratively updated based on computations of the gradient, for which an adjoint method with automatic code generation (Corliss et al., 2001) was used.

The LBFGS algorithm is one of the most popular quasi-Newton methods. It is based on generating an approximation of the inverse of Hessian matrix from computations of the gradient. For problems with large number of model parameters, like the problem this section is dealing with, it becomes impossible to even store the Hessian matrix. The LBFGS algorithm stores only a set of vectors instead of the whole Hessian matrix, and produces search direction with great convergence efficiency.

Let $g_k$ be the $k^{th}$ step gradient calculated from the adjoint method, $L$ be the maximum number of vectors to be stored, which has to be fixed as constant, and $k$ be the current iteration number. The Nocedal LBFGS algorithm is as follows.

1. If $k \leq L$, set $incr = 0$ and $bound = k$; else set $incr = k - L$ and $bound = L$

2. $q_{\text{bound}} = g_k$

3. For $i = bound - 1, \dots, 0$

$$\begin{cases} j = i + incr \\ \alpha_i = \rho_j s_j^T q_{i+1} \\ q_i = q_{i+1} - \alpha_i y_j \end{cases}$$

$$r_0 = \widetilde{H}_0^{-1} \times q_0$$

4. For $i = 0, 1, \dots, bound - 1$

$$\begin{cases} j = i + incr \\ \beta_j = \rho_j y_j^T r_i \\ r_{i+1} = r_i + s_j(\alpha_i - \beta_i) \end{cases}$$

5. $d_k = -r_{\text{bound}}$

where

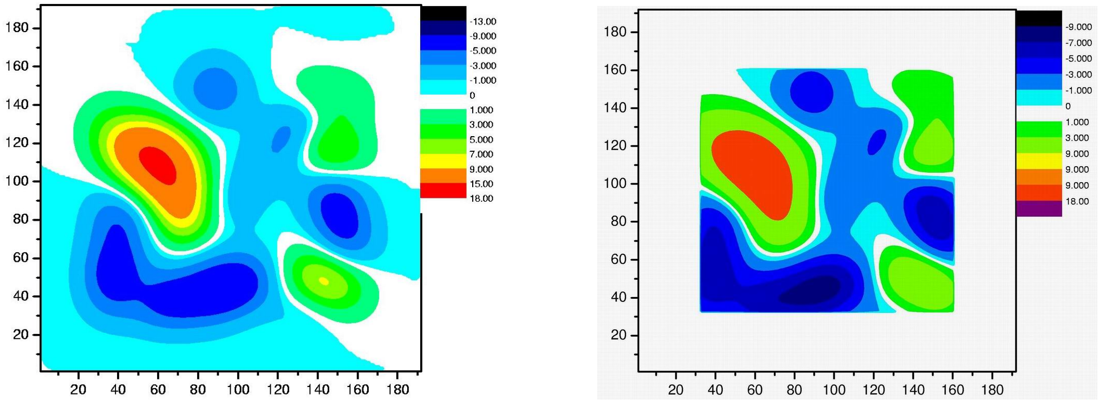
$$\begin{aligned} s_j &= m_{j+1} - m_j \\ y_j &= g_{j+1} - g_j \\ \rho_j &= 1/(y_j^T s_j) \end{aligned}$$

$d_k$ is the search direction given by $-\widetilde{H}_k^{-1} \times g_k$ if Newton's method is used.

Fig. 20 (right) shows the gradient of the objective function with respect to $Y_1$. The gradient of the objective function to $Y_1$ clearly reflects the regions of high sensitivity. Comparing the right and left sub-figures in Fig. 20, it may concluded that the mismatch function is most sensitive to changes in the values of the random variables that are near facies boundaries. The width of the region of sensitivity depends on the width of the transition region and its choice affects the rate of convergence.

Fig. 21(a) is the output gradient of the objective function with respect to the Gaussian random field $Z_1$. Because the scheme of list convolution was used, the

Gaussian random fields $Z_1$ and $Z_2$ both have greater dimensions than the true gridblocks. From Eq. 28, the gradient of the objective function with respect to $Z_1$ can also be calculated by convolution of the Gaussian template with the gradient of objective function to $Y_1$. So the field in Fig. 20(right) was convolved with the Gaussian template and the result is plotted in Fig. 21(b). As expected, Fig. 21(a) and Fig. 21(b) have very similar values. The difference of the two plots is in the frame areas, which are largely due to numerical error brought by the difference in the list convolution routine and its adjoint routine. The plot with sensitivity values in the frame area is more reasonable.



(a) Gradient of objective function to $Z_1$ field from adjoint code.

(b) Gradient of objective function to $Z_1$ field from chain rule.

**Figure 21:** The resulting gradient field of $Z_1$ from convolution of the Gaussian template with the gradient of objective function to $Y_1$ field.

In applying the LBFGS method, the method would frequently become stuck at a fairly large value of the objective function, and that the line search implementation was not trivial. After experimenting with several methods, a simple investigation of the behavior of the objective function in the descent direction for the first iteration was performed. The reduction of the objective function in the first search direction is shown in Fig. 22. The discontinuous reduction in the objective function curve is due to a switching of lithotypes in different regions of the threshold map, when the modification on the threshold line parameters caused the lines to intersect in a different way.

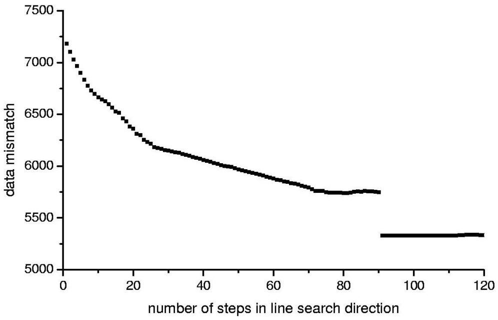


**Figure 22:** Shape of initial objective function in first search direction.

Finally the training image is shown together with the final facies map in Fig. 23. The facies in 95.6% of the gridblocks in the final result matched the facies in the training image.

The minimization procedure was reported 200 times; each procedure started with unconditional realizations of the parameters describing locations of the threshold lines and unconditional realizations of the random variables on the grid. The Randomized Maximum Likelihood (RML) method was used for the sampling, as it seems to do a relatively good job of sampling for this type of problem (see Liu and Oliver, 2003a).

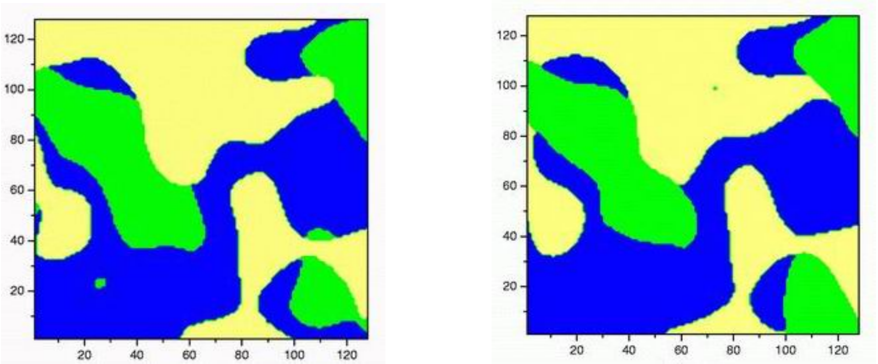


**Figure 23:** Comparison of the facies map generated from the minimization of the objective function (right) with the training image (left).

In approximately half of the cases the resulting value of the objective function seemed satisfactory (less than 2000). Fig. 24 summarizes the distribution of realizations of the orientations of the first two threshold lines.

Although the orientation of the threshold lines were sampled randomly from a uniform distribution, the facies were assigned in a non-random and non-uniform manner. As a result, it was necessary to arrange the threshold lines in the order $\theta_1 < \theta_2 < \theta_3$. Once this is done, the distributions of orientations are no longer uniform. It would be unlikely, for example, for the smallest of the three angles ($\theta_1$) to be close to $\pi$, and this is seen in Fig. 24. From the two plots, it is not apparent that the estimate of the posterior distribution for $\theta_1$ or $\theta_2$ (indicated by the black squares) is significantly different from the prior distribution (indicated by the black triangles). If there had been rotational symmetry to the threshold plots, the orientations to cluster along lines in cross-plots of variables should be expected. Cross-plots of the threshold line parameters from the 109 accepted conditional samples failed, however, to show any obvious patterns. In order to honor the proportions of facies correctly, there must be some fairly strong constraints on the relationships among the threshold line parameters, but they are not obvious from the conditional realizations. 2500 conditional threshold models were generated in addition to the 109 models, and the cross-plots are presented in Fig. 25. Again, no obvious patterns are revealed. It would be easy, however, to use the realizations generated in this procedure in a Monte Carlo method, in which case the true sampling space of model parameters does not have to be known.

### 6.4.3 Experiment with automatic adjoint system generator

Among the methods of optimization of large number of model parameters, the adjoint method is one of the most powerful methods for computing the gradient of the

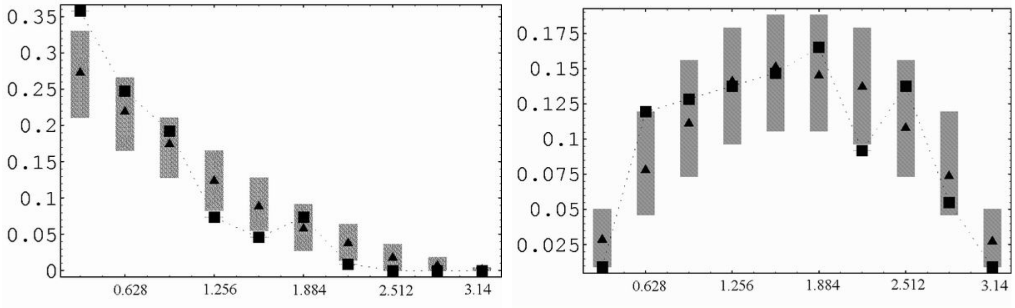
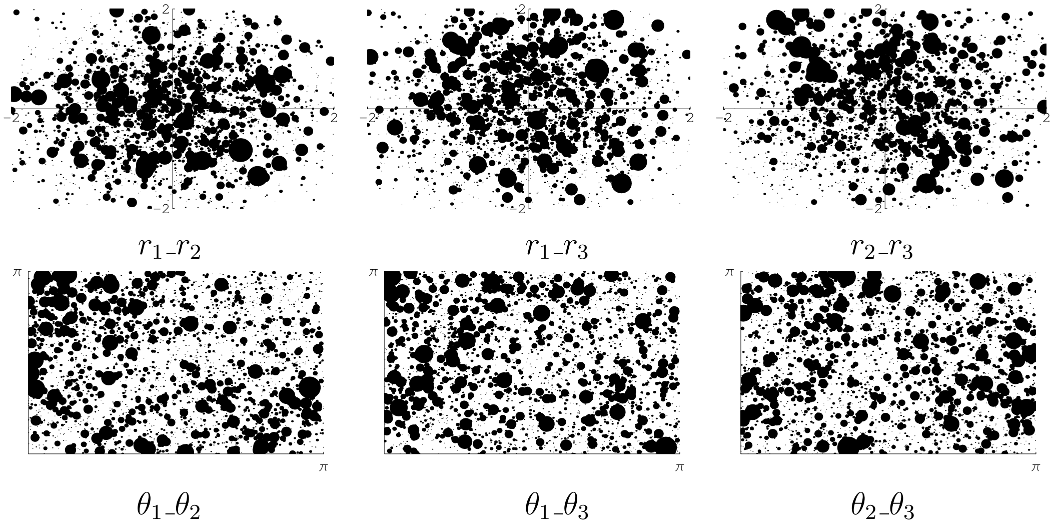


**Figure 24:** Comparison of the estimated pdf (squares) for $\theta_1$ (left) and $\theta_2$ (right) to the prior pdf (triangles) The width of the gray bars indicates the variability due to limited sample size of 109 ordered sets of 3 orientations from a uniform distribution. 10% are higher and 10% are lower.



$r_1\_r_2$ $r_1\_r_3$ $r_2\_r_3$

$\theta_1\_\theta_2$ $\theta_1\_\theta_3$ $\theta_2\_\theta_3$

**Figure 25:** Cross-plots of the conditional threshold line models.

objective function with respect to model parameters. However, construction of the adjoint system for large history matching problems is not trivial.

Software tools for automatic generation of code to compute derivatives using the adjoint method have been developed at several national laboratories in the U.S. and in France (Corliss et al., 2001). Among these software tools, TAPENADE[1] was designed for "reverse mode AD", i.e. computation of gradient of the objective function with respect to model parameters based on adjoint method. Providing a Fortran source program, the name of the top routine to be differentiated, the dependent output variables whose derivatives are required and the independent input variables with respect to which it must differentiate, this tool returns the reverse (adjoint) differentiated program. TAPENADE can be either utilized as a server (JAVA servlet) or be installed locally as a set of JAVA classes (JAR archive). In that case it is run by a simple command line, which can be included into a Makefile.

Here, an example of generating reverse mode of a Fortran 77 subroutine by the TAPENADE online server is presented. The subroutine CNVLVE does array convolution by list convolving an input template named MASK with the input array, ARRAY. The input variables M and N are the number of rows and number of columns of ARRAY respectively. MMASK and NMASK are the dimensions of the MASK. OUTPUT is the result from convolution with the same dimensions as ARRAY.

[1] Available at www-sop.inria.fr/tropics/tapenade.html

```
!—- List Convolve Code ————————————————————*
SUBROUTINE CNVLVE( ARRAY, M, N, MASK, MMASK, NMASK, OUTPUT
)
INTEGER M, N, MMASK, NMASK REAL ARRAY(M,N),
MASK(MMASK,NMASK), OUTPUT(M,N) INTEGER I, J, IM, JM, MOFF,
NOFF REAL SUM
! Calculate MOFF and NOFF. These offsets into the image are used
! to leave an edge around the output image. We leave them alone;
! the user must explicitly do something about it, either setting
! it to zero, to the mean of the image, or whatever.
MOFF = (MMASK-1) / 2
NOFF = (NMASK-1) / 2
! Scan through ARRAY. At each position, evaluate the sum of the
! products of array pixels with the mask. When we've looked at
! all the pixels, put this sum into the output array.
DO J = NOFF+1, N-NOFF
DO I = MOFF+1, M-MOFF
SUM = 0.0
DO JM = 1, NMASK
DO IM = 1, MMASK
SUM = SUM + ARRAY(I+IM-(MOFF+1),J+JM-(NOFF+1))
* MASK(IM,JM)
END DO !IM
END DO !JM
OUTPUT(I,J) = SUM
```

```
(cont'd)
END DO ! I
END DO ! J
RETURN
END
```

The adjoint code generated from TAPENADE first repeats the programs in the source code CNVLVE and pushes intermediate computation results into a memory stack. In the second part of the adjoint code, it pops out those intermediate variables from the memory stack. Though some of the stack memory "pushes" and "pops" are unnecessary for obtaining the desired gradients, some are vital information and had to be saved from the forward computation. The memory stack operation subroutines such as PUSHINTEGER4, POPINTEGER4 etc., are provided by TAPENADE together with the generated adjoint code. However, these stack operation subroutines are in the C language. This brought two problems to WINDOWS system users. First, they have to own both Fortran and C compilers to possibly compile their Fortran source code and C package. Second, the visual Fortran platforms for WINDOWS system, such as COMPAQ VISUAL STUDIO and MICROSOFT VISUAL STUDIO, are good at calling Fortran subroutines by C, but not the other way around. In this research, all the computations using adjoint code from TAPENADE were achieved on IBM Regatta high performance computers.

```
! Generated by TAPENADE (INRIA, Tropics team)
! Version 2.0.2 -(Id: 1.11 vmp Stable - Tue Mar 11 10:35:48 MET 2003)
! Differentiation of adjoint in reverse (adjoint) mode:
! gradient, with respect to input variables: array mask
! of linear combination of output variables: output
! ————————————————————————————
SUBROUTINE CNVLVE_B(array, arrayb, m, n, mask, mmask, nmask, output,
outputb)
INTEGER m, mmask, n, nmask
REAL array(m, n), arrayb(m, n), mask(mmask, nmask), output(m, n),outputb(m,
n)
INTEGER adfrom, adfrom0, adto, i, ii, ii0, im, j, jm, moff, noff REAL sum, sumb
moff = (mmask-1)/2
noff = (nmask-1)/2
adFrom0 = noff + 1
DO j=adFrom0,n-noff
adFrom = moff + 1
DO i=adFrom,m-moff
sum = 0.0
DO jm=1,nmask
DO im=1,mmask
sum = sum + array(i+im-(moff+1), j+jm-(noff+1))*mask(im,jm)
ENDDO
CALL PUSHINTEGER4(im - 1)
ENDDO
```

(cont'd)

```
CALL PUSHINTEGER4(jm - 1)
output(i, j) = sum
ENDDO
CALL PUSHINTEGER4(i - 1)
CALL PUSHINTEGER4(adFrom)
ENDDO
CALL PUSHINTEGER4(j - 1)
CALL PUSHINTEGER4(adFrom0)
DO ii0=1,n
DO ii=1,m
arrayb(ii, ii0) = 0.0
ENDDO
ENDDO
CALL POPINTEGER4(adFrom0)
CALL POPINTEGER4(adTo)
DO j=adTo,adFrom0,-1
CALL POPINTEGER4(adFrom)
CALL POPINTEGER4(adTo)
DO i=adTo,adFrom,-1
sumb = outputb(i, j)
outputb(i, j) = 0.0
CALL POPINTEGER4(adTo)
DO jm=adTo,1,-1
CALL POPINTEGER4(adTo)
```

```
(cont'd)
DO im=adTo,1,-1
arrayb(i+im-(moff+1), j+jm-(noff+1)) = arrayb(i+im-(moff+1
), j+jm-(noff+1)) + mask(im, jm)*sumb
ENDDO
ENDDO
ENDDO
ENDDO
END
```

In reservoir history matching problems, construction of the adjoint system for obtaining the gradient of the objective function with respect to model parameters always takes great effort. To minimize the effort required for generation of the adjoint code, the automatic gradient generator, TAPENADE, was applied to this problem. It managed to calculate the gradient of facies mismatch with respect to all the model paramters ($Z_1, Z_2, R$ and $\Theta$). Because TAPENADE is designed as a very general tool for differentiation of any kind of Fortran statements, the adjoint code it generated is not "plug and play" and needs to be edited carefully before use. Thus for complicated problems, it still takes a lot of work to understand its adjoint computation. Later application of the adjoint code generator for computing the gradient of the production rate mismatch with respect to the random Gaussian fields was not successful.

### 6.4.4 Limitations on perturbation of threshold slope angles

There are some potential problems here with the way the threshold lines are defined. The initial set of the slope angles are generated from a uniform distribution $U(0, \pi)$ and defined as double precision numbers, so it is highly unlikely for any two of the slope angles to have the same value, or for any of them to exactly equal to $\frac{\pi}{2}$. During perturbation, however, the threshold lines need to be freely rotated and sometimes problems may occur in calculation.

1. Parallel: When any of the two threshold lines are parallel, there will be only six areas in the threshold map, and the coordinates of one of the intersections will become extremely large. This problem can be avoided by checking the values of $\theta$ and $\delta\theta$ when updating model parameters. If $|\theta_i - \theta_j| < \epsilon'$, then

   $$\theta_i = \theta_i + \text{Sign}(\delta\theta_i)\epsilon' \tag{49}$$

   $$\theta_j = \theta_j + \text{Sign}(\delta\theta_j)\epsilon' \tag{50}$$

   where $\epsilon'$ should not be greater than $10^{-8}$.

   However, associated with thresholds getting parallel, there is a facies switching problem as illustrated in Fig. 26. The threshold map on the left is before perturbation and the one on the right is after one threshold took an exaggerated slope angle increase. By adjusting one slope angle to be greater than another, at least 4 areas have been dramatically shifted (area 2, 3, 6 and 7). Consequently, the facies assigned to the gridblocks mapped in these area were changed and the objective function soared abruptly. The gradient based optimization process usually does not accept increasing of the objective function, this caused the threshold line sticking in the position of nearly parallel with another.

2. $\theta_i = \frac{\pi}{2}$: From the definition of threshold lines in Eq. 40, the $\cos\theta$ term in the denominator can not equal to zero, i.e. $\theta_i$ should avoid being equal to $\frac{\pi}{2}$ in double precision. The solution to this problem is the same as in Eq. 49. No facies switching problem is involved here.

3. $\theta_i \leq 0$ or $\theta_i \geq \pi$: The equations for sensitivity calculations in the last section requires $\theta \in (0, \pi)$. So another check should be applied besides those mentioned

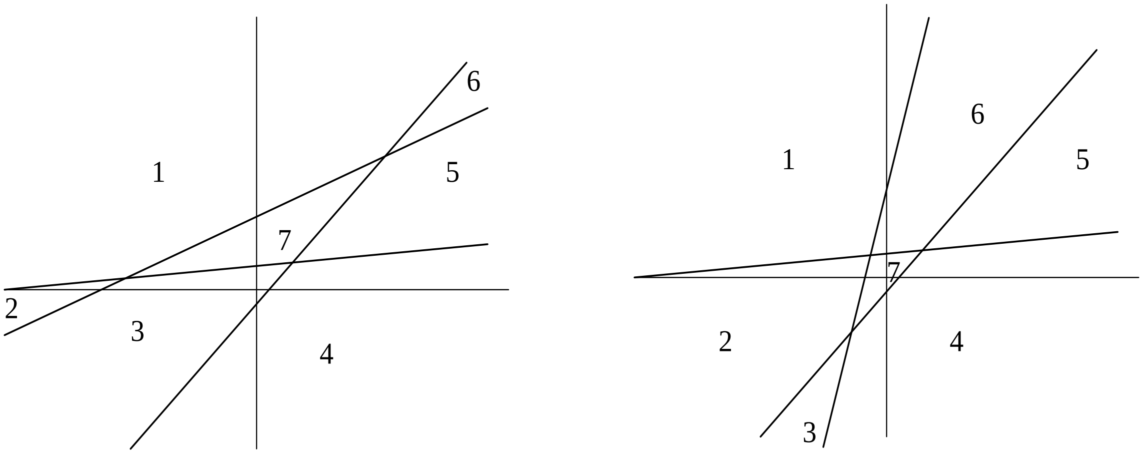


**Figure 26:** Facies switching caused by changing relative value of the slope angles.

in the previous two items. After updating the model parameters,

$$\theta_i = \begin{cases} \theta_i + \pi & \text{if } \theta_i < 0 \\ \pi - \epsilon' & \text{if } \theta_i = 0 \\ \epsilon' & \text{if } \theta_i = \pi \\ \theta_i - \pi & \text{if } \theta_i > \pi. \end{cases}$$

There is also a facies switching problem related with a slight rotation passing through the slope angle 0 or $\pi$, which is illustrated in Fig. 27. The slope angle of one of the threshold lines was a little less than $\pi$ in the figure on the left, and was increased to be greater than $\pi$ as shown on the right. Though the value of the slope angle can be fixed by subtracting $\pi$, the labelled regions from 1 to 6 are altered. Again, this causes an increase in the objective function and the minimization process will be stuck in this situation. However, the solution is more obvious here than in item 1. If the line passes through 0 or $\pi$ from positive direction, i.e. $\theta_i$ increases, the labels are switched on facies areas clockwise, and vice versa.

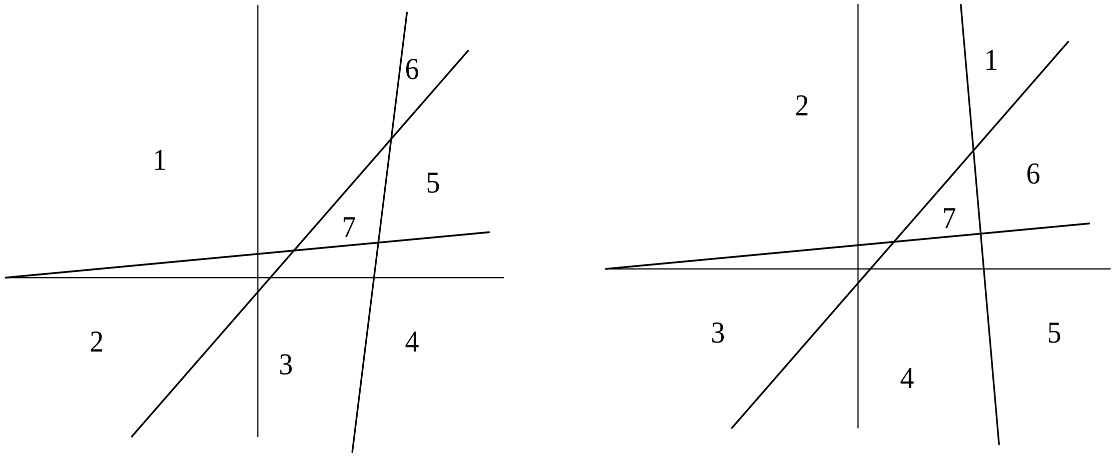


**Figure 27:** Rotation of facies assignment due to one slope angle increased greater than $\pi$.

## 6.5 Estimation of covariance parameters

This section presents the study on estimation of the probability distribution of the covariance parameters from a given training image. Although most of the softwares on geostatistical analysis and simulation provides functions for estimation of the covariance parameters, such as the covariance type, the principle ranges, and the anisotropy angle, it is an unusual problem to estimate the covariance parameters for the underlying Gaussian fields in truncated plurigaussian approach. The uniqueness of the problem lies in two aspects:

1. The direct measurement in the training image is the facies type, which is a discontinuous indicator variable. It is straightforward to generate the experimental variogram from the training image for the facies, however, the facies types do not relate to any of the Gaussian fields linearly because of the facies assignment scheme in the threshold map.

2. The facies type at each location is decided by two or more Gaussian variables in the truncated plurigaussian approach. The idea of estimation of the spatial

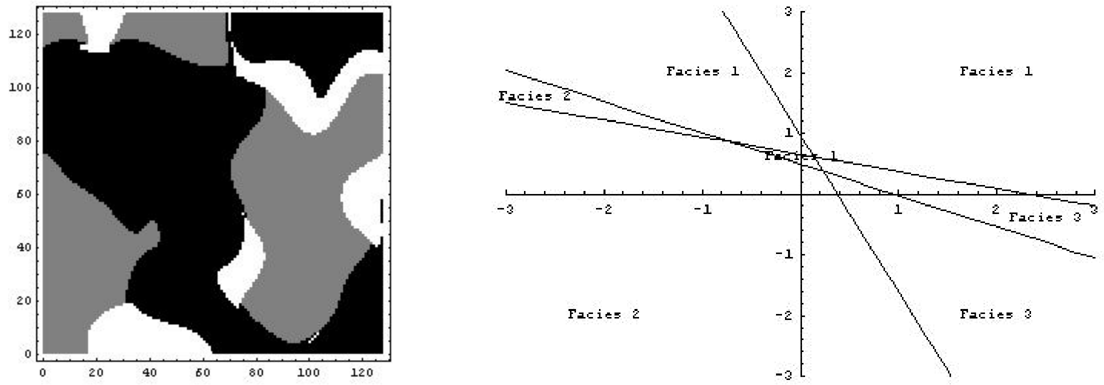
correlation of the Gaussian fields is to solve an underdetermined problem.

Therefore, the training image barely provides a qualitative spatial correlation of the Gaussian fields.

This study takes the Baysian approach in estimation of the geostatistical parameters for the Gaussian fields. The a posteriori PDF of the model parameters are generated using the randomized maximum likelihood (RML) algorithm by minimization of the objective function as shown in Eq. 48. The objective function is composed of the squared model mismatch and the squared data mismatch. Unlike the previous study on estimation of the threshold line parameters, the covariance parameters are unknown and are to be estimated in this section.

### 6.5.1 Known threshold map

Previous experience in estimation of the threshold parameters has shown that the diagonal elements of the model covariance term in the objective function cannot just be the variance of the prior distribution of the model parameters. The random variables deciding the Gaussian fields, $Z_1$ and $Z_2$, are not in the same category with the threshold line parameters, and have much less impact on the objective function. The prior distributions of the underlying random variables $Z_1$ and $Z_2$ both are Gaussian with mean 0 and variance 1. Although the prior distribution of the distances of the threshold lines to the origin ($\{r_1, r_2, r_3\}$) is the positive half of $N(0, 1)$, their corresponding weighting term in the prior model covariance matrix was set as 1000. When the weighting terms for the threshold parameters are low ($1 - 500$) in the prior model covariance matrix, the perturbations to model parameters in the optimization process were exclusively made on the threshold line parameters, and the modifications to the random variables $Z_1$ and $Z_2$ are too small to take any effect within a reasonable number of optimization iterations.



(a) The training image

(b) The "true" threshold map

**Figure 28:** The given training image for estimation of covariance parameters, and the threshold map that used to generated the training image.

When the geostatistical parameters of the covariances of the Gaussian fields are considered, the relative weighting of the covariance parameters to the threshold parameters becomes an issue to investigate. So the study for estimation of the covariance parameters starts with the assumption that the threshold map used for generating the training image ("true threshold map") is known.

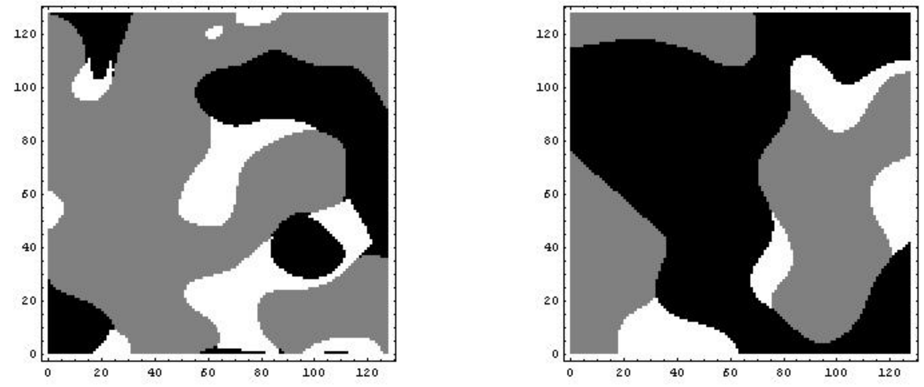
Fig. 28(a) shows the training image, in which the black area represents facies 1, grey for facies 2, and white for facies 3. The dimensions of the training image are 128 × 128, with each gridblock size 40 ft × 40 ft. Both the Gaussian fields $Y_1$ and $Y_2$ for generating the training image have isotropic Gaussian type covariance and a range of 1536 feet. The correlation ranges for both Gaussian fields are 30% of the facies map width, such that there is enough periodicity for the spatial correlation of the Gaussian fields, while avoiding generation of small facies clusters that are difficult to match. The "true threshold map" is shown in Fig. 28(b).

From the observation of a training image, prior estimations of the covariance parameters become available. Assume the prior estimations of the principle ranges are in Gaussian distributions with the mean the same as the true ranges (1536 ft) and the standard deviations as 10% of the range distances. The prior estimation of the anisotropy angles of both the Gaussian fields are also in Gaussian distributions with the same mean as the true angle (0 degrees for isotropic Gaussian fields), and 5 degrees standard deviation.

Although the dimensions of the Gaussian fields $Y_1$ and $Y_2$ are $128 \times 128$, the dimensions of the underlying model fields $Z_1$ and $Z_2$ are much larger ($204 \times 204$) because of the correlation length of the covariances and the moving average algorithm used for generating Gaussian fields. The correlation ranges and the anisotropy angles are changing in the optimization process. However, the dimensions of $Z_1$ and $Z_2$ have to be fixed throughout the matching process, as $Z_1$ and $Z_2$ are model parameters to be carried through and estimated. Therefore the dimensions of the covariance templates $L_T^1$ and $L_T^2$ for the moving average have to be large enough to allow possible overestimation of the ranges without losing much accuracy for simulation of the facies map. For every iteration, the model parameters are updated and $L_T^1$ and $L_T^2$ are recomputed.

The number of model parameters is 83,238 ($\{Z_1, Z_2, a_{11}, a_{12}, \theta_{c1}, a_{21}, a_{22}, \theta_{c2}\}$), and the number of data is 16,384 ($128 \times 128$). For minimization of the objective function, the adjoint method is used for computation of the gradient of the objective function to the model parameters, and the LBFGS algorithm is used for computation of the search direction. Fig. 29(a) shows a facies map realization simulated from a set of prior model parameters. The initial number of gridblocks that has the facies type different from the training image (facies mismatch) is 11,500, which is 70.2% of the total gridblocks. After 57 iterations, the facies mismatch reduced to 738 (4.5%). The final facies map simulated from the conditioned model parameters is shown in Fig. 29(b).

