Report Title:	**Mapping of Reservoir Properties and Facies Through Integration of Static and Dynamic Data**
Report Type:	Final Technical Report
Reporting Period Start Date:	October 1, 2000
Reporting Period End Date:	September 30, 2004
Principal Authors:	Albert C. Reynolds, Dean S. Oliver, Yannong Dong, Ning Liu, Guohua Gao, Fengjun Zhang & Ruijian Li
Date Report Issued:	December 2004
DOE Award Number:	DE-FC26-00BC15309

Petroleum Engineering Department
The University of Tulsa
600 South College Avenue
Tulsa, Oklahoma 74104

Contents

List of Figures

ABSTRACT

Knowledge of the distribution of permeability and porosity in a reservoir is necessary for the prediction of future oil production, estimation of the location of bypassed oil, and optimization of reservoir management. The volume of data that can potentially provide information on reservoir architecture and fluid distributions has increased enormously in the past decade. The techniques developed in this research will make it easier to use all the available data in an integrated fashion. While it is relatively easy to generate plausible reservoir models that honor static data such as core, log, and seismic data, it is far more difficult to generate plausible reservoir models that honor dynamic data such as transient pressures, saturations, and flow rates. As a result, the uncertainty in reservoir properties is higher than it could be and reservoir management can not be optimized. In this project, we have developed computationally efficient automatic history matching techniques for generating geologically plausible reservoir models which honor both static and dynamic data. Specifically, we have developed methods for adjusting porosity and permeability fields to match both production and time-lapse seismic data and have also developed a procedure to adjust the locations of boundaries between facies to match production data. In all cases, the history matched rock property fields are consistent with a prior model based on static data and geologic information. Our work also indicates that it is possible to adjust relative permeability curves when history matching production data.

EXECUTIVE SUMMARY

In the performance of research for this project, we have developed algorithms and software for making optimal use of oilfield logs, production data, and time-lapse seismic data to better map the flow properties of the reservoir. The method efficiently adjusts gridblock values of permeability and porosity in a reservoir simulation model until the predictions from the model match the observations as closely as possible. For reservoirs whose properties are determined primarily by the geolgic facies with unknown boundary locations, we have developed methods for estimating the boundary locations. Although time-lapse seismic data is not widely available, the data are shown to be effective for mapping flow properties in the regions between wells, even when the signal to noise ratio is fairly low.

In our work, data integration is done within the framework of Bayesian statistics which provides a theoretical basis for the automatic history matching of multiphase flow production data to construct estimates or realizations of reservoir variables that are consistent with time-lapse seismic data, production data and static data obtained from logs, cores and geologic and geophysical interpretation. Because we will always use a reservoir simulator to calculate production data for a given reservoir description, it is convenient to include reservoir simulator gridblock log-permeabilities and porosities in the set of reservoir variables to be estimated by history-matching, but reservoir variables may also include parameters that characterize relative permeability curves and the locations of boundaries between facies.

In the Bayesian setting, the automatic history matching of production and/or time-lapse seismic data requires the minimization of an objective function which consists of the sum of a regularization term and production data mismatch terms. The regularization term encapsulates prior geologic information on the distribution of rock properties and facies distributions.

Early in the life of this project, we implemented a Gauss-Newton method and a modified Levenberg-Marquardt method for automatic history matching. Our automatic history matching procedure is founded on the development and implementation of a discrete adjoint method for calculating the sensitivity of three-phase flow production data to reservoir

variables (model parameters.) To the best of our knowledge, this is the first time that an adjoint method has been coupled with a simulator based on a fully-implicit, three-dimensional, finite-difference simulator to calculate the sensitivity of pressure, producing gas-oil ratio and producing water-oil ratio production data to reservoir variables under three-phase flow conditions. However, it is feasible to calculate the sensitivity coefficients with the adjoint method only when the total number of data to be history matched is small. As our objective was to develop an automatic history matching procedure that is applicable to large scale history matching problems, we were forced to develop and implement an optimization algorithm that is far more efficient than the Gauss-Newton method or the traditional Levenberg-Marquardt modification of the Gauss-Newton method, which are efficient only for cases where the number of data or number of parameters are small. Here, large scale history matching problems refer to cases where the number of parameters adjusted range from several hundred to tens of thousands and the number of production and seismic data is also large (several dozen to several thousands). For large scale problems, we implemented a version of a scaled limited memory Broyden-Fletcher-Goldfarf-Shanno (LBFGS) algorithm to obtain a history match by minimizing the appropriate objective function. In some cases, we found that a straightforward implementation of the LBFGS algorithm results in a history-matched reservoir model that is inconsistent with geological information. To avoid such problems, we introduced scaling, damping and constraints into the LBFGS algorithm. With these procedures, the resulting LBFGS has proved to be a robust and computationally efficient algorithm for large scale history matching problems. The reliability of the LBFG algorithm is illustrated with specific history matching examples.

We have also introduced uncertainty into the relative permeability curves under the assumption that relative permeability curves can be described by power law models. Under this assumption, we have demonstrated that it may be possible to adjust prior relative permeability curves obtained from laboratory data when history matching production data.

We have also developed techniques to integrate time-lapse seismic data into the reservoir description. Because time-lapse seismic data covers the areal extent of the reservoir and is related to vertically averaged pressure and fluid saturations, it is clear that matching these seismic data, simultaneously with production data, will reduce the uncertainty in our estimates of the rock property fields. We have developed an adjoint method so that the LBFGS algorithm can also be used to adjust permeability and porosity fields by history matching time-lapse seismic data; we provide examples to illustrate the utility of this procedure.

Facies (defined here as regions of relatively uniform petrophysical properties) are common features of all reservoirs. Typically, the variation of any rock property between facies is

much greater than the variation of the property within a specific facies. Because of this, it is important to estimate facies boundaries in the history matching process. In this work, we have developed a truncated plurigaussian model for the generation of facies maps. Unlike previous implementations of this method, we have used intersecting lines as thresholds. With this approach, we show that it is still possible to (1) generate a rich variety of textures and shapes, (2) estimate the locations of the threshold lines and (3) generate approximations of the sensitivity coefficients needed to condition reservoir models to facies distributions by history matching production data.

Chapter 1

INTRODUCTION

The goal of this project was to develop computationally efficient automatic history matching techniques for generating geologically plausible models that honor both static and dynamic data. The emphasis was on developing techniques that are applicable for large scale reservoir simulation models where the number of reservoir variables is large and the number of dynamic data to be matched can also be large. For such large scale problems, the feasibility of automatic history matching requires the solution of two related problems, a robust and efficient optimization algorithm and an efficient method for computing the sensitivity of data to reservoir variables that are required by the optimization algorithm. Here, reservoir variables include reservoir simulation gridblock porosities and log-permeabilities and may also include parameters defining power law relative permeability curves as well as parameters defining a model for the distribution of geological facies. A major accomplishment of this project was the development and implementation of a limited-memory Broyden-Fletcher-Goldfarb-Shanno (LBFGS) for matching dynamic data in Bayesian framework (Zhang and Reynolds, 2002a; Gao and Reynolds, 2004). At each iteration, this algorithm requires no sensitivities other than the calculation of the gradient of the objective function which we wish to minimize to obtain a history match of production data. This objective function includes both a production data mismatch term and a model mismatch term where the model mismatch term is included to constrain the history-matched model to static data. The gradient of the objective function with respect to the reservoir variables can be efficiently calculated using the implementation of the adjoint method developed by Li et al. (2003a) for coupling with a three-phase flow reservoir simulator based on a fully-implicit finite difference method. While the LBFGS algorithm uses the adjoint method to compute the necessary gradients, it may occasionally require special implementations such as data damping to control overshooting. Rescaling of model parameters and a judicious choice of a line search

algorithm may further improve its computational efficiency.

Because of the limited gradient information required by LBFGS algorithm and the reliability of the adjoint, it became clear during this research project that, at least conceptually, one could history-match time-lapse seismic data as well as production data. Thus, even though the original goal of this research project did not include the incorporation of seismic data, the adjoint method was extended to include the calculation of the gradient of time-lapse seismic data with respect to permeability and porosity fields. With this modification, we were able to apply the LBFGS algorithm to simultaneously history match both production (pressures, GOR and WOR) and time-lapse seismic data (Dong and Oliver, 2003).

Time-lapse seismic is the process of repeating 3D seismic surveys over a producing reservoir to monitor changes in saturation and pressure. The potential impact on reservoir engineering and reservoir management is large because time-lapse seismic may allow direct imaging of rock properties that are closely related to vertically averaged fluid saturations and pressure. This is much different from the current limitation of measurements of these quantities at well locations. In general, seismic images are sensitive to the spatial variation of two distinct types of reservoir properties (Arenas et al., 2001):

- Non-time-varying *static* geologic properties such as lithology, porosity, cementation, and shale content.
- Time-varying *dynamic* fluid-flow properties such as fluid saturation and pore pressure.

If data were available from only one 3D seismic survey, it would not be possible to differentiate between the effects of static features and those due to changes in saturation and pressure. By comparing the data from 3D surveys acquired at different times in the same location, however, it is possible to eliminate the effects of unknown static properties to focus on the dynamic changes in production related properties.

The simplest, most direct method of using time-lapse seismic data is to qualitatively monitor reservoir changes due to production. In this approach, one simply identifies regions in which the amplitude or impedance has changed with time and attributes these changes to changes in saturation, pressure, or temperature. The first tests of this concept were carried out by Arco in the Holt Sand fireflood from 1981 to 1983 (Hughes, 1998). Similar studies have been reported by Cooper et al. (1999) at the Foinhaven Field and Lumley et al. (1999) at the Meren Field in Nigeria. The primary objectives at Foinhaven were simply to map fluid movements and to identify by-passed oil. The authors of the study concluded that the time-lapse signal qualitatively agreed with the expected reservoir performance. At Meren, the goal was to identify pathways of injected water, sealing faults, and compartments that

may have by-passed oil. The authors concluded that the data was successful in achieving these objectives.

The other, more difficult, approach is to use the time-lapse data to estimate the reservoir flow parameters, such as permeability and porosity. In studies to date, the sensitivity of time-lapse seismic data to changes in model parameters has either been computed by the finite-difference method (Huang et al., 1997, 1998, 2001; van Ditzhuijzen et al., 2001) or the gradient simulator method (Landa and Horne, 1997). It is not feasible to compute sensitivity coefficients using either of these methods when the number of model parameters is large, however. The only reasonable approach is to use the adjoint method to integrate seismic impedance data into our objective function and to compute the sensitivity of data to model parameters and this is the approach followed in this work.

Although the simplest approach to integrate static data is to lump all static data into a prior geostatistical model and then apply Bayes theorem to derive a posteriori pdf which integrates static data, production data and seismic data, we have also developed methodology to incorporate more detailed geology using a facies model. Facies (defined here as regions of relatively uniform petrophysical properties) are common to all reservoir. Because the variation of rock properties between facies is normally far greater than the variations within an individual facies, knowledge of the locations of boundaries between facies diminishes the uncertainty in the prediction of reservoir performance and increases our ability to optimize reservoir management. In this work, we developed a truncated pluri-Gaussian model for the description of reservoir boundaries. The truncated plurigaussian is attractive for modeling facies for the following 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 are well-suited to current in history matching codes.

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

In the course of this project, we have developed procedures to solve two aspects of the history matching problem. The first problem 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 other 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 problem is adjustment of the facies boundaries for a fixed set of geostatistical model parameters.

While our primary focus has been on estimation of log-permeability and porosity fields and facies boundaries, we have also investigated the estimation of parameters defining power-law relative permeability curves from production data. The basic idea is not new; Archer and Wong (1973) used a trial and error approach (manual history matching) to estimate relative permeability curves by history matching laboratory core flood data and Sigmund and McCaffery (1979) used automatic history matching (nonlinear regression) for the same purpose. However, both authors estimated only the two parameters defining the shape of the relative permeability curves. Later, Kerig and Watson (1987), Kerig and Watson (1986) and Watson et al. (1988) applied splines to obtain a more general representation of relative permeability curves. Of the spline functions they considered, B-splines were found superior in that the coefficients of the B-spline representations of the relative permeability curves represent the independent adjustable parameters, and with parabolic B-splines, it is easy to constrain the B-splines to be monotonic although there is no guarantee that they will be concave up.

The most direct precursors of our work on estimating relative permeability curves is the work of Lee and Seinfeld (1987) and Yang and Watson (1991). Lee and Seinfeld (1987) considered the simultaneous estimation of the absolute permeability field and relative permeabilities for a two-dimensional, two-phase flow oil-water system. They assumed power law relative permeability curves and assumed that the end point values of relative permeabilities were known. Thus only the two exponents in the power law relative permeability functions were estimated. In the specific examples considered, they matched pressure and water cut data at wells producing from an oil reservoir under waterflood. Tikhonov (1963) regularization was used to stabilize the nonlinear least squares problem. Matching of data was accomplished by a three-step process with the steepest descent algorithm applied for minimization of the objective function which includes the sum of squared data mismatch terms. In the application of steepest descent, the gradient of the objective function was calculated using the adjoint method (Chen et al., 1974; Chavent et al., 1975). Yang and Watson (1991) considered the estimation of relative permeability curves using a Bayesian approach with relative permeability functions modeled as a linear combination of B-splines. In this approach, the objective function to be minimized is the sum of two terms, a production data mismatch term and a term which measures the deviation from a prior relative permeability model. (A prior model would typically be developed from laboratory core floods or by analogy with similar reservoirs.) They considered only homogeneous reservoirs and

assumed that all physical properties except relative permeabilities were known. Minimization of the objective function was accomplished with a Broyden-Fletcher-Goldfarb-Shanno optimization algorithm. They illustrated their methodology by applying it to a synthetic two-dimensional, two-phase flow waterflooding problem with a single injection well and a single producing well. They matched pressure data at both wells and WOR data at the producing well. A thorough review of other work on the estimation of relative permeability curves can be found in Reynolds et al. (2004).

In this project, we consider the simultaneous estimation of the absolute permeability field and relative permeability curves from three-phase flow production data. Irreducible water saturation, critical gas saturation and residual oil saturations are assumed to be known. The two-phase relative permeability curves for an oil-gas system and the two-phase relative permeability curves for an oil-water system are represented by power law models. The three-phase oil relative permeability curve is calculated from the two sets of two-phase curves using Stone's Model II. The adjoint method is applied to three-phase flow problems to calculate the sensitivity of production data to the absolute permeability field and the parameters defining the relative permeability functions. Using the calculated sensitivity coefficients, absolute permeability and relative permeability fields are estimated by automatic history matching of production data. A prior model for absolute permeability and relative permeability parameters is assumed to provide regularization, i.e., Bayesian estimation is applied to generate estimates. Model parameters, which are estimated by automatic history matching of production data, consist of gridblock absolute log-permeabilities and the parameters defining the relative permeability curves. As the example problems considered here are relatively small, we apply the Levenberg-Marquardt algorithm to do history matching. As noted above the estimation of relative permeability from production data is not a new problem, but to the best of our knowledge, our work represents the first attempt to estimate three-phase relative permeabilities by history matching.

As noted previously, our focus is on large scale problems where the number of model parameters and number of model parameters are both large. One way to attempt to deal with such problems is via reparameterization by reducing the number of model parameters (reservoir variables). This type of approach has a long history beginning with the pioneering work on zonation of Jacquard and Jain (1965) and Jahns (1966) and continuing today with work on adaptive parameterization; see, for example, Grimstad et al. (2001). During the first year of this project report, we investigated reparameterization based on subspace methods (Kennett and Williamson, 1988; Oldenburg et al., 1993; Oldenburg and Li, 1994; Reynolds et al., 1996) to reduce the number of parameters directly estimated in the optimization

process. Our results on this approach were summarized in Abacioglu et al. (2001) and are presented in detail in Abacioglu (2001). These references considered only single-phase problems. Based on these results, we believed that algorithms that require only the gradient of the objective function and do not directly form the matrix of sensitivity coefficients would prove to be more efficient than subspace methods for large scale history matching problems. Three possibilities exist, truncated Newton (Nash, 1985), nonlinear conjugate gradient, and quasi-Newton methods. As discussed later, our intuition suggests that it will be difficult to implement the truncated Newton method in an efficient way for large scale problems and thus, we focused on nonlinear conjugate gradient and quasi-Newton methods. Quasi-Newton methods have been employed previously in automatic history matching-problems by Yang and Watson (1988), Masumoto (2000) and Savioli and Grattoni (1992). While these studies generally found that a self-scaling BFGS method is more robust and computationally efficient than CG, steepest descent and standard unscaled BFGS algorithms, the number of reservoir variables estimated was less than 25 in all examples considered. Thus, it is impossible to draw any conclusions from their results about how these algorithms will perform when the number of reservoir variables to be estimated range from several hundred to tens of thousands.

Quasi-Newton (variable metric) methods, which are based on generating an approximation to the inverse of the Hessian matrix, require only the gradient of the objective function and thus avoid the computation of individual sensitivity coefficients needed to directly form the Hessian matrix. Here, only the Broyden-Flecher-Goldfarb-Shanno (BFGS) quasi-Newton method is considered since it has proved to be more robust in practice than other algorithms; see Kolda et al. (1998). It is well known that scaling can improve the convergence attributes of quasi-Newton methods, and numerous suggestions have been made for calculating scaling factors; see, in particular, Oren and Luenberger (1974) and Shanno and Phua (1978). We identify scaling procedures that have worked well for the limited number of history matching problems that we have tried. For large scale optimization problems, memory requirements may be reduced by replacing the BFGS algorithm with the limited memory Broyden-Fletcher-Goldfarb-Shanno (LBFGS) algorithm in the form developed and implemented by Nocedal (1980).

Similar to quasi-Newton methods, the nonlinear conjugate gradient algorithm does not require computation of the Hessian matrix; only the gradient of the objective function is required. Here, we use the Polak-Ribière form of the nonlinear conjugate gradient algorithm; see Fletcher (1987). Makhlouf et al. (1993) applied the conjugate gradient method to estimate 450 gridblock permeabilities by history matching production data obtained under multiphase flow conditions. In one example, 110 iterations were required to obtain conver-

gence. In the second example considered, 222 iterations were required to obtain convergence. However, the authors apparently did not use preconditioning. Because the efficiency of the conjugate gradient (CG) method depends primarily on the preconditioner used, we attempted to develop preconditioners to obtain an efficient implementation of the nonlinear conjugate gradient algorithm. Although the two preconditioners we tried improved the performance of the CG method, the resulting algorithms are still significantly less robust than appropriately scaled BFGS and LBFGS algorithms.

It is important to note that for cases where the number of data and or the number of model parameters is small, the Gauss-Newton (GN) method or a modified Levenberg-Marquardt (MLM) method can be used successfully to history match production data (pressures, water-oil ratios and gas-oil ratios.) In fact, these algorithms have been applied in our work on estimating the parameters defining power law relative permeability. When either the MLM or GN method is used, all sensitivities necessary to form the Hessian are computed. If the number of model parameters (reservoir variables) to be estimated is small, the sensitivity of production data to model parameters can be computed by the so-called gradient simulator method (Anterion et al., 1989; Yeh, 1986), whereas, if the number of dynamic data to be history matched is small but the number of model parameters is large, sensitivities can be computed with the adjoint method (Chen et al., 1974; Chavent et al., 1975) using the multiphase, multidimensional implementation of the adjoint method developed by Li et al. (2003a).

All results presents in this report are given in oil field units.

Chapter 2

HISTORY MATCHING OF DATA

2.1 Model Estimation and Simulation

Here, we introduce the basic notation for the history matching problem in a Bayesian setting. This includes definition of the model or model parameters and the a posteriori probability density function (pdf) for these parameters. This pdf, which is conditional to observed data, determines the set of plausible reservoir descriptions. We discuss the computation of the maximum a posteriori (MAP) estimate of reservoir variables. The MAP estimate is the model which maximizes the a posteriori pdf and is thus conveniently referred to as the most probable model. A method for sampling this pdf to generate multiple realizations of reservoir variables is also discussed.

2.1.1 The Prior Model

The reservoir is assumed to occupy the volume Ω. Although it is not necessary, for simplicity, we often use test examples where the reservoir is assumed to be a rectangular parallelepiped which occupies the region

$$\Omega = \{(x, y, z) \mid 0 < x < L_x, 0 < y < L_y, 0 < z < L_z\}. \tag{2.1}$$

The forward model is a fully-implicit finite-difference simulator based on a block centered grid. The principle permeability directions are assumed to be aligned with the coordinate directions so that the permeability tensor is diagonal. Fluid properties are assumed to be known. Given two-phase oil-water and two-phase oil-gas relative permeabilities, the three-phase oil relative permeability is constructed from Stone's Model II; see Aziz and Settari (1979). Wellbore constraints are handled using the equation of Peaceman (1983).

Although many examples are presented in terms of a rectangular reservoir, non-rectangular reservoirs can be considered using two features of the simulator. First, grid cells can be made inactive by assigning zero permeabilities and nonzero porosities to those cells. Secondly, even though the simulator is based on using a specified number of "layers" in the z direction, the top depth of the reservoir can be a function of (x, y) and the vertical dimension (Δz) can be different for each gridblock in a "layer." This provides significant flexibility in defining the geometry of the reservoir; for example, an anticline can be modeled.

Here, we assume model parameters or reservoir variables include only gridblock porosities, horizontal log-permeabilities, vertical log-permeabilities, the skin factor at each well and parameters defining power law relative permeability curves. In a later chapter, we discuss our work on estimating the location of the boundaries between facies and this model parameters can include parameters defining the boundaries between facies. For simplicity in exposition, we write the vector of model parameters for the case where permeability is areally isotropic so that $k_x = k_y$. However, the methodology presented applies to the fully anisotropic case. For the areally isotropic case, the model parameters may include the reservoir simulator gridblock porosities, horizontal log-permeabilities, vertical log-permeabilities, well skin factors and parameters defining power law relative permeability curves. We estimate at most one skin factor per well, so the number of skin factors to be estimated is equal to the number of wells. Thus, if there are N simulator gridblocks and N_w wells, the total number of model parameters is equal to $N_m = 3N + N_w + N_{rp}$, where N_{rp} is the number of parameters defining relative power law relative permeability curves; see Chapter3. Specifically, the model (or vector of model parameters, or set of reservoir variables) is given by

$$m = [m_\phi^T, m_k^T, m_{k_z}^T, m_s^T, m_{rp}^T]^T, \tag{2.2}$$

where m_ϕ is an N-dimensional column vector with its jth entry equal to the porosity of gridblock j, m_k is an N-dimensional column vector with its jth entry equal to the horizontal log-permeability for gridblock j, m_{k_z} is an N-dimensional column vector with its jth entry equal to the vertical log-permeability for gridblock j, m_s is an N_w dimensional column vector with its jth entry given by the skin factor at the jth well, and m_{rp} is the vector of parameters defining the power law relative curves. These reservoir parameters are modeled as random variables, so m is a random vector. From a purely history matching point of view, we wish to construct an estimate of m from production and time lapse seismic data (dynamic data) and static data. However, there are an infinite number of models which will give equally reasonable matches of the data, and it is desirable to define a procedure for generating a particular estimate or to characterize the uncertainty in reservoir descriptions. From both the philosophical and practical points of view (see Tarantola (1987) and Omre et al. (1993)), the