Comparing Fig. 29(b) with the training image in Fig. 28(a), the major features and most of the small features in the training image are reproduced in the final facies



(a) The facies map from the prior model

(b) The final facies map realization

**Figure 29:** The facies map realizations from the prior model estimation and from the final model conditioned to the training image.

| Parameters | $a_{11}$ | $a_{12}$ | $\theta_{c1}$ | $a_{21}$ | $a_{22}$ | $\theta_{c2}$ |
|---|---|---|---|---|---|---|
| Prior | 1530.2 | 1180.6 | 0.10173 | 1289.4 | 1384.2 | -0.10913 |
| Final | 1240.2 | 862.4 | -1.09170 | 1148.6 | 1222.8 | -1.43212 |

**Table 3:** The covariance parameters before and after matching to the training image.

map. However, the final covariance model parameters reflect increased uncertainty from the prior model PDF. The prior and the final covariance parameters are listed in Table 3. All of the correlation ranges from the conditioned realization are smaller than their prior value, and the anisotropy for both Gaussian fields has increased.

To observe the a posteriori distribution of the covariance parameters, 200 sets of the prior model parameters $\{Z_1, Z_2, a_{11}, a_{12}, a_{21}, a_{22}, \theta_{c1}, \theta_{c2}\}$ are generated from the same prior PDF as the previous prior realization, and optimized to match the training image in Fig. 28(a). Although some sets of the initial model parameters are more difficult to be adjusted than others to match the training image, all the 200 final covariance parameters are used for analysis of the a posteriori distributions of the covariance parameters, regardless of the final facies mismatch value. Fig. 30 shows the histograms of the 200 sets of the covariance parameters. All the distributions of the ranges are biased to values lower than those that generated the training image, and the spreads are wider than the prior distributions. The distributions for the anisotropy angles seem to center at 0 degrees, but also have much wider spread than the priors.

Figure 31 shows the cross-plots of the conditional covariance parameters. Each sub-figure contains 200 points, and the size of each point is decided by the quality of the match. For conditional parameters that are optimized to yield lower objective function values, the dots are larger, and vice versa. These plots indicate not only the conditional parameter distributions, but also the scale of the final objective function. From the cross-plots, the clusters of bigger dots generally coincide with the mode in the histograms of the ranges. Although the histogram of $\theta_{c1}$ has the mode value at 0 degrees, the bigger dots seem to be uniformly distributed from $-\pi$ to $\pi$. The histogram of $\theta_{c2}$ indicates the mode at 1.3, but the cross-plots show that the bigger dots are clustered at -2.

This case study revealed that it is difficult to accurately estimate the covariance parameters, at least for this scale of problem where the size of the field is about three times the correlation range in each direction. The patterns of facies distribution in the training image are relatively insensitive to the covariance parameters, such that a training image generated from two isotropic Gaussian fields is sufficiently duplicated by facies maps that are from anisotropic Gaussian fields with much smaller correlation ranges.

### 6.5.2 Impact of correlation ranges to facies patterns

The example for estimation of the covariance parameters of the two Gaussian fields from a training image demonstrated the wide conditional distribution of the ranges when the correlation ranges are large in comparison to the field size (30%). On the

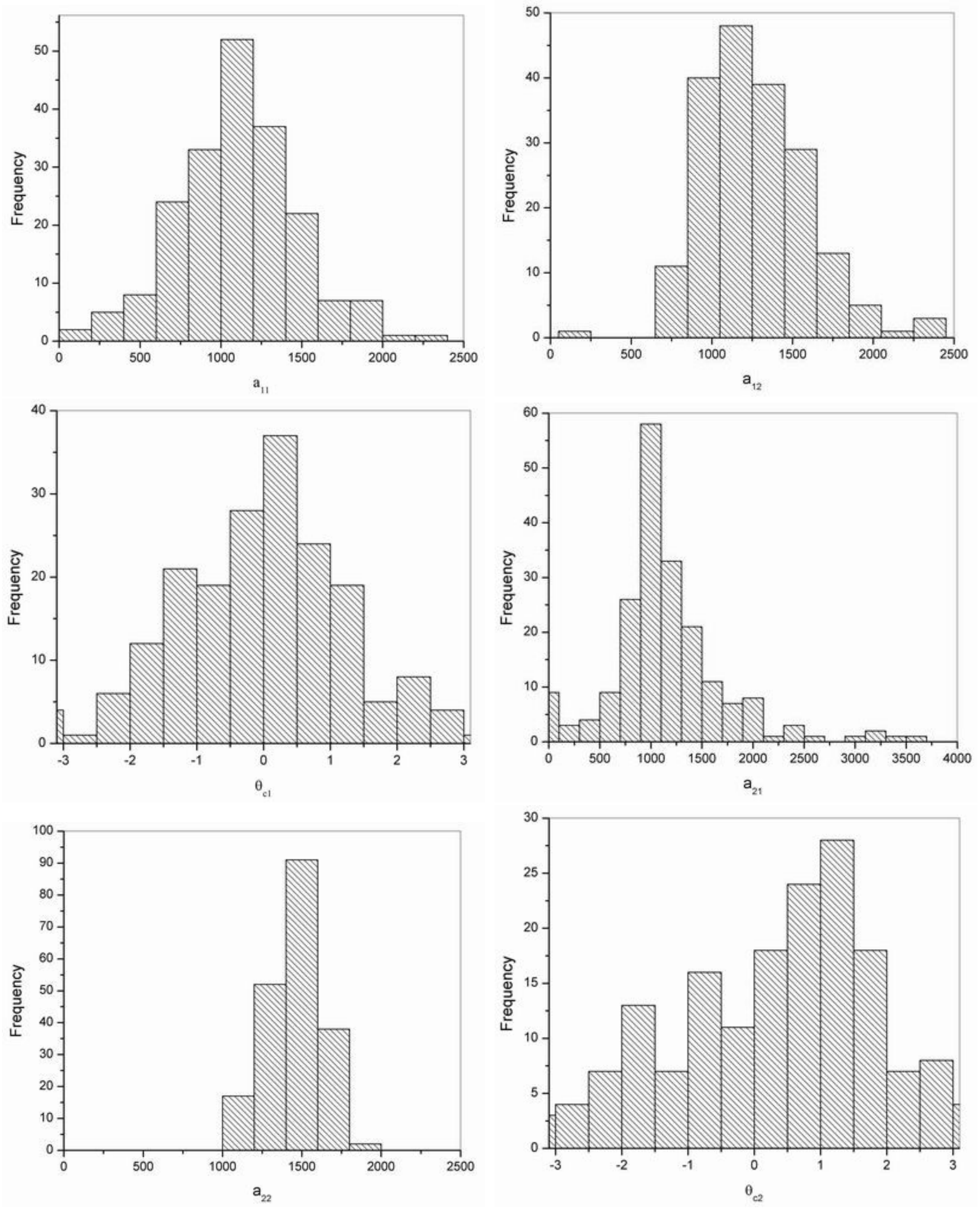
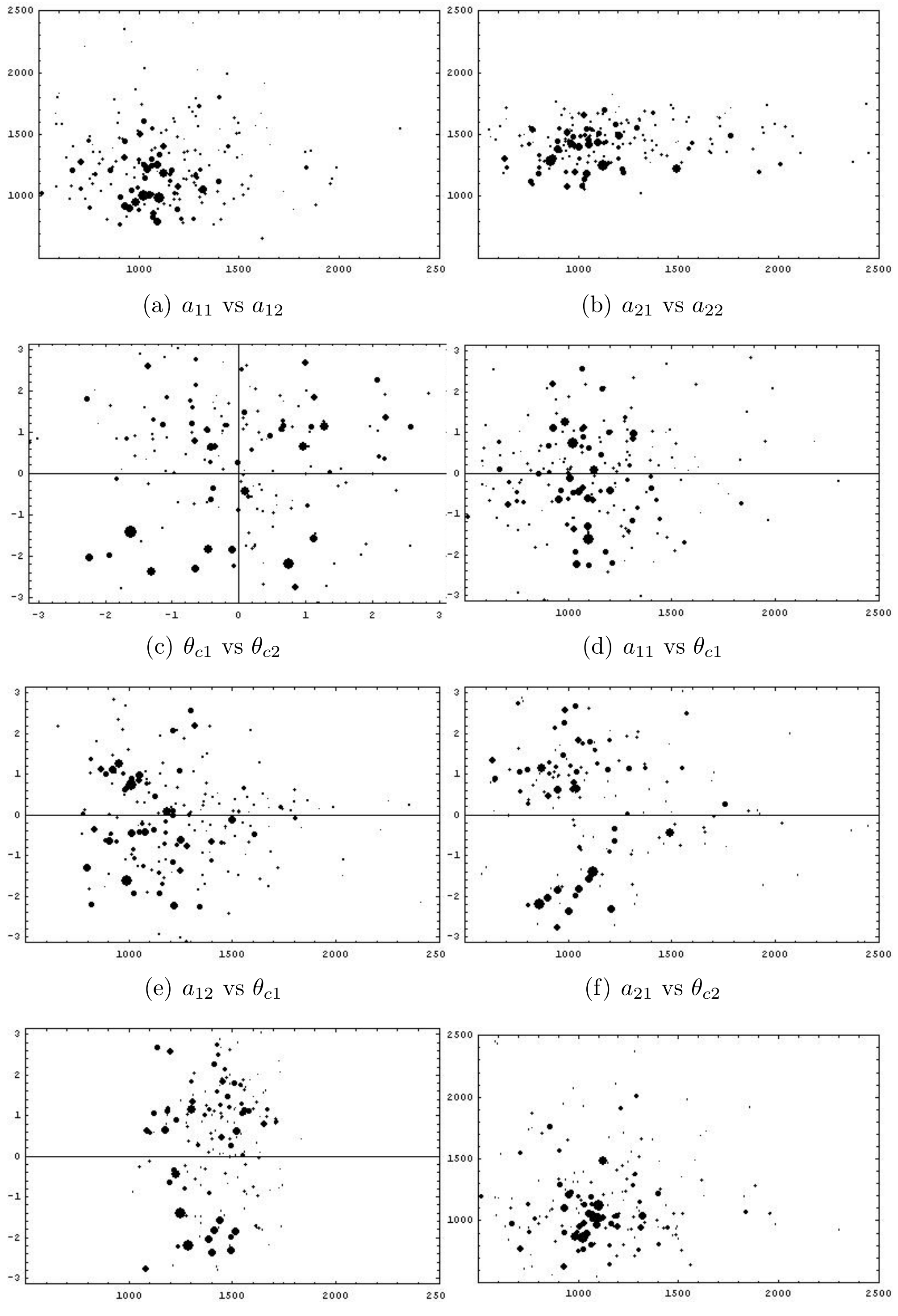


**Figure 30:** Histograms of the 200 conditioned covariance models.



(a) $a_{11}$ vs $a_{12}$

(b) $a_{21}$ vs $a_{22}$

(c) $\theta_{c1}$ vs $\theta_{c2}$

(d) $a_{11}$ vs $\theta_{c1}$

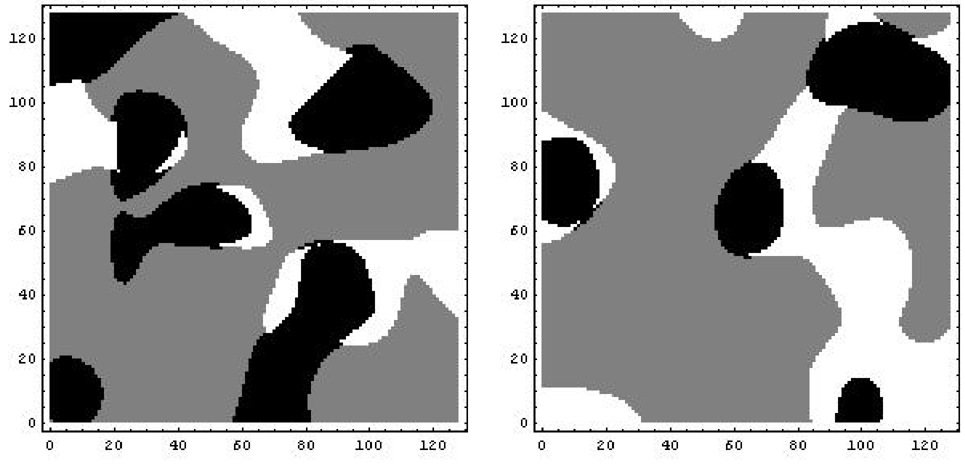
(e) $a_{12}$ vs $\theta_{c1}$

(f) $a_{21}$ vs $\theta_{c2}$

(g) $a_{22}$ vs $\theta_{c2}$

(h) $a_{11}$ vs $a_{21}$

**Figure 31:** Cross-plots of the conditional covariance parameters. The size of the dots is proportional to the matching quality to the training image.



(a)

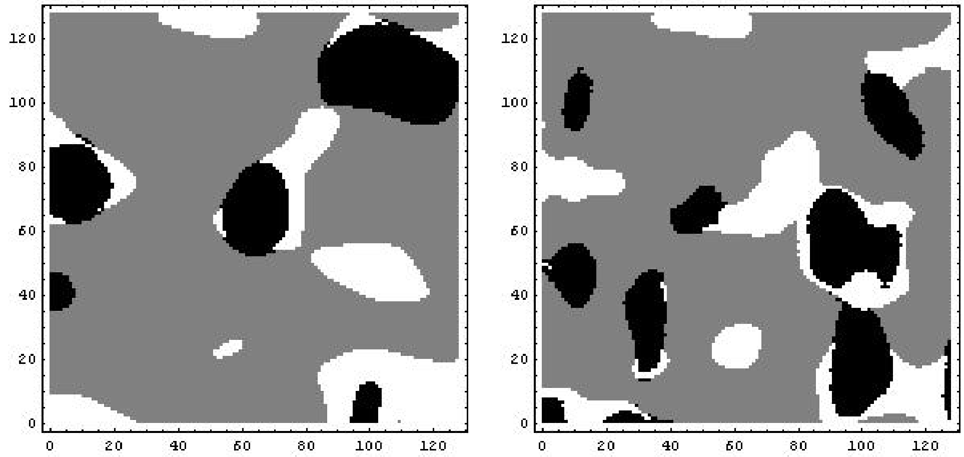
(b)

**Figure 32:** Two unconditional facies realizations generated from two isotropic Gaussian fields. The ranges of both Gaussian fields are 1536 ft.

other hand, the patterns in the facies distribution do not depend much on the spatial correlations of the Gaussian fields. The same set of covariance parameters yields a great variety of facies patterns. The two unconditional facies realizations shown in Fig. 32 both are generated from isotropic Gaussian fields with range 1536, and the only difference is the random variables $Z_1$ and $Z_2$. The field size is 5120 ft × 5120 ft. Without knowing the actual covariance parameters used in each facies map, it is easy to conclude that the two facies maps have distinct spatial correlations. The proportion of facies 1 (in black) is significantly greater in Fig. 32(a) than in Fig. 32(b).

Fig. 33 shows two facies realizations from the same set of $Z_1$ and $Z_2$ with those in Fig. 32(b). The Gaussian field $Y_1$ for Fig. 33(a) is anisotropic with the principle ranges in the horizontal direction (0 degree) 1536 ft, and vertical direction (90 degree) 768 ft. The Gaussian field $Y_2$ for Fig. 33(a) is isotropic with range 1536 ft. Although the Gaussian field $Y_1$ for Fig. 33(a) has anisotropy ratio of 2, it does not make the spatial structure in the facies map distinct from Fig. 32(b).

Shrinking the horizontal correlation range of $Y_2$ ($a_{21}$) from 1536 to 768, the facies

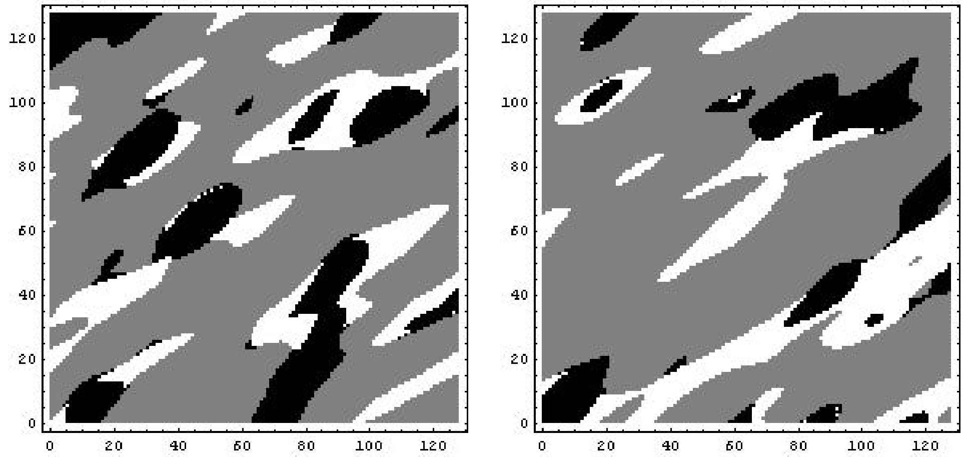


(a) (b)

**Figure 33:** Two unconditional facies realizations generated from anisotropic covariance fields. For the facies map on the left: $a_{11} = 1536$, $a_{12} = 768$, $\theta_{c1} = 0$, $a_{21} = a_{22} = 1536$. For the facies map on the right: $a_{11} = 1536$, $a_{12} = 768$, $\theta_{c1} = 0$, $a_{21} = 768$, $a_{22} = 1536$, $\theta_{c1} = \pi/2$.

map in Fig. 33(a) is rearranged as shown in Fig. 33(b). Both of the Gaussian fields lying behind the facies map have anisotropic ratio of 2. Moreover, as $Y_2$ is the vertical coordinate in the threshold map, it extends horizontally along the coordinate of $Y_1$, and the anisotropy features are expected to be enhanced. However, comparing Fig. 33(b) and Fig. 32(a) without knowing their actual covariance parameters, it is hard to tell which one is more anisotropic.

The anisotropy becomes observable in the facies maps when the anisotropy ratios are significantly larger than 1. The two facies maps in Fig. 34 have the same set of covariance parameters: $a_{11} = 1536$ ft, $a_{12} = 384$ ft, $\theta_{c1} = \pi/6$, $a_{21} = 1536$ ft, $a_{22} = 384$ ft, and $\theta_{c2} = \pi/4$, but they still appear to be different in spatial structure. The proportion of facies 1 (in black) is greater in Fig. 34(a) than in Fig. 34(b). The reason seems to be that the principle range of 1536 ft for both Gaussian fields is too large in comparison to the field size (30%), such that the features of spatial correlation are not repeated sufficiently. Reducing the principle range from 1536 to



(a)

(b)

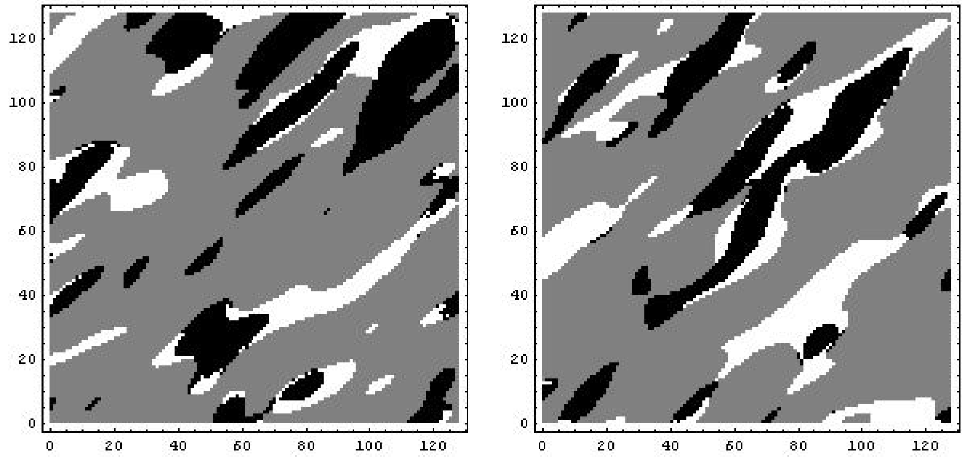
**Figure 34:** Two unconditional facies realizations generated from highly anisotropic Gaussian fields. $a_{11} = 1536$ ft, $a_{12} = 384$ ft, $\theta_{c1} = \pi/6$, $a_{21} = 1536$ ft, $a_{22} = 384$ ft, and $\theta_{c2} = \pi/4$.

1280, and keeping all other parameters as in Fig. 34, the spatial structures of the two unconditional facies realizations in Fig. 35 seem a lot more similar.

### 6.5.3 Experiments on training images with shorter correlation lengths

The facies map in Fig. 35(a) is used as a training image to examine the uncertainty distribution of the estimated covariance parameters. The correlation range of facies in Fig. 35(a) should be small enough to allow sufficient periodicity to be observed. The truncation scheme for generating the training image is assumed to be known, which is shown in Fig. 28(b). The unconditional ranges for the Gaussian fields are generated from the Gaussian prior PDFs with the "true" mean and the standard deviation of 10% of the mean. The prior distribution of the unconditional anisotropy angles are Gaussian with mean at the truth and the standard deviation of 5 degrees.

The mismatches of the 200 final facies maps to the training image are plotted in the histogram as shown in Fig. 36. The mode of the histogram is around 10% of the total number of grid blocks, and the optimization process seemed to be more difficult



(a) (b)

**Figure 35:** Two unconditional facies realizations generated from highly anisotropic Gaussian fields. $a_{11} = 1280$ ft, $a_{12} = 384$ ft, $\theta_{c1} = \pi/6$, $a_{21} = 1280$ ft, $a_{22} = 384$ ft, and $\theta_{c2} = \pi/4$

than in the case with larger correlation ranges. Figure 37 shows the histograms of the final 200 covariance parameters. The distribution of $a_{11}$ is multi-modal with modes around 300 ft and 900 ft. The distribution of $a_{12}$ has the mode around 300 ft. The distribution of $a_{21}$ is similar with that of $a_{11}$, but the greater mode is around 1000 ft. The distribution of $a_{22}$ is much narrower than other ranges. Although the distribution of the 200 conditioned $a_{22}$ appears to have two modes, the value difference between the two modes is small. So the multi-modal appearance can be attributed to the insufficient number of samples. The real mode is around 300 ft. The smaller mode in the histograms of the principle ranges, $a_{11}$ and $a_{21}$, could be attributed to the switching of the longer and shorter ranges. However, the distributions of the shorter ranges, $a_{12}$ and $a_{22}$, do not appear to have peaks around 1000 ft, which indicates that isotropic covariance fields with ranges around 300 ft could also reproduce the training image. This might be feasible as the features of the facies distributions also depend on the truncation schemes.

The histograms of both the anisotropy angles are multi-modal. The mode with

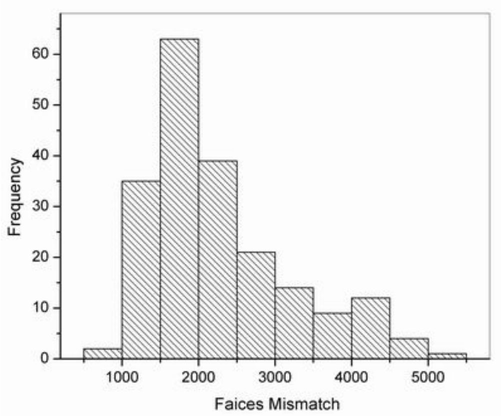


**Figure 36:** The distribution of the facies mismatch from the final 200 facies realizations.

higher frequency for $\theta_{c1}$ is at 0.6, and the other mode is at 2.3. The two modes are $\pi/2$ apart, which should be the result of principle range reduced below the secondary range. The mode with higher frequency for $\theta_{c2}$ is at 0.8, and the other mode is at 2.4, which is also about $\pi/2$ apart.

Figure 38 shows the cross-plots of the geostatistical parameters, in which the dots indicate the final values of the facies mismatch. The bigger the size of the dot, the better match its corresponding geostatistical parameters achieved. The size of the dots shown in Fig. 38 are comparable in size with the ones in Fig. 31. It appears to have less good matches with highly anisotropic and shorter correlated Gaussian fields, but more features are revealed instead of just blobs as in Fig. 31. All the good matches in Fig. 38(c) are clustered around the true values, with the uncertainty distribution wider than the prior. The distributions of the estimated ranges represented by the bigger dots are also wider than the prior, and shifted to lower values.

In practice, the prior estimations of the geostatistical parameters from the given training image may have larger uncertainties than in the case studied above. When the initial estimation is far away from the truth, it is often harder to minimize the facies mismatch in the optimization process. Therefore, when the uncertainties in the

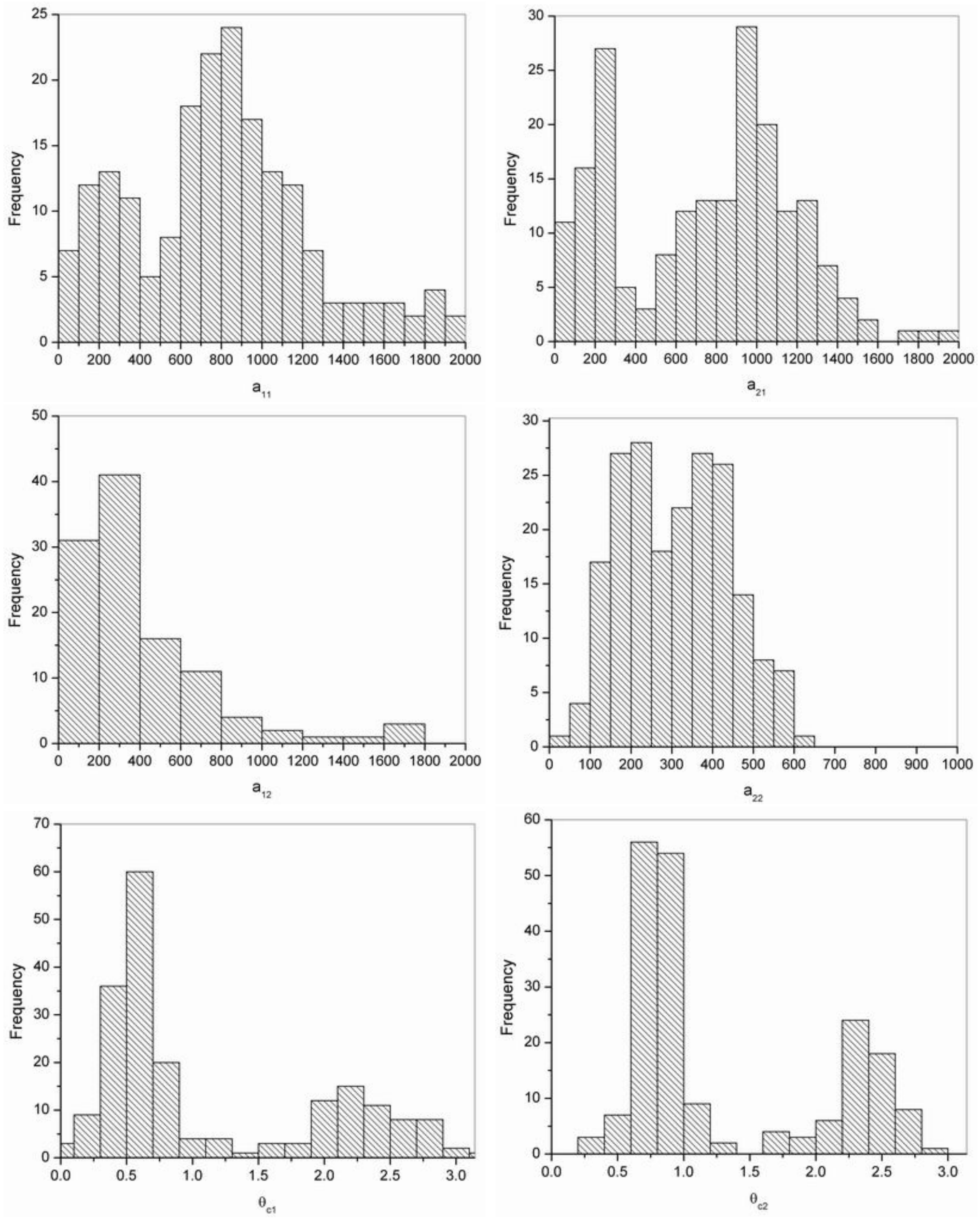
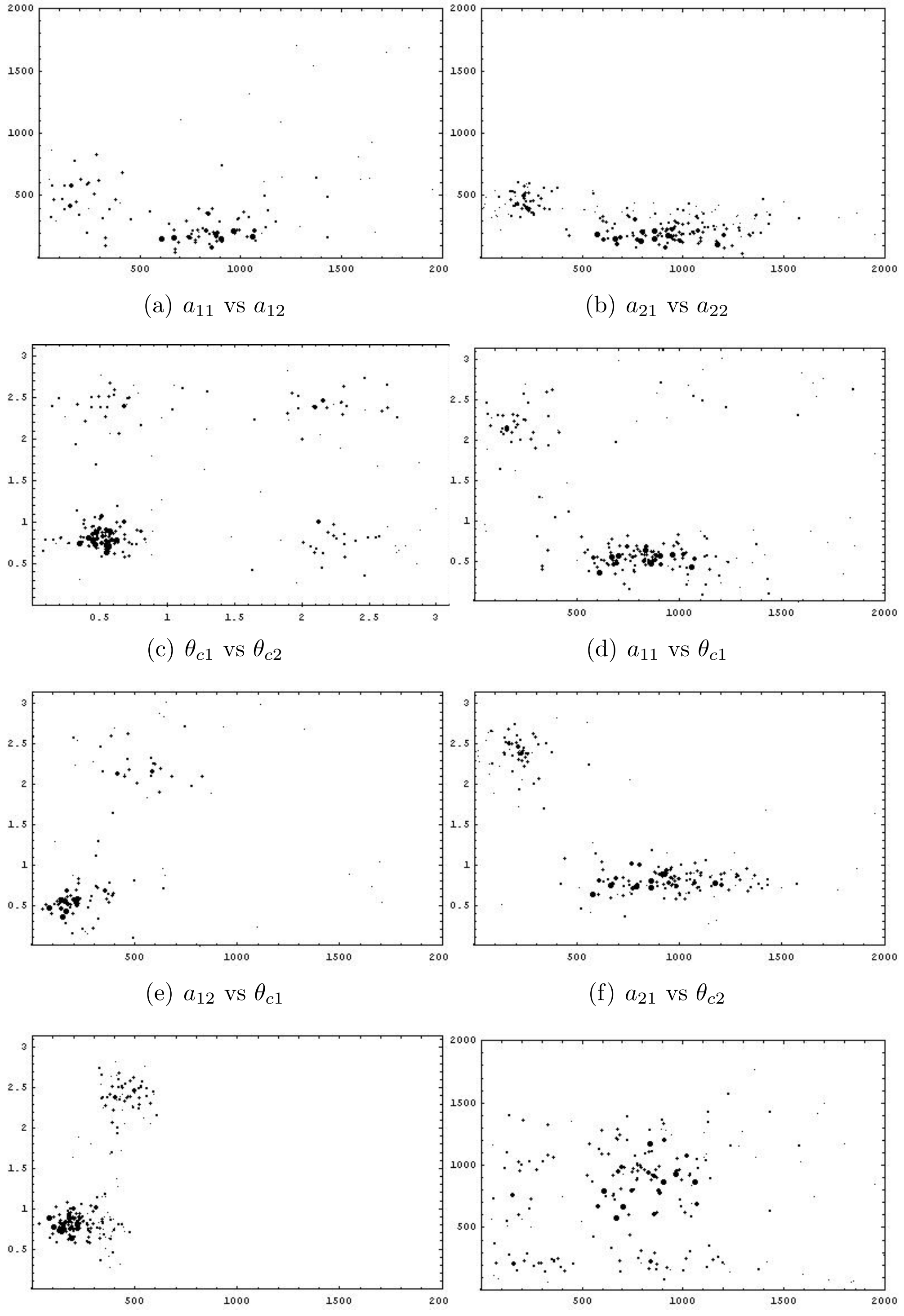


**Figure 37:** Histograms of the 200 conditioned covariance models estimated from the training image with shorter correlation range. The prior PDFs of the correlation ranges are Gaussian with the mean at the "truth" and standard deviation of 10% of the mean. The prior PDFs of the anisotropy angles are Gaussian with the mean at the "truth" and standard deviation of 5 degrees.



(a) $a_{11}$ vs $a_{12}$

(b) $a_{21}$ vs $a_{22}$

(c) $\theta_{c1}$ vs $\theta_{c2}$

(d) $a_{11}$ vs $\theta_{c1}$

(e) $a_{12}$ vs $\theta_{c1}$

(f) $a_{21}$ vs $\theta_{c2}$

(g) $a_{22}$ vs $\theta_{c2}$

(h) $a_{11}$ vs $a_{21}$

**Figure 38:** Cross-plots of the conditional covariance parameters from the training image with shorter correlation range. The size of the dots is proportional to the quality of the match to the training image.

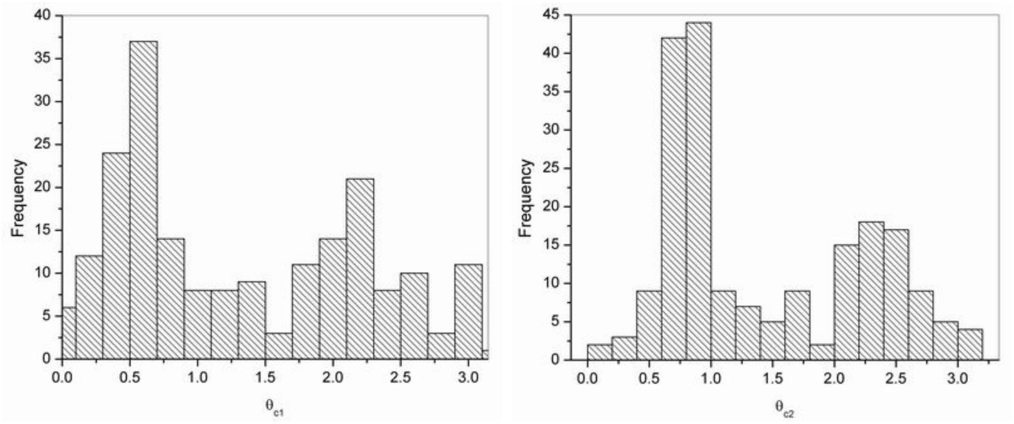
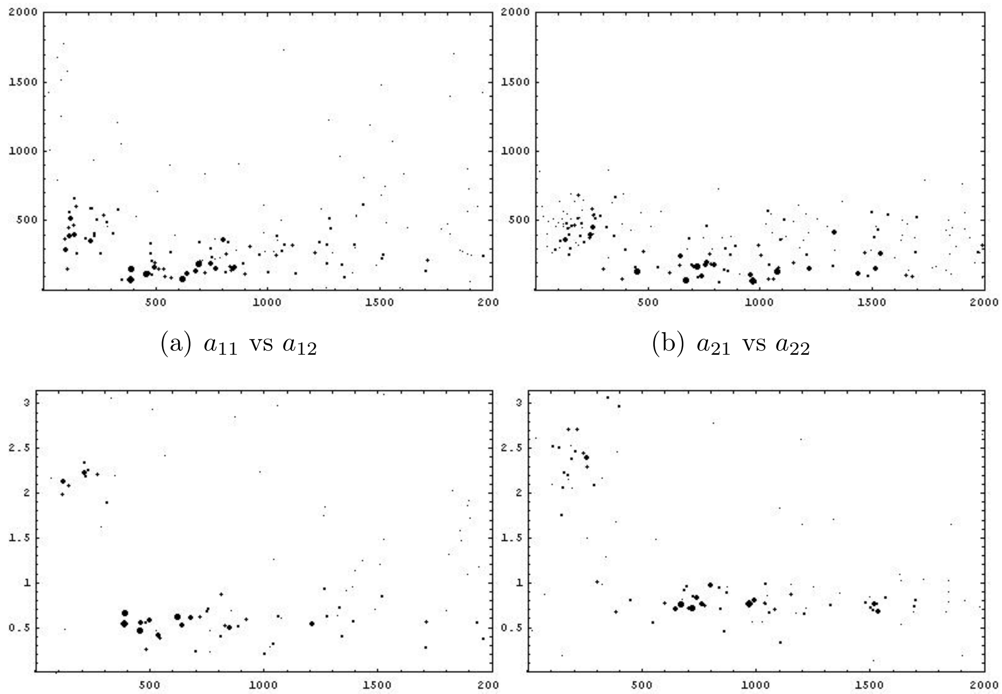


**Figure 39:** Histograms of the 200 conditioned covariance models estimated from the training image with shorter correlation range. The prior PDFs of the correlation ranges are Gaussian with the mean at the "truth" and standard deviation of 50% of the mean. The prior PDFs of the anisotropy angles are Gaussian with the mean at the "truth" and standard deviation of 45 degrees.

prior estimations are increased, fewer sets of conditional geostatistical model parameters could be obtained with low facies mismatch, but the conditional realizations of the geostatistical model parameters from the wider prior PDFs should be a better representation of the true a posteriori PDFs.

The unconditional realizations of the correlation ranges are generated from the truncated Gaussian distributions with the mode at the "truth" and the standard deviation of 50% of the mode values. As there is about 19.7% of the possibility that the prior ranges could be negative, the negative values generated were abandoned and new samples were drawn from the positive part of the prior Gaussian distributions. The prior distributions of the anisotropy angles are Gaussian with modes at the "truth" ($\pi/6$ and $\pi/4$, respectively) and standard deviation of $\pi/4$. The covariance parameters generated from the prior PDFs with greater uncertainties are optimized to match the training image in Fig. 35(a).

Figure 39 shows the histograms of the conditioned anisotropy angles. There are



(a) $a_{11}$ vs $a_{12}$

(b) $a_{21}$ vs $a_{22}$

(c) $a_{11}$ vs $\theta_{c1}$

(d) $a_{21}$ vs $\theta_{c2}$

**Figure 40:** Cross-plots of the conditional covariance parameters from the training image with shorter correlation range. The prior covariance parameters are generated from wide uncertainty distributions.

again two peaks that separated by $\pi/2$ in each histogram. The experimental a posteriori distributions are centered at the true anisotropy angles with less uncertainties than in the prior distributions.

The difficulty in matching the training image using the prior geostatistical parameters from wide uncertainty distributions is reflected by the sparsity of bigger dots in the cross plots shown in Fig. 40. The conditioned estimates of the Gaussian correlation ranges are biased to much lower values than the truth.

# CHAPTER VII

# GRADIENT METHOD IN MATCHING PRODUCTION DATA

The only locations in the reservoir at which facies can actually be observed are the well bores with coring operations. A training image from geologists can be used to describe expected facies distribution patterns and features based on the sedimentary environment and formation outcrops, but does not carry any local accuracy. In the previous study, probability density functions for the geostatistical parameters, such as the ones deciding truncation schemes and the ones deciding the covariance of the Gaussian fields, were estimated from a given training image. The geostatistical parameters were then sampled from the PDFs in a Bayesian scheme for simulation of facies conditional to logs and/or production data.

When the geostatistical parameters are fixed, realizations of the facies distribution map can be generated by minimizing the objective function with respect to the random fields $Z_1$ and $Z_2$. The objective function in this case is:

$$\begin{aligned} O(m) &= \frac{1}{2}[P(m) - P_{\mathrm{obs}}]^T C_P^{-1}[P(m) - P_{\mathrm{obs}}] + \frac{1}{2}[F(m) - F_{\mathrm{obs}}]^T C_F^{-1}[F(m) - F_{\mathrm{obs}}] \\ &\quad + \frac{1}{2}(m - m_{\mathrm{pr}})^T C_M^{-1}(m - m_{\mathrm{pr}}) \\ &= \frac{1}{2}[d(m) - d_{\mathrm{obs}}]^T C_D^{-1}[d(m) - d_{\mathrm{obs}}] + \frac{1}{2}(m - m_{\mathrm{pr}})^T C_M^{-1}(m - m_{\mathrm{pr}}), \qquad (51) \end{aligned}$$

where the prior model $m_{\mathrm{pr}} = \{Z_{1,pr}, Z_{2,pr}\}$ is composed of random deviates with mean 0 and variance 1, so the prior model covariance matrix $C_M$ is an identity matrix. $P_{\mathrm{obs}}$ is the observed production data, such as the bottom-hole pressures, the water-oil ratios (WORs), the gas-oil ratios (GORs), and the production rates. $P(m)$ is the simulated production data. $F_{\text{obs}}$ is the facies observation at a few cored locations. Both $C_P$ and $C_F$ are diagonal covariance matrices of the data observations for independent data errors, and its diagonal elements are the variances of measurement errors. The facies observations are very accurate so its variance is much smaller than that of production data. The mismatch to the facies data dominates the objective function before all the facies observations are matched, so that the facies type at well locations are forced to exactly honor the facies observations. The production data and the facies data are combined and denoted as $d_{\text{obs}}$.

## 7.1 Gradient derivation

The variation of rock properties within a facies is assumed to be negligible in comparison with that between different facies types, so the porosity and the permeability values at each gridblock are determined by the facies type. The gradient of the objective function $O(m)$ with respect to the model parameters is:

$$\begin{aligned} g &= \nabla_m O(m) \\ &= G_P(m)^T C_P^{-1}[P(m) - P_{\text{obs}}] + G_F(m)^T C_F^{-1}[F(m) - F_{\text{obs}}] + C_M^{-1}(m - m_{\text{pr}}) \\ &= g_P + g_F + g_m \\ &= \nabla_m O_d(m) + \nabla_m O_m(m). \end{aligned} \tag{52}$$

When the geostatistical model has been chosen, generating a facies realization that matches the production data and logging data requires optimization with respect to the random fields $Z_1$ and $Z_2$.

The gradient of the data mismatch with respect to $Z_1$ and $Z_2$ can be derived by the chain rule as:

$$\begin{aligned}\nabla_{Z_1} O_d(m) &= \nabla_{Z_1}[O_P(m) + O_F(m)] \\ &= L_1 \cdot [\nabla_{Y_1} K(m) \cdot \nabla_K O_P(m) + \nabla_{Y_1}\phi(m) \cdot \nabla_\phi O_P(m) \\ &\quad + \nabla_{Y_1} O_F(m)], \qquad (53)\end{aligned}$$

$$\begin{aligned}\nabla_{Z_2} O_d(m) &= \nabla_{Z_2}[O_P(m) + O_F(m)] \\ &= L_2 \cdot [\nabla_{Y_2} K(m) \cdot \nabla_K O_P(m) + \nabla_{Y_2}\phi(m) \cdot \nabla_\phi O_P(m) \\ &\quad + \nabla_{Y_2} O_F(m)], \qquad (54)\end{aligned}$$

where $L_1$ and $L_2$ are square roots of the covariance matrices $C_{Y1}$ and $C_{Y2}$. In practice, because the dimensions of the model are so large, we use convolution of a single row as described by Oliver (1995). $K(m)$ is the permeability field and $\phi(m)$ is the porosity field. The gradients of the objective function with respect to the permeability and the porosity fields, $\nabla_K O_P(m)$ and $\nabla_\phi O_P(m)$, were obtained from the adjoint method for general automatic history matching of reservoir property fields (see Li et al., 2003b).

As the permeability field $K(m)$ is a vector of $N_g$ elements, where $N_g$ is the number of gridblocks, $\nabla_{Y_1} K^T$ and $\nabla_{Y_2} K^T$ are both $N_g \times N_g$ matrices. Sensitivity of $K$ to $Y_1$ at the $i$th gridblock, $Y_{1,i}$, is a vector of all zeros but the $i$th element. Equation 55 shows the gradient of $K$ about $Y_{1,1}$:

$$\nabla_{Y_{1,1}} K = \begin{bmatrix} \frac{\partial K_1}{\partial Y_{1,1}} \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \qquad (55)$$

which indicates that a perturbation to the Gaussian random field $Y_1$ at gridblock 1 only impacts the permeability at the gridblock 1. Consequently, the gradients of the property fields with respect to the Gaussian fields, $\nabla_{Y_j} K(m)$ and $\nabla_{Y_j} \phi(m)$ for $j = 1, 2$, are diagonal matrices:

$$\nabla_{Y_j} K^T = \begin{bmatrix} \frac{\partial K_1}{\partial Y_{j,1}} & 0 & \cdots & 0 \\ 0 & \frac{\partial K_2}{\partial Y_{j,2}} & \cdots & 0 \\ \vdots & & \ddots & \\ 0 & 0 & \cdots & \frac{\partial K_{N_g}}{\partial Y_{j,N_g}} \end{bmatrix},$$

$$\nabla_{Y_j} \phi^T = \begin{bmatrix} \frac{\partial \phi_1}{\partial Y_{j,1}} & 0 & \cdots & 0 \\ 0 & \frac{\partial \phi_2}{\partial Y_{j,2}} & \cdots & 0 \\ \vdots & & \ddots & \\ 0 & 0 & \cdots & \frac{\partial \phi_{N_g}}{\partial Y_{j,N_g}} \end{bmatrix}.$$

The diagonal elements are computed by chain rule:

$$\begin{aligned} \frac{\partial K_i(m)}{\partial Y_{1,i}} &= \frac{dK_i}{dd_l} \frac{\partial d_l}{\partial Y_{1,i}}, \\ \frac{\partial \phi_i(m)}{\partial Y_{1,i}} &= \frac{d\phi_i}{dd_l} \frac{\partial d_l}{\partial Y_{1,i}}, \\ \frac{\partial K_i(m)}{\partial Y_{2,i}} &= \frac{dK_i}{dd_l} \frac{\partial d_l}{\partial Y_{2,i}}, \\ \frac{\partial \phi_i(m)}{\partial Y_{2,i}} &= \frac{d\phi_i}{dd_l} \frac{\partial d_l}{\partial Y_{2,i}}, \end{aligned} \tag{56}$$

for $i = 1, \ldots, N_g$. The $d_l$ terms are the distance of $(Y_{1,i}, Y_{2,i})$ to the closest threshold line, which is computed in Eq. 23.

The property fields $K$ and $\phi$ are regarded as homogenous within a facies region, but the values are discontinuous across the facies boundaries. To compute the gradient of the property fields with respect to the Gaussian fields, the property fields have to be differentiable. As the facies observations are only available at a few cored locations, it is not possible to tell whether the facies type corresponding to any $(Y_{1,i}, Y_{2,i})$ is correct. Therefore the transition zone of the permeability and porosity values is made on both

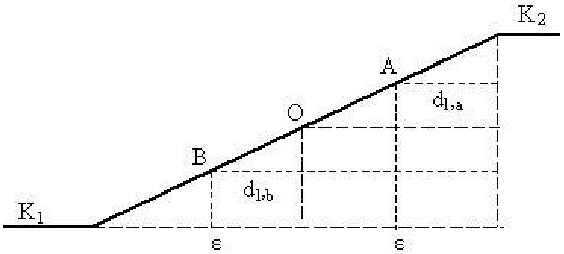


**Figure 41:** The gradient of the permeability is derived from this linear interpolation model. $K_1$ and $K_2$ are permeability values assigned to two adjacent regions in the threshold map. Point O is the cross-section with the threshold line, which is also the middle point of the transition zone in this 1-D plot.

sides along the closest threshold line to point $(Y_{1,i}, Y_{2,i})$. If the property fields in the transition zone were defined by the square root of the distance to the threshold line as:

$$K(d_l) = \frac{K_1 + K_2}{2} + Sign(d_l)\frac{K_2 - K_1}{2}\sqrt{\frac{|d_l|}{\epsilon}}, \text{ for } |d_l| \leq \epsilon \ ,$$

and $K_1$ and $K_2$ are permeability values assigned to each facies, then the gradient $\frac{dK_i}{dd_l}$ goes to infinite when the distance to the threshold line approaches to zero. Therefore linear interpolation was chosen for the transition zone of the property fields.

Let the width of the transition zone on each side of the threshold line be $\epsilon$, the permeability along the direction perpendicular to a threshold line is:

$$K(d_l) = K_1 - \frac{1}{2}(K_1 - K_2)(1 - \frac{|d_l|}{\epsilon}), \text{ for } |d_l| \leq \epsilon \tag{57}$$

where $K_1$ is the assigned facies permeability at the same side of the threshold line with $(Y_{1,i}, Y_{2,i})$, and $K_2$ is the assigned permeability on the other side. $\epsilon$ is the absolute distance to the threshold line. In Fig. 41, the permeability at point B can be computed by Eq. 57, and $K_A$ should be computed by:

$$K_A = K_2 - \frac{1}{2}(K_2 - K_1)(1 - \frac{|d_{l,a}|}{\epsilon}), \tag{58}$$

as A is on the same side with $K_2$.

The gradient $\frac{dK_i}{d|d_l|}$ in the linear interpolation case is:

$$\frac{dK_i}{d|d_l|} = \begin{cases} \frac{K_1-K_2}{2\epsilon} & \text{for } |d_l| \leq \epsilon \\ 0 & \text{for } |d_l| > \epsilon. \end{cases} \tag{59}$$

Substituting Eq. 59 into Eq. 56, the gradient of the permeability at the $i$th grid with respect to $Y_1$ at the $i$th grid is:

$$\begin{aligned} \frac{\partial K_i(m)}{\partial Y_{1,i}} &= \frac{dK_i}{d|d_l|}\frac{d|d_l|}{dd_l}\frac{\partial d_l}{\partial Y_{1,i}} \\ &= Sign(d_l)\frac{K_1-K_2}{2\epsilon}\sin(\theta_l-\frac{\pi}{2}). \end{aligned} \tag{60}$$

Similarly,

$$\begin{aligned} \frac{\partial \phi_i(m)}{\partial Y_{1,i}} &= \frac{d\phi_i}{d|d_l|}\frac{d|d_l|}{dd_l}\frac{\partial d_l}{\partial Y_{1,i}} \\ &= Sign(d_l)\frac{\phi_1-\phi_2}{2\epsilon}\sin(\theta_l-\frac{\pi}{2}), \end{aligned}$$

$$\begin{aligned} \frac{\partial K_i(m)}{\partial Y_{2,i}} &= \frac{dK_i}{d|d_l|}\frac{d|d_l|}{dd_l}\frac{\partial d_l}{\partial Y_{1,2}} \\ &= -Sign(d_l)\frac{K_1-K_2}{2\epsilon}\cos(\theta_l-\frac{\pi}{2}), \end{aligned}$$

$$\begin{aligned} \frac{\partial \phi_i(m)}{\partial Y_{1,i}} &= \frac{d\phi_i}{d|d_l|}\frac{d|d_l|}{dd_l}\frac{\partial d_l}{\partial Y_{1,i}} \\ &= -Sign(d_l)\frac{\phi_1-\phi_2}{2\epsilon}\cos(\theta_l-\frac{\pi}{2}). \end{aligned}$$

The third term of $\nabla_{Z_1}O_d(m)$ in the Eq. 53 is the gradient of facies mismatch about $Z_1$. The computation of $\nabla_{Y_1}O_F(m)$ is similar with the case where a training image is given. $O_F$ is the squared difference between the simulated facies and facies observations. When the simulated and the observed facies are the same type, the difference factor $f_i = 0$, otherwise $f_i = 1$. As the gradient $\nabla_{Y_1}O_F(m) = G_{F,Y_1}(m)^T C_F^{-1} f$, the key is to compute the sensitivity of the facies mismatch to $Y_1$: $G_{F,Y_1}$, which is an

$N_F \times N_g$ sparse matrix with maximum one non-zero element in each row. The non-zero elements are the sensitivities of the facies difference $f$ at the facies observation locations with respect to $Y_1$ at the corresponding grid. The following pseudo code describes the computation of $\frac{df_i}{dY_{1,i}}$.

```
DO $l = 1,\ N_l$
    $d(l) = \frac{\tan(\theta_l - \frac{\pi}{2})Y_{1,i} - Y_{2,i} - \tan(\theta_l - \frac{\pi}{2})\frac{r_l}{\cos\theta_l}}{\sqrt{1+\tan^2(\theta_l - \frac{\pi}{2})}} = \sin(\theta_l - \frac{\pi}{2})Y_{1,i} - \cos(\theta_l - \frac{\pi}{2})Y_{2,i} + r_l$
END DO.
$d_l = \min(d)$
IF $F_i = F_{\mathrm{obs,i}}$ THEN
    $\frac{df_i}{dY_{1,i}} = 0$
ELSE
    IF $|d_l| > \epsilon$ THEN
        $\frac{df_i}{dY_{1,i}} = 0$
    ELSE
        IF $(F_{ab,i} = F_{\mathrm{obs,i}})$ THEN
            $\frac{df_i}{dY_{1,i}} = \frac{d_l}{|d_l|}\frac{1}{2\sqrt{\epsilon|d_l|}}\sin(\theta_l - \frac{\pi}{2})$
        ELSE
        $\frac{df_i}{dY_{1,i}} = 0$
        END IF
    END IF
END IF
```

$F_{ab,i}$ is the facies type on the other side of the closest threshold line. When the facies at grid $i$ decided by $(Y_{1,i}, Y_{2,i})$ does not match the facies observation at that location, but the other side of the threshold line has the correct facies type, i.e. $F_{ab,i} = F_{\mathrm{obs,i}}$, then there exists a transition zone on the side of the threshold line that is closer to $(Y_{1,i}, Y_{2,i})$.

The non-zero terms in $\nabla_{Y_1} O_F(m) = G_{F,Y_1}(m)^T C_F^{-1} f$ are computed as:

$$\frac{\partial O_F(m)}{\partial Y_{1,i}} = \frac{d_l}{|d_l|} \frac{1}{2\sigma_F^2 \epsilon} \sin(\theta_l - \frac{\pi}{2}).$$

Similarly, the gradient of the facies mismatch with respect to $Y_2$: $\nabla_{Y_2} O_F(m)$, is a very sparse vector with a maximum of $N_F$ non-zero elements. The non-zero elements are computed as:

$$\frac{\partial O_F(m)}{\partial Y_{2,i}} = \frac{d_l}{|d_l|} \frac{1}{2\sigma_F^2 \epsilon} \sin(\theta_l - \frac{\pi}{2}).$$

## *7.2 Generate initial model*

If the initial model for history matching of production data does not honor the facies observations, the production data mismatch is enormous and difficult to minimize. Knowing the geological and geostatistical model, the initial model set $Z_1$ and $Z_2$ can be generated in the following procedures.

1. Generate two multivariate Gaussian deviates $Z_1$ and $Z_2$ from $N(\mathbf{0}, I)$.

2. Simulate the facies map from $Z_1$ and $Z_2$, and check whether the simulated facies at observation locations match all the facies observations.

3. For simulated facies that do not honor the facies observations, generate new random variables $Z_1$ and $Z_2$ in patches of appropriate size (proportional to the correlation ranges of the Gaussian fields) and replace the patches to the variables centered at the observation locations that do not match.

4. Run the simulation again and check the matching of the facies observation. If there is still a facies mismatch, repeat step 3. If not, output the random fields $Z_1$, $Z_2$ as a set of initial model.

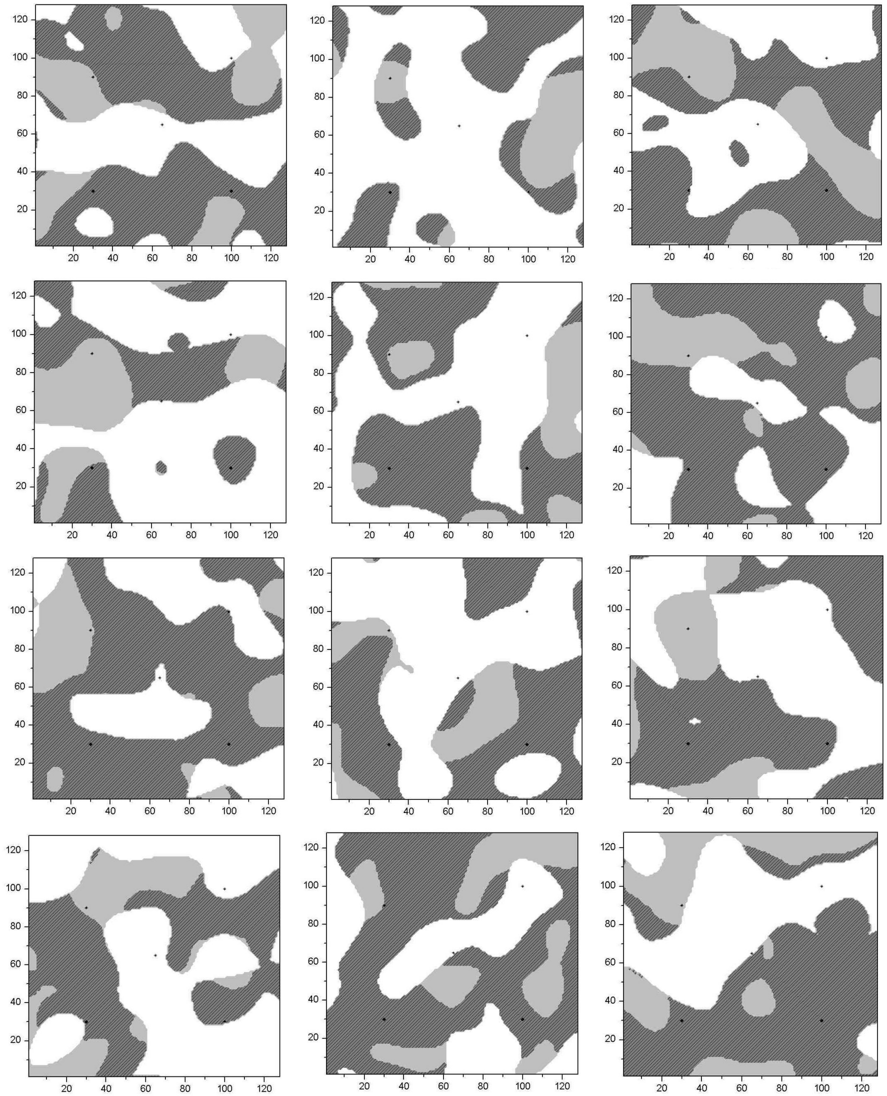


**Figure 42:** Twelve initial facies maps that honor the facies observations. The initial models are then used for matching the production data, while preserving the correct facies at observation locations.

Twelve initial models were generated and the initial facies maps are shown in Fig. 42.

As the assignment of facies to the seven regions in the threshold map is ad hoc and non-differentiable at this stage, the true facies arrangement is assumed to be known as a part of the geostatistical model. The history matching process then uses the true threshold map to do the truncation. One potential problem with fixing the threshold map and the facies assignment is that the grids with Gaussian random variables far from threshold lines may be difficult to be modified to the correct facies. To make this point clear, the true threshold map is shown in Fig. 43 with the pairs of Gaussian variables $(Y_1, Y_2)$ at the observation locations of the true facies map. The regions in the threshold map are assigned three types of facies: $F = 1, 2$, or 3. Both the facies observations with facies $F = 1$ can be close to facies 2 and 3 in the true facies map. An initial map can be very possibly generated matching the facies observations, but with the pairs of Gaussian variables in a different region of the threshold map. For instance, if either of the two pairs of $(Y_1, Y_2)$ giving facies 1 is in the top region where $Y_2$ has large positive value, the observation location will be difficult to be adjusted to facies 2, as it is not likely for that point to move around the corner with the facies 3 region and get down to the lower facies 1 region. When there are production data at this facies observation location, it has small chance for convergence to the production data because the facies types at the near well grids are difficult to correct.

Therefore the initial model should honor the threshold regions at facies observation locations to make the convergence easier. The procedures for generating the initial models that not only match the facies observations, but match the threshold regions of the facies observations are shown as follows:

1. Generate two multivariate Gaussian deviates $Z_1$ and $Z_2$ from $N(\mathbf{0}, I)$.

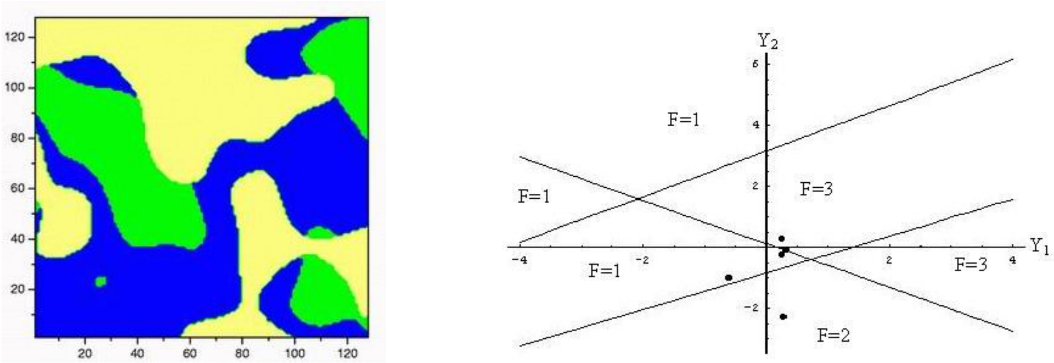


**Figure 43:** The true facies map and the true threshold map with the Gaussian variables $(Y_1, Y_2)$ at each facies observation of the true facies map.

2. Simulate the facies map from $Z_1$ and $Z_2$, then check whether the pairs of the Gaussian variables $(Y_1, Y_2)$ at observation locations are in the correct region.

3. For Gaussian pairs at observation locations that are not in the correct region in the threshold map, generate new random variables in patches of appropriate size (proportional to the correlation ranges of the Gaussian fields) and replace the patches to the variables centered at the observation locations that do not match.

4. Run the simulation again and check the matching of the Gaussian variable region. If there is mismatch, repeat step 3. If not, output the random fields $Z_1$, $Z_2$ as a set of initial model.

## *7.3 Investigation on convergence*

The convergence of the objective function is largely dependent on the transition zone width chosen for the optimization process. Each grid has two Gaussian variables $(Y_1, Y_2)$, and it has non-zero gradients of the objective function with respect to both Gaussian variables only if it is in the transition zone. When the transition zone is wide, grids that are far from the boundaries of facies regions respond to the perturbation to model parameters $Z_1$ and $Z_2$. The advantage of wide transition zone might be that modifications to more grids can be made in each iteration. However, as the transition is just made to compute the gradient and the objective function does not consider the transition zone, gradients with a narrower width should provide a more accurate adjustment direction. Fig. 44 shows the objective function along the search direction in the first optimization iteration. The top row is with the transition width of 0.2, and the bottom row is for a transition width of 1.0. The two figures in the left column include points with non-zero facies mismatch at facies observations. As all five facies observations are at well locations, a facies mismatch causes great production data mismatch from the wells. The objective function increases by a factor of 20 when the step size is slightly too large.

The reduction of the objective function before the facies alteration is small relative to its magnitude when the facies at well locations are different from the observed facies. Therefore the plot of objective function vs the number of iterations appears flat. Looking closely at the trend of the objective function with respect to the step size along the search direction, the flat parts are plotted out as shown in the right column of Fig. 44. Comparing the two figures with transition widths of 0.2 and 1.0 respectively, the maximum possible reduction of the objective function in the first step is greater for the case with wider transition width.

When the proposed model does not satisfy both Wolfe conditions (see Kolda et al., 1998) at the same time, a quadratic fit will be made to reduce the objective function by optimizing the step size. The quadratic function is of the form:

$$q(\alpha) = a\alpha^2 + b\alpha + c, \tag{61}$$

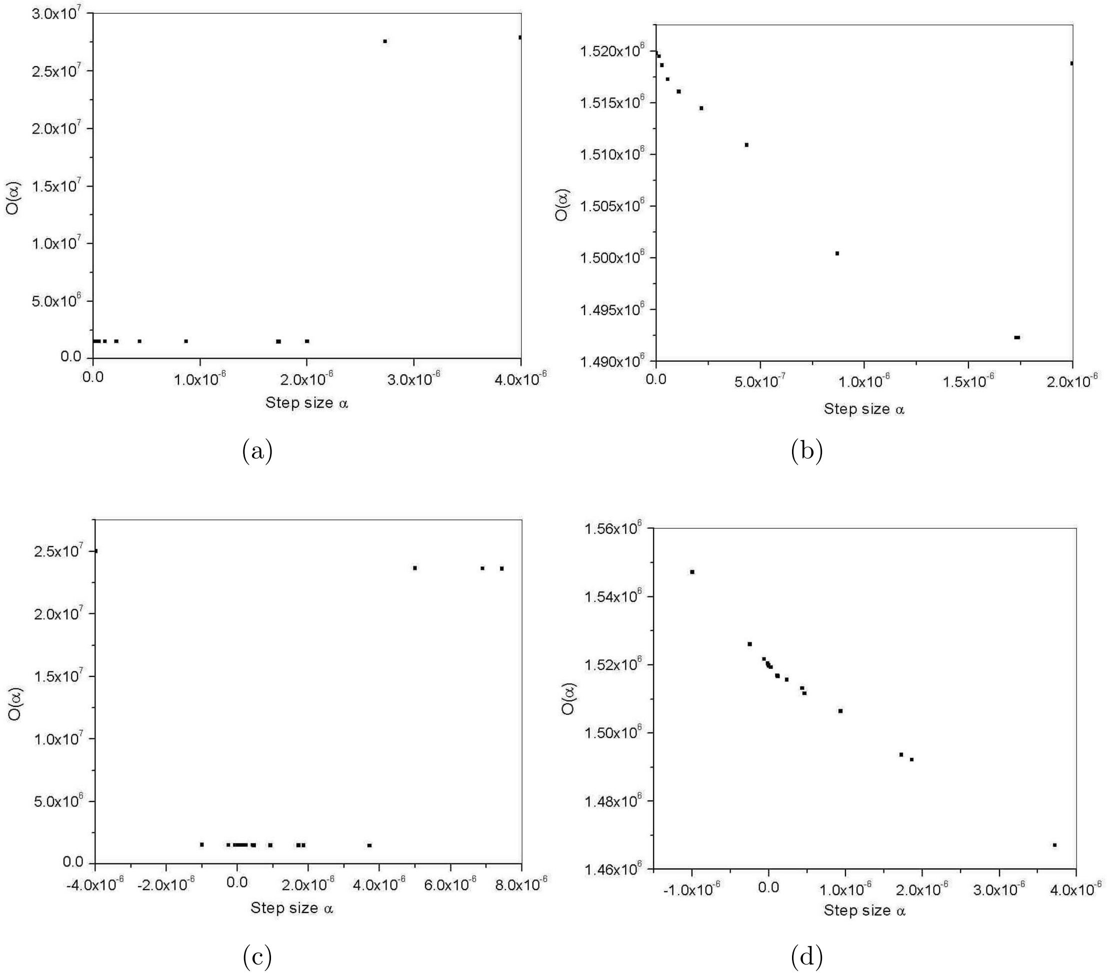


**Figure 44:** All four figures are the objective function along the first search direction. The transition zone width in (a) and (b) is 0.2, in (c) and (d) is 1.0. The figures on the right column are amplifications of the flat region in the figures on the left column.