most challenging part of the inverse problem is the determination of a representative pdf for reservoir parameters. Similar to other work in automatic history matching (Oliver, 1994; Chu et al., 1995a; He et al., 1997; Wu et al., 1999), we follow ideas that can be found in Tarantola (1987) and simply assume that a prior geostatistical model for $m_r = [m_\phi, m_k^T, m_{k_z}^T]^T$ can be constructed from static data, e.g. log, core and other geologic data. We also assume a prior model can be constructed for the well skin factors and the parameters defining relative permeability curves. One may be able to construct a prior model for well skin factors from completion information or from well test data and analogue reservoirs or laboratory data can be used to generate a prior model for power law relative permeability curves. These prior model can of course be highly uncertain. In fact, the point of history matching production and/or seismic data is to reduce the uncertainty in our reservoir model. In our work, we use a prior geostatistical model based on the assumption that m_r can be represented by a multivariate Gaussian distribution with a given mean and covariance matrix. In practice, the prior covariance matrix for the rock property fields can be generated from semivariograms by assuming that porosity and horizontal and vertical permeability can be modeled as stationary random functions. In our implementation, we make this assumption and then apply the Xu et al. (1992) screening hypothesis to generate the prior covariance matrix for m_r; see, Chu et al. (1995b). In the prior model, each well skin factor is treated as an independent Gaussian variable with specified mean and variance. If the skin factor was estimated by fitting pressure data with a classical well testing model solution using nonlinear regression, then the estimate of the skin factor would be its prior mean and its variance can be constructed directly from the same information used to construct confidence intervals. The prior model for set of parameters defining relative permeability curves assumes these parameters can be modeled as independent Gaussian random variables.

The vector of prior means is given by

$$m_{\text{prior}} = \begin{bmatrix} m_{\phi,\text{prior}} \\ m_{k,\text{prior}} \\ m_{k_z,\text{prior}} \\ m_{s,\text{prior}} \\ m_{rp,\text{prior}} \end{bmatrix}. \tag{2.3}$$

C_ϕ denotes the prior covariance matrix for m_ϕ, C_k denotes the prior covariance matrix for m_k and C_{k_z} denotes the prior covariance for m_{k_z}. $C_{\phi,k}$ denotes the cross covariance matrix between ϕ and m_k, C_{ϕ,k_z} denotes the cross covariance matrix between ϕ and m_{k_z} and C_{k,k_z} denotes the cross covariance matrix between m_k and m_{k_z}. We let C_s denote the $N_w \times N_w$ model covariance matrix for the vector of well skin factors and let C_{rp} denote the prior

covariance matrix for relative permeability parameters. Our assumptions imply that C_s and C_{rp} are diagonal matrices. The prior model covariance matrix is given by the $N_m \times N_m$ matrix

$$C_M = \begin{bmatrix} C_\phi & C_{\phi,k} & C_{\phi,k_z} & O & O \\ C_{\phi,k} & C_k & C_{k,k_z} & O & O \\ C_{\phi,k_z} & C_{k,k_z} & C_{k_z} & O & O \\ O & O & O & C_s & O \\ O & O & O & O & C_{rp} \end{bmatrix}, \tag{2.4}$$

where the O's denote null submatrices of the appropriate size. If porosity is uncorrelated with horizontal and vertical permeability, then $C_{\phi,k}$ and C_{ϕ,k_z} are null matrices. If horizontal and vertical permeability are not correlated, then C_{k,k_z} is also a null matrix.

The prior pdf for m is then given by

$$\pi_p(m) = a \exp\Big\{ -\frac{1}{2}(m - m_{\text{prior}})^T C_M^{-1}(m - m_{\text{prior}})\Big\}, \tag{2.5}$$

where a is the normalizing constant. Note the model which has the highest probability based on Eq. 2.5 is $m = m_{\text{prior}}$, thus it is convenient to think of m_{prior} as the best estimate of the model based on static data.

2.1.2 The a Posteriori Probability Density Function

We wish to determine the conditional pdf for m given observed production data and/or time-lapse seismic data. Here, we consider only three types of production data, wellbore pressure (p_{wf}), producing water-oil ratio (WOR) and producing gas-oil ratio (GOR). The WOR and GOR data are not actually measured directly but are constructed from rate measurements. Nevertheless, we will refer to the values of WOR and GOR as measured or observed data. The column vector $d_{\text{obs,w}}$ contains all observed WOR data that will be used as conditioning data. The column vector $d_{\text{obs,g}}$ contains the set of GOR conditioning data and $d_{\text{obs,p}}$ contains all conditioning pressure data. Throughout, the N_d dimensional column vector d_{obs} includes all data that will be used to condition the model m. This may include one type of data, e.g., only GOR data or multiple types of data, e.g., pressure, WOR and GOR data and time-lapse seismic data. Although, we do not consider it here, d_{obs} may even include hard data, e.g., observed porosities at wells.

Pressure measurements errors are modeled as independent identically distributed Gaussian random variables with mean zero and the variance of the ith measurement error given by $\sigma^2_{p,i}$. Often, we assume the variances of all pressure measurement errors are equal. GOR measurement errors are modeled as independent Gaussian random variables with mean zero

and variance $\sigma^2_{g,i}$ where the subscript g, i refers to the ith GOR measurement error. Similarly, measurement errors in the WOR ratio are assumed to be independent random variables with mean zero with the variance of the ith WOR measurement error given by $\sigma_{w,i}$. Although we refer to these errors as measurement errors, they must also account for modeling or processing errors, for example, if the total phase production rates are measured at a central location and then back allocated to individual wells. In the case of time-lapse seismic data, the covariance matrix for the "measurement errors" also include modeling/processing errors and these errors may be correlated (Aannonsen et al., 2002). The necessity to have an estimate of measurement errors is a significant problem when integrating data as the measurement errors will determine the relative weights of different types of data in the objective function that must be minimized to obtain a history match of dynamic data. Aannonsen et al. (2002) used a moving average to estimate true data and then generated an estimate of measurement error by subtracting the estimated true data from the observed data in order to construct an estimate of the covariance of measurement errors for both production and seismic data. His work indicates that this is a challenging problem and one worthy of further investigation.

In synthetic cases, WOR ratio measurement errors are often modeled by the procedure introduced by Wu et al. (1999). In this model, the WOR measurement error depends on the magnitude of the measurement. Specifically, the variance of a particular measurement error is defined as

$$\mathrm{Var}(e_{WOR}) = WOR^2_{\mathrm{obs}}\epsilon_o + \frac{1}{q^2_{o,\mathrm{obs}}}\max\left[\epsilon^2_w q^2_{\mathrm{w,obs}},\, \sigma^2_{\mathrm{w,min}}\right], \tag{2.6}$$

where e_{WOR} denotes the error in the "measurement" of WOR constructed from the observed oil and water rates, $q_{o,\mathrm{obs}}$ and $q_{\mathrm{w,obs}}$. Here, ϵ_m denotes the relative measurement error for the flow rate of phase m. For example, if the relative measurement error in the oil flow rate is two per cent, then $\epsilon_o = 0.02$. The term $\sigma_{\mathrm{w,min}}$ is used so that we do not prescribe unrealistically small measurement errors for the WOR when the WOR is small. To use this model, one must specify values of ϵ_w, ϵ_o, and $\sigma_{\mathrm{w,min}}$.

The three diagonal matrices, $C_{D,p}$, $C_{D,w}$ and $C_{D,g}$, respectively, denote the covariance matrices for pressure data measurement errors, WOR measurement errors and GOR measurement errors, and $C_{D,s}$ denotes the covariance matrix for time-lapse seismic data.

If the total number of dynamic data to be history matched is N_d, i.e., the dimension of

$d_{\rm obs}$ is N_d, then the overall data covariance matrix is given by the following $N_d \times N_d$ matrix:

$$C_D = \begin{bmatrix} C_{D,p} & O & O & O \\ O & C_{D,w} & O & O \\ O & O & C_{D,g} & O \\ O & O & O & C_{D,s} \end{bmatrix}. \tag{2.7}$$

We of course do not need to use all types of data as conditioning data. For example, if we wish to history match only GOR data, then $d_{\rm obs} = d_{\rm obs,g}$ and $C_D = C_{D,g}$.

For a given model m, d denotes the predicted, true or calculated data corresponding to $d_{\rm obs}$. If m is the true reservoir from which $d_{\rm obs}$ was obtained and there are no measurement errors, then $d = d_{\rm obs}$. As d depends on the model, we write

$$d = g(m), \tag{2.8}$$

to represent the operation of calculating d given m. In our work, Eq. 2.8 represents the operation of running the reservoir simulator to calculate d.

Bayes' theorem (see Tarantola (1987)) implies that the a posteriori pdf for the model m conditional to the observed data is proportional to the product of the prior pdf and the likelihood function for the model, and is thus given by

$$f(m|d_{\rm obs}) = a \exp\{-O(m)\}, \tag{2.9}$$

where a is the normalizing constant and

$$O(m) = \frac{1}{2}\Big[\big(m - m_{\rm prior}\big)^T C_M^{-1}\big(m - m_{\rm prior}\big) + \big(g(m) - d_{\rm obs}\big)^T C_D^{-1}\big(g(m) - d_{\rm obs}\big)\Big]. \tag{2.10}$$

2.1.3 Construction of the MAP Estimate and Realizations

The maximum a posteriori (MAP) estimate is denoted by m_∞ and is defined to be the model that maximizes the pdf of Eq. 2.9, or equivalently minimizes the objective function of Eq. 2.10. Although gradient based methods appear to be the only feasible way to construct a minimum of $O(m)$, there is no guarantee that Eq. 2.10 has a unique global minimum, or that a gradient-based optimization procedure will converge to a global minimum. In fact, if a gradient method is applied to minimize $O(m)$, it is important to check the results to ensure that the method did not converge to a local minimum which yields an unacceptable match of production data, or unreasonable reservoir properties. Thus, we wish a MAP estimate to satisfy two criteria (i) the geological features and distribution of rock properties should be plausible when compared with the features encapsulated in the prior geostatistical

model and (ii) the map estimate should predict dynamic data that is in good agreement with the observed dynamic data that we are trying to history match. An indication of whether we have obtained an acceptable match of observed conditioning data is provided by the value of the objection function at the MAP estimate, m_∞, obtained from the optimization algorithm. Specifically, when the data is linearly related to m, it is shown in Tarantola (1987) that the $O(m_\infty)$ has a chi-squared distribution with expectation $N_d/2$ and variation $N_d/2$. Assuming that a value of $O(m_\infty)$ that differs more than five standard from its expected value is unreasonable, we do not accept m_∞ as a legitimate history-matched model unless

$$\frac{N_d}{2} - 5\sqrt{\frac{N_d}{2}} \leq O(m_\infty) \leq \frac{N_d}{2} + 5\sqrt{\frac{N_d}{2}}. \tag{2.11}$$

or equivalently,

$$1 - 5\sqrt{\frac{2}{N_d}} \leq \frac{2}{N_d} O(m_\infty) \leq 1 + 5\sqrt{\frac{2}{N_d}}. \tag{2.12}$$

Although this result can be rigourously established only when the relation of Eq. 2.8 is linear, i.e.,

$$d = Gm, \tag{2.13}$$

where G is an $N_d \times N_m$ matrix, computational results suggest we should be able to satisfy Eq. 2.12 even when the relation between data and the model is nonlinear.

The MAP estimate effectively represents the mean of an approximation to the a posteriori pdf based on a linearization about the MAP estimate. As such, the MAP estimate is normally too smooth to provide a plausible reservoir model. Moreover, if one wishes to assess the uncertainty in the model or in future reservoir performance predictions, it is desirable to generate multiple realizations of the model. In order for these models to give an accurate characterization of the uncertainty, they should represent a correct sample of the conditional pdf of Eq. 2.9. The most common way to do this is to apply the method proposed by Oliver et al. (1996) and Kitanidis (1995). In our work this method is referred to as the randomized maximum likelihood method. To generate a realization with this procedure, we calculate an unconditional realization m_{uc} from

$$m_{\text{uc}} = m_{\text{prior}} + C_M^{1/2} z_M, \tag{2.14}$$

where z_M is N_m-dimensional column vector of independent standard random normal deviates. The matrix $C_M^{1/2}$ is a square root of C_M and is normally chosen as $C_M^{1/2} = L$ where

$$C_M = LL^T, \tag{2.15}$$

is the Cholesky decomposition of C_M. For large problems, generation of the Cholesky decomposition is not feasible, and we apply sequential Gaussian co-simulation to generate an unconditional realization of the model; see Gómez-Hernández and Journel (1992). Similarly a realization of the data is generated from

$$d_{\rm uc} = d_{\rm obs} + C_D^{1/2} z_D, \tag{2.16}$$

where z_D is an N_d-dimensional column vector of standard random normal deviates. The conditional realization of m is then obtained by minimizing

$$O_r(m) = \frac{1}{2}(m - m_{\rm uc})^T C_M^{-1}(m - m_{\rm uc}) + \frac{1}{2}(d - d_{\rm uc})^T C_D^{-1}(d - d_{\rm uc}). \tag{2.17}$$

Similar to the results which gave Eq. 2.11, it can be argued (Zhang et al., 2001a)) that $O(m)$ can be approximated as a chi-squared distribution with expectation given by $E(O(m)) = N_d$ and standard deviation given approximately by $\sigma(O(m)) \approx \sqrt{2N_d}$. Virtually all samples should be within five standard deviations of the mean. Thus, if applying an optimization algorithm to minimize Eq. 2.17 gives a result m_c, we accept m_c as a legitimate realization if and only if

$$N_d - 5\sqrt{2N_d} \leq O(m_c) \leq N_d + 5\sqrt{2N_d}. \tag{2.18}$$

Eq. 2.18 has proved to be very reliable for single-phase flow history matching synthetic problems, but for multiphase flow problems, for a long time, we encountered situations where we were unable to decrease the objective function to a value consistent with Eq. 2.18. In this report, we discuss damping, rescaling and constrained optimization procedures that enable us to obtain conditional realization satisfying Eq. 2.18 using the LBFGS optimization algorithm.

It is important to note that the goal of this project was to develop reliable computational techniques for history matching large scale problems, not uncertainty analysis. Thus, in most examples considered we do not generate multiple realizations. Instead, either the MAP estimate or a single conditional realization is calculated using the randomized maximum likelihood method.

2.1.4 Gauss-Newton and Levenberg-Marquardt Algorithms

The Gauss-Newton method with restricted step has often been used to minimize $O(m)$; see Chu et al. (1995a). However, if the initial guess for the model yields a very poor match of the observed production data, a straightforward application of the method may converge extremely slowly or may converge to a model which yields an unacceptable match of production data; see Wu et al. (1999). Moreover the model obtained may exhibit undershooting

and overshooting, e.g., unreasonably high or low values of rock properties. Wu et al. (1999) overcame this problem by using an artificially high value for the variance of data measurement errors at early iterations. For problems which are sufficiently small so that it feasible to compute the sensitivity coefficient matrix, we can overcome this problem using the modified Levenberg-Marquardt algorithm introduced for Bayesian estimation problems by Bi (1999). For large scale problems, we extend the data damping procedure used in Wu et al. (1999) together with rescaling to overcome overshooting and undershooting problems.

Here, we avoid this difficulty by using a form of the Levenberg-Marquardt algorithm introduced by Bi (1999). This algorithm can be written in two different forms. The first comes from a modification of the standard Gauss-Newton method and is given by

$$\left[(1+\lambda_l)C_M^{-1} + G_l^T C_D^{-1} G_l\right]^{-1} \delta m^{l+1} = -\left[C_M^{-1}(m^l - m_{\text{prior}}) + G_l^T C_D^{-1}(g(m^l) - d_{\text{obs}})\right], \quad (2.19)$$

$$m^{l+1} = m^l + \alpha_l \delta m^{l+1} \quad (2.20)$$

where $\alpha_l = 1$. Here l, as either a subscript or superscript, refers to the iteration index. The matrix G_l denotes the $N_d \times N_m$ sensitivity coefficient matrix evaluated at m^l. The entry in the ith row and jth column of G_l represents the sensitivity of the ith calculated data g_i to the jth model parameter evaluated at m^l, i.e., this entry is $\partial g_i(m^l)/\partial m_j$, where m_j is the jth entry of m. If $O(m^{l+1}) < O(m^l)$, we set $\lambda_{l+1} = \lambda_l/10$, and if the objective function does not decrease, we increase the Levenberg-Marquardt parameter by a factor of 10. We start with an initial value of $\lambda = 10,000$. For the multiphase flow problems we have considered to date, this simple procedure works well.

2.2 LM Versus Gauss-Newton

We consider two-dimensional flow problem where water is injected into an oil reservoir with initial pressure above bubble point pressure. Water is injected via a well near the center of the reservoir and four wells near the corners of the reservoir are produced at a constant oil rate. All fluid and rock properties are assumed known except for the gridblock values of absolute permeability. Observed data consist of eight values of wellbore pressure at each well uniformly spaced in time with the first pressure measurements occurring at 30 days. We wish to history match these pressure data to generate the MAP estimate of the log-permeability field. The model consists of the gridblock values of $\ln(k)$ on a $21 \times 21 \times 1$ reservoir simulation grid. The grid is areally uniform, i.e., $\Delta x = \Delta y$. In the history matching procedure, the covariance matrix C_M was generated using an isotropic spherical variogram with a sill of 0.5 and a range equal to about $4\Delta x$.

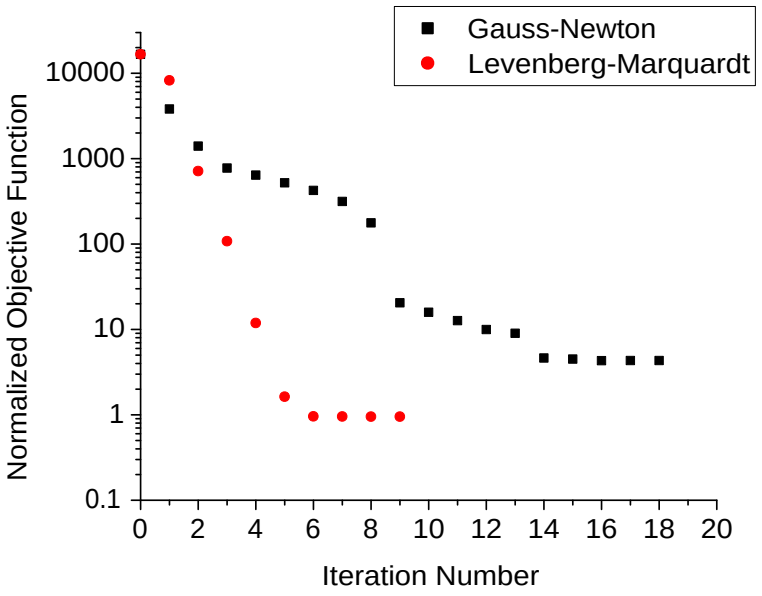


Figure 2.1: Behavior of the normalized objective function for Gauss-Newton and Levenberg-Marquardt algorithms.

History matching was done by minimizing the objective function of Eq. 2.10 using the modified Levenberg-Marquardt (MLM) algorithm of Eq. 2.18 and with the Gauss-Newton method of obtained by setting $\lambda_l = 0$ for all l in Eq. 2.10. Fig. 2.2 shows the behavior of the normalized objective function during the iterations of the two optimization algorithms. Here the normalized objective function is defined by

$$O_N(m) = 2\frac{O(m)}{N_d}, \tag{2.21}$$

where the objective function $O(m)$ is given by Eq. 2.10. Because $N_d = 32$, Eq. 2.12 indicates that expected value of the normalized objective function evaluated at the MAP estimate should be around unity and should satisfy

$$O_N(m_\infty) \leq 1 + 5\sqrt{\frac{2}{N_d}} = 2.25. \tag{2.22}$$

The results of Fig. 2.1 indicate that Eq. 2.22 is satisfied by the MAP estimate obtained with the MLM algorithm but is not satisfied using the MAP estimate obtained with the Gauss-Newton method even though more iterations were done with the Gauss-Newton algorithm. Much more importantly, Fig. 2.2 shows that the MAP estimate of $\ln(k)$ obtained from the Gauss-Newton method is rough in the sense that small values of $\ln(k)$ on the order of 3.0 or

less occur in gridblocks close to gridblocks where $\ln(k)$ is 4.5 or greater. This roughness is a consequence of ill-conditioning which tends to result in overly rough models at early iterations when the Gauss-Newton method is used. Note the MAP estimate obtained with the MLM algorithm is much smoother or more continuous (Fig 2.3) because the MLM iteration matrix is better conditioned at early iterations and large changes in the model parameters are damped. (In the MLM algorithm, the value of λ_l was decreased by a factor of 10 at each iteration.) (Strictly speaking we should refer to the models obtained as estimates of the MAP estimate as the estimated models are different and there is no assurance that either permeability field is a global minimum of $O(m)$. Nevertheless, we will simply refer to a model obtained by applying an optimization algorithm to minimize Eq. 2.10 as a MAP estimate instead of an estimate of a MAP estimate.)

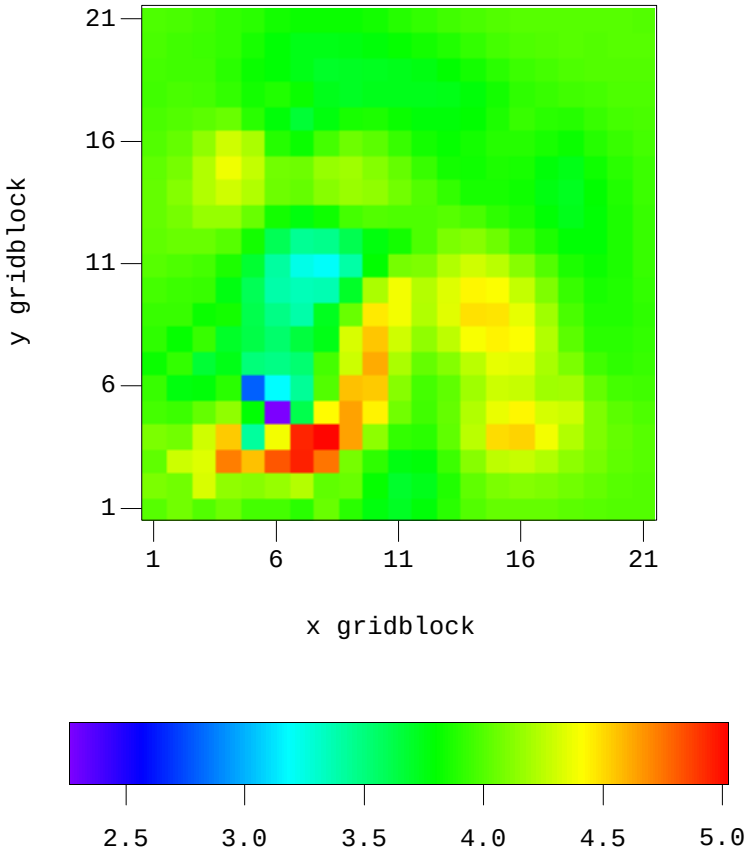


Figure 2.2: MAP estimate from Gauss-Newton method.