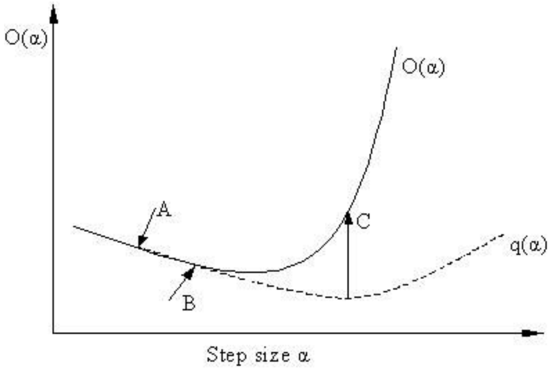


**Figure 45:** A schematic plot of the quadratic fit to an objective function curve with the typical shape for this minimization problem. The quadratic fit gives a higher objective function value than that from the Newton-Raphson iteration.

where the coefficients a, b, and c are computed as:

$$\begin{aligned} a &= \frac{O(m_k + \hat{\alpha}_k d_k) - \nabla O(m_k)\hat{\alpha}_k - O(m_k)}{\hat{\alpha}_k^2}, \\ b &= \nabla O(m_k), \\ c &= O(m_k). \end{aligned} \tag{62}$$

The step size $\hat{\alpha}_k$ is computed from the Newton-Raphson iteration right before the quadratic fit. $d_k$ is the search direction along which we try to find a minimum of the objective function. $m_k$ is the current model from which the search direction is computed. $O(m_k)$ is the objective function at the current model $m_k$. $\nabla O(m_k)$ is the gradient of the objective function about model parameters at $m_k$. Minimizing $q(\alpha)$ gives

$$\alpha_k = -\frac{\nabla O(m_k)\hat{\alpha}_k^2}{2[O(\hat{\alpha}_k d_k) - \nabla O(m_k)\hat{\alpha}_k - O(m_k)]}, \tag{63}$$

Both curves of the objective function with different transition zone width are nearly linear before abrupt jumps. Figure 45 is a schematic plot showing the typical shape of the objective function along the search direction for this type of minimization problem. Point A represents the starting model $m_k$, and B is the temporary model by Newton-Raphson search. Although the objective function has been reduced from point A to point B, the change in the model parameters might not be large enough to satisfy the second Wolfe condition. Therefore a quadratic fit is made through points A and B. The step size corresponding to point C is at the minimum point in the quadratic fit function $q(\alpha)$, but gives a higher objective function than both point A and B. In this case, the new model at point C is not acceptable. Instead the step size from the quadratic fit is repeatedly cut by a factor of 10 until the objective function from the proposed model become less than $O(m_k)$.

The process of optimization is fairly complicated. A flow chart (Fig. 46) is provided to give a better illustration of the structure of the code developed along this study.

## *7.4 Constrained optimization*

There are two types of data available in the history matching problem for geologic facies. One type is the production data, such as the pressure, the rates, the WOR, the GOR, and the logging interpretations. The other type is the facies observation data from cores, which are regarded as hard data. The process of optimizing the facies model to match the production data is similar to common automatic history matching problems. The hard data in our problem, however, are not the same type as the model parameters. The hard data are the facies observations, and the underlying model parameters $Z_1$ and $Z_2$ have to be constrained to the facies observations while matching the production data.

There are two aspects to consider on matching the facies observations. One is in generating initial models, which have been discussed in one of the previous sections. The other is in maintaining the facies types at observation locations in the process of optimization.

In the line search for the optimized step size, one Newton-Raphson iteration is

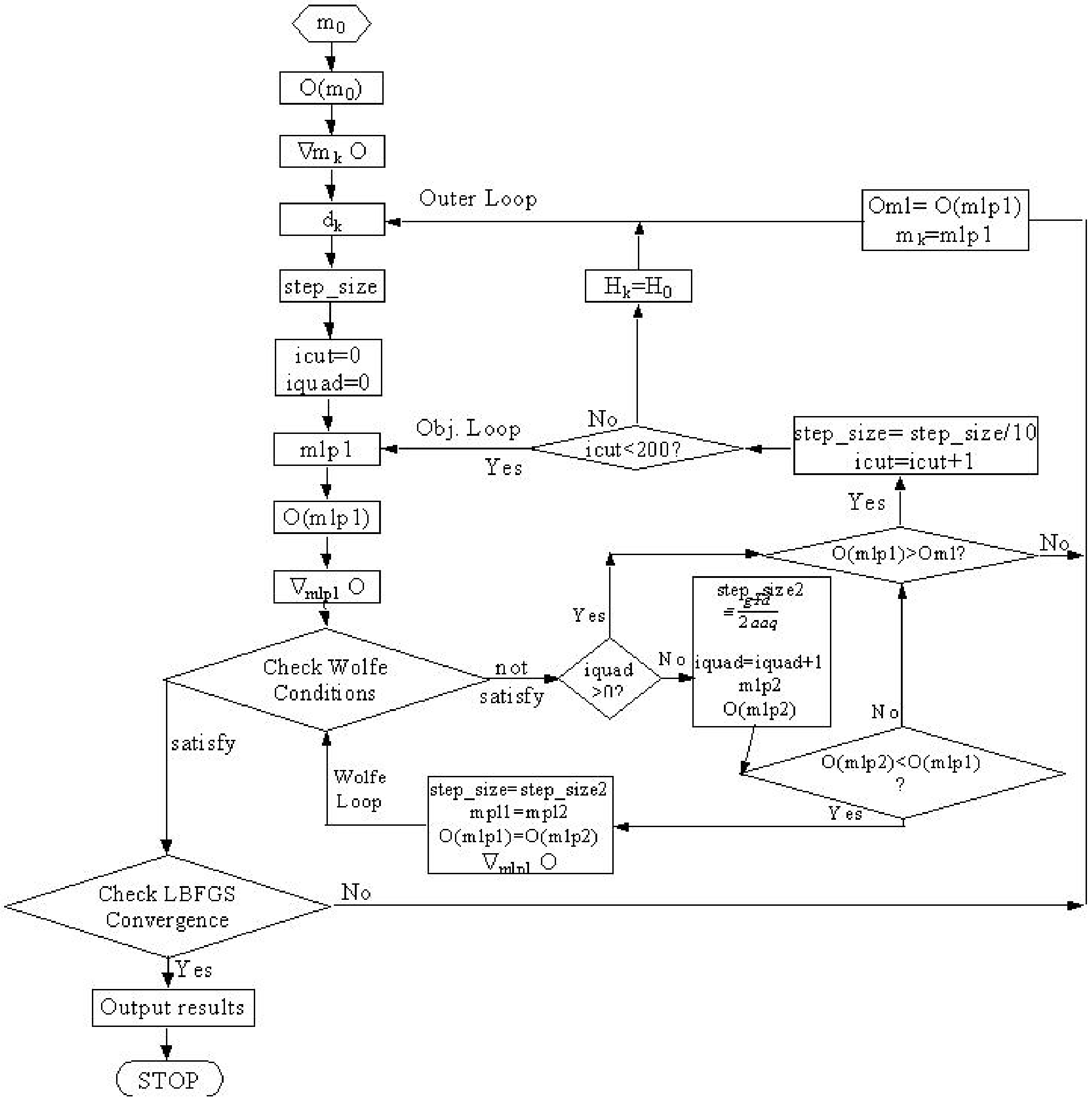


**Figure 46:** Flow chart for the automatic history matching process.

taken as:

$$\alpha_1 = -\frac{(\nabla O(m_k))^T d_k}{d_k^T H(m_k) d_k}, \tag{64}$$

where the computation of the denominator $d_k^T H(m_k) d_k$ requires calculation of $G_k d_k$. The term $G_k d_k$ is computed in this project by the finite difference method:

$$\begin{aligned} Gd_k &= \|d_k\| \frac{dO(m_k)}{d\alpha} \\ &\approx \|d_k\| \frac{O(m_k + \epsilon d_k) - O(m_k)}{\epsilon \|d_k\|} \\ &= \frac{O(m_k + \epsilon d_k) - O(m_k)}{\epsilon}, \end{aligned} \tag{65}$$

where $\epsilon$ is a small constant chosen based on the infinity norm of $d_k$:

$$\epsilon = \frac{10^{-3}}{\|d_k\|_\infty}. \tag{66}$$

The model $m_k + \epsilon d_k$ may not match the facies observation, which makes it meaningless and impractical to compute the objective function knowing it will be discarded. Therefore the facies mismatch is checked for the model $m_k + \epsilon d_k$. If it is non-zero, then $\epsilon$ is reduced: $\epsilon = \epsilon/10$; otherwise, the model is put into the simulator to compute the objective function $O(m_k + \epsilon d_k)$.

After the step size is computed from the line search, again the facies mismatch is checked or the model $m_k + \alpha_1 d_k$. If it is non-zero, then $\alpha_1 = \alpha_1/10$, otherwise, the new model $m_k + \alpha_1 d_k$ is put into the simulator to compute the objective function $O(m_k + \alpha_1 d_k)$. The procedure for checking facies mismatch is much more efficient than running the reservoir simulation.

## 7.5 A case study

This truncated pluri-gaussian simulation scheme was tested on a synthetic 2-D field history matching problem with three facies on 128 by 128 gridblocks. Both Gaussian fields are assigned Gaussian type covariance and have the same range of 30% of the

| index | Facies 1 | Facies 2 | Facies 3 |
|---|---|---|---|
| Litho-type | dolomite | shale | sand |
| Permeability (md) | 74 | 6 | 372 |
| Porosity | 0.18 | 0.146 | 0.25 |

**Table 4:** Properties for each of the litho-facies in the synthetic problem.

| Well name | Well 1 | Well 2 | Well 3 | Well 4 | Well 5 |
|---|---|---|---|---|---|
| Well type | producer | producer | producer | producer | injector |
| Well location | (30,30) | (30,90) | (100,30) | (100,100) | (65,65) |
| Facies type | 1 | 2 | 1 | 3 | 3 |
| Constant rate (STB/day) | 2000 | 250 | 2000 | 2000 | 4000 |

**Table 5:** Production conditions for all five wells in the field.

field width. Three lithofacies, dolomite, shale, and sand, are distributed throughout the field and have very distinct properties. Table 4 lists the permeability and porosity values for each of the facies types. Table 5 describes the production conditions for all five wells in this field. In Fig. 47, the true facies field is shown with well locations. The darkest area represents facies 1, light grey area for facies 2 and white area for facies 3. Draw-down tests are simulated at the four producing wells which last three days and where 20 bottom hole pressure data were collected from each well. Besides the bottom hole pressure data, it is also assumed the facies at well locations are observed.

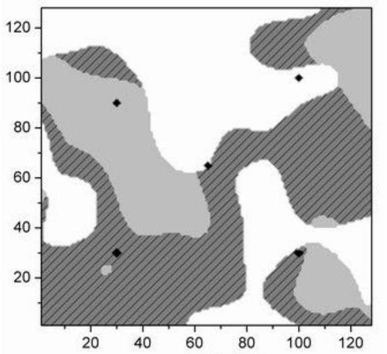


**Figure 47:** The “true” facies map with well locations.

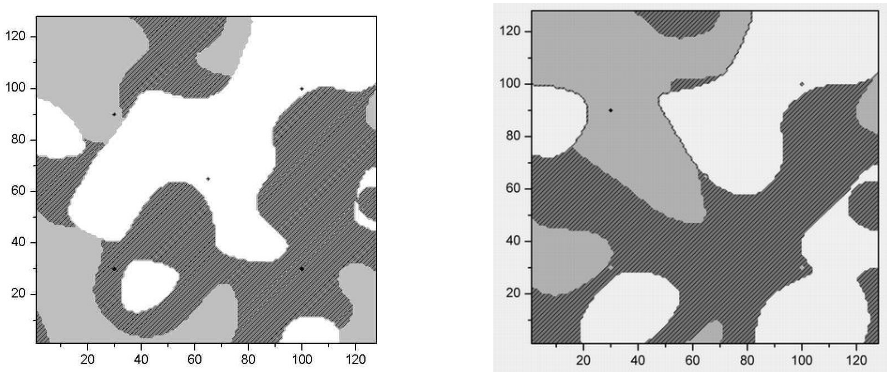
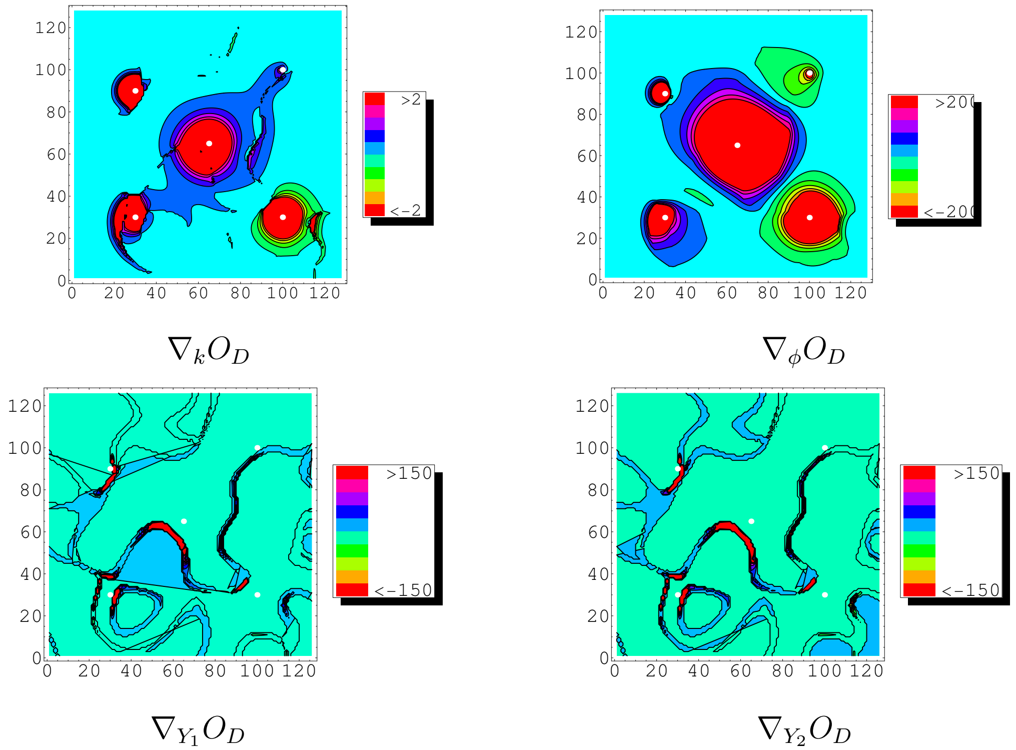


**Figure 48:** The initial facies map and the final facies map after convergence. The objective function reduced to 2% of the original objective function at the final model.

In this problem, the total number of pressure data is 100, and the total number of model parameters is 73734. The number of model parameters is more than twice the number of gridlocks because the moving average method required a margin of $Z_1$ and $Z_2$ fields outside of the facies grid. The width of the margins is determined by the range of the correlation of $Y_1$ and $Y_2$. For such a large scale problem, we chose the adjoint method to compute the gradient of pressure mismatch with respect to the model parameters $Z_1$ and $Z_2$. The limited memory BFGS method (Nocedal, 1980) (L-BFGS) is a quasi-Newton method used in computing the search direction. The L-BFGS method was found to be the most efficient minimization method for automatic history matching in terms of saving computational time and memory (Zhang and Reynolds, 2002).

An initial model has been generated matching the regions of facies observations. The initial facies map is shown on the left of Fig. 48. After 13 LBFGS iterations, the objective function reduced to 2% of the initial value. The facies map for the converged model is shown on the right of Fig. 48. The production data is from the true case shown in Fig. 23.

The intermediate results for computing the gradient of the squared data mismatch



$\nabla_k O_D$

$\nabla_\phi O_D$

$\nabla_{Y_1} O_D$

$\nabla_{Y_2} O_D$

**Figure 49:** The gradient of the objective function with respect to intermediate parameters. The first row are output from the normal adjoint computation. The second row is the gradient of the objective function with respect to each of the two Gaussian fields $Y_1$ and $Y_2$.

$O_d$ with respect to model parameters $Z_1$ and $Z_2$ are shown in Fig. 49. The gradient of the squared data mismatch with respect to the permeability and the porosity fields are computed from the adjoint code of Li (2001). Then the chain rule is applied to compute the gradient of the squared data mismatch with respect to each of the Gaussian fields. As well 4 at the upper right corner is surrounded by facies 3, which has very high permeability, the sensitivity of rock properties to production data is relatively small.

The final matching of production data for each well is shown in Fig. 50. The production data for well 2 has the best match, mostly because the gradient is large in the low permeability region. The final simulated data from well 4 is further away from the observation data, in comparison with the simulated data from the initial model. The reason might be that the magnitude of the gradient around well 4 is relatively small compared to the magnitude in the neighborhood of other wells, thus the facies modification close to well 4 is dominated by the gradients from other wells. If the objective function can be further reduced, the gradients from the data mismatch at well 4 will finally dominate, and the facies distribution around well 4 will be improved, leading to reductions of data mismatch of well 4.

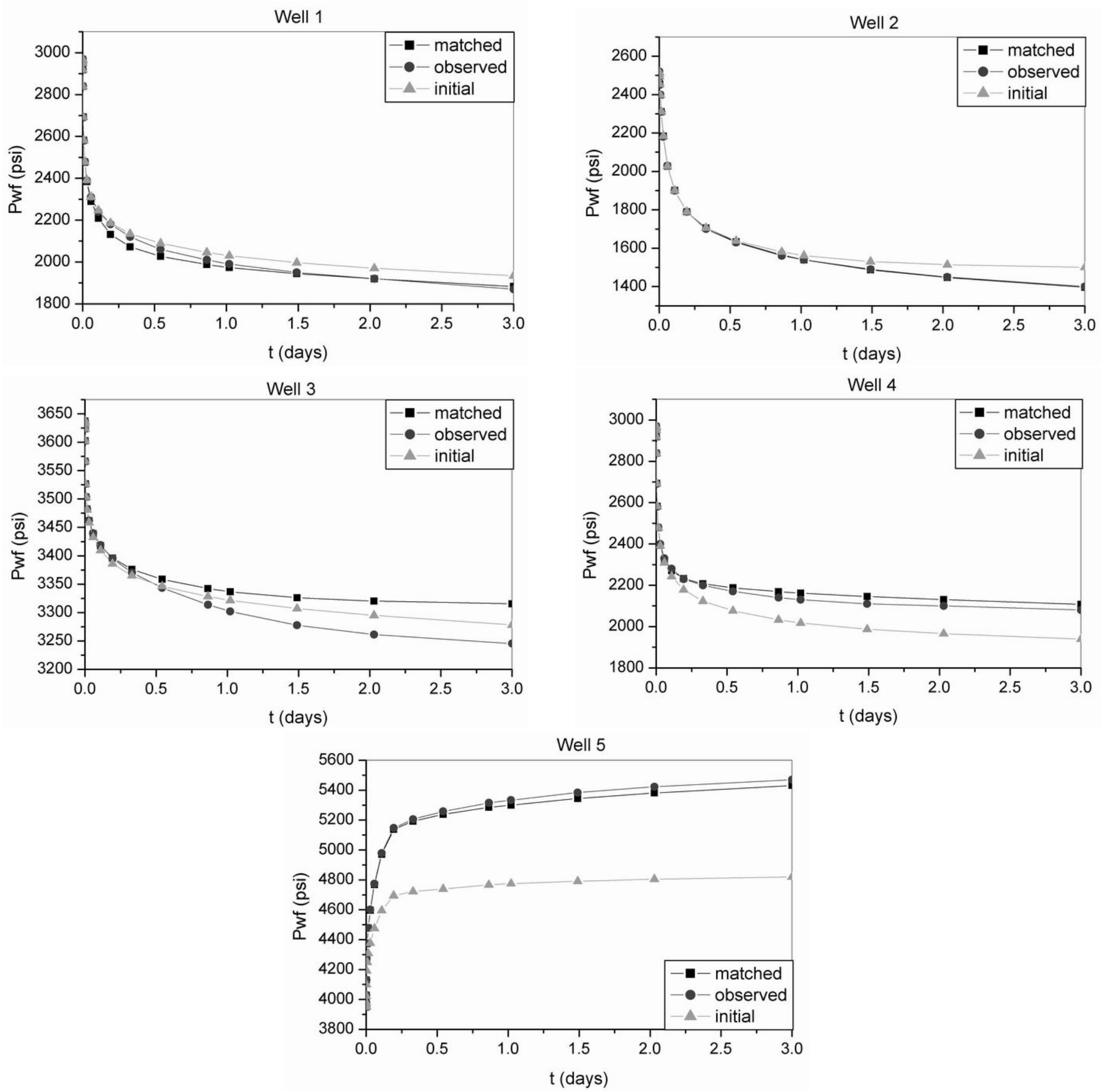


**Figure 50:** Comparison of the production data from the initial and the final model with the observation data.

# CHAPTER VIII

# ENSEMBLE KALMAN FILTER FOR HISTORY MATCHING

In this chapter, the potential of the ensemble Kalman filter (EnKF) method to the problem of estimation of reservoir facies distributions is investigated. Traditional automatic history matching tools such as described in previous chapters are not widely available, largely because of the high computational cost in minimization of the objective function (which represents the weighted mismatch of the estimated reservoir property model to the observed data, and to the empirical initial model), and because of the complexity of the algorithms. The ensemble Kalman filter method takes only one simulation run for each ensemble member, and the simulations can be run simultaneously on parallel processors. After the last data are assimilated, the ensemble contains reservoir models that are not identical but all honor given observation data.

This chapter first presents a comprehensive review of important results from previous studies on applications of EnKF. Then it focuses on the algorithm formulations and the theoretical framework. Two experimental case studies have been done to investigate the feasibility of applying EnKF with the truncated pluri-gaussian method in automatic history matching. The data in the first case are the facies observations at five well locations, where the wells are drilled one-at-a-time. A reservoir flow simulator is not required for matching facies observations. A 1-D water flood model is used in the second case with one injector, one producer, and one observation well. An ensemble of 41 reservoir models with unknown facies boundaries was generated matching frequent production data observations.

The ensemble Kalman filter method was first introduced by Evensen (1994). It begins with an ensemble of initial models (typically 40-100) consistent with prior knowledge of the initial state and its probability distribution (uncertainty). After running a simulation on each of the models up to the next observation time, the covariance update is computed directly from the ensemble of states, i.e.,

$$C^p_{y,k} = \frac{1}{N_e - 1} \sum_{i=1}^{N_e} (y^p_{k,i} - \bar{y}^p_k)(y^p_{k,i} - \bar{y}^p_k)^T, \tag{67}$$

where the subscript $k$ indicates the number of observation time points the simulations have been run to, the index $i$ is for numbering of the ensemble members, the superscript $p$ means present and $u$ means updated. $N_e$ is the total number of state vectors in the ensemble. $\bar{y}^p_k$ is the mean of the $N_e$ ensemble members at the $k$th step. Although the covariance update is still too large to store in computer memory, it is never necessary to be computed explicitly as only a few columns are needed for computation of Kalman gain.

EnKF has great potential in reservoir characterization for two major reasons. First, any reservoir simulator can be used in the EnKF history matching system without excessive work. This idea becomes clear after the illustration of the EnKF formulation in section 8.2. Second, with the increase in deployment of permanent downhole sensors for monitoring pressure, flow rate and other variables, large amounts of data become available with small time intervals in between. Assimilation of these data is a problem of continuous model updating. The Ensemble Kalman filter is by nature well suited to solving such a problem.

## 8.1 Theoretical framework

### 8.1.1 The standard Kalman filter

The Kalman filter method consists of two sequential steps. One is a forecast based on solution of the dynamical equations for flow and transport in the reservoir. The other is data assimilation to update the model by correcting the variables describing the state of the system to honor the observations. The model for the Kalman filter is referred to as the state vector, which contains all the uncertain and dynamic variables that define the state of the system. At a certain time step $i$, the state vector for the reservoir model is expressed as:

$$y^i = [(m^i)^T, d(m^i)^T]^T, \tag{68}$$

where $m^i$ consists of variables for rock properties and flow system in every gridblock, $d(m^i)$ is the simulated data from the previous simulation run. The size of $d(m^i)$ does not have to be constant and it depends on the number of observation data at time step $i$.

In the standard Kalman filter, the updated state vector after each observation assimilation is a linear combination of the measurement and the predicted model state vector:

$$y^u = y^p + K(d_{\text{obs}} - Hy^p), \tag{69}$$

where $y^p$ is the present state vector, $y^u$ is the updated state vector, the weighting term $K$ is called the Kalman gain. $H$ is the measurement operator that extracts the simulated data from the state vector $y^p$. Therefore it has 1s corresponding to data and 0s elsewhere. The state vector in the true model gives the true data, i.e.

$$d_{\text{obs}} = Gm_{\text{true}} + \epsilon = Hy_{\text{true}} + \epsilon, \tag{70}$$

where $\epsilon$ is a vector of unknown observation errors, and $E[\epsilon\epsilon^T] = C_D$. $m_{\text{true}}$ is the true reservoir property field, and $G$ is the sensitivity matrix between reservoir model and observation data. Traditional automatic history matching estimates $m_{\text{true}}$ knowing $d_{\text{obs}}$. Computation of the sensitivity matrix $G$ required the greatest computational effort. The Kalman gain is computed as:

$$K = C_y H^T (H C_y H^T + C_D)^{-1}, \tag{71}$$

where $C_y$ is the covariance matrix for the state vector $y$:

$$C_y = \begin{bmatrix} C_M & C_M G^T \\ G C_M & G C_M G^T \end{bmatrix}. \tag{72}$$

The best estimate of $y$ at each step of data assimilation is the state vector that minimizes the objective function

$$S(y) = \frac{1}{2}(Hy - d_{\text{obs}})^T C_D^{-1} (Hy - d_{\text{obs}}) + \frac{1}{2}(y - y^p)^T C_y^{-1} (y - y^p). \tag{73}$$

Taking the derivative of $S(y)$ with respect to $y$, and let it equal to zero, the best estimate of $y$ is obtained.

$$< y >= y^p + C_y H^T (H C_y H^T + C_D)^{-1} (d_{\text{obs}} - H y^p) \tag{74}$$

Comparing Eq. 71 with Eq. 74, it is clear that the Kalman filer estimation minimizes the objective function. Because

$$C_y H^T = \begin{bmatrix} G C_M \\ G C_M G^T \end{bmatrix}, \tag{75}$$

and $H C_y H^T = G C_M G^T$, the best estimate of a reservoir model in terms of $C_M$ and $G$ is

$$< m >= m^p + C_M G^T (G C_M G^T + C_D)^{-1} (d_{\text{obs}} - G m^p). \tag{76}$$

Traditional history matching normally matches all the observation data at the same time. Therefore $d_{\text{obs}}$ contains all observations, and $C_D$ is the covariance matrix for all the data. The posteriori model covariance after all data assimilation is

$$\begin{aligned} C_{M'} &= (C_M^{-1} + G^T C_D^{-1} G)^{-1} \\ &= C_M - C_M G^T (C_D + G C_M G^T)^{-1} G C_M. \end{aligned} \tag{77}$$

In Kalman filter, the posteriori covariance needs to be computed for each model update. The covariance matrix is typically very large and involves large matrix inversion computation. The sensitivity $G$ is also large and expensive to compute for nonlinear problems.

### 8.1.2 Extended Kalman filter

For nonlinear problems, the relation between the model and the data can be expressed as a function of the model

$$d_{\mathrm{m}} = g(m) + \epsilon, \tag{78}$$

where $g(\cdot)$ is a nonlinear function. The sensitivity $G$ can be approximated by the tangent linear operator (Jacobian) of $g(m)$. Therefore the extended Kalman filter is an approximate method that samples from the linearized model space.

$$y^u = y^p + K(d_{\mathrm{obs}} - H y^p),$$

where the Kalman gain is computed from

$$K = C_y H^T (H C_y H^T + C_D)^{-1}.$$

The state covariance matrix $C_y$ has to be computed from

$$C_y = \begin{bmatrix} C_M & C_M G^T \\ G C_M & G C_M G^T \end{bmatrix}.$$

When the size of the state vectors is $N_y$, the dimensions of the state covariance matrix $C_y$ is $N_y \times N_y$. Therefore this approach is limited to very small problems.

### 8.1.3 Ensemble Kalman filter

The idea of the ensemble Kalman filter for continuous model updating is propagating an ensemble of initial reservoir models along time to assimilate data, and the statistical information carried among the models at each observation time is used to update the model covariance. The ensemble of state vectors is denoted by $\Psi$:

$$\Psi = \{y_1, y_2, \ldots, y_{N_e}\},$$

where $N_e$ is the number of ensemble members; $y_i$ for $i = 1, \ldots, N_e$ are state vectors. Each of the state vectors in the ensemble Kalman filter contains all the uncertain and dynamic variables that define the state of the system. At a certain time step $t_k$ for $k = 1, \ldots, N_t$, the state vector for the reservoir model is expressed as:

$$y_k = [(m_k)^T, d(m_k)^T]^T,$$

where $m_k$ consists of variables for rock properties and flow system in every gridblock, $d(m_k)$ is the simulated data from the model state $m_k$. The number of simulated data in the vector $d(m_k)$ does not have to be constant since it depends on the number of observation data at time step $t_k$.

The methodology of ensemble Kalman filter for data assimilation consists of two sequential steps. One is the forecast forward in time based on solution of the dynamical equations for flow and transport in the reservoir:

$$y^p_{k,j} = f(y^u_{k-1,j}), \text{ for } j = 1, N_e,$$

where $f(\cdot)$ is the reservoir simulator. The superscript $p$ indicates the "predicted" state. This step does not modify the rock properties, but replaces the pressure, saturation, and simulated data in the predicted state vector. The initial ensemble for $k = 1$ refers to the collection of initial state vectors, which are sampled from the prior probability density function of the state vector before any data assimilation.

The second step is model updating by correcting the variables describing the state of the system to honor the observations. The update to each ensemble member is made using the Kalman update formula:

$$y_j^u = y_j^p + K_e(d_j - Hy_j^p), \text{ for } j = 1, \ldots, N_e$$

where the superscript $u$ denotes "updated", and $K_e$ is the ensemble Kalman gain. $H$ is the measurement operator that extracts the simulated data from the state vector $y^p$:

$$H_k = \begin{bmatrix} 0 & I \end{bmatrix}.$$

$d_j$ is the observation data at current time plus random error that has the same distribution with the measurement error:

$$d_j = d_{\text{obs}} + \epsilon_j, \text{ for } j = 1, \ldots, N_e.$$

The ensemble Kalman gain is computed as:

$$K_e = C_{\Psi,e}H^T(HC_{\Psi,e}H^T + C_D)^{-1},$$

where the covariance matrix of the state vectors $C_{\Psi,e}$ at any time can be estimated from the ensemble members by the standard statistical definition:

$$C_{\Psi,e} = \frac{1}{N_e - 1}\sum_{i,j=1}^{N_e}(y_i^p - \bar{y}^p)(y_j^p - \bar{y}^p)^T,$$

where the indices $i$ and $j$ are for numbering of the ensemble members. $\bar{y}^p$ is the mean of the $N_e$ ensemble members at the current data assimilation step. If the size of each state vector is $N_y$, the covariance matrix $C_{\Psi,e}$ is $N_y \times N_y$. It is not possible to compute or store $C_{\Psi,e}$ except for problems that are quite small. Fortunately, the formulation of the ensemble Kalman gain allows the computation of $HC_{\Psi,e}$ instead of $C_{\Psi,e}$ itself. $HC_{\Psi,e}$ is the last $N_d$ rows of $C_{\Psi,e}$.

## *8.2 Application of the ensemble Kalman filter*

This section is intended to provide a general idea of the application of EnKF to the history matching of the facies distributions. In the truncated pluri-gaussian method for simulation of a facies map, the static reservoir variables to be estimated are the Gaussian fields that define the facies. Observation data can be facies at well locations, and/or production data such as production/injection rate, down-hole pressure, GOR, WOR, etc. As the idea is identical for either hard data or production data, in the following explanation the simulated data are denoted $d_{\mathrm{sim}}$.