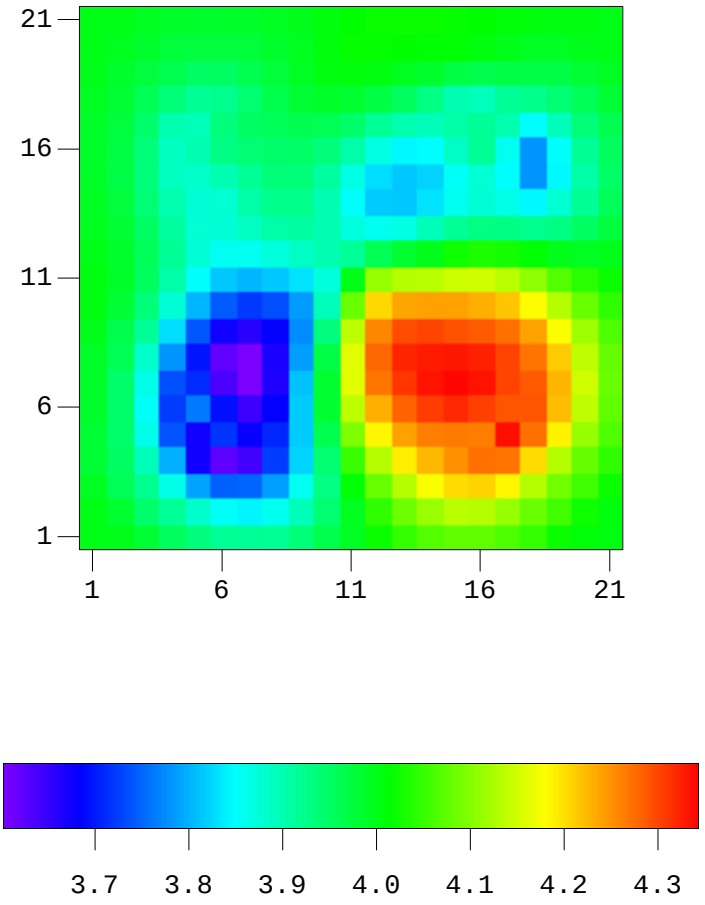


Figure 2.3: MAP estimate from the modified Levenberg-Marquardt algorithm.

2.3 Calculation of Sensitivities with the Adjoint Method

It is not feasible to apply automatic history matching of production data unless one has an efficient method for computing sensitivities that can be coupled with a multiphase reservoir simulator. One key contribution of our work is the development and implementation of an adjoint procedure that can be coupled with a fully implicit finite-difference simulator in order to calculate sensitivities needed in gradient based optimization routines. The equations that must be solved to compute sensitivity coefficients with the adjoint method are presented in this section. The procedure given here is also used to compute the gradient of the objective function if the LBFGS or nonlinear conjugate gradient optimization algorithms are used in history matching.

2.3.1 The Reservoir Simulator

The simulator used is based on a fully-implicit, finite-difference formulation of the three-phase flow, black-oil equations expressed in a x–y–z coordinate system which apply on Ω; see Eq. 2.1. Suppose there are N_x, N_y, N_z gridblocks in the $x-$, $y-$ and $z-$ directions respectively. Let N be the total number of gridblocks, i.e., $N = N_x \times N_y \times N_z$. At each of the N gridblocks, three basic finite-difference equations apply. These three equations represent the mass balance for each of the three components, i.e., oil, gas and water. In addition, a constraint is applied at each of the N_w wells to yield N_w additional equations. At each well at each time step, either an individual phase flow rate, the total flow rate or the wellbore pressure may be specified as a well constraint. In the results considered in this work, capillary pressures are assumed to be negligible. The fully-implicit, black-oil simulator (CLASS-Chevron's Limited Applications Simulation System) used in this work was provided by Chevron.

For gridblock i, the primary variables that are solved for are case dependent. Table 2.1 summarizes the different cases and the primary variables solved for in each case. In the column entitled "Equations", Sum denotes the total mass balance equation (i.e., the summation of the oil, gas and water equations); Oil represents the oil mass balance equation and Gas represents the gas mass balance equation.

Phases		Equations	Unknowns	Auxiliary equation
O-W-G	$S_g > 0$	Sum, Oil, Gas	p, S_o, S_g	$S_w = 1 - S_o - S_g$; R_s from PVT table
	$S_g = 0$	Sum, Oil, Gas	p, S_o, R_s	$S_g = 0$; $S_w = 1 - S_o - S_g$
O-W		Sum, Oil	p, S_o	$S_w = 1 - S_o$
W-G		Sum, Gas	p, S_g	$S_w = 1 - S_g$
O-G	$S_g > 0$	Sum, Gas	p, S_g	$S_o = 1 - S_g$; R_s from PVT table
	$S_g = 0$	Sum, Gas	p, R_s	$S_g = 0$; $S_o = 1 - S_g$

Table 2.1: Equations and unknowns solved for in the simulator.

At each time step, we can output p, S_o, S_g, S_w and R_s of each individual gridblock from CLASS. From these primary variables, we can calculate all the derivatives required for constructing the adjoint system based on the PVT table. In addition to the gridblock variables, the flowing wellbore pressure, $p_{wf,l}$ at the lth well at a specified depth is also a primary variable. We let y^n denote a column vector which contains the set of primary variables (pressures and saturations) at time step n. At gridblock i, the finite-difference

equation for component u can be written as

$$f_{u,i}(y^{n+1}, y^n, m) = 0, \tag{2.23}$$

for $u = o, w, g$ and $i = 1, \ldots, N$. The well constraints are represented by

$$f_{wf,l}(y^{n+1}, y^n, m) = 0, \tag{2.24}$$

for $l = 1, 2, \ldots, N_w$. For simplicity, we let

$$f_{u,i}^{n+1} = f_{u,i}(y^{n+1}, y^n, m), \tag{2.25}$$

and

$$f_{wf,l}^{n+1} = f_{wf,l}(y^{n+1}, y^n, m), \tag{2.26}$$

then Eqs. 2.23 and 2.24 can be rewritten as

$$f_{u,i}^{n+1} = 0, \tag{2.27}$$

and

$$f_{wf,l}^{n+1} = 0, \tag{2.28}$$

respectively. If the flowing wellbore pressure at well l at the datum depth at time t^{n+1} is specified to be equal to $p_{wf,l,0}^{n+1}$, then Eq. 2.25 is simplified to

$$f_{wf,l}^{n+1} = p_{wf,l}^{n+1} - p_{wf,l,0}^{n+1} = 0. \tag{2.29}$$

In CLASS, the three equations for three-phase problem that are solved at gridblock i are

$$f_{1,i}^{n+1} = f_{o,i}^{n+1} + f_{w,i}^{n+1} + f_{g,i}^{n+1} = 0 \tag{2.30}$$

$$f_{2,i}^{n+1} = f_{o,i}^{n+1} = 0 \tag{2.31}$$

$$f_{3,i}^{n+1} = f_{g,i}^{n+1} = 0 \tag{2.32}$$

where $i = 1, 2, \cdots, N$. If the following three equations

$$f_{1,i}^{n+1} = f_{o,i} = 0 \tag{2.33}$$

$$f_{2,i}^{n+1} = f_{w,i}^{n+1} = 0 \tag{2.34}$$

$$f_{3,i}^{n+1} = f_{g,i}^{n+1} = 0 \tag{2.35}$$

instead of Eqs. 2.30 through 2.32 are used to construct the Jacobian matrix, then we will have trouble for some situations to do incomplete LU decomposition of the Jacobian matrix

in order to use orthomin which is an iterative solver. From Table 2.1, we can see no matter which case happens, the pressure is always one of the primary variables. Hence, in the Jacobin matrix, the derivative of a certain equation at ith gridblock with respect to the pressure at the ith gridblock is always the diagonal element. The location of this entry depends on how you order the primary variables. If the pressure is ordered as the first primary variable in each gridblock as people usually do, then in the Jacobian matrix, every third diagonal entry will be $df_{1,i}/dp_i$ where i is the gridblock index. If $f_{1,i} = f_{o,i}$, then the derivative $df_{1,i}/dp_i$ is zero whenever oil saturation $S_{o,i}$ is zero. Because if this is the case, then every individual term involved in $df_{1,i}/dp_i$ is related to oil saturation by either relative permeability or S_o itself and becomes zero. The subroutine we used to perform the incomplete LU decomposition will perform the operation of dividing the row of Jacobian matrix by the diagonal element. Therefore, if the situation presented above happens, e.g., in the gas cap area, then this subroutine will be terminated because of the illegal math operation.

Eq. 2.28 and Eqs. 2.30 through 2.32 represent a system of N_e equations where

$$N_e = 3N + N_w. \tag{2.36}$$

These N_e equations are solved to obtain the values of the primary variables at time $t^{n+1} = t^n + \Delta t^n$. For wells at which the flowing bottom-hole pressure is specified, phase flow rates at each well are computed by Peaceman's equation (Peaceman, 1983). The component flow rates from the perforated layer k of well l (at gridblock (i, j, k)) at time step $n + 1$ can be evaluated as

$$q^{n+1}_{o,i,j,k} = WI_{i,j,k} \left(\frac{k_{ro}}{B_o \mu_o} \right)^{n+1}_{i,j,k} (p^{n+1}_{i,j,k} - p^{n+1}_{wf,l,k}), \tag{2.37}$$

$$q^{n+1}_{w,i,j,k} = WI_{i,j,k} \left(\frac{k_{rw}}{B_w \mu_w} \right)^{n+1}_{i,j,k} (p^{n+1}_{i,j,k} - p^{n+1}_{wf,l,k}), \tag{2.38}$$

and

$$\begin{aligned} q^{n+1}_{g,i,j,k} &= WI_{i,j,k} \left(\frac{k_{rg}}{B_g \mu_g} \right)^{n+1}_{i,j,k} (p^{n+1}_{i,j,k} - p^{n+1}_{wf,l,k}) + R^{n+1}_{so,i,j,k} q^{n+1}_{o,i,j,k} \\ &= WI_{i,j,k} \left(\frac{k_{rg}}{B_g \mu_g} + R_s \frac{k_{ro}}{B_o \mu_o} \right)^{n+1}_{i,j,k} (p^{n+1}_{i,j,k} - p^{n+1}_{wf,l,k}). \end{aligned} \tag{2.39}$$

The rates $q^{n+1}_{o,i,j,k}$ and $q^{n+1}_{w,i,j,k}$ are in units of STB/Day, and $q^{n+1}_{g,l,k}$ has units of SCF/Day. Here, layer k means the wellbore gridblock with z-direction gridblock index equal to k. The well index term $WI_{i,j,k}$ is the geometry part of productivity index and it is defined by

$$WI_{i,j,k} = \frac{0.00708 \triangle z_k \sqrt{k_{x,i,j,k} k_{y,i,j,k}}}{\ln(r_{o,l,k}/r_{w,l,k}) + s_{l,k}}, \tag{2.40}$$

and $r_{o,l,k}$ is defined

$$r_{o,l,k} = \frac{0.28073 \triangle x_i \sqrt{1 + \frac{k_{x,i,j,k}}{k_{y,i,j,k}} \left(\frac{\triangle y_j}{\triangle x_i}\right)^2}}{1 + \sqrt{k_{x,i,j,k}/k_{y,i,j,k}}}. \tag{2.41}$$

Here, $r_{w,l,k}$ is the wellbore radius of the well l at layer k and $s_{l,k}$ is the skin factor for well l at layer k.