In this study, all the geostatistical models for generating the two Gaussian fields are assumed to be known, and the static variables to be modified in history matching are the random Gaussian fields $Y_1$ and $Y_2$. As the hard data measurements do not depend on the dynamic states of the reservoir fluid flow, the state vector for cases with only facies measurements is $y_j = \{Y_1, Y_2, d_{\mathrm{sim}}\}$. The facies measurements can be assimilated one-at-a-time to simulate the process of sequential well placement, in which case $d_{\mathrm{sim}}$ is the facies type of the simulated facies field at the current observation location. When there are production data in $d_{\mathrm{sim}}$, the state vector includes the pressure and the saturation in every gridblock, $y_j = \{Y_1, Y_2, P, S, d_{\mathrm{sim}}\}$. Both Gaussian fields have the same size as the reservoir grid, therefore the size of the state vector is $N_y = 4 \times n_{\mathrm{grid}} + n_d$, where $n_d$ is the number of data obtained at each observation time.

### 8.2.1 The initial ensemble

The initial state vector contains the Gaussian fields from the prior PDF that is decided by geostatistical information, the pressure and the saturation at the earliest measurement time, and the observation data. The procedures for generating the initial ensemble are listed below.

- Compute the covariance templates $L_1$ and $L_2$ from the given geostatistical parameters, such as the types of the covariance, the ranges, the principle directions, and so on. Here denote the dimensions of $L_1$ are $n_{\mathrm{L1x}}$ and $n_{\mathrm{L1y}}$. Similarly, the dimensions of $L_2$ are $n_{\mathrm{L2x}}$ and $n_{\mathrm{L2y}}$. More details on construction of the covariance templates have been provided in the previous section on covariance template construction.

- Generate an ensemble of 40 independent random multivariate Gaussian vectors $\{Z_1, Z_2\} \subset N(\mathbf{0}, I)$. The size of $Z_1$ is $(n_x + n_{\mathrm{L1x}} - 1) \times (n_y + n_{\mathrm{L1y}} - 1)$, and the size of $Z_2$ is $(n_x + n_{\mathrm{L2x}} - 1) \times (n_y + n_{\mathrm{L2y}} - 1)$, for reservoir models with dimensions $n_x \times n_y$.

- For $index = 1, N_e$

  1. Choose the vectors $Z_1$ and $Z_2$ that both ranked *index* in their ensemble group.

  2. Convolve $L_1$ with $Z_1$ in the ensemble to generate the initial $Y_1$ field using the moving average method as described in Eq. 12. Convolve $L_2$ with $Z_2$ to generate the initial $Y_2$ field.

  3. Truncate the Gaussian fields $\{Y_1, Y_2\}$ to generate an initial facies map. The thresholds and facies assignment on the threshold map are also geostatistical parameters that have been decided before history matching.

  4. Assign the permeability and the porosity fields to the facies map. In this study, the permeability and the porosity are assumed homogenous within each facies region, but distinct among the facies.

  5. Set the initial pressure $P_0$ and the initial saturation $S_0$ to each gridblock.

6. Run reservoir simulation until the first observation time. Output the pressure and the saturation at every gridblock. Record the simulated data $\mathrm{d}_{\mathrm{sim}}$.

7. Merge the variables to form a state vector $y_{index} = \{Y_1, Y_2, P, S, d_{\mathrm{sim}}\}$

### 8.2.2 Computation of the ensemble Kalman gain

The ensemble Kalman gain works as a weighting term for assimilation of the observation data in the model updating. As the number of data at each observation time $N_d$ is normally small, the computation of the inversion term $(HC_{\Psi,e}H^T + C_D)^{-1}$ is fast. The covariance of the state vectors can also be written in matrix form as

$$C_{\Psi,e} = \frac{1}{N_e - 1}[\Delta y_1, \Delta y_2, \ldots, \Delta y_{N_e}] \cdot \begin{bmatrix} \Delta y_1 \\ \Delta y_2 \\ \vdots \\ \Delta y_{N_e} \end{bmatrix}, \tag{79}$$

but in fact only the product $HC_{\Psi,e}$ is required. So we define

$$\begin{aligned} A &= H[\Delta y_1, \Delta y_2, \ldots, \Delta y_{N_e}] \\ &= H\Delta\Psi. \end{aligned} \tag{80}$$

Because of the structure of $H$, the matrix $A$ consists of the last $N_d$ rows of $\Delta\Psi$. Express the ensemble Kalman gain in terms of $A$:

$$K_e = \frac{1}{N_e - 1}\Delta\Psi A^T (\frac{1}{N_e - 1} A A^T + C_D)^{-1}. \tag{81}$$

### 8.2.3 Update the ensemble of states

Burgers et al. (1998) suggested that without adding measurement error to the observation, the resulting model variance is too low. Thus a vector of random error

$\epsilon \subset N(0, C_D)$ is added to the observations for each state vector update. The $H$ matrix does not really need to be constructed, because $Hy_j^p$ is just the simulated data $d_{\mathrm{sim,j}}$. The updated state vectors are computed as

$$y_j^u = y_j^p + K_e(d_{\mathrm{obs}} + \epsilon_j - d_{\mathrm{sim,j}}). \tag{82}$$

Equation 82 is applied to each of the state vectors.

## *8.3 Matching facies observations*

One concern with the application of the ensemble Kalman filter to problems of history matching of facies is that the facies type is a discontinuous indicator variable, but the Kalman filter method has an underlying assumption of Gaussian distributed continuous state variables. In this preliminary study, experiments with the application of the ensemble Kalman filter on a 2-D field with five wells drilled in sequence are performed. One facies observation is assimilated to improve the reservoir model after each well is drilled. The key issue in this case is to account for the difference between the observed facies and the simulated facies when updating the states. In a second problem, the spatial correlation for one Gaussian field is anisotropic with the ratio of correlation lengths in two principle directions is 2.0. Eighteen facies observations are made over the field. The objective is to test the sampling capacity of the EnKF with larger number of hard data constraints with general geostatistical features.

### 8.3.1 Five facies observations

The reservoir model is $128 \times 128$. Both Gaussian fields used to generate the facies are known to be isotropic with the variogram range of about the width of 20 gridblocks. Three facies are present in the field, which are denoted as facies 1, facies 2, and facies 3. In the computation of the difference between the observed and the simulated facies, the facies are assigned values 1, 2, and 3 respectively according to their notations.

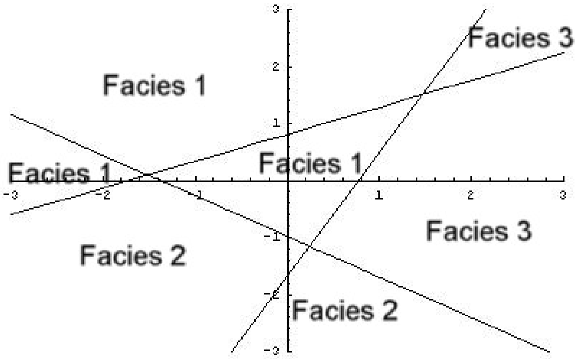


**Figure 51:** The true threshold map used for generation of the true facies map and all the facies realizations for the case of matching five facies observations.

| Well | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| x | 30 | 30 | 100 | 100 | 65 |
| y | 30 | 90 | 30 | 100 | 65 |
| facies | 3 | 1 | 3 | 1 | 1 |

**Table 6:** Facies observations from each of the five wells.

Prior facies maps are generated by truncating two prior Gaussian fields $Y_1$ and $Y_2$ using the threshold map shown in Fig. 56. Although the observation data are facies, and no reservoir simulation is needed in this example, it is assumed that the wells are drilled one-by-one such that the facies data are assimilated one-at-a-time. The facies observations are listed in Table 9 with the well number and locations.

In this problem, the state vectors are formulated as $\{Y_1, Y_2, F_{\text{sim}}\}$. The size of the state vectors is $2 \times nGrid + 1$, where $nGrid$ is the total number of gridblocks in the reservoir model. Generation of the initial states here is simpler than introduced in the last section. After computing the ensemble of Gaussian fields, truncation is applied to generate facies maps for each ensemble state. The $F_{\text{sim}}$ is then read from each facies map corresponding to the location of current facies observation. As the facies is an indicator variable, no observation error is considered in this case study. As a result, the updated data in the state vectors always match facies observations. But, because the relation between the Gaussian fields and the facies map is not linear; the model variables are not always consistent with the observation after updating. Modifications to the Gaussian variables in the update step is a linear approximation to the prediction of simulated facies. Multiple iterations on assimilation of the same data are necessary to have the state vectors match the observation. The nonlinearity is mainly because the facies arrangement in the truncation threshold map is discontinuous. Increasing or decreasing of the Gaussian variables $Y_1$ and $Y_2$ may not necessarily increase/decrease the values assigned to each facies type.

For the same reason, updating the Gaussian fields to match the current facies observation may quite possibly destroy the match to the previous facies observations. Thus after assimilating all the data, the state vectors are checked to see if all the state vectors match the facies data. If not, the ensemble Kalman filter is applied to the state vectors to assimilate the data once again. The flow chart in Fig. 52 provides a detailed structure of the ensemble Kalman filter application to this test problem.

The ensemble in this example contains 41 state vectors. After the assimilation of all the facies observations at the first time, five state vectors do not honor all the data, thus the assimilation was applied once again to all the current 41 state vectors. After assimilating the data twice, all the states honor the data. In Figs. 53 and 54, the facies maps are shown from the first six states in the ensemble. The black regions indicate facies 1, gray for facies 2, and white for facies 3. The columns on the left show the initial facies maps, and the columns on the right show the final facies maps. Well locations are marked by round dots in each facies map. Although none of the facies maps from different state vectors seems alike, all the final facies maps have facies types at well locations that agree with the observation data. Comparing the initial facies maps with the corresponding final ones on their right, the facies around wells were shifted to match the data while preserving some of the initial features.

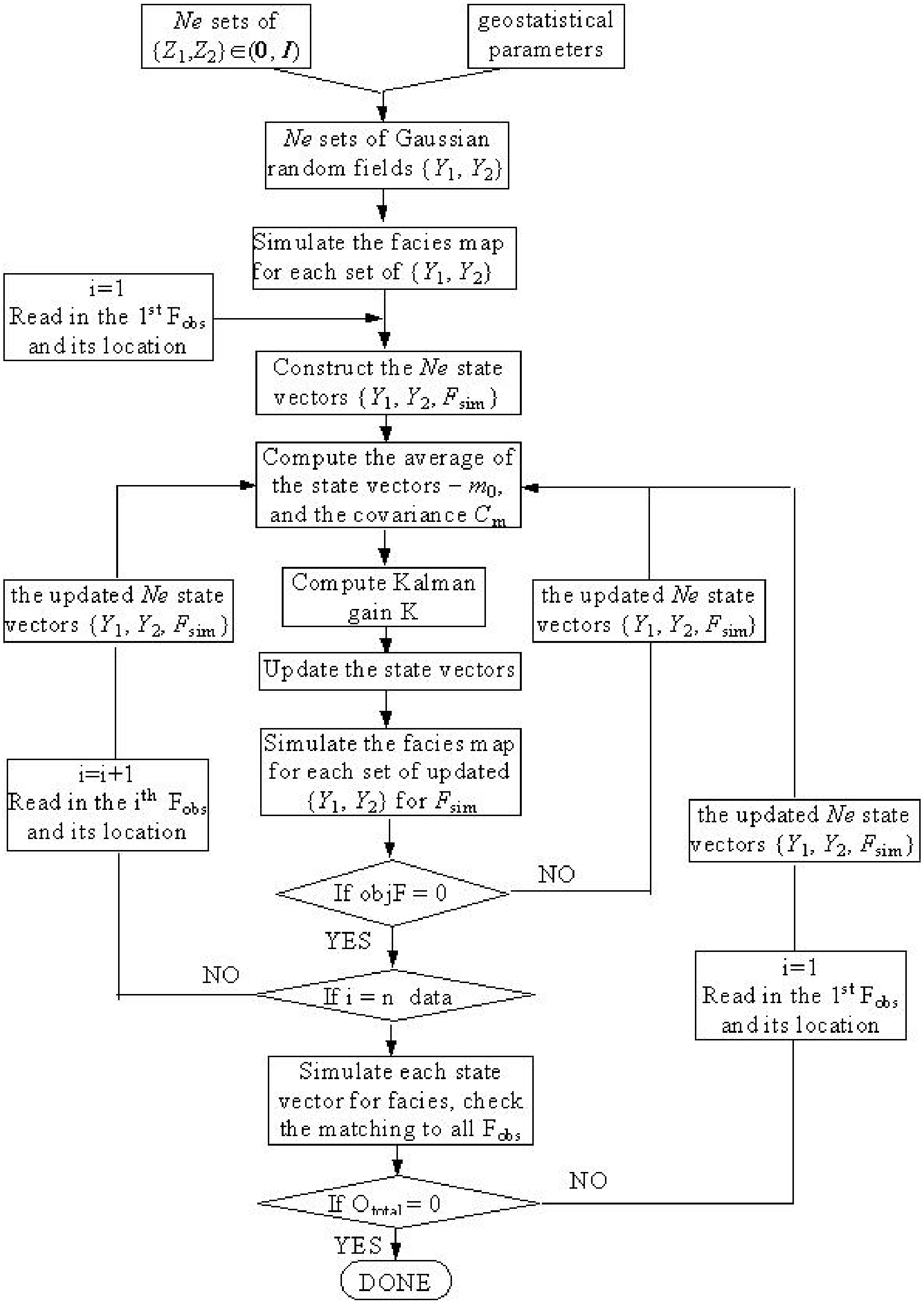
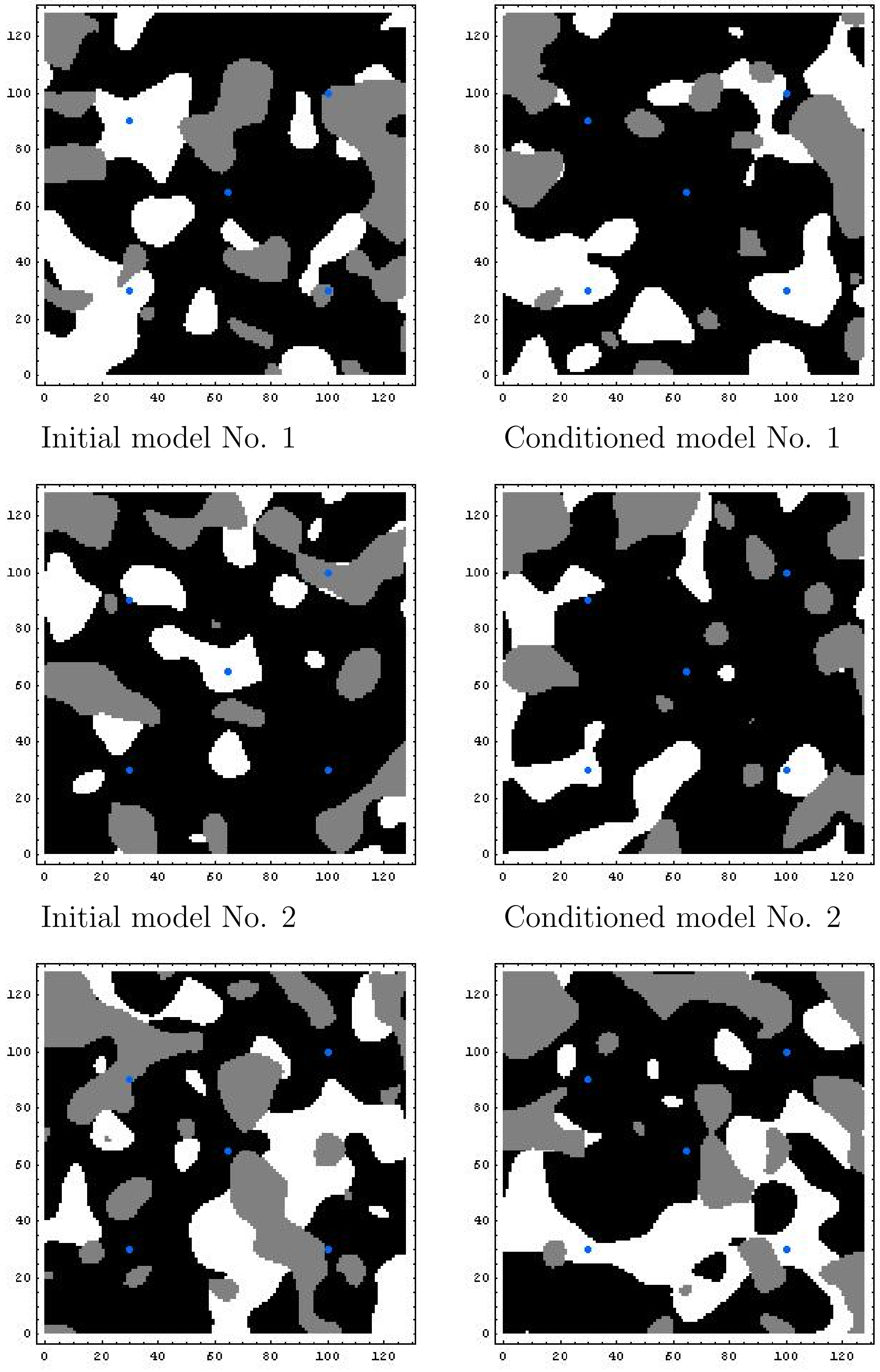

*Ne* sets of
$\{Z_1, Z_2\} \in (\mathbf{0}, \boldsymbol{I})$
geostatistical parameters
*Ne* sets of Gaussian random fields $\{Y_1, Y_2\}$
Simulate the facies map for each set of $\{Y_1, Y_2\}$
i=1
Read in the 1st $F_{obs}$ and its location
Construct the *Ne* state vectors $\{Y_1, Y_2, F_{sim}\}$
Compute the average of the state vectors – $m_0$, and the covariance $C_m$
Compute Kalman gain K
the updated *Ne* state vectors $\{Y_1, Y_2, F_{sim}\}$
the updated *Ne* state vectors $\{Y_1, Y_2, F_{sim}\}$
Update the state vectors
Simulate the facies map for each set of updated $\{Y_1, Y_2\}$ for $F_{sim}$
i=i+1
Read in the ith $F_{obs}$ and its location
the updated *Ne* state vectors $\{Y_1, Y_2, F_{sim}\}$
If objF = 0
NO
YES
NO
If i = n data
i=1
Read in the 1st $F_{obs}$ and its location
Simulate each state vector for facies, check the matching to all $F_{obs}$
NO
If $O_{total} = 0$
YES
DONE

**Figure 52:** The flow chart for history matching of facies map using ensemble Kalman filter.

One of the popular methods for risk analysis in reservoir engineering is to look at the statistics of an ensemble of possible reservoir models. Although the facies are discontinuous variables, for the purpose of statistics, all 41 final states were taken and the average, median, and variance of the facies maps were computed, which are plotted in Fig. 55. The darkest shade is facies 1 and the lightest shade is facies 3. The top figure is the average facies map. The shades at well locations are either black or white as the facies data are either 1 or 3. To some extent, the darkness of the regions provides information on the possibility of certain facies. The figure in the middle shows the median facies type at every location. The facies type in each gridblock is decided by the median of the facies from the corresponding gridblocks of the 41 facies map. It should be the most probable facies map. The bottom figure is the variance of the facies maps. The variance is zero at well locations.



Initial model No. 1

Conditioned model No. 1

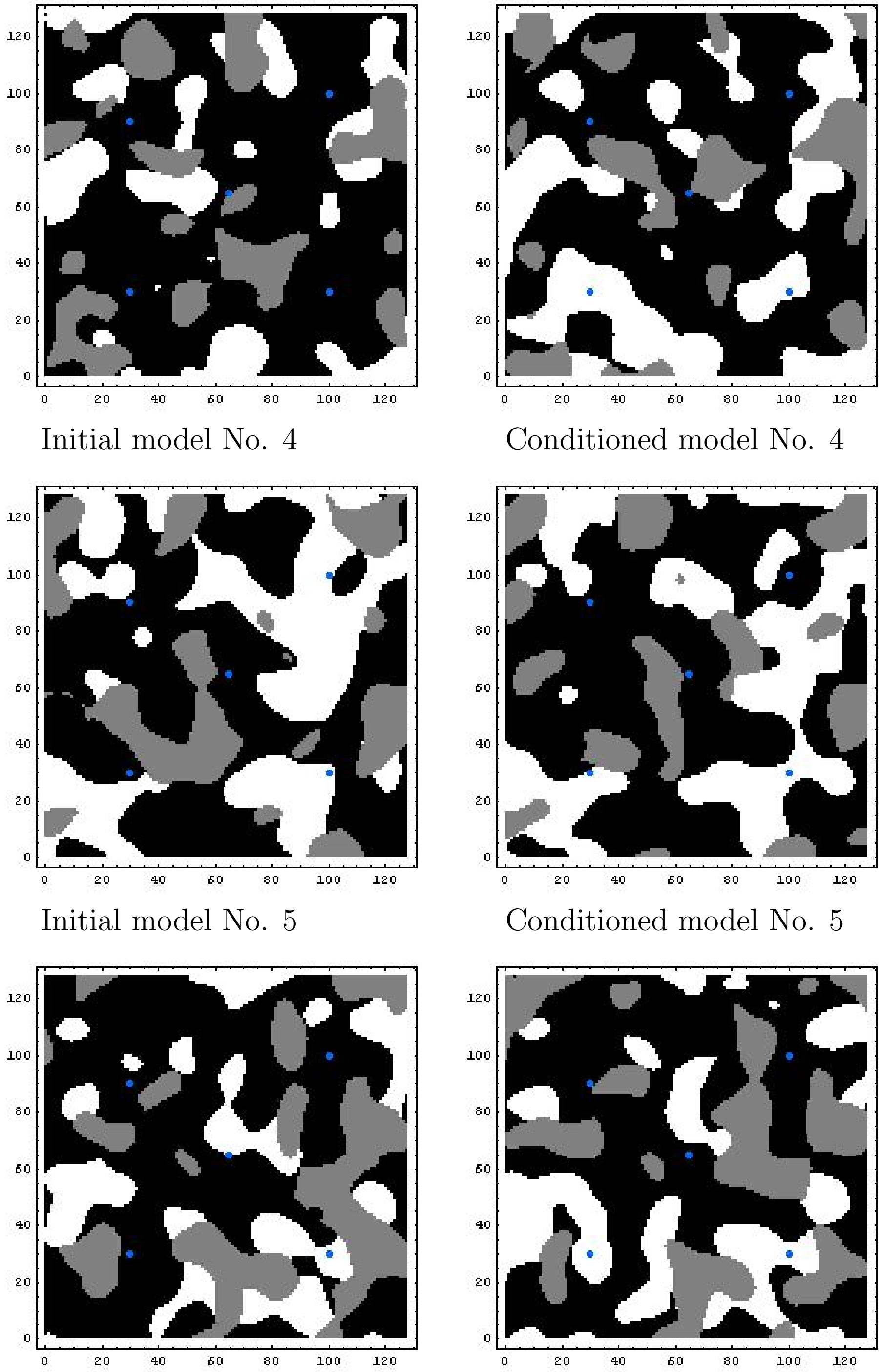
Initial model No. 2

Conditioned model No. 2

Initial model No. 3

Conditioned model No. 3

**Figure 53:** Facies maps from the first three states in the ensemble. The column on the left shows the initial facies maps, and the column on the right shows the final facies maps. Well locations are marked by round dots in each facies map.



Initial model No. 4

Conditioned model No. 4

Initial model No. 5

Conditioned model No. 5

Initial model No. 6

Conditioned model No. 6

**Figure 54:** Facies maps from the 4th, 5th, and 6th states in the ensemble. The column on the left shows the initial facies maps, and the column on the right shows the final facies maps. Well locations are marked by round dots in each facies map.

| Well | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| x | 17 | 49 | 81 | 113 | 33 | 65 | 97 | 17 | 49 | 81 | 113 | 33 | 65 | 97 | 17 | 49 | 81 | 113 |
| y | 25 | 25 | 25 | 25 | 45 | 45 | 45 | 65 | 65 | 65 | 65 | 85 | 85 | 85 | 105 | 105 | 105 | 105 |
| facies | 1 | 2 | 1 | 3 | 2 | 3 | 3 | 1 | 2 | 1 | 3 | 3 | 1 | 1 | 1 | 3 | 1 | 1 |

**Table 7:** Facies observations from each of the 18 wells for the second case of matching facies observations.

### 8.3.2 Eighteen facies observations

The second case is also based on a $128 \times 128$ grid. The Gaussian field $Y_1$ is anisotropic with the principle direction in NW 30°. The range in the principle direction is about the width of 20 gridblocks, which is twice the range in the perpendicular direction. The second Gaussian field $Y_2$ is isotropic with range of about the width of 20 gridblocks. The locations and facies types of the 18 facies observations are shown in Table 7, and the true facies field is shown in Fig. 57.

In this case, the facies observations contain all three types of facies. The facies assignment to the regions in the threshold map has been nonlinear, which made it difficult to correct the Gaussian variables $(Y_1, Y_2)$ of a grid to match the facies observation at that location. In the second case of matching facies observations, the threshold map is divided into three regions by the assignment of facies types. The upper left regions are assigned to facies 1, the bottom regions are assigned to facies 2, and the upper right regions are assigned to facies 3. The algorithm of ensemble Kalman filter modifies the Gaussian variables $Y_1$ and $Y_2$ based on a linear approximation of the relation of the Gaussian variables with the facies mismatch. In the instance where the observation is facies 2 and the simulated facies from the ensembles contain facies 1 and 3, the corrections to the Gaussian variables lead to

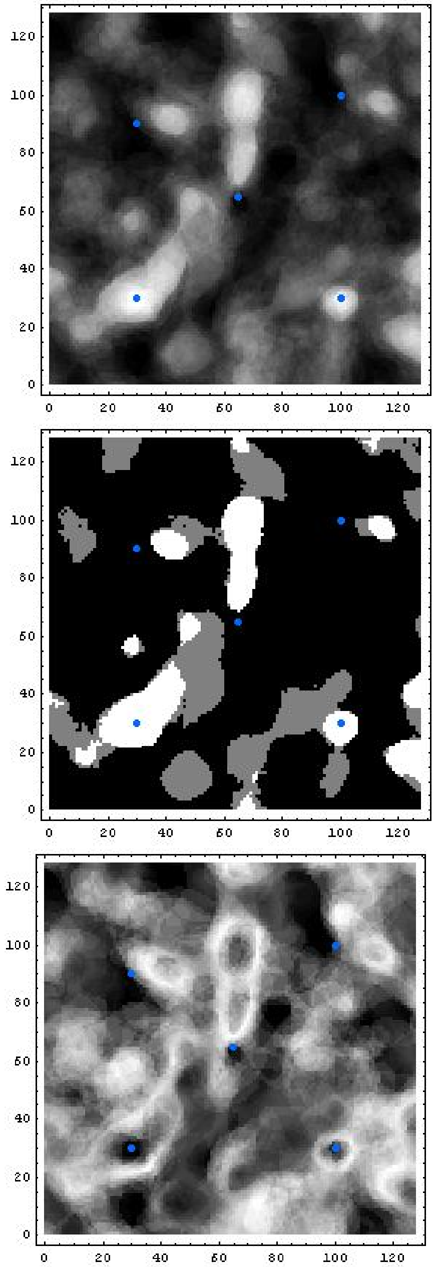


**Figure 55:** The average, median, and variance of the 41 final facies maps.

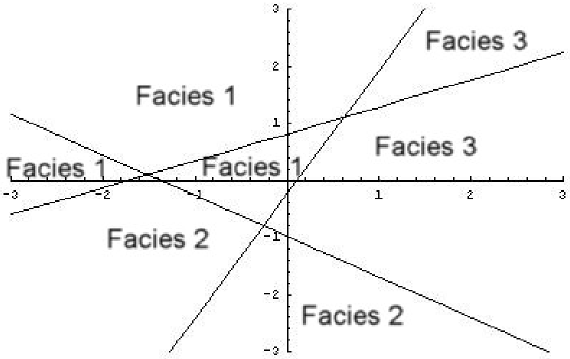


**Figure 56:** The true threshold map used for generation of the true facies map and all the facies realizations.

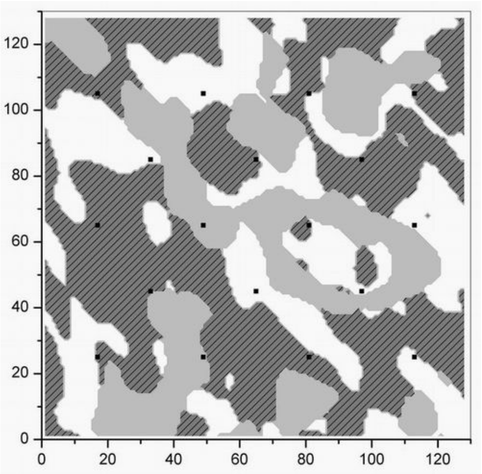


**Figure 57:** The true facies map with all the well locations denoted by black dots. The wells are numbered 1 through 18 from the lower-left corner to upper-right corner by rows.

the area between the facies 1 regions and facies 3 regions. Unfortunately, there is no facies 2 region transition between the facies 1 region and facies 3 region, so the Gaussian variables oscillate at the threshold line between facies 1 and facies 3 without finding the correct region.

The solution to this problem is to define the facies mismatch $f$ instead of the simulated facies in the state vectors. The facies mismatch is defined as:

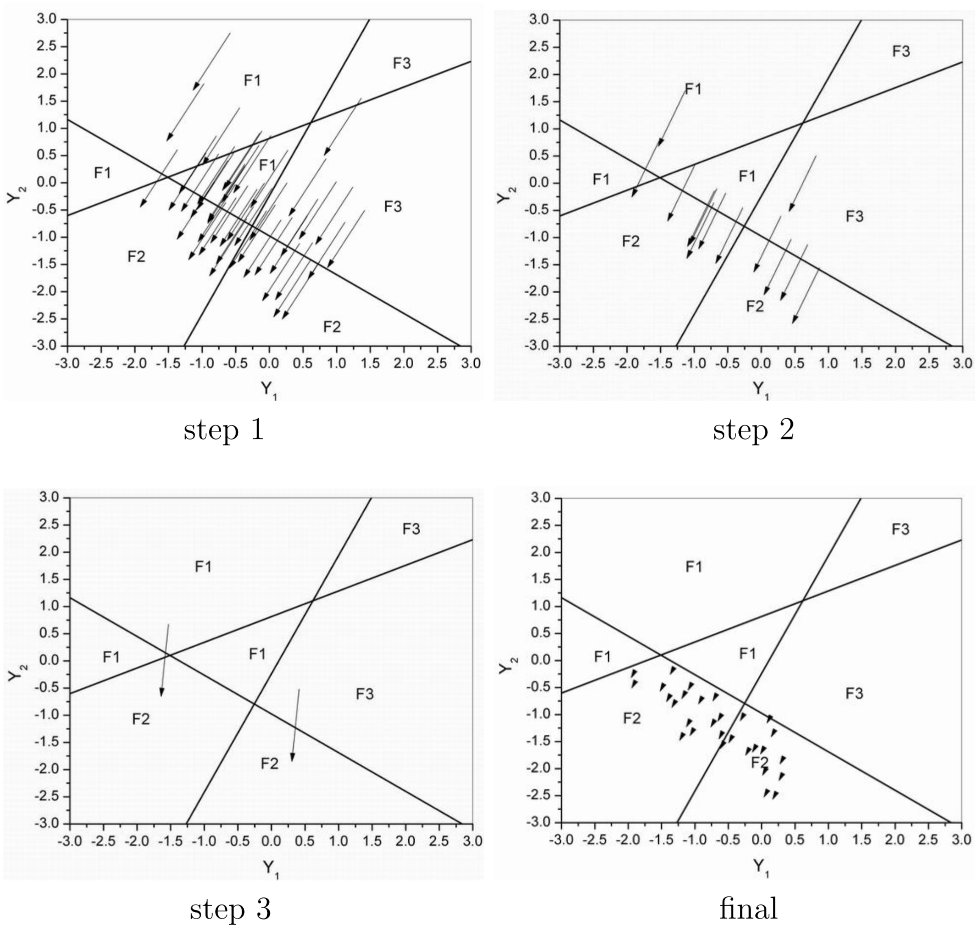
$$f = \begin{cases} 0, & F_{\text{sim}} = F_{\text{obs}} \\ 1, & F_{\text{sim}} \neq F_{\text{obs}}. \end{cases}$$

Consequently, the state vector update step becomes:

$$\begin{aligned} y_j^u &= y_j^p + K_e(0 - f) \\ &= y_j^p - K_e f, \end{aligned}$$

where the zero in the first line results from taking the difference between the observed facies and itself. The first plot in Fig. 58 shows the modification in the first iteration to the Gaussian variables at well 2 from each of the ensemble members that does not match the observed facies. The facies assigned to each region has been labeled as $F_1$, $F_2$, and $F_3$. The threshold map shows there is less probability for the unconditional pdf of the Gaussian variables to fall into facies 2 regions as they are further from the origin. 45 of the 50 initial ensemble members have non-zero facies mismatch. In the first step of model update, 31 ensemble members are corrected to simulate facies 2 at the observation grid $(49, 25)$. Three iterations are required for all the ensemble members to simulate facies 2 at grid $(49, 25)$. The bottom right plot shows the final locations of the Gaussian variables at grid $(49, 25)$.

It was not possible to match all facies observations using 50 ensemble members, the size of the ensemble was increased to 120. 112 final facies maps are obtained matching



step 1

step 2

step 3

final

**Figure 58:** After replacing the simulated facies $F_{\text{sim}}$ with the facies mismatch $f$ in the state vectors, the problem of facies assignment nonlinearity in the threshold map is solved. The second facies observation is matched by all 50 ensemble members in three update steps. The thick lines in each plot are the threshold lines. The arrows point from the starting locations of the Gaussian variables before update to the end locations after update.

all 18 facies observations. Some of the ensemble members contain pairs of Gaussian variables that are too difficult to be modified to the regions with correct facies type in the threshold map, and they are deleted during the assimilation iterations. Figures 60 and  61 show the first 12 facies maps from the initial ensemble of 120. Figures 62 and 63 show the corresponding final facies maps that matched all 18 facies observations.

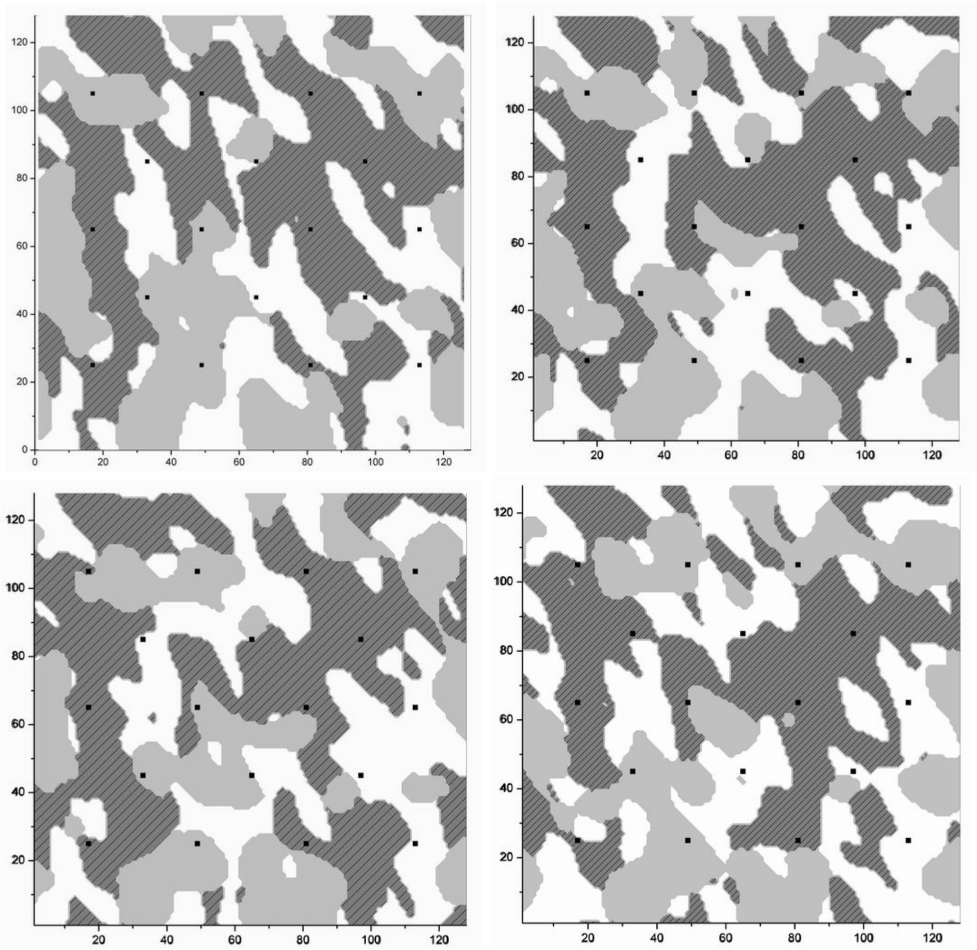


**Figure 59:** The first 4 facies maps from the 50 ensemble members. As the number of ensemble members is too small for the number of variables in the state vectors in order to honor the 18 facies observations, only the facies at wells 1-10 are matched. The wells are denoted by black dots, and numbered 1-18 by rows from the bottom-left corner.

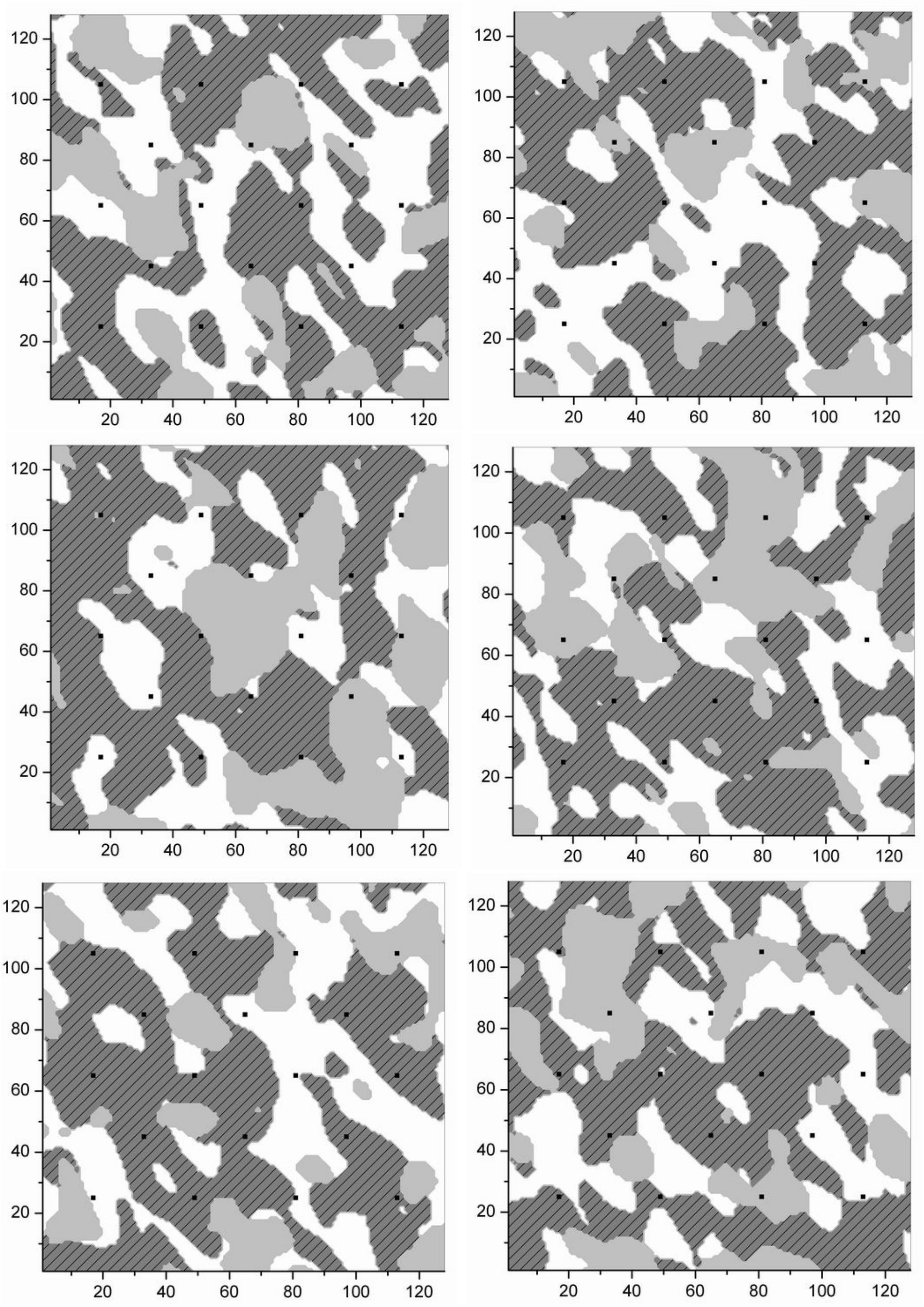


**Figure 60:** The initial facies map realizations 1-6. The black dots are well locations.

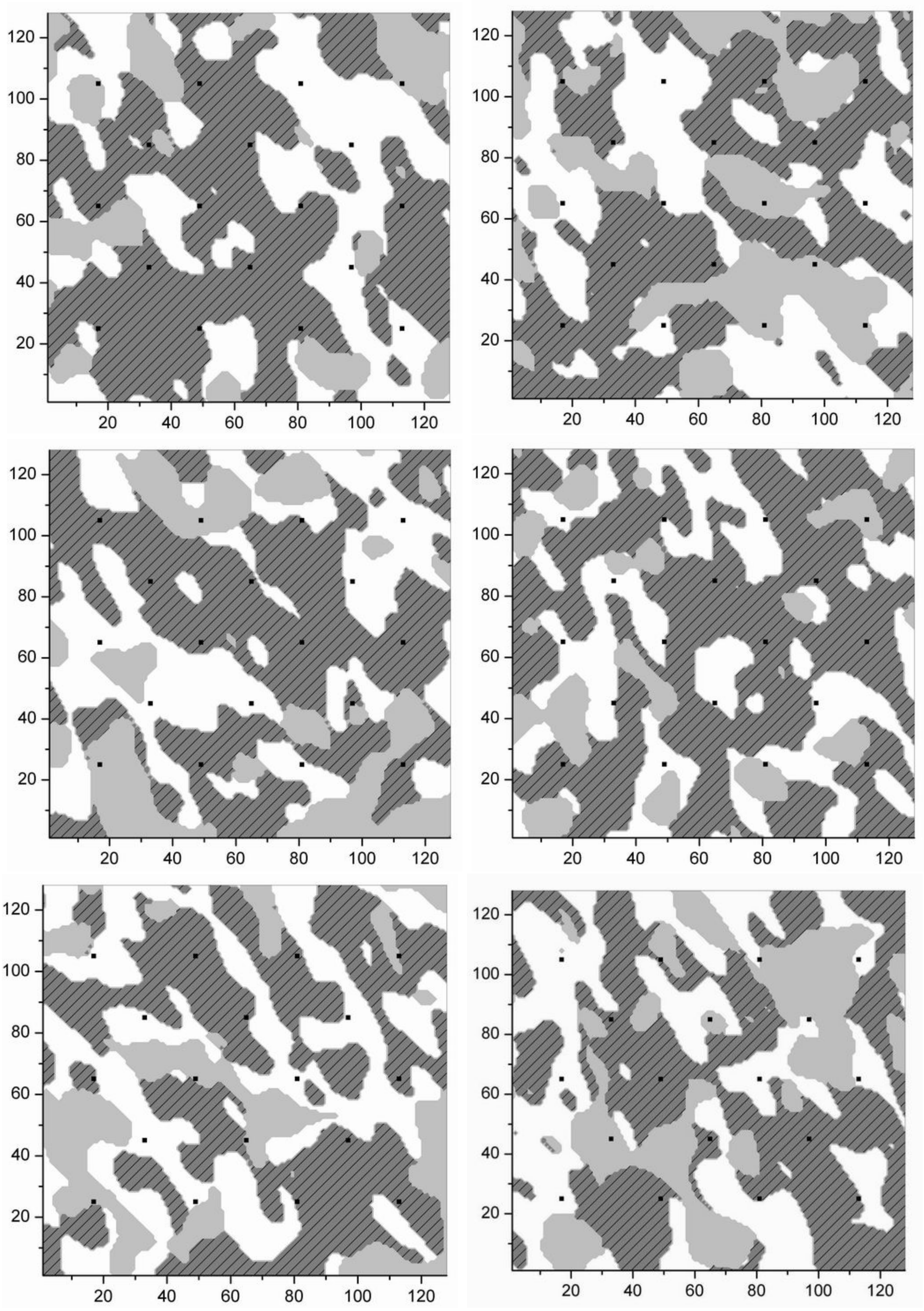


**Figure 61:** The initial facies map realizations 7-12. The black dots are well locations.

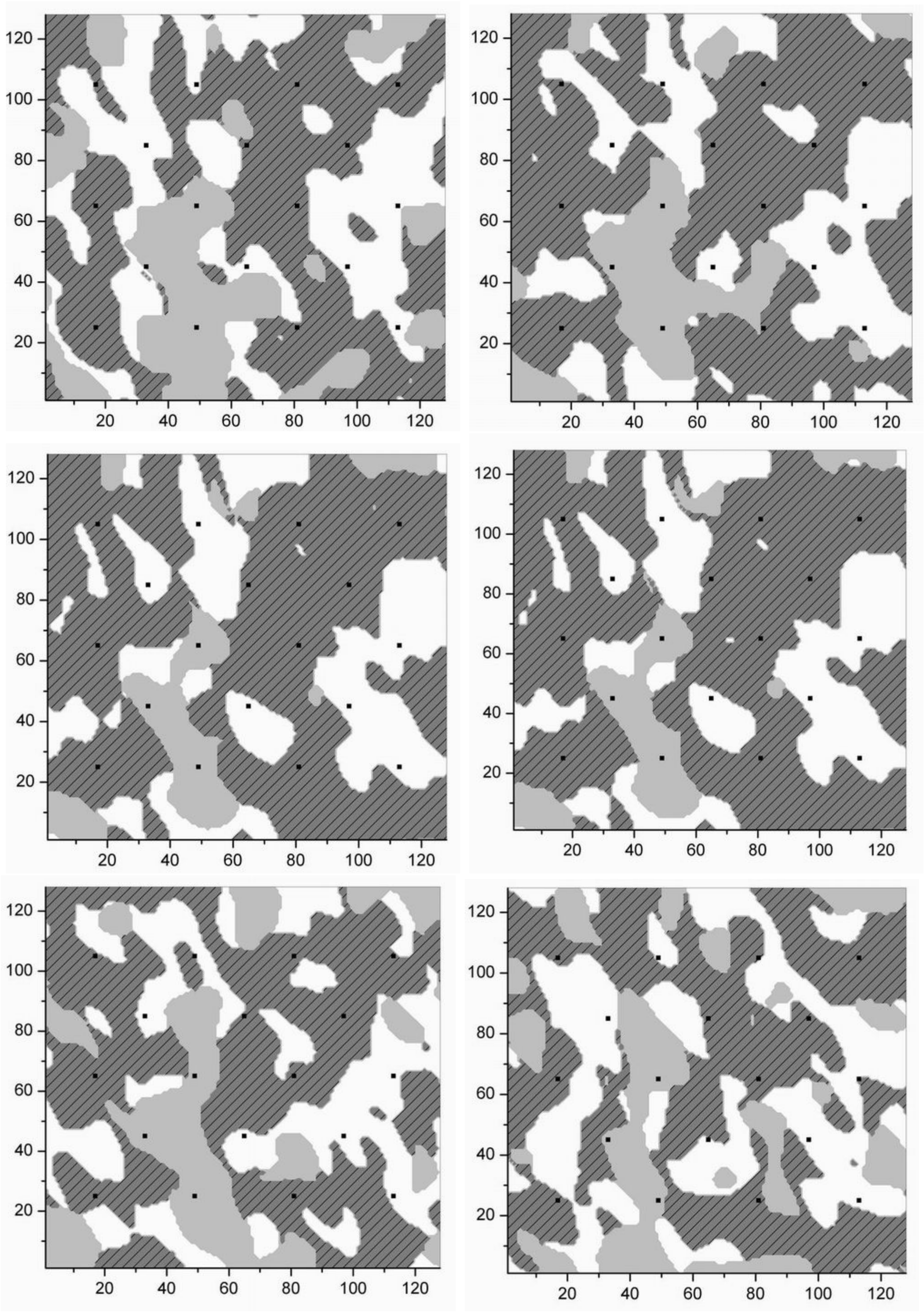


**Figure 62:** The facies map realizations 1-6 that match the facies observations from all 18 wells.

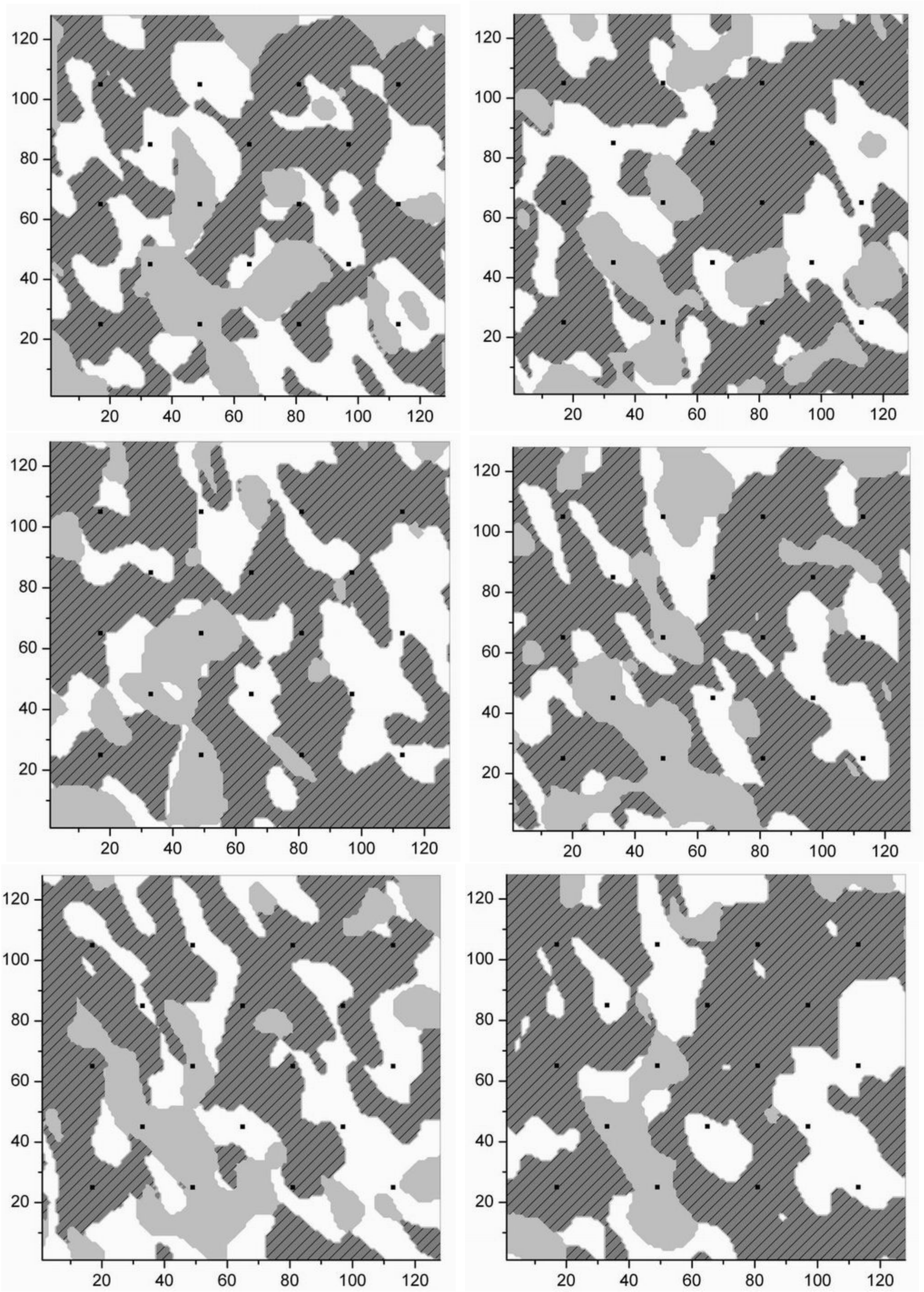


**Figure 63:** The facies map realizations 7-12 that match the facies observations from all 18 wells.

## *8.4 Matching production data*

The purpose of this section is to test the ensemble Kalman filter method on a history matching problem with discontinuous and distinct porosity and permeability fields. As a preliminary experiment, a 1-D reservoir model with an injector at one end and a producer at the other was chosen. One permeability and one porosity are assigned to each facies. The reservoir model has 32 gridblocks in length. Water is injected in gridblock 1 at a constant bottom-hole pressure of 4500 psig. A producer is in gridblock 32 with constant bottom-hole pressure of 1900 psig. An observation well is located in the 20th gridblock to measure the water saturation at that location. Measurements were recorded every 10 days until WOR exceeded 50, at which time the wells are shut-in. Four data are observed at each time step including water injection rate, saturation observation, oil production rate, and water rate. The state vectors are constructed as $Y_1, Y_2, P, S, d_{\text{sim}}$. 41 ensemble members are used to estimate the covariance between the state parameters.

The Gaussian fields used to define the facies both have the same Gaussian-type covariance and an effective correlation range of about the length of 13 grids. Figure 64 shows the truncation to the Gaussian fields. Most of the gridblocks have $(Y_1, Y_2)$ values that fall into regions assigned facies 1, a few fall into the region assigned to facies 2, and even fewer fall into the region assigned to facies 3. Each facies has distinct rock properties. Facies 1 is assigned a permeability of 74 md and a porosity of 0.18. Facies 2 has permeability 20 md and porosity 0.146. Facies 3 has permeability of 372 md and porosity 0.25. The property fields are decided once the facies type for every grid is obtained.

In traditional history matching, the reservoir properties that describe the reservoir model are modified to match data. In this case, the reservoir parameters are the

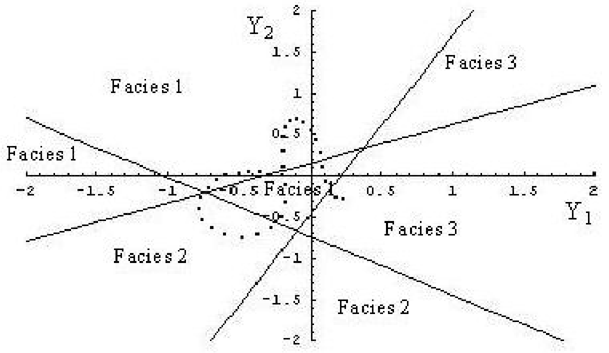


**Figure 64:** The truncation threshold map with the true Gaussian fields, from which the observed production data are generated. The Gaussian fields $(Y_1, Y_2)$ are shown as a dotted curve.

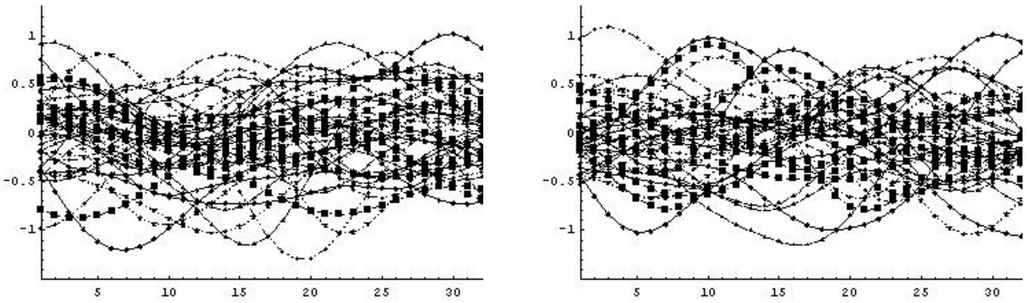
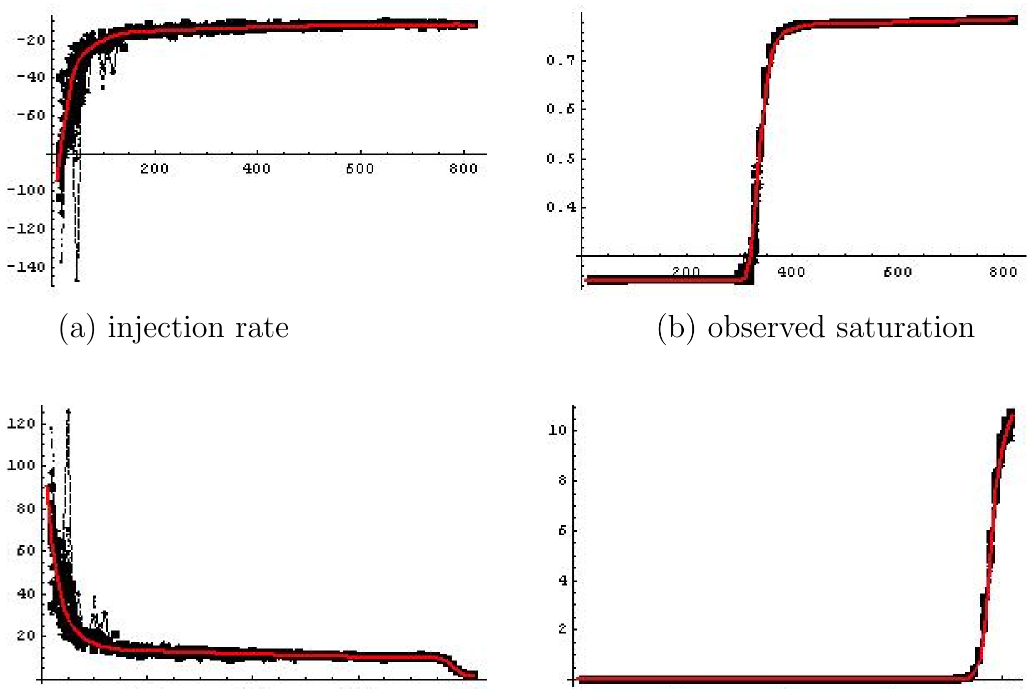


**Figure 65:** The initial ensemble of Gaussian fields. The plot on the left shows all 41 initial $Y_1$, and the plot on the right shows all 41 initial $Y_2$. Both Gaussian fields seem to scattered in the correct distribution $N(\mathbf{0}, I)$.

Gaussian fields $Y_1, Y_2$. They determine the facies map as well as the rock properties. Figure 65 shows the ensembles of both initial Gaussian fields. The plot on the left shows all 41 initial $Y_1$, and the plot on the right shows all 41 initial $Y_2$. Both ensembles seem to have correct mean and standard deviation.

One of the advantages of the ensemble Kalman filter over traditional history matching is that it is only necessary to run the simulation once for each ensemble member. At early time of history matching when not many data have been assimilated, the simulated data from the ensemble are distributed widely, reflecting a large



(a) injection rate

(b) observed saturation

(c) oil rate

(d) water rate

**Figure 66:** The production data from 41 states are compared with the observation data, which is in thick line. The x-axes is the time in days.

uncertainty of reservoir models. The uncertainty decreases as more data are assimilated. Figure 66 shows the evolution of the distribution of each type of production data from all the states. The injection rate and the oil production rate both have wide distribution in the first 100 days. The uncertainty is significantly reduced after 200 days for both plots. The saturation profile from the observation well at grid 20 does not show much uncertainty, largely because the water front moves to the 20th grid after at least 250 days, when all the states become similar. The same thing happens with the water production rate. All the states have water breakthrough almost at the same time.

Figure 67 shows the Gaussian fields after data assimilation. Comparing with Fig. 65, the uncertainty has reduced significantly. It is not clear why the final states in the ensemble do not come closer to matching the true Gaussian fields. The plot on

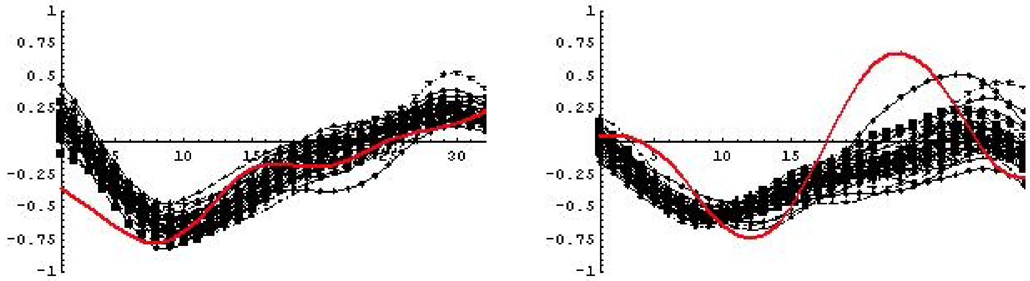


**Figure 67:** The ensemble of 41 final Gaussian fields $Y_1$ and $Y_2$. The thick line in each plot is the true Gaussian field.

the left of Fig. 68 shows the final facies maps with the true facies map in thick line. There is larger uncertainty at the injection end of the reservoir as well as at facies boundaries. All the states have correct facies type away from facies boundaries. The plot on the right compares the median of the ensemble facies maps with the truth. There are two grids out of 32 that do not have the correct facies types. To have a better statistical view of the 41 ensemble states, the variance of the facies maps were also computed, which is shown in Fig. 69. For the grids near the injector, the pressure and the saturation reach about constant when the grids become behind the water front in early days of simulation. Insufficient number of data are assimilated by then, and the covariance of pressure and saturation is small in these grids among the states. Therefore there is more uncertainty in grids near the injector. Grids 6, 14, and 30 are the facies boundary locations in the true case, and they have variance peaks for the states. The peak at grid 30 is about four times as large as the other two at facies boundaries. That is simply because the difference between facies 1 and 3 is regarded as 2, however, the difference between facies 1 and 2 is regarded as 1. In fact, the difference should all be 1 if they are not same facies types. So the variance is actually not much higher around grid 30 than around the other two peaks.

Finally, the saturation profiles of the states along the 1-D reservoir model are shown. Saturations at the 200th, 400th, 600th, and 820th day are output from the

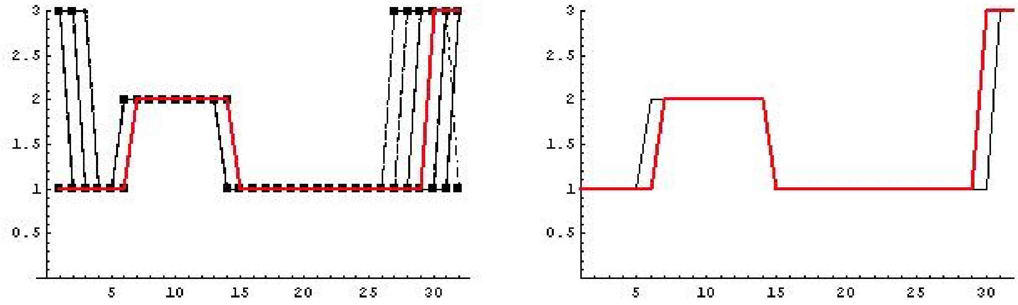


**Figure 68:** Comparison of the final facies maps from all the states with the truth.

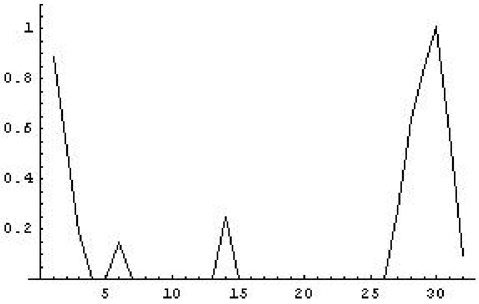
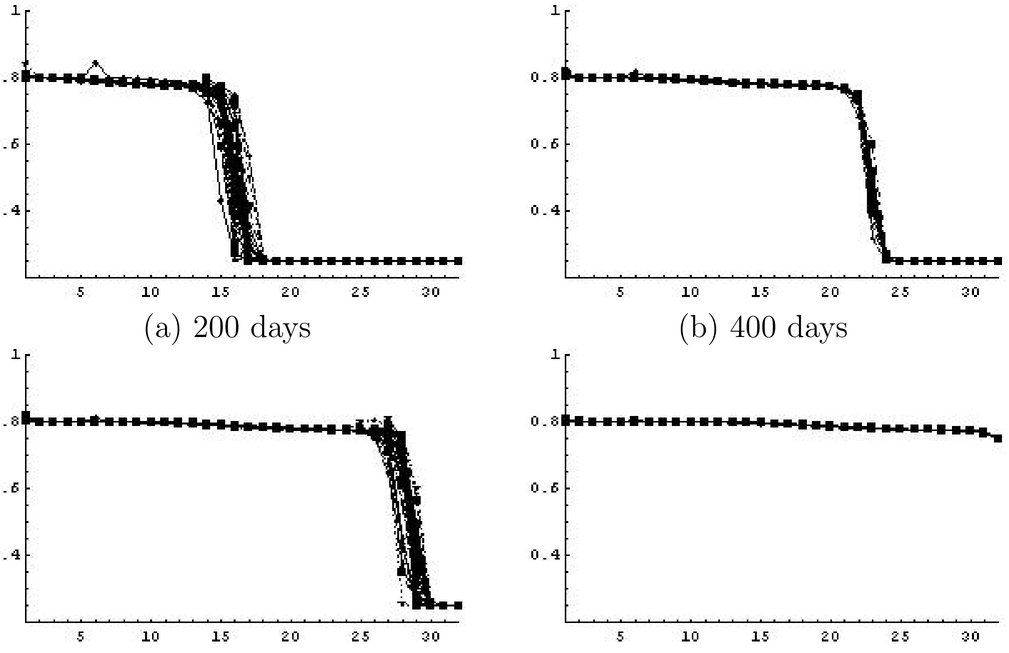


**Figure 69:** The variance of the final facies maps from all 41 states.

simulation to illustrate the propagation of water fronts from all states. The spread at the water front has been reduced at the 200th day, and is further reduced at the 400th day. At the 600th day the water fronts from the states reached a facies boundary from facies 1 to 3, and the spread increased. After water front breakthrough, saturation profiles are nearly all identical.