The complete system of equations can formally be written as

$$f^{n+1} = f(y^{n+1}, y^n, m) = \begin{bmatrix} f_{1,1}^{n+1} \\ f_{o,1}^{n+1} \\ f_{g,1}^{n+1} \\ f_{1,2}^{n+1} \\ \vdots \\ f_{g,N}^{n+1} \\ f_{wf,1}^{n+1} \\ \vdots \\ f_{wf,N_w}^{n+1} \end{bmatrix} = 0, \tag{2.42}$$

where

$$m = [m_1, m_2, \cdots, m_{N_m}]^T, \tag{2.43}$$

and

$$y^{n+1} = [p_1^{n+1}, S_{o,1}^{n+1}, x_1^{n+1}, p_2^{n+1}, \cdots, p_i^{n+1}, S_{o,i}^{n+1}, x_i^{n+1}, \cdots, x_N^{n+1}, p_{wf,1}^{n+1}, \cdots, p_{wf,N_w}^{n+1}]^T, \tag{2.44}$$

where

$$x_i^{n+1} = \begin{cases} S_{g,i}^{n+1} & \text{for} \quad S_{g,i} > 0 \\ R_{s,i}^{n+1} & \text{for} \quad S_{g,i} = 0 \end{cases} \tag{2.45}$$

Eq. 2.42 is solved by the Newton-Raphson method which can be written as

$$J^{n+1,k} \delta y^{n+1,k+1} = -f^{n+1,k} \tag{2.46}$$

$$y^{n+1,k+1} = y^{n+1,k} + \delta y^{n+1,k+1}, \tag{2.47}$$

where k is the Newton-Raphson iteration index, n is the time step index and

$$J^{n+1,k} = \left[\nabla_{y^{n+1}} (f^{n+1})^T\right]_{y^{n+1,k}}^T, \tag{2.48}$$

is the Jacobian matrix evaluated at $y^{n+1,k}$, which represents the kth approximation for y^{n+1}. The initial guess for y^{n+1} is chosen as the solution at the previous time step, i.e.,

$$y^{n+1,0} = y^n. \tag{2.49}$$

2.3.2 Adjoint Equations

We define a general scalar function by

$$\beta = \beta(y^1, ..., y^L, m), \tag{2.50}$$

where L corresponds to the last time step t^L at which one wishes to compute sensitivity coefficients. The objective is to compute the derivatives of β with respect to the model parameters m. We obtain an adjoint functional J by adjoining Eq. 2.42 to the function β:

$$J = \beta + \sum_{n=0}^{L} (\lambda^{n+1})^T f^{n+1}, \tag{2.51}$$

where λ^{n+1} is the vector of adjoint variables at time step $n+1$, and is given by

$$\lambda^{n+1} = \left[\lambda_1^{n+1}, \quad \lambda_2^{n+1}, \quad \ldots, \quad \lambda_{N_e}^{n+1}\right]^T. \tag{2.52}$$

Taking the total differential of Eq. 2.51, and doing some simple rearranging gives

$$\begin{aligned} dJ &= d\beta + \sum_{n=0}^{L} \left\{ (\lambda^{n+1})^T [\nabla_{y^{n+1}} (f^{n+1})^T]^T dy^{n+1} + [\nabla_m (f^{n+1})^T]^T dm \right\} \\ &\quad + \sum_{n=0}^{L} (\lambda^{n+1})^T [\nabla_{y^n} (f^{n+1})^T]^T dy^n \\ &= d\beta + BT + \sum_{n=1}^{L} \{ [(\lambda^n)^T [\nabla_{y^n} (f^n)^T]^T \\ &\quad + (\lambda^{n+1})^T [\nabla_{y^n} (f^{n+1})^T]^T] dy^n + (\lambda^n)^T [\nabla_m (f^n)^T]^T dm \}, \end{aligned} \tag{2.53}$$

where

$$BT = (\lambda^{L+1})^T \left\{ [\nabla_{y^{L+1}} (f^{L+1})^T]^T dy^{L+1} + [\nabla_m (f^{L+1})^T]^T dm \right\} + (\lambda^1)^T \left[\nabla_{y^0} (f^1)^T\right]^T dy^0. \tag{2.54}$$

The total differential of β can be written as

$$d\beta = \sum_{n=1}^{L} [\nabla_{y^n} \beta]^T dy^n + [\nabla_m \beta]^T dm. \tag{2.55}$$

The initial conditions are fixed, so

$$dy^0 = 0. \tag{2.56}$$

Choosing

$$\lambda^{L+1} = 0, \tag{2.57}$$

it follows that $BT = 0$. Using this result and Eq. 2.55 in Eq. 2.53 and rearranging the resulting equation gives

$$\begin{aligned} dJ = \sum_{n=1}^{L} \Big[\Big\{ (\lambda^n)^T [\nabla_{y^n} (f^n)^T]^T + (\lambda^{n+1})^T [\nabla_{y^n} (f^{n+1})^T]^T \\ + [\nabla_{y^n} \beta]^T \Big\} dy^n \Big] + \Big\{ [\nabla_m \beta]^T + \sum_{n=1}^{N} (\lambda^n)^T [\nabla_m (f^n)^T]^T \Big\} dm. \end{aligned} \tag{2.58}$$

To obtain the adjoint system, the coefficients multiplying dy^n in Eq. 2.58 are set equal to zero; i.e., we require that the adjoint variables satisfy

$$(\lambda^n)^T [\nabla_{y^n} (f^n)^T]^T + (\lambda^{n+1})^T [\nabla_{y^n} (f^{n+1})^T]^T + [\nabla_{y^n} \beta]^T = 0. \tag{2.59}$$

Taking the transpose of Eq. 2.59, gives the adjoint system

$$\boxed{\big[\nabla_{y^n} (f^n)^T\big] \lambda^n = -\big[\nabla_{y^n} (f^{n+1})^T\big] \lambda^{n+1} - \nabla_{y^n} \beta.} \tag{2.60}$$

where

$$\nabla_{y^n} [f^n]^T = \begin{bmatrix} \frac{\partial f^n_{1,1}}{\partial p^n_1} & \frac{\partial f^n_{w,1}}{\partial p^n_1} & \cdots & \frac{\partial f^n_{g,N}}{\partial p^n_1} & \frac{\partial f^n_{wf,1}}{\partial p^n_1} & \cdots & \frac{\partial f^n_{wf,N_w}}{\partial p^n_1} \\ \frac{\partial f^n_{1,1}}{\partial S^n_{w,1}} & \frac{\partial f^n_{w,1}}{\partial S^n_{w,1}} & \cdots & \frac{\partial f^n_{g,N}}{\partial S^n_{w,1}} & \frac{\partial f^n_{wf,1}}{\partial S^n_{w,1}} & \cdots & \frac{\partial f^n_{wf,N_w}}{\partial S^n_{w,1}} \\ \frac{\partial f^n_{1,1}}{\partial S^n_{g,1}} & \frac{\partial f^n_{w,1}}{\partial S^n_{g,1}} & \cdots & \frac{\partial f^n_{g,N}}{\partial S^n_{g,N}} & \frac{\partial f^n_{wf,1}}{\partial S^n_{g,1}} & \cdots & \frac{\partial f^n_{wf,N_w}}{\partial S^n_{g,1}} \\ \frac{\partial f^n_{1,1}}{\partial p^n_2} & \frac{\partial f^n_{w,1}}{\partial p^n_2} & \cdots & \frac{\partial f^n_{g,N}}{\partial p^n_2} & \frac{\partial f^n_{wf,1}}{\partial p^n_2} & \cdots & \frac{\partial f^n_{wf,N_w}}{\partial p^n_2} \\ \vdots & \vdots & \cdots & \vdots & \vdots & \cdots & \vdots \\ \frac{\partial f^n_{1,1}}{\partial S^n_{g,N}} & \frac{\partial f^n_{w,1}}{\partial S^n_{g,N}} & \cdots & \frac{\partial f^n_{g,N}}{\partial S^n_{g,N}} & \frac{\partial f^n_{wf,1}}{\partial S^n_{g,N}} & \cdots & \frac{\partial f^n_{wf,N_w}}{\partial S^n_{g,N}} \\ \frac{\partial f^n_{1,1}}{\partial p^n_{wf,1}} & \frac{\partial f^n_{w,1}}{\partial p^n_{wf,1}} & \cdots & \frac{\partial f^n_{g,N}}{\partial p^n_{wf,1}} & \frac{\partial f^n_{wf,1}}{\partial p^n_{wf,1}} & \cdots & \frac{\partial f^n_{wf,N_w}}{\partial p^n_{wf,1}} \\ \vdots & \vdots & \cdots & \vdots & \vdots & \cdots & \vdots \\ \frac{\partial f^n_{1,1}}{\partial p^n_{wf,N_w}} & \frac{\partial f^n_{w,1}}{\partial p^n_{wf,N_w}} & \cdots & \frac{\partial f^n_{g,N}}{\partial p^n_{wf,N_w}} & \frac{\partial f^n_{wf,1}}{\partial p^n_{wf,N_w}} & \cdots & \frac{\partial f^n_{wf,N_w}}{\partial p^n_{wf,N_w}} \end{bmatrix}, \tag{2.61}$$

where $f_{1,i}$ is given by Eq. 2.30 and

$$\nabla_{y^n}[f^{n+1}]^T = \begin{bmatrix} \frac{\partial f_{1,1}^{n+1}}{\partial p_1^n} & \frac{\partial f_{w,1}^{n+1}}{\partial p_1^n} & \cdots & \frac{\partial f_{g,N}^{n+1}}{\partial p_1^n} & \frac{\partial f_{wf,1}^{n+1}}{\partial p_1^n} & \cdots & \frac{\partial f_{wf,N_w}^{n+1}}{\partial p_1^n} \\ \frac{\partial f_{1,1}^{n+1}}{\partial S_{w,1}^n} & \frac{\partial f_{w,1}^{n+1}}{\partial S_{w,1}^n} & \cdots & \frac{\partial f_{g,N}^{n+1}}{\partial S_{w,1}^n} & \frac{\partial f_{wf,1}^{n+1}}{\partial S_{w,1}^n} & \cdots & \frac{\partial f_{wf,N_w}^{n+1}}{\partial S_{w,1}^n} \\ \frac{\partial f_{1,1}^{n+1}}{\partial S_{g,1}^n} & \frac{\partial f_{w,1}^{n+1}}{\partial S_{g,1}^n} & \cdots & \frac{\partial f_{g,N}^{n+1}}{\partial S_{g,N}^n} & \frac{\partial f_{wf,1}^{n+1}}{\partial S_{g,1}^n} & \cdots & \frac{\partial f_{wf,N_w}^{n+1}}{\partial S_{g,1}^n} \\ \frac{\partial f_{1,1}^{n+1}}{\partial p_2^n} & \frac{\partial f_{w,1}^{n+1}}{\partial p_2^n} & \cdots & \frac{\partial f_{g,N}^{n+1}}{\partial p_2^n} & \frac{\partial f_{wf,1}^{n+1}}{\partial p_2^n} & \cdots & \frac{\partial f_{wf,N_w}^{n+1}}{\partial p_2^n} \\ \vdots & \vdots & \cdots & \vdots & \vdots & \cdots & \vdots \\ \frac{\partial f_{1,1}^{n+1}}{\partial S_{g,N}^n} & \frac{\partial f_{w,1}^{n+1}}{\partial S_{g,N}^n} & \cdots & \frac{\partial f_{g,N}^{n+1}}{\partial S_{g,N}^n} & \frac{\partial f_{wf,1}^{n+1}}{\partial S_{g,N}^n} & \cdots & \frac{\partial f_{wf,N_w}^{n+1}}{\partial S_{g,N}^n} \\ \frac{\partial f_{1,1}^{n+1}}{\partial p_{wf,1}^n} & \frac{\partial f_{w,1}^{n+1}}{\partial p_{wf,1}^n} & \cdots & \frac{\partial f_{g,N}^{n+1}}{\partial p_{wf,1}^n} & \frac{\partial f_{wf,1}^{n+1}}{\partial p_{wf,1}^n} & \cdots & \frac{\partial f_{wf,N_w}^{n+1}}{\partial p_{wf,1}^n} \\ \vdots & \vdots & \cdots & \vdots & \vdots & \cdots & \vdots \\ \frac{\partial f_{1,1}^{n+1}}{\partial p_{wf,N_w}^n} & \frac{\partial f_{w,1}^{n+1}}{\partial p_{wf,N_w}^n} & \cdots & \frac{\partial f_{g,N}^{n+1}}{\partial p_{wf,N_w}^n} & \frac{\partial f_{wf,1}^{n+1}}{\partial p_{wf,N_w}^n} & \cdots & \frac{\partial f_{wf,N_w}^{n+1}}{\partial p_{wf,N_w}^n} \end{bmatrix}, \tag{2.62}$$