Although the ensemble Kalman filter is designed for linear problems, it worked satisfactorily for the simple test problem largely because of the frequent data assimilation from the early production period. The small corrections to the state vector made at any time step can be adequately approximated as a linear system. Therefore, this method is ideally suited for the assimilation of time series of data from permanent sensors.

The results from the 1-D water flood problem have shown underestimation of uncertainty in the final distribution of realizations. The ensemble members of the two



(a) 200 days

(b) 400 days

(c) 600 days

(d) 820 days

**Figure 70:** The saturation profile at the 200th, 400th, 600th, and 820th day.

Gaussian fields are in close agreement with each other and in a narrower distribution than the variability of the true Gaussian fields. It seems that the ensemble converges to a small subspace of the correct probability distribution.

The purpose of this second example is to demonstrate the application of the ensemble Kalman filter to the problem of history matching production data for a reservoir with unknown facies boundaries. The true reservoir model is 50 × 50 with four producers and one injector as shown in Fig. 71. The facies in dark grey is facies 1, the light grey areas are facies 2, and the white areas are facies 3. The rock properties for each facies are shown in Table 10. Bottom hole pressures are fixed at 5000 psia for well 1 and at 1500 psia for producers. The field is produced for 195 days and the rates are recorded every 15 days from day 15. Well 3 has water breakthrough on day 183. In this case, there are 14 data for each assimilation step, which includes 1 injection rate, 8 production rates, and 5 facies observations. Every member of the

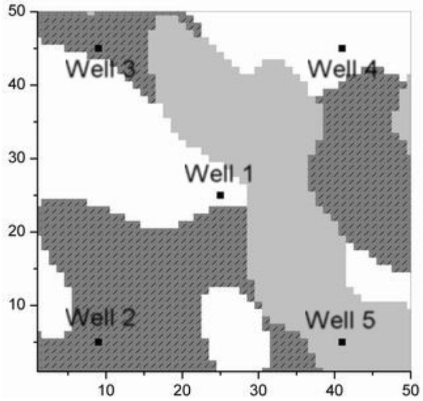


**Figure 71:** The true facies map for the 2-D case study of matching both the facies observations and the production data. This facies map is a $50 \times 50$ square taken from the $128 \times 128$ true facies map in the case study of matching 18 facies observations.

| index | Facies 1 | Facies 2 | Facies 3 |
|---|---|---|---|
| Permeability (md) | 174.0 | 80.0 | 372.0 |
| Porosity | 0.18 | 0.146 | 0.25 |

**Table 8:** Properties of the litho-facies in the synthetic problem.

initial ensemble honors facies observations at the well locations. Six out of 100 initial facies fields for matching production data are shown in Fig. 72.

The facies observations in the ensemble states are kept, because updating of the Gaussian fields from matching production data may change the facies type at well locations. Once the facies type at a well location is wrong, the Kalman correction to the Gaussian fields can be large, and may cause over-shoot of the Gaussian variables. An iteration step over the facies observations is made after each model update to ensure the rock properties at well locations are always correct.

Figure 73 shows 6 out of the 100 facies maps from the final ensemble members after assimilating all the production data. Each of the final facies maps shown has kept some features from the initial state, but in general has developed common features among the ensemble members. Some of the common features do not exist in the true facies map.

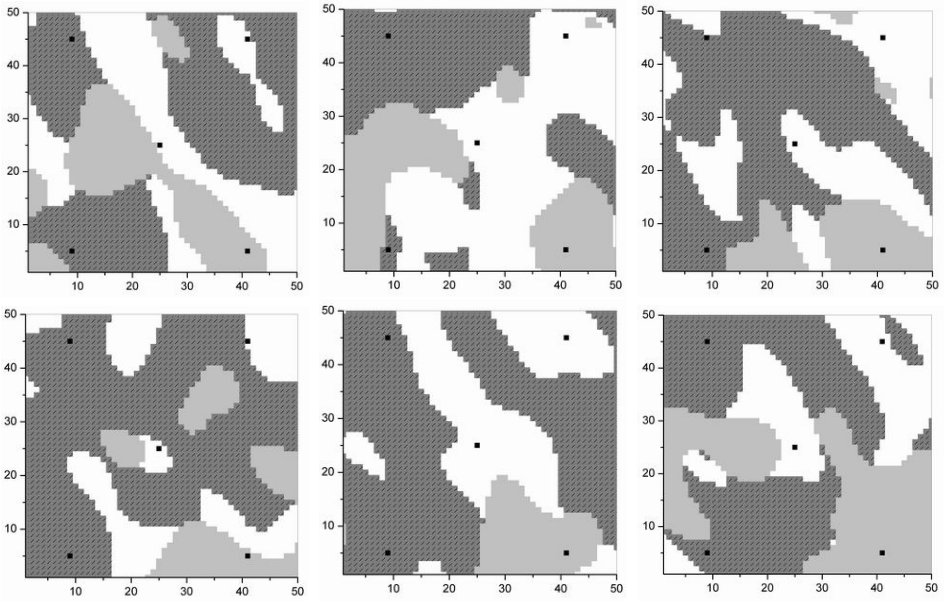


**Figure 72:** The initial facies maps from the ensemble members 1, 20, 40, 60, 80, and 100 that matched the facies observations in the previous case study. Well locations are denoted by the black dots.

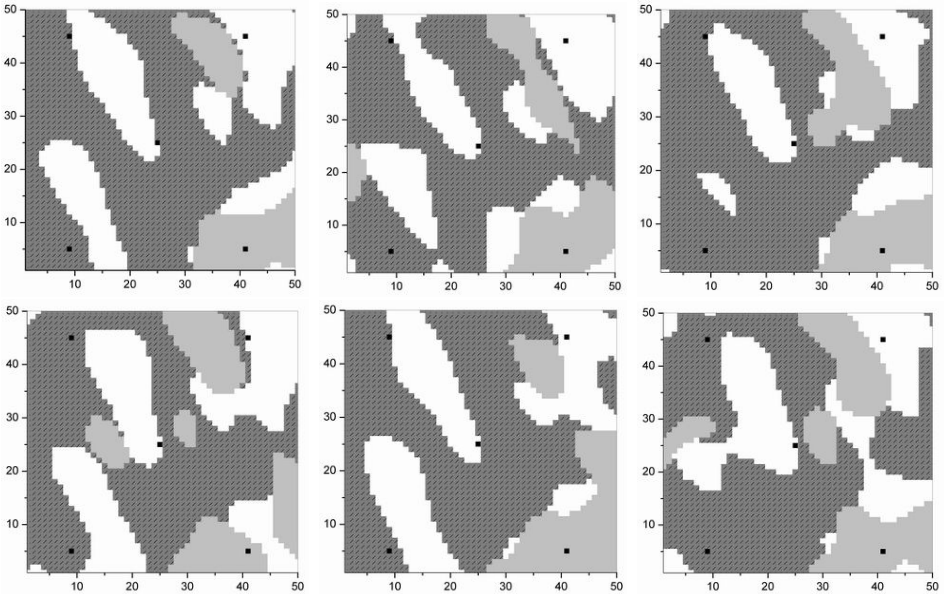


**Figure 73:** The final facies maps from the ensemble members 1, 20, 40, 60, 80, and 100 after matched both the production data and the facies observations. Well locations are denoted by the black dots.

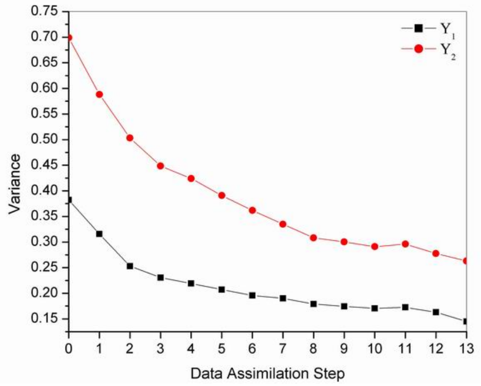


**Figure 74:** The averaged variance of the Gaussian fields vs. the number of data assimilations.

The variability of the final ensemble has obviously been reduced and the subspace spanned by the ensemble members does not seem to include the true facies map. The variance of the Gaussian variables can be used to quantify the reduction in variability due to data assimilation. At every gridblock, the variance over the ensemble is computed, then the average variance over all grids is computed, and the results are plotted as shown in Fig. 74. The unconditional PDF for both Gaussian fields should have variance close to 1. After matching the facies observations, the average variances for the two Gaussian fields have decreased from 1 to less that 0.4, and 0.7, respectively. After assimilating the production data, the variances decreased to less than 0.3. The reduction of the average variance for both Gaussian fields in all the ensemble members indicate that the variability among the ensemble members has reduced and the ensemble members become more and more similar with data assimilations.

The mean of the 100 ensemble Gaussian fields $Y_1$ and $Y_2$ should contain numbers close to zero without obvious patterns if the ensembles are not correlated. The first row of Fig. 75 shows the mean of the 100 initial Gaussian fields $Y_1$ (left) and $Y_2$ right. Areas with mean values greater than 1.5 are shown in white, and less than -1.5 in black.

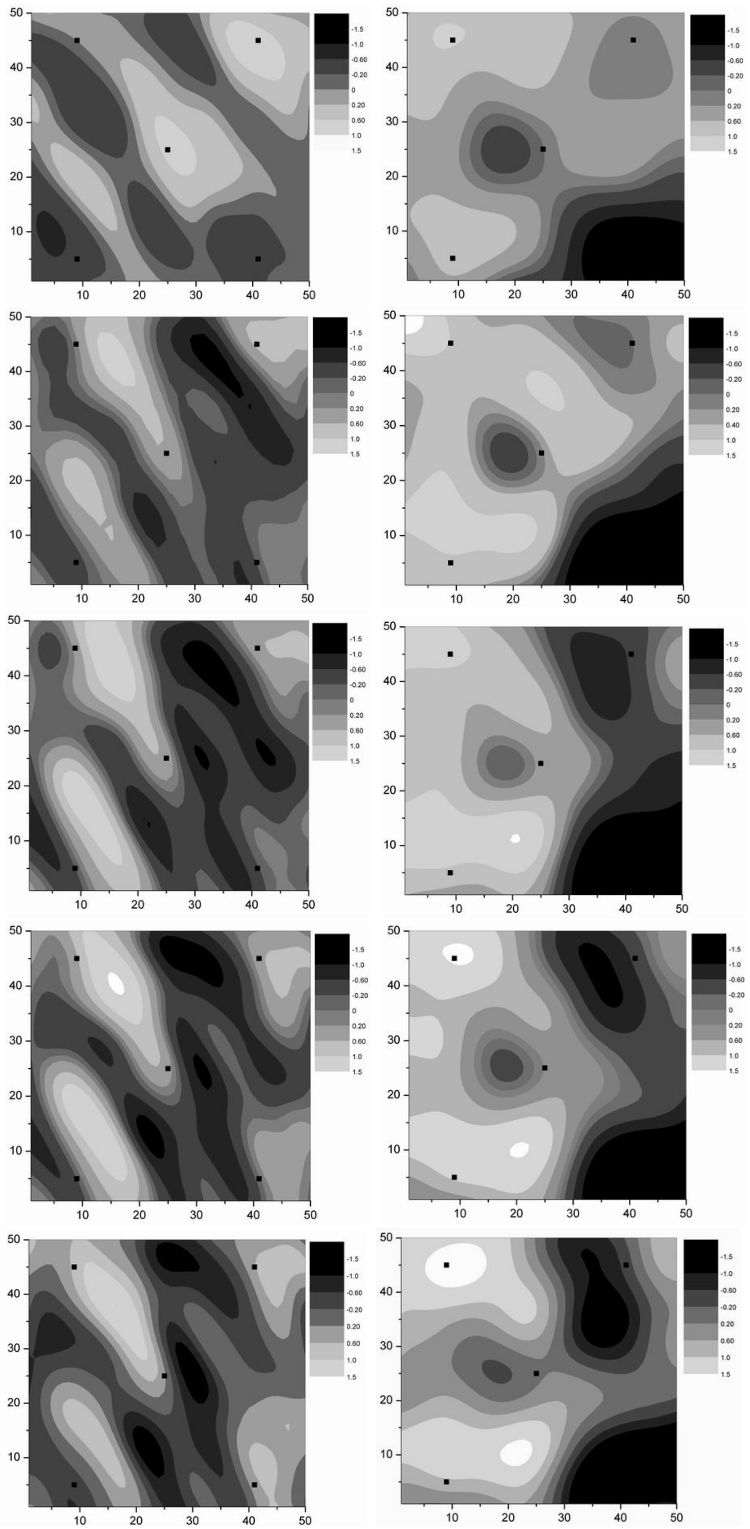


**Figure 75:** The ensemble means of the 100 $Y_1$ (left) and $Y_2$ (right) for initial (top), first (second row), fourth (third row), 12th (fourth row), and 13th (bottom) data assimilation.

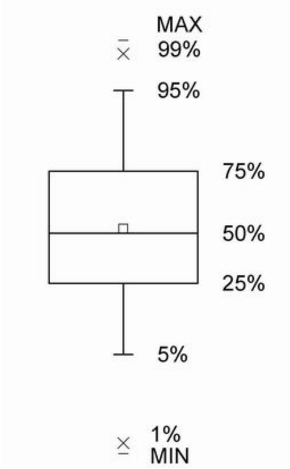


**Figure 76:** Key for interpretation of box plots. The square in the center of the box indicates the mean value.

Box plots are used to represent the distribution of the simulated production data from all the 100 ensemble members over the 195 days of production. Figure 76 shows the key for interpretation of box plots. The simulated rates and the observed data are plotted together in Figs. 77 through 82. The box plot on the left of Fig. 77 shows the injection rates from the initial ensemble conditional only to facies observations. The observed injection rate is plotted as the thick line. The distribution of injection rates from the 100 final reservoir models is much narrower than the initial distribution and almost centered at the observed data (Fig. 77, right).

The distributions of the oil rates from the final ensemble are shown in Figs. 78 to 82; they are all much narrower and closer to the observed data. Figure 82 shows the water rate of well 3 over the 195 days production history from both the initial and the final ensembles. Only a few of the initial 100 states have water breakthrough within 195 days. After data assimilation, almost all the reservoir models have breakthrough in 195 days.

The histograms in Fig. 83 compare the squared data mismatch from the initial ensemble with that from the final ensemble. On average, the squared data mismatch

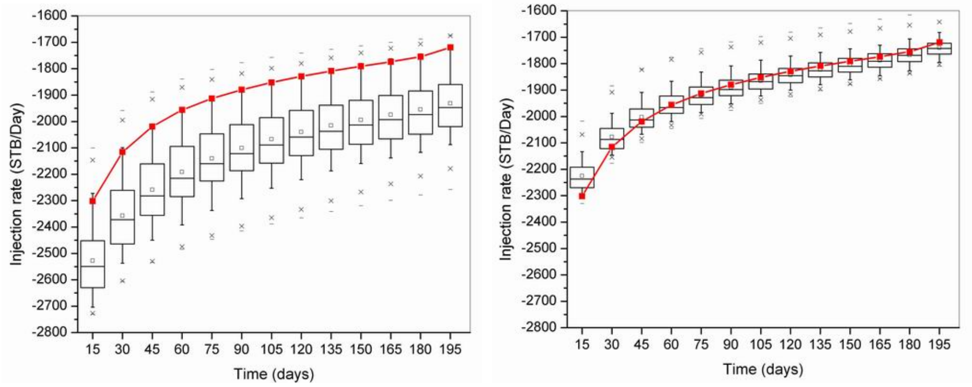


**Figure 77:** The injection rate over the 195 days production history from the initial ensemble (left) and the final ensemble (right). The thick line shows the observed data.

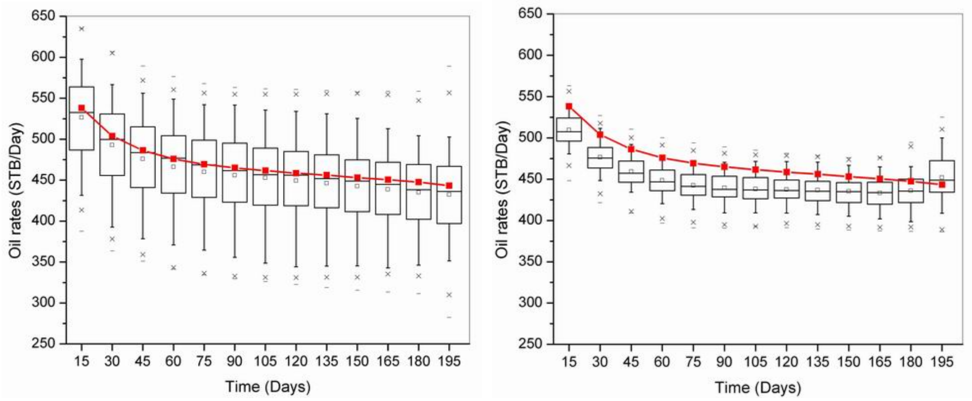


**Figure 78:** The oil rate of well 2 over the 195 days production history from the initial ensemble (left) and the final ensemble (right). The thick line shows the observed data.

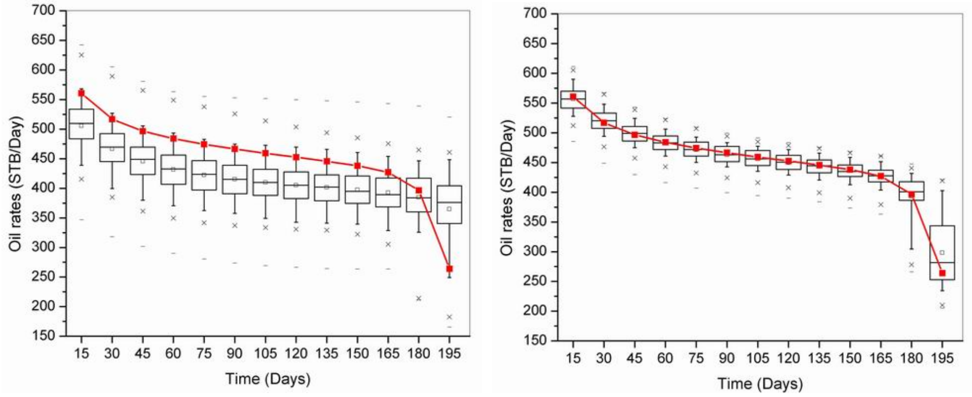


**Figure 79:** The oil rate of well 3 over the 195 days production history from the initial ensemble (left) and the final ensemble (right). The thick line shows the observed data.

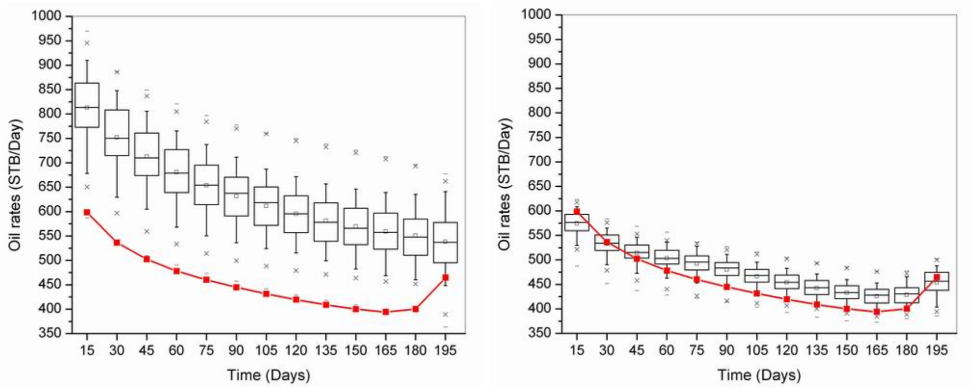


**Figure 80:** The oil rate of well 4 over the 195 days production history from the initial ensemble (left) and the final ensemble (right). The thick line shows the observed data.

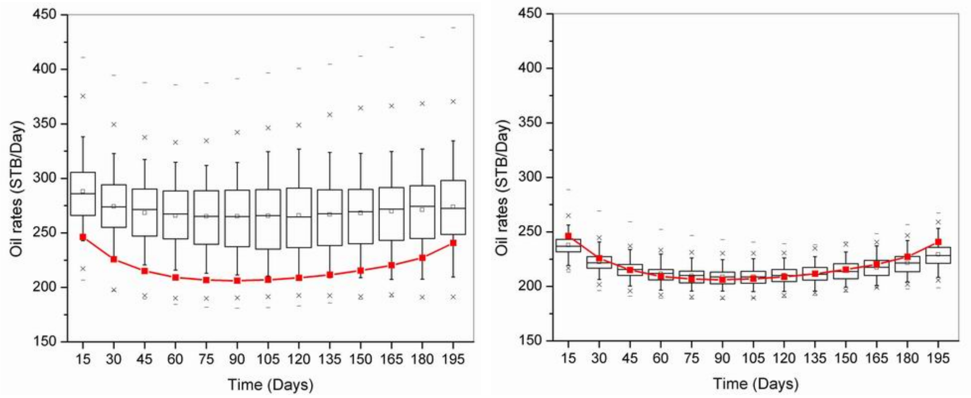


**Figure 81:** The oil rate of well 5 over the 195 days production history from the initial ensemble (left) and the final ensemble (right). The thick line shows the observed data.

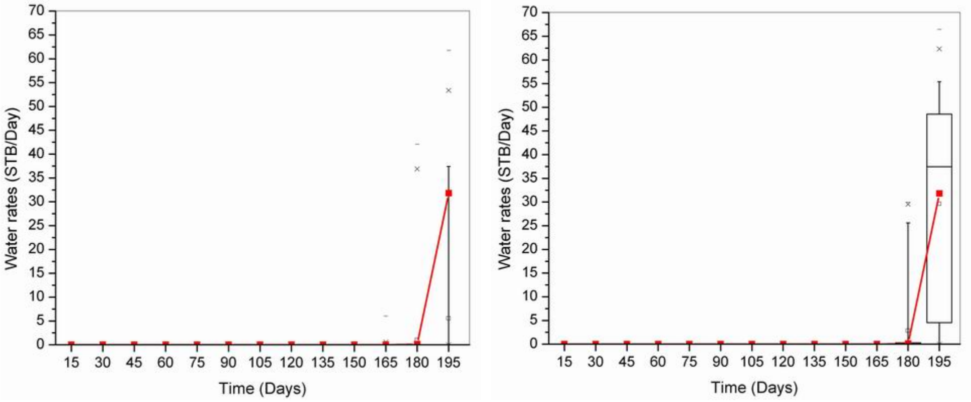


**Figure 82:** The water rate of well 3 over the 195 days production history from the final ensemble. The thick line shows the observed data. There is no breakthrough from any of the 100 initial ensemble member within 195 days.

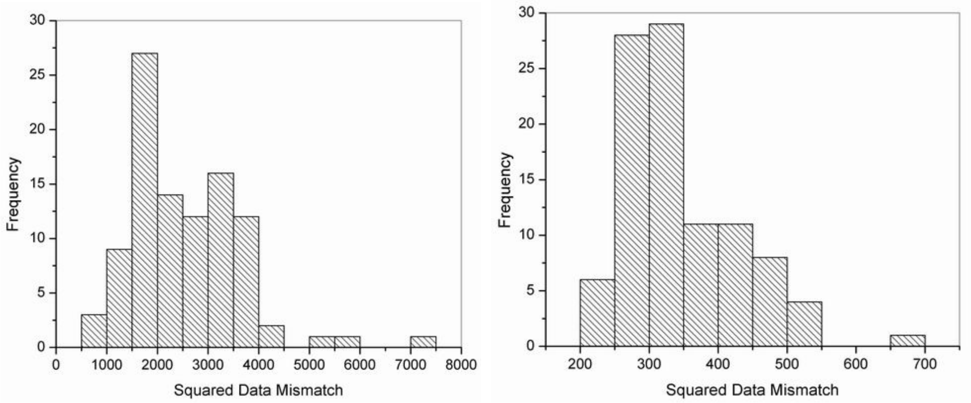


**Figure 83:** The histograms of squared data mismatch from the 100 initial ensemble states (left) and the final ensemble states (right).

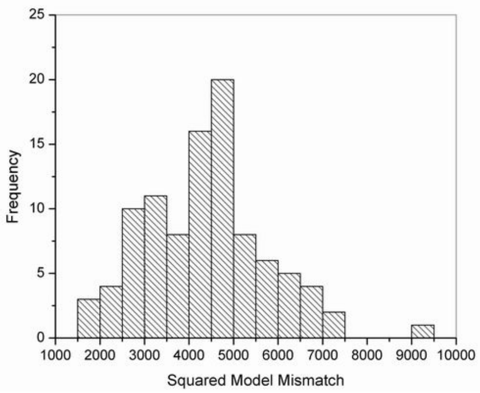


**Figure 84:** The histogram of squared model mismatch from the 100 initial ensemble states (left) and the final ensemble states (right).

in average has been reduced to 16.5% of the initial values. The best reduction is to 4.9% and the worst is to 49.2%.

# CHAPTER IX

# COMPARISON OF ENKF WITH GRADIENT METHODS ON HISTORY MATCHING PROBLEMS

The objective of this chapter is to compare the performance of the ensemble Kalman filter (EnKF) to the performance of a gradient-based minimization method for the problem of estimation of facies boundaries in history matching. Because of the approximate nature of the EnKF, the realizations from one ensemble tend to underestimate the uncertainty especially for problems that are highly nonlinear. In this study, the distributions of reservoir model realizations from 20 independent ensembles are compared with the distributions from 20 randomized maximum likelihood (RML) realizations for a 2D water-flood model with one injector and four producers.

## *9.1 Review of the Randomized Maximum Likelihood.*

A standard method for quantifying the uncertainty in reservoir simulation predictions is to generate multiple conditional reservoir model realizations, and predict future performance of each. The Randomized Maximum Likelihood (RML) method generates realizations conditional to nonlinear data from unconditional realizations in a Gaussian random field by a process of minimization. It has been shown to have good sampling properties for history matching problems with highly nonlinear relationship between the data and the model parameters (see Liu and Oliver, 2003b). If the prior covariance of the model parameters and the variance of the observed data are known, matched models can be generated as follows:

1. Generate an unconditional realization of the model parameters, $m_u \leftarrow N[m_{\text{pr}}, C_M]$.
2. Generate a realization of the data, $d_u \leftarrow N[d_{\text{obs}}, C_D]$.
3. Compute the set of model variables, $m$, that minimizes the function:

$$\begin{aligned} O(m) = & \frac{1}{2}(m - m_u)^T C_M^{-1}(m - m_u) \\ & + \frac{1}{2}(g(m) - d_u)^T C_D^{-1}(g(m) - d_u) \end{aligned}$$

Solving a minimization problem is required to generate each matched model. In this study, the limited memory BFGS method is used in computing the search direction, and the gradients of the objective function with respect to the permeability and the porosity fields, $\nabla_k O$ and $\nabla_\phi O$, were obtained from the adjoint method for general automatic history matching of reservoir property fields (see Li et al., 2003a).

## *9.2 Matching hard data and production data*

The test case is a reservoir model on a 50 × 50 × 1 grid. The dimensions of each gridblock is 30 ft×30 ft×20 ft. The covariance of the random Gaussian field $Y_1$ is Gaussian type with the principle direction 60° east of north. The range in the principle direction is one third of the field width, and twice the range in the perpendicular direction. The covariance of the random Gaussian field $Y_2$ is same with that of $Y_1$ except that the anisotropy angle is 45° east of north. Figs. 85(a) and 85(b) show a pair of unconditional Gaussian fields $(Y_1, Y_2)$ with the specified covariances. Three facies are present in the field, which are denoted as facies 1, facies 2, and facies 3. An unconditional facies map as shown in Fig. 85(d) is generated by truncating the two unconditional Gaussian fields $Y_1$ and $Y_2$ using a truncation scheme in Fig. 85(c). The facies in dark grey is facies 1, in light grey is facies 2, and in white is facies 3. The covariances of the two Gaussian fields and the truncation scheme are assumed to be known during history matching, and used to simulate all the facies realizations.

The true facies field is shown in Fig. 86. There is one injector near the center and four producers at the corners. The facies observations are listed in Table 9 with the well number and locations. The rock properties are constant within a facies type, but distinct among the facies. Table 10 presents the permeability and the porosity for each facies type. The injection rate is fixed at 4600 rb/day for well 1, and the production rates are fixed at 1300 reservoir barrel of total fluid per day for wells 2, 3, and 4, respectively. Well 5 is in a low productivity region, and the production rate is fixed at 600 rb/day. The initial reservoir pressure is 3800 psia, which is far above the bubble point of 500 psia. The field is produced for 80 days and the bottom-hole-pressure at all wells are recorded at day 2, and every 10 days beginning at day 10. There are 45 bottom-hole-pressure measurement and 5 facies observations from all wells. The measurement error for pressure data is assumed to be distributed normally with mean 0 and standard deviation of 3 psi. The facies observations are assumed to be exact. The same reservoir model was used to evaluate both the RML and the EnKF methods. Because of small differences in the way dates are entered for the two reservoir simulators CLASS and ECLIPSE, the "observed" data from the two simulators are only compared with the "true" data from the same simulator.

### 9.2.1 Gradient approach.

The traditional approaches to history matching assimilate all the data at the same time. Because the simulation of the process is so expensive, an efficient method of modifying the model parameters to match the observations is required. In this work, the adjoint method is used to compute the gradient of the objective function with respect to the model variables $Y_1$ and $Y_2$. The adjoint system is complicated in development, and dependent on the specific reservoir simulator for which it was developed. The adjoint system used was built for the Chevron Limited Application Simulation System (CLASS) (Li et al., 2003a; Zhang and Reynolds, 2002).

The history matching starts with the initial models having the correct facies types at well locations. The matching of facies data is done efficiently using the ensemble Kalman filter method. During the process of history matching to production data, the step-size of the modifications to the model parameters is restricted to ensure that the facies at well locations are maintained to be correct.

Figure 87 uses box plots to compare the distributions of the simulated BHP data from the 20 accepted RML realizations to the observed data. The "boxes" includes the range from $\mathrm{P}_{25}$ to $\mathrm{P}_{75}$, while the lines include the range from $\mathrm{P}_{10}$ to $\mathrm{P}_{90}$. The distributions of the simulated bottom-hole pressure from the 20 RML models are much wider than expected, based on the assumption that the magnitude of the noise is about 3 psi. As the facies type is an indicator variable, in order to compute the gradient of the objective function with respect to the facies, a transition zone was added at facies boundaries. This transition zone is only for the purpose of gradient approximation, and does not exist in the simulation process. Unlike history matching of permeability and porosity, the minimization of the objective function for history matching of the discontinuous facies often stops at a relatively high objective function value, because the gradient is only approximately correct. A typical minimization required approximately 11 iterations. One hundred initial models were generated, but only the 20 models with the lowest final objective function value were used.

### 9.2.2 EnKF for history matching.

The states variables in the Kalman filter consist of two Gaussian fields for facies description, pressure and saturation fields, simulated bottom-hole pressure data from each well, and simulated facies at each well location: $\{Y_1, Y_2, P, S, P_{\mathrm{wf,sim}}, F_{\mathrm{sim}}\}$. 40 state vectors are included in an ensemble. The initial ensemble of state vectors are conditional to the facies observations using EnKF. Unless care is taken, updating of the Gaussian fields from matching production data may change the facies type at well locations. Once the facies type at a well location is wrong, the Kalman correction to the Gaussian fields can become large, and may cause over-shoot of the Gaussian variables. An EnKF iteration for facies observations is made after each model update to ensure the updated rock properties at well locations are always correct.

The first row of Fig. 88 shows 4 out of the 40 initial facies maps, and the second row shows the corresponding facies maps simulated from the final ensemble members after assimilating all the production data. The initial facies maps all have unique local structures, and the final facies maps have developed common features among the ensemble members. Some of the common features do not exist in the true facies map.

Box plots are used to represent the distributions of the simulated production data from all the 40 ensemble members over the 80 days of production. The reservoir simulator ECLIPSE is used in the EnKF approach for adaption to supercomputers. The simulated bottom-hole pressure and the observed data are plotted together in Fig. 89. The box plots in the first column show the bottom-hole pressure from the initial ensemble conditional only to facies observations. The box plots in the second column show bottom-hole pressure after assimilation to pressure data. The observed injection rate is plotted in the thick line. In the second column of Fig. 89, the distributions of the bottom-hole pressure from the 40 final reservoir models are much narrower than the initial distributions and are centered at the observed data.