and

$$\nabla_{y^n}\beta = \left[\frac{\partial\beta}{\partial p_1^n}, \frac{\partial\beta}{\partial S_{w,1}^n}, \frac{\partial\beta}{\partial S_{g,1}^n}, \frac{\partial\beta}{\partial p_2^n}, \cdots, \frac{\partial\beta}{\partial S_{g,N}^n}, \frac{\partial\beta}{\partial p_{wf,1}^n}, \cdots, \frac{\partial\beta}{\partial p_{wf,N_w}^n}\right]^T. \tag{2.63}$$

Note that when we set up the adjoint system, we use the water equation f_w, instead of the oil equation f_o as in CLASS, as the second equation in order to use the previous code developed by Ruijian Li without modifying it too much. Our results indicate that using f_w instead of f_o as the second equation does not affect the accuracy of the adjoint solutions. When we construct the adjoint system, the entries of the y vector are always p, S_o, S_g and p_{wf}, i.e.,

$$y = [p_1, S_{o,1}, S_{g,1}, p_2, \cdots, p_i, S_{o,i}, S_{g,i}, \cdots, S_{g,N}, p_{wf,1}, \cdots, p_{wf,N_w}], \tag{2.64}$$

whereas in the forward simulator, Eq. 2.43 is used. Our results indicate that this does not affect the accuracy of the adjoint solutions.

Eq. 2.60 with initial condition 2.57 is solved backwards in time for $n = L, L-1, \ldots, 1$. Note that the forward simulation equation is solved forward in time. Also note that the coefficients in Eq. 2.60 are independent of the adjoint variable λ, which means that the adjoint equation is linear. Therefore, solving the adjoint system is cheaper in terms of the computation cost than solving the forward simulation equation which is nonlinear. In the above equations, $\nabla_{y^n}(f^n)^T$ and $\nabla_{y^n}(f^{n+1})^T$ are $N_e \times N_e$ matrices, and $\nabla_{y^n}\beta$ is an N_e-dimensional column vector.

The matrix given by Eq. 2.62 is a diagonal band matrix which is only related to the accumulation terms in the reservoir simulation equations. Note that the coefficient matrix $(\nabla_{y^n}(f^n)^T)$ (Eq. 2.61) in the adjoint system is simply the transpose of the Jacobian matrix

of Eq. 2.48 evaluated at y^n when the equations and primary variables used to construct adjoint system are the same as used in the forward equations. As the adjoint system is solved backwards in time, information needed in these matrices (Eqs. 2.61 and 2.62) must be saved from the forward simulation run. In our code, we write all these primary variables to disk to save memory. For details on these equations for computing the derivatives $\nabla_{y^n}(f^n)^T$, $\nabla_{y^n}(f^{n+1})^T$, and $\nabla_{y^n}\beta$ in the adjoint equation, Eq. 2.60, see Li (2001).

As a summary, the adjoint system has the following properties:

(i) the adjoint system is solved backward in time;

(ii) the adjoint system is linear;

(iii) the coefficient matrix in the adjoint system is the transpose of the Jacobian matrix used for solving the forward equations only if the adjoint system is fully consistent with the forward equation, i.e., in each gridblock the same equations and primary variables are used to construct the adjoint system and the flow equation system.

Considering J as a function of m , we can write its total differential as

$$dJ = (\nabla_m J)^T dm. \tag{2.65}$$

By comparing Eq. 2.58 and Eq. 2.65, it follows that the desired sensitivity coefficients for J, or equivalently, β, are given by

$$\boxed{\nabla_m J = \nabla_m \beta + \sum_{n=1}^{L} [\nabla_m (f^n)^T](\lambda^n),} \tag{2.66}$$

where

$$\nabla_m [f^n]^T = \begin{bmatrix} \frac{\partial f^n_{1,1}}{\partial m_1} & \frac{\partial f^n_{w,1}}{\partial m_1} & \frac{\partial f^n_{g,1}}{\partial m_1} & \frac{\partial f^n_{1,2}}{\partial m_1} & \cdots & \frac{\partial f^n_{g,N}}{\partial m_1} & \frac{\partial f^n_{wf,1}}{\partial m_1} & \cdots & \frac{\partial f^n_{wf,N_w}}{\partial m_1} \\ \frac{\partial f^n_{1,1}}{\partial m_2} & \frac{\partial f^n_{w,1}}{\partial m_2} & \frac{\partial f^n_{g,1}}{\partial m_2} & \frac{\partial f^n_{1,2}}{\partial m_2} & \cdots & \frac{\partial f^n_{g,N}}{\partial m_2} & \frac{\partial f^n_{wf,1}}{\partial m_2} & \cdots & \frac{\partial f^n_{wf,N_w}}{\partial m_2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial f^n_{1,1}}{\partial m_{N_m}} & \frac{\partial f^n_{w,1}}{\partial m_{N_m}} & \frac{\partial f^n_{g,1}}{\partial m_{N_m}} & \frac{\partial f^n_{1,2}}{\partial m_{N_m}} & \cdots & \frac{\partial f^n_{g,N}}{\partial m_{N_m}} & \frac{\partial f^n_{wf,1}}{\partial m_{N_m}} & \cdots & \frac{\partial f^n_{wf,N_w}}{\partial m_{N_m}} \end{bmatrix}, \tag{2.67}$$

and

$$\nabla_m \beta = \left[\frac{\partial \beta}{\partial m_1}, \frac{\partial \beta}{\partial m_2}, \cdots, \frac{\partial \beta}{\partial m_{N_m}} \right]^T. \tag{2.68}$$

The matrix $\nabla_m [f^n]^T$ is an $N_m \times N_e$ sparse matrix and $\nabla_m \beta$ is an N_m-dimensional column vector. In Eq. 2.66, the gradient $\nabla_m \beta$ involves the partial derivatives of β with respect to the

model parameters. If the jth model parameter does not explicitly appear in the expression for β, then $\partial\beta/\partial m_j = 0$. For example, if $\beta = p_{wf}^n$, then we set $\nabla_m \beta = 0$ in Eq. 2.66.

To apply a conjugate gradient (Makhlouf et al., 1993) or variable metric method (Yang and Watson, 1988), we need only compute the gradient of the objective function and this can be done by setting $\beta = O(m)$ in the adjoint procedure. In this case, one only needs to solve the adjoint system Eq. 2.60 once and substitute the resulting adjoint solutions to Eq. 2.66 to obtain the gradient.

To apply the adjoint method to calculate the sensitivity of the variable β to model parameters m, one needs to solve the adjoint system equation Eq. 2.60 to obtain the adjoint variable λ, and then use Eq. 2.66 to calculate sensitivity coefficients. If we consider permeabilities (k_x, k_y and k_z) and porosities (ϕ) in each individual gridblock, i.e.,

$$m_{k_x} = k_x = [\ k_{x,1},\ \ k_{x,2},\ \ \cdots,\ \ k_{x,N}\ ]^T, \tag{2.69}$$

$$m_{k_y} = k_y = [\ k_{y,1},\ \ k_{y,2},\ \ \cdots,\ \ k_{y,N}\ ]^T, \tag{2.70}$$

$$m_{k_z} = k_z = [\ k_{z,1},\ \ k_{z,2},\ \ \cdots,\ \ k_{z,N}\ ]^T, \tag{2.71}$$

and

$$m_\phi = \phi = [\ \phi_1,\ \ \phi_2,\ \ \cdots,\ \ \phi_N\ ]^T, \tag{2.72}$$

then from Eq. 2.66, the equations to calculate the derivatives with respect to k_x, k_y, k_z and ϕ are given by

$$\nabla_{k_x} J = \nabla_{k_x}\beta + \sum_{n=1}^{L} [\nabla_{k_x}(f^n)^T](\lambda^n), \tag{2.73}$$

$$\nabla_{k_y} J = \nabla_{k_y}\beta + \sum_{n=1}^{L} [\nabla_{k_y}(f^n)^T](\lambda^n), \tag{2.74}$$

$$\nabla_{k_z} J = \nabla_{k_z}\beta + \sum_{n=1}^{L} [\nabla_{k_z}(f^n)^T](\lambda^n), \tag{2.75}$$

and

$$\nabla_{\phi} J = \nabla_{\phi}\beta + \sum_{n=1}^{L} [\nabla_{\phi}(f^n)^T](\lambda^n), \tag{2.76}$$

where β is p_{wf}, GOR, WOR at some specified time step L, the whole data mismatch part of the objective function $O_d(m)$ or any other terms for which we wish to calculate sensitivities.