A single ensemble of 800 state members was also generated and assimilated to the observed data. Column 3 of Fig. 89 shows the distributions of the simulated BHP from all the 800 matched final models. The matching quality is clearly at least as good as that using 40 ensemble members, as the width of the boxes is very narrow.

## *9.3 Discussions*

This test problem is somewhat unusual in that the objective function for the problem, which includes facies data mismatch, is not differentiable, so it was necessary to introduce an approximate objective function with a transition between facies. The quasi-Newton minimization method should have converged fairly quickly near the minimum, but did not do so in this case. The final values of the objective function were also larger than expected. The 20 best realizations from 100 minimizations are elected. A typical history matching minimization required 11 iterations, with each iteration requiring CPU time equivalent to approximately 5 forward simulation runs. The effort required to generate the 20 independent realizations using LBFGS was approximately 100 starting models × 11 iterations per model × 5 simulations per iterations giving 5500 simulations, or 275 simulations per history matched model.

The ensemble Kalman filter method with 40 members in an ensemble required 40 simulation runs to generate 40 history matched models, but it is clear from Fig. 88 that the variance within an ensemble is too small, so the 40 models are not independent realizations. Realizations from the EnKF method, as judged by the median value of the objective function (6220), are much better than those from the LBFGS method which gave a larger median value for the objective function (21,300).

The tendency for the variance within an ensemble to reduce to a level that is smaller than required by the data has been previously noted for the EnKF method (Houtekamer and Mitchell, 1998). Although the quality of the matching to the observations is satisfactory for all the 40 ensemble members, the significant reduction of the variations among the ensemble of matched facies models lead us to investigate the capability of EnKF in uncertainty quantification.

In practice, the number of state members in an ensemble is empirically chosen between 40 and 100. When the number is too small, the variation carried from the initial ensemble is not sufficient in approximation of the state covariances for each model updating. Consequently, the matched models may become highly correlated, and clustered at a small subspace of the real model probability distribution. In that case, the 40 final models from the same ensemble are equivalent to one well matched RML realization for uncertainty quantification. On the other hand, each ensemble member requires one reservoir simulation run plus some computational overhead for data assimilation. The EnKF approach becomes more computationally expensive the more ensemble members are included.

Twenty ensembles were also generated with 40 state members in each, such that the total number of states was 800. Each of the matched models in the 20 small ensembles and the large ensemble of 800 members was simulated in ECLIPSE for water cut prediction from each of the four producers at day 140. The distributions of the predicted water cut from each ensemble are plotted in box-plots as shown in Fig. 90. The boxes numbered from 1 to 20 are distributions from the ensembles with 40 members. The last box in each plot at number 21 is the distribution from the 800 member ensemble. The thick lines are predicted water cut from the true model, where the 80 days production data were generated. None of the distributions of the 20 small ensembles covers the predicted water cut for all four producers. Some of the ensembles predicted very small uncertainties, such as number 3 and 18. And none of the small ensembles yields distributions that are large enough to represent the distributions from all other small ensembles.

The water cut predictions from the large ensemble of 800 state members cover the true water cut from each well. They also largely cover the distributions from the small ensembles. However, except for Well 4, the truth is far from the center of the distributions and the predictions are heavily biased. Well 5 does not have water breakthrough in the true model, and only 2 out of the 800 predictions are correct.

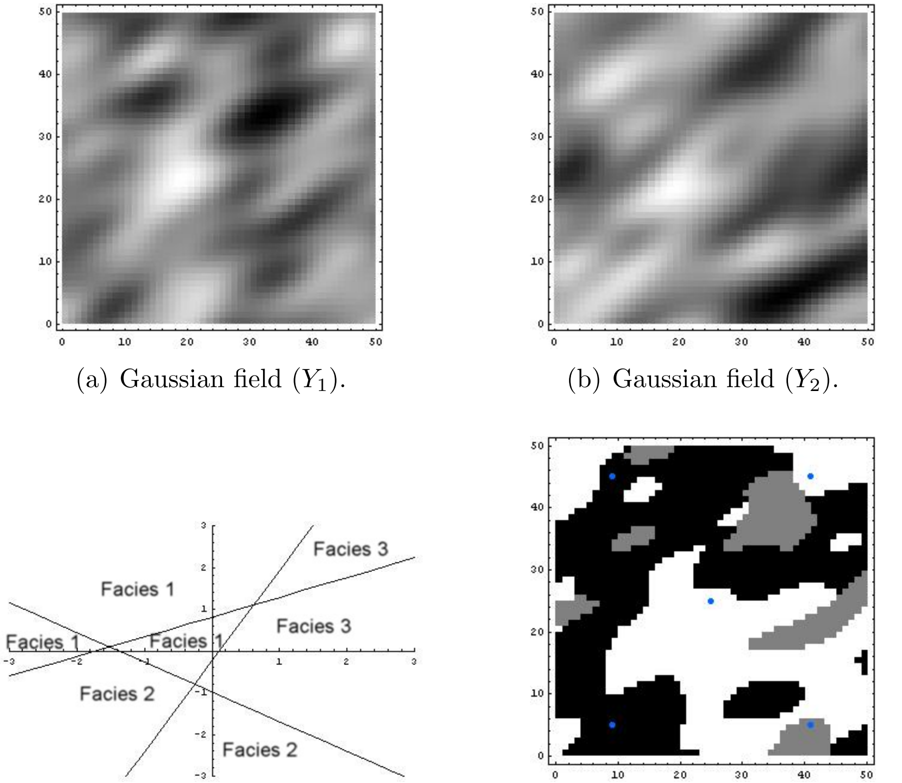
Not all of the history matched realizations are of equivalent quality — with quality in this case measured by the magnitude of the data mismatches after data assimilation. The higher quality realizations have smaller sums of squared mismatch between predicted and observed data. The limited memory BFGS method to history match pressure data for 100 reservoir models is used, but only retained the 20 with the lowest value of the objective function after minimization. The median value of the objective function from the best 20 was 21,300. The median value of the final objective function from the ensemble Kalman filter method with 40 members in an ensemble was 6220. The median value from EnKF using a single ensemble of 800 was 4780. It is not surprising that better matches to the data are obtained from the larger ensemble, simply because there are more degrees of freedom. It is somewhat surprising that the minimization method that used the adjoint to compute the gradient achieved the poorest matches to data. This has not been the typical experience with other history matching problems, and may be attributed in this case to the necessity of introducing a transition region between facies so that the gradient is computed from an approximation to the actual objective function.

| Well | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| x | 25 | 9 | 9 | 41 | 41 |
| y | 25 | 5 | 45 | 45 | 5 |
| facies | 3 | 1 | 1 | 3 | 2 |

**Table 9:** Facies observations from each of the five wells.

| index | Facies 1 | Facies 2 | Facies 3 |
|---|---|---|---|
| Permeability (md) | 174.0 | 80.0 | 372.0 |
| Porosity | 0.18 | 0.146 | 0.25 |

**Table 10:** Properties of each the litho-facies in the synthetic problem.



(a) Gaussian field ($Y_1$).

(b) Gaussian field ($Y_2$).

(c) Threshold map.

(d) Calculated facies field map.

**Figure 85:** Simulation of lithofacies distribution in the field by truncation of random Gaussian fields $Y_1$ and $Y_2$ using intersecting line thresholds.

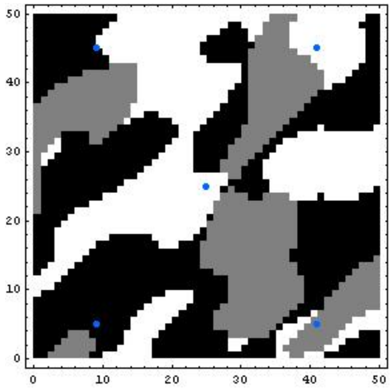
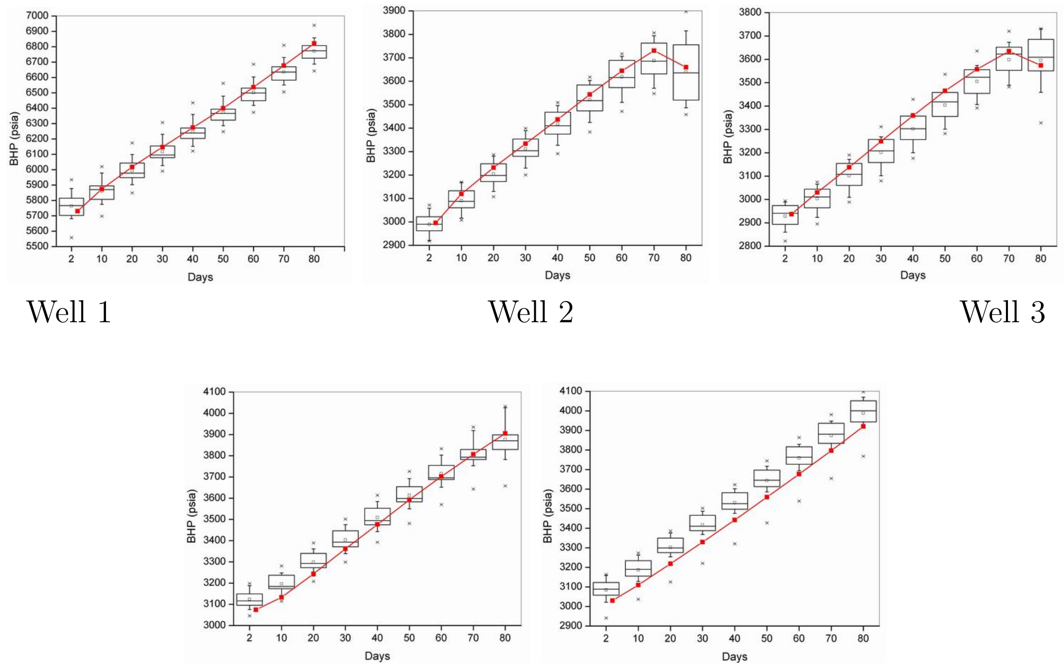


**Figure 86:** The true facies field of the synthetic model contains three facies types. The five dots are well locations.



Well 1 Well 2 Well 3

Well 4 Well 5

**Figure 87:** The box plots present the distributions of the simulated bottom hole pressure from 20 history matched models. The thick lines are the observed bottom pressure data from each well.

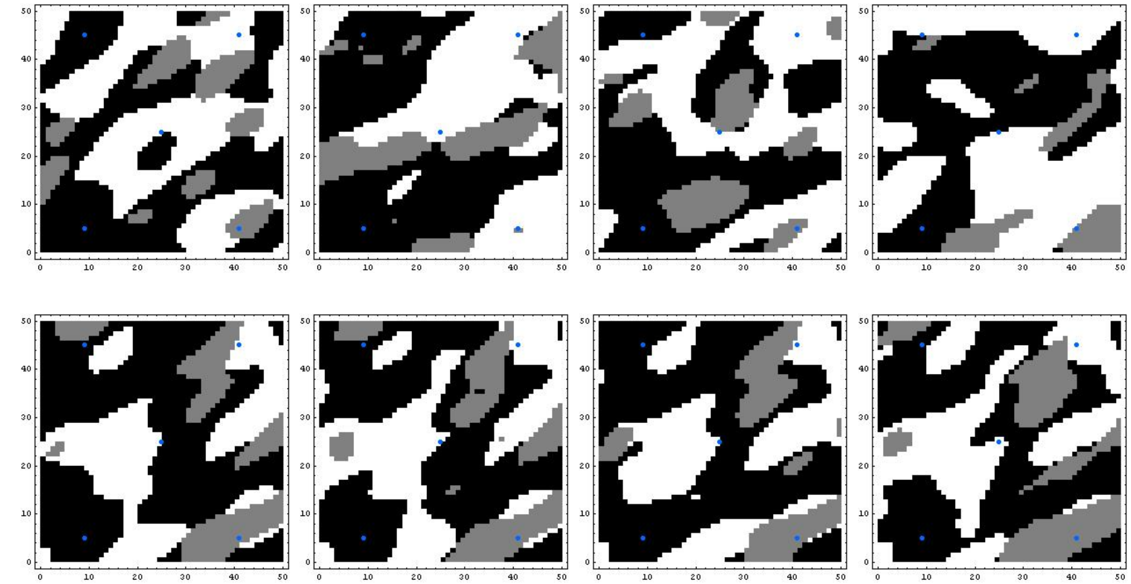


**Figure 88:** The first four initial facies map in an ensemble of 40 members (top row), and the corresponding final facies map after history matched to production data and hard data (bottom row). The dots in each facies map are well locations.

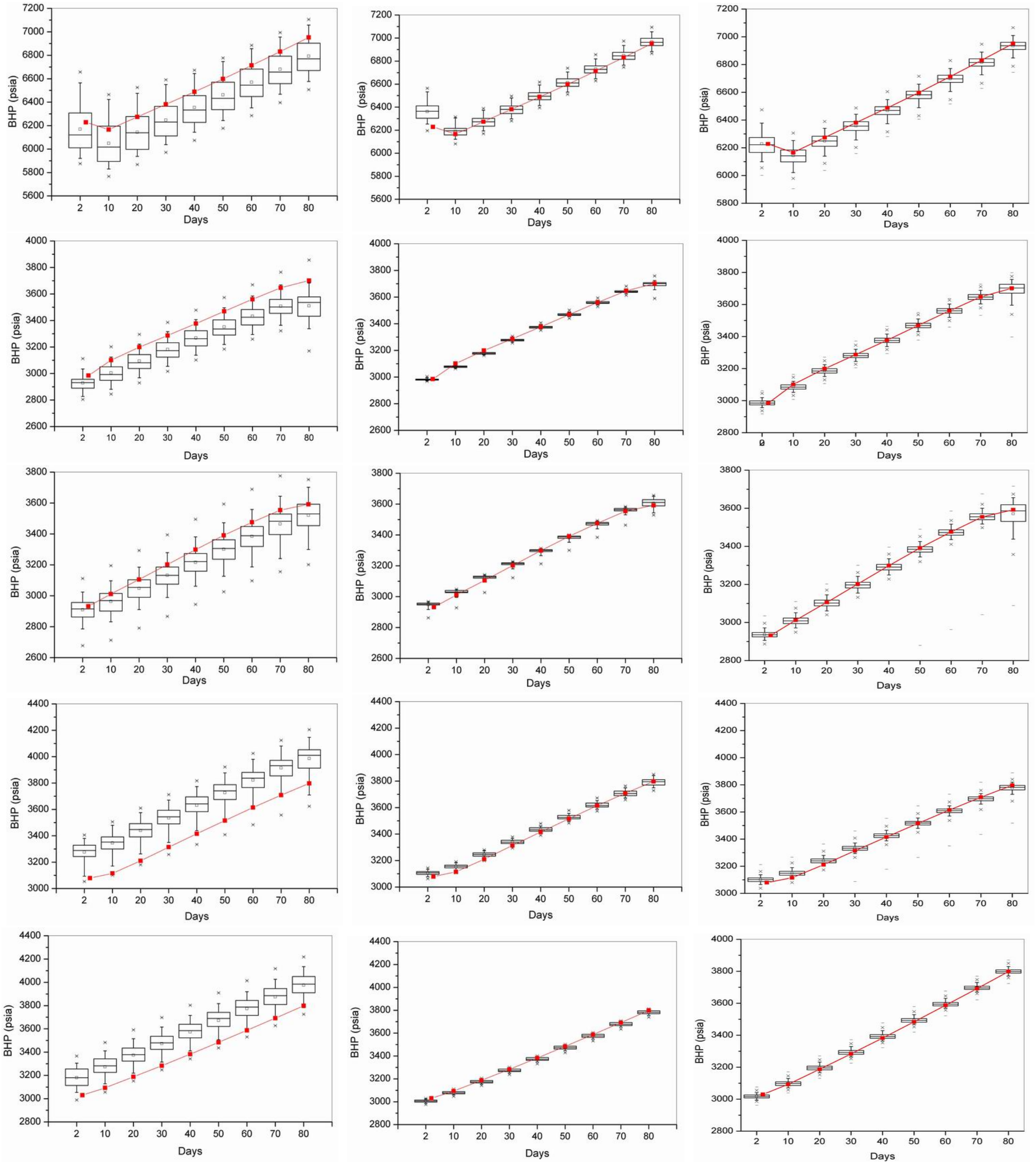
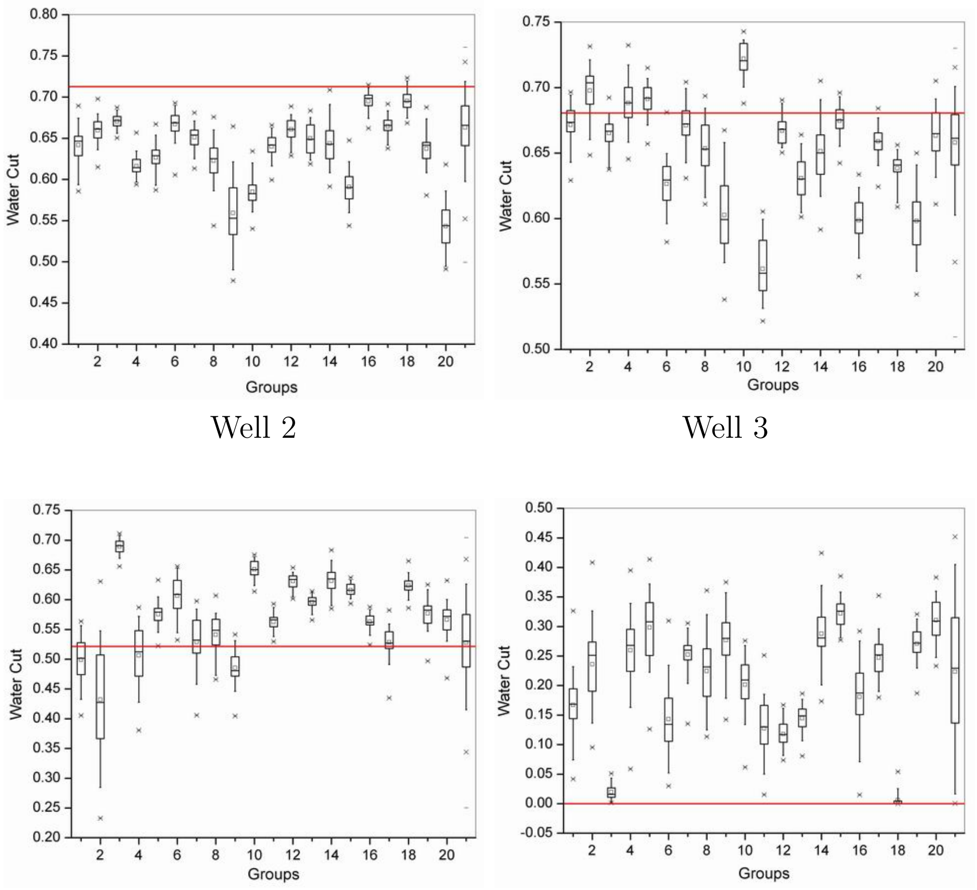


**Figure 89:** Distributions of the simulated bottom hole pressure from the initial ensemble of 40 (first row), and from the history matched ensemble (second row). The observed data from each well is plotted in a thick line for comparison. The box plots shown are for well 1 through 4.



Well 2

Well 3

Well 4

Well 5

**Figure 90:** Distributions of water cut prediction on day 140 from 20 ensemble groups. The straight lines are water cut predicted from the true model.

# CHAPTER X

# CONCLUSIONS

This study is devoted to history matching of geologic facies distributions to production data and facies observations. The history matching problem was solved in two steps. In the first step, the geostatistical parameters, such as the parameters deciding the threshold lines, the ranges and the anisotropy angles of the Gaussian fields, were estimated from an empirical training image based on the formation sedimentation environment. A gradient based method was used for minimization of the facies mismatch, and the gradient is approximated by adding a transition zone to facies boundaries. As certain combinations of the geostatistical parameters are more likely to reproduce some geological structures than others, the randomized maximum likelihood method was used to generate a large number of realizations of geostatistical parameters for the probability distributions. In the second step, the geostatistical parameters were sampled from the a posteriori probability distributions and fixed. The Gaussian random fields were optimized to match the facies and production data observations in two distinct approaches: the gradient method and the EnKF method.

The truncated pluri-Gaussian method for simulation of geologic facies boundaries proved to be very useful for history matching facies locations and ensuring that the shapes honored production data. On the facies map, the boundaries between different facies are the most sensitive region to changes in model parameters. Using intersecting lines as thresholds, the facies map can be easily adjusted if the optimal direction for change is known. The adjoint method was useful in computing the gradient of the facies mismatch. In a case study of estimating the Gaussian random fields and the truncation line parameters to match the training image, approximately 73,000 variables were adjusted to match over 16,000 facies observations. 95.6% of the final facies map matched the training image. The Gaussian covariance parameters, such as the ranges and the anisotropy angles, were also estimated from the training image using the adjoint approach. In the study, the number of model parameters was 83,238 ($\{Z_1, Z_2, a_{11}, a_{12}, \theta_{c1}, a_{21}, a_{22}, \theta_{c2}\}$), and the number of data was 16,384 (128 × 128). After 57 iterations, the facies mismatch reduced to 738 (4.5% of the total gridblocks) from the initial number of 11,500 (70.2%).

The traditional approach for history matching to production data and facies observations modifies the model parameters to assimilate all data at once. The adjoint system is developed to compute the gradient of the objective function with respect to the Gaussian random fields. A 2-D, 2-phase flow case study was made using the simulator CLASS and the adjoint system developed for CLASS. The cost of reducing the objective function by 98% was 13 LBFGS iterations, or approximately 65 simulation runs.

This study also investigated the applicability of the ensemble Kalman filter to the problem of history matching of facies locations. Facies are indicator variables, and hence not differentiable or Gaussian. By a proper choice of state variables, however, for instance using the Gaussian fields instead of the facies and using an appropriate definition of facies mismatch instead of facies type, history matching of facies locations is possible with EnKF. The results from both case studies are satisfactory, which establishes the usefulness of EnKF in solving reservoir history matching problems with geologic facies.

The relationships between the data and the state variables are nonlinear in both the history matching and the facies matching examples. When matching facies observations, it was necessary to enforce the data constraints iteratively. Fortunately, there was no need to compute the gradient or the state covariance explicitly, so the cost of the iterations was negligible.

In the history matching problem, the 2-D facies maps at the end of data assimilation are much different from the initial facies maps, while consistent with the rates assimilated in early time. This fact is encouraging for the use of the ensemble Kalman filter method in reservoir history matching. When the early production data need to be reinforced to the ensemble states, it may be necessary to iterate on the data assimilation with the final states which would require rerunning the reservoir simulations from early time.

The squared data mismatch for history matching was on average reduced to 16% of its initial value after assimilating production data. The uncertainty of the model variables conditional to data appears to be underestimated, as the subspace spanned by the final states does not include the true facies map.

The EnKF algorithm was highly efficient from the standpoint of computational cost and software development. Once the algorithm is fully understood, it only takes a few days to write the EnKF code for data assimilation. The amount of computation time for generating 100 reservoir facies models conditional to production data is the time required for 100 reservoir simulations plus some overhead for the update of Kalman states after each data assimilation. Although the uncertainty from a single ensemble appears to have been underestimated, comparison of uncertainty estimation between the EnKF and the gradient based history matching algorithms should probably be based on the distributions from multiple ensembles.

The ensemble Kalman filter method outperformed the gradient-based minimization method in both computational efficiency and applicability for the problem of estimation of facies boundaries in history matching. It took approximately 5500 equivalent simulation runs to obtain the 20 accepted RML model realizations, and only 800 simulation runs for the 20 independent EnKF model realizations. The quality of the matching from the EnKF model realizations are better than those from the gradient approach. The number of 40 state vectors in an ensemble is insufficient for uncertainty quantification, as obvious correlations have been developed among the state vectors during data assimilation.

Better quality matches were obtained from the larger ensemble, where 800 simulation runs were made, than the 20 independent ensembles with 40 state vectors in each. Based on the experience in this problem, it appears that it might be more efficient to use one large ensemble to access uncertainty, than to use many small ensembles.

# Bibliography

Anderson, J. L., An ensemble adjustment Kalman filter for data assimilation, *Monthly Weather Review*, **129**(12), 2884–2903, 2001.

Anterion, F., B. Karcher, and R. Eymard, Use of parameter gradients for reservoir history matching, SPE-18433, in *10th SPE Reservoir Simulation Symp.*, pp. 339–354, 1989.

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

Bissell, R., Y. Sharma, and J. E. Killough, History matching using the method of gradients: Two case studies, SPE-28590, *SPE 69th Annual Technical Conference and Exhibition*, pp. 275–289, 1994.

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

Carter, R. D., L. F. Kemp, A. C. Pierce, and D. L. Williams, Performance matching with constraints, *Soc. Petrol. Eng. J.*, **14**(4), 187–196, 1974.

Chavent, G. M., M. Dupuy, and P. Lemonnier, History matching by use of optimal control theory, *Soc. Petrol. Eng. J.*, **15**(1), 74–86, 1975.

Chen, W. H., G. R. Gavalas, J. H. Seinfeld, and M. L. Wasserman, A new algorithm for automatic history matching, *Soc. Petrol. Eng. J.*, pp. 593–608, 1974.

Chu, L., A. C. Reynolds, and D. S. Oliver, Computation of sensitivity coefficients for conditioning the permeability field to well-test data, *In Situ*, **19**(2), 179–223, 1995.

Corliss, G., C. Faure, A. Griewank, L. Hascoet, and U. Naumann (eds.), *Automatic Differentiation of Algorithms, from Simulation to Optimization*, LNCSE, Springer, 2001.

Corser, G. P., J. E. Harmse, B. A. Corser, M. W. Weiss, and G. L. Whitflow, Field test results for a real-time intelligent drilling monitor, SPE-59227, in *Proceedings of the 2000 IADC/SPE Drilling Conference*, 2000.

Deschamps, T., T. Grussaute, D. Mayers, and R. Bissell, The results of testing six different gradient optimisers on two history matching problems, in *Proceedings of the 6th European Conference on the Mathematics of Oil Recovery*, pp. B–24, 1998.

Eisenmann, P., M.-T. Gounot, B. Juchereau, and S. J. Whittaker, Improved Rxo measurements through semi-active focusing, SPE-28437, in *Proceedings of the SPE 69th Annual Technical Conference and Exhibition*, 1994.

Evensen, G., Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics, *Journal of Geophysical Research*, **99**(C5), 10,143–10,162, 1994.

Evensen, G., The ensemble Kalman filter: Theoretical formulation and practical implementation, *Ocean Dynamics*, 2003.

Fletcher, R., *Practical Methods of Optimization*, second edn., John Wiley & Sons, New York, 1987.

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

Galli, A., H. Beucher, G. Le Loc'h, B. Doligez, and H. Group, The pros and cons of the truncated Gaussian method, in *Geostatistical Simulations*, pp. 217–233, Kluwer Academic, Dordrecht, 1994.

Gosselin, O., A. C. S. van den Berg, and S. D. Chowdhury, A gradient-based approach for history-matching of both production and 4D seismic data, Tech. rep., preprint, 2001.

Gu, Y. and D. S. Oliver, History matching of the PUNQ-S3 reservoir model using the ensemble Kalman filter, SPE-89942, in *Proceedings of the SPE Annual Technical Conference and Exhibition*, 2004.

He, N., D. S. Oliver, and A. C. Reynolds, Conditioning stochastic reservoir models to well-test data, SPE-38655, in *1997 SPE Annual Technical Conference and Exhibition*, 1997.

Houtekamer, P. L. and H. L. Mitchell, Data assimilation using an ensemble Kalman filter technique, *Monthly Weather Review*, **126**(3), 796–811, 1998.

Houtekamer, P. L. and H. L. Mitchell, A sequential ensemble Kalman filter for atmospheric data assimilation, *Monthly Weather Review*, **129**(1), 123–137, 2001.

Kolda, T. G., D. P. O'Leary, and L. Nazareth, BFGS with update skipping and varying memory, *SIAM J. on Optimization*, **8**(4), 1060–1083, 1998.

Landa, J. L., M. M. Kamal, C. D. Jenkins, and R. N. Horne, Reservoir characterization constrained to well test data: A field example, *SPE Reservoir Evaluation and Engineering*, **3**(4), 325–334, 2000.

Lantuéjoul, C., *Geostatistical Simulation: Models and Algorithms*, Springer, Berlin, 2002.

Le Loc'h, G., H. Beucher, A. Galli, B. Doligez, and H. Group, Improvement in the truncated Gaussian method: Combining several Gaussian Functions, in *Proceedings of ECMOR IV, Fourth European Conference on the Mathematics of Oil Recovery*, 1994.

Le Loc'h, G. and A. Galli, Truncated plurigaussian method: Theoretical and practical points of view, in *Geostatistics Wollongong '96*, (edited by E. Y. Baafi and N. A. Schofield), vol. 1, pp. 211–222, Kluwer Academic, 1997.

Li, R., *Conditioning Geostatistical Models to Three-Dimensional Three-Phase Flow Production Data by Automatic History Matching*, Ph.D. thesis, University of Tulsa, Tulsa, Oklahoma, 2001.

Li, R., A. C. Reynolds, and D. S. Oliver, History matching of three-phase flow production data, *SPE Journal*, **8**(4), 328–340, 2003a.

Li, R., A. C. Reynolds, and D. S. Oliver, Sensitivity coefficients for three-phase flow history matching, *Journal of Canadian Petroleum Technology*, **42**(4), 70–77, 2003b.

Liu, N. and D. S. Oliver, Automatic history matching of geologic facies, SPE 84594, *Proceedings of the 2003 SPE Annual Technical Conference and Exhibition*, pp. 1–15, 2003a.

Liu, N. and D. S. Oliver, Evaluation of Monte Carlo methods for assessing uncertainty, SPE 84936, *SPE Journal*, **8**(2), 188–195, 2003b.

Liu, N. and D. S. Oliver, Automatic history matching of geologic facies, SPE 84594, *SPE Journal*, **9(4)**, 1–15, 2004.

Nævdal, G., L. M. Johnsen, S. I. Aanonsen, and E. H. Vefring, Reservoir monitoring and continuous model updating using ensemble Kalman filter, SPE-84372, *Proceedings of SPE Annual Technical Conference and Exhibition*, 2003.

Nocedal, J., Updating quasi-Newton matrices with limited storage, *Math. Comp.*, **35**, 773–782, 1980.

Oliver, D. S., Incorporation of transient pressure data into reservoir characterization, *In Situ*, **18**(3), 243–275, 1994.

Oliver, D. S., Moving averages for Gaussian simulation in two and three dimensions, *Mathematical Geology*, **27**(8), 939–960, 1995.

Omre, H., B. K. Hegstad, and H. Tjelmeland, Alternative history matching approaches, Tech. rep., Department of Mathematical Sciences, Norwegian University of Science & Technology, Trondheim, Norway, 1996.

Rahon, D., G. Blanc, and D. Guérillot, Gradients method constrained by geological bodies for history matching, in *Proceedings of the 5th European Conference on the Mathematics of Oil Recovery*, pp. 283–293, 1996.

Rahon, D., P. F. Edoa, and M. Masmoudi, Inversion of geological shapes in reservoir engineering using well-tests and history matching of production data, *1997 SPE Annual Technical Conference and Exhibition, SPE-38656*, pp. 141–150, 1997.

Reichle, R. H., D. B. McLaughlin, and D. Entekhabi, Hydrologic data assimilation with the ensemble Kalman filter, *Monthly Weather Review*, **130**(1), 103–114, 2002.

Shirriff, K., Generating fractals from Voronoi diagrams, *Computers & Graphics*, **17**(2), 165–167, 1993.

Tarantola, A., *Inverse Problem Theory: Methods for Data Fitting and Model Parameter Estimation*, Elsevier, Amsterdam, The Netherlands, 1987.

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

Woodbury, A. D., Bayesian updating revisited, *Mathematical Geology*, **21**, 285–308, 1989.

Zhang, F. and A. C. Reynolds, Optimization algorithms for automatic history matching of production data, *Proceedings of 8th European Conference on the Mathematics of Oil Recovery*, 2002.

Zhang, F., A. C. Reynolds, and D. S. Oliver, Evaluation of the reduction in uncertainty obtained by conditioning a 3D stochastic channel to multiwell pressure data, *Mathematical Geology*, **34**(6), 715–742, 2002.

Zimmerman, D. A., G. de Marsily, C. A. Gotway, M. G. Marietta, C. L. Axness, R. Beauheim, R. Bras, J. Carrera, G. Dagan, P. B. Davies, D. P. Gallegos, A. Galli, J. Gomez-Hernandez, S. M. Gorelick, P. Grindrod, A. L. Gutjahr, P. K. Kitanidis, A. M. Lavenue, D. McLaughlin, S. P. Neuman, B. S. Ramarao, C. Ravenne, and Y. Rubin, A comparison of seven geostatistically based inverse approaches to estimate transmissivities for modeling advective transport by groundwater flow, *WRR*, **34**(6), 1373–1413, 1998.