In order to calculate the gradient of the objective function, we consider β as the whole data mismatch part of the objective function, i.e.,

$$\beta = O_d(m) = \frac{1}{2}(g(m) - d_{\text{obs}})^T C_D^{-1}(g(m) - d_{\text{obs}}), \tag{2.77}$$

or in the case of stochastic simulation of m,

$$\beta = O_d(m) = \frac{1}{2}(g(m) - d_{\rm uc})^T C_D^{-1}(g(m) - d_{\rm uc}). \tag{2.78}$$

Thus, we have

$$\begin{aligned}
\nabla_{y^n}\beta &= \nabla_{y^n}\{\frac{1}{2}(g(m) - d_{\rm obs})^T C_D^{-1}(g(m) - d_{\rm obs})\} \\
&= \Big[\nabla_{y^n}(g(m) - d_{\rm obs})^T\Big] C_D^{-1}(g(m) - d_{\rm obs}) \\
&= \nabla_{y^n}[g(m)]^T C_D^{-1}(g(m) - d_{\rm obs}).
\end{aligned} \tag{2.79}$$

In the case of β given by Eq. 2.78, the $d_{\rm obs}$ in Eq. 2.79 should be replaced by $d_{\rm uc}$. The matrix $\nabla_{y^n}[g(m)]^T$ is an $N_e \times N_d$ matrix and defined as

$$\nabla_{y^n}[g(m)]^T = \begin{bmatrix}
\frac{\partial g_1}{\partial p_1^n} & \frac{\partial g_2}{\partial p_1^n} & \cdots & \frac{\partial g_{N_d}}{\partial p_1^n} \\
\frac{\partial g_1}{\partial S_{w,1}^n} & \frac{\partial g_2}{\partial S_{w,1}^n} & \cdots & \frac{\partial g_{N_d}}{\partial S_{w,1}^n} \\
\frac{\partial g_1}{\partial S_{g,1}^n} & \frac{\partial g_2}{\partial S_{g,1}^n} & \cdots & \frac{\partial g_{N_d}}{\partial S_{g,N}^n} \\
\frac{\partial g_1}{\partial p_2^n} & \frac{\partial g_2}{\partial p_2^n} & \cdots & \frac{\partial g_{N_d}}{\partial p_2^n} \\
\vdots & \vdots & \cdots & \vdots \\
\frac{\partial g_1}{\partial S_{g,N}^n} & \frac{\partial g_2}{\partial S_{g,N}^n} & \cdots & \frac{\partial g_{N_d}}{\partial S_{g,N}^n} \\
\frac{\partial g_1}{\partial p_{wf,1}^n} & \frac{\partial g_2}{\partial p_{wf,1}^n} & \cdots & \frac{\partial g_{N_d}}{\partial p_{wf,1}^n} \\
\vdots & \vdots & \cdots & \vdots \\
\frac{\partial g_1}{\partial p_{wf,N_w}^n} & \frac{\partial g_2}{\partial p_{wf,N_w}^n} & \cdots & \frac{\partial g_{N_d}}{\partial p_{wf,N_w}^n}
\end{bmatrix}. \tag{2.80}$$

The entries of vector $g(m)$ represent production data. The vector may contain entries like p_{wf}, GOR and WOR or any combination of these three kinds of production data. Details for calculating each entry of matrix $\nabla_{y^n}[g(m)]^T$ can be found in Li (2001). It turns out many columns of this matrix are zero. Only the columns corresponding to data that are measured at time n are nonzero. After we evaluate the matrix $\nabla_{y^n}[g(m)]^T$, we multiply $C_D^{-1}(g(m) - d_{\rm obs})$ by this matrix to obtain $\nabla_{y^n}\beta$. Once we have $\nabla_{y^n}\beta$, we can apply Eq. 2.60 to compute the adjoint variables.

To apply Eq. 2.66 to compute the derivatives, we need to evaluate $\nabla_m\beta$ first. The vector $\nabla_m\beta$ is given by

$$\begin{aligned}
\nabla_m\beta &= \nabla_m O_d(m) \\
&= \nabla_m\big\{\frac{1}{2}(g(m) - d_{\rm obs})^T C_D^{-1}(g(m) - d_{\rm obs})\big\} \\
&= \big[\nabla_m(g(m) - d_{\rm obs})^T\big] C_D^{-1}(g(m) - d_{\rm obs}) \\
&= \nabla_m[g(m)]^T C_D^{-1}(g(m) - d_{\rm obs}).
\end{aligned} \tag{2.81}$$

In the case of β given by Eq. 2.78, the $d_{\rm obs}$ in Eq. 2.81 should be replaced by $d_{\rm uc}$. The matrix $\nabla_m[g(m)]^T$ is an $N_m \times N_d$ matrix and defined as

$$\nabla_m[g(m)]^T = \begin{bmatrix} \frac{\partial g_1}{\partial m_1} & \frac{\partial g_2}{\partial m_1} & \cdots & \frac{\partial g_{N_d}}{\partial m_1} \\ \frac{\partial g_1}{\partial m_2} & \frac{\partial g_2}{\partial m_2} & \cdots & \frac{\partial g_{N_d}}{\partial m_2} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{\partial g_1}{\partial m_{N_m}} & \frac{\partial g_2}{\partial m_{N_m}} & \cdots & \frac{\partial g_{N_d}}{\partial m_{N_m}} \end{bmatrix}. \tag{2.82}$$

The vector $g(m)$ is the calculated production data vector. For the history matching problems considered here, an entry of g will correspond to p_{wf}, GOR or WOR. The formulas for calculation of elements in the matrix $\nabla_m[g(m)]^T$ when g represents production data are straightforward and can be found in Li (2001). After we compute $\nabla_m\beta$, we can use Eq. 2.66 to compute the derivatives of the objective function with respect to model parameters, i.e., the gradient of the objective function.

2.4 Results, Example History Match

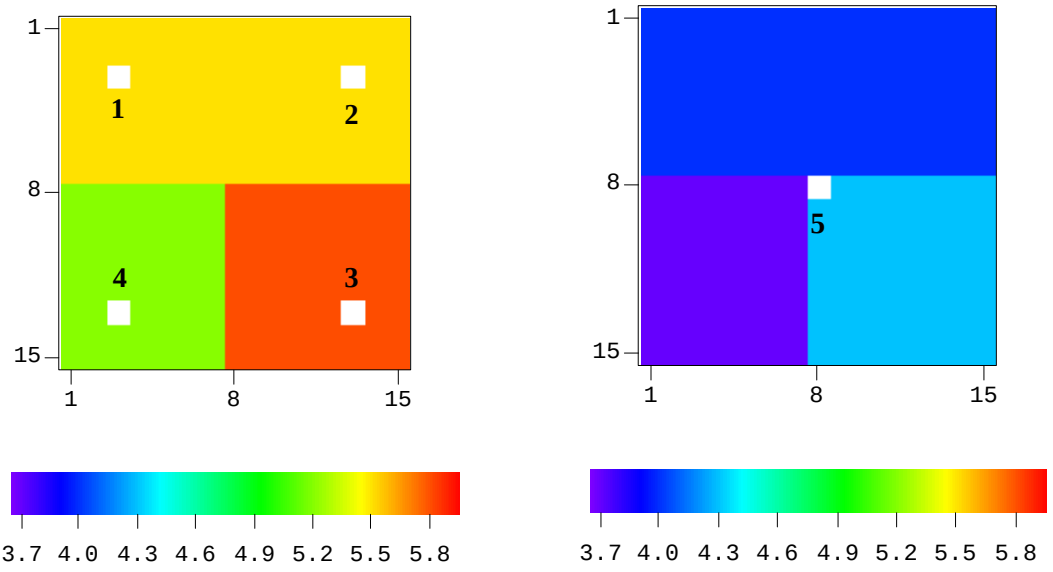
For relatively small problems, it is appropriate to use the Modified Levenberg-Marquardt (MLM) algorithm to history match production. In this section, we illustrate the validity of the history matching procedure by considering a simple problem. In the next chapter we consider algorithms (LBFGS and nonlinear conjugate gradient) which do not require computation of all sensitivities and can thus be applied to history match large scale problems. The synthetic example presented here is one of several that were used to verify that the adjoint method gives sufficiently accurate sensitivity coefficients for use in the optimization algorithms. This was verified by comparing sensitivities computed with the adjoint methods with those computed with the finite-difference methods. (In some previous publications, the finite-difference method was referred to as the direct method, see, for example, Chu and Reynolds (1995) or He et al. (1997).) For all cases that we have considered, the adjoint method and finite-difference method gave results that agreed to two significant digits. Thus plots of the sensitivities obtained from the two methods are graphically indistinguishable and we do not present results.

2.4.1 True Reservoir

The areal extent of the reservoir is 600 feet by 600 feet and contains two layers. Layer 1 refers to the top layer and layer 2 refers to the bottom layer. The thickness of each layer is uniform

and equal to 30 feet. An uniform $15 \times 15 \times 2$ finite difference grid with $\Delta x = \Delta y = 40$ ft and $\Delta z = 30$ ft is used in all reservoir simulation runs.

Each layer of the true model consists of three permeability regions. Fig. 2.4 shows the distribution of values of horizontal log-permeability in layers 1 and 2. (The white gridblocks in Fig. 2.4(a) indicate gridblocks in which producing wells are completed.). For layer 1, $\ln(k) = 5.2$ ($k = 181$ md) in the lower left quadrant, $\ln(k) = 5.8$ ($k = 330$ md) in the lower right quadrant and $\ln(k) = 5.5$ ($k = 245$ md) in the upper half. For layer 2, $\ln(k) = 3.7$ ($k = 40$ md) in the lower left quadrant, $\ln(k) = 4.3$ ($k = 74$ md) in the lower right quadrant and $\ln(k) = 4.0$ ($k = 55$ md) in the upper half. Fig. 2.4(b) shows the distribution of values of horizontal log-permeability in layer 2. In this figure, the white gridblock shows the location of a water injection well. Water is injected only into layer 2.

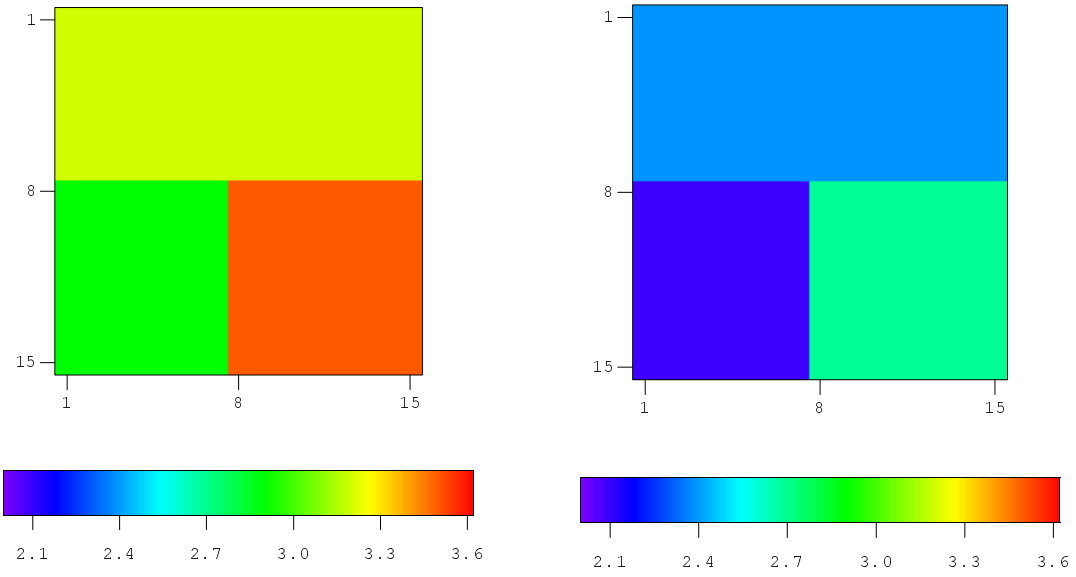


(a) Layer 1.

(b) Layer 2.

Figure 2.4: Horizontal log-permeability for true model and well completions.

In the truth case, vertical permeability is equal to one-tenth horizontal permeability in the top layer and is equal to two-tenths of horizontal permeability in the bottom layer. Thus, for layer 1, $\ln(k_z) = 2.9$ ($k_z = 18$ md) in the lower left quadrant, $\ln(k_z) = 3.5$ ($k_z = 33$ md) in the lower right quadrant and $\ln(k_z) = 3.2$ ($k_z = 24.5$ md) in the upper half. For layer 2, $\ln(k_z) = 2.1$ ($k_z = 8$ md) in the lower left quadrant, $\ln(k_z) = 2.7$ ($k_z = 15$ md) in the lower right quadrant and $\ln(k_z) = 2.4$ ($k_z = 11$) in the upper half. Figs. 2.5(a) and 2.5(b),



(a) Layer 1.

(b) Layer 2.

Figure 2.5: Vertical log-permeability for true model.

respectively, show the distribution of values of vertical log-permeability in layers 1 and 2.

In this example, reservoir porosity is assumed to be uniform with $\phi = 0.22$. Even though the truth case consists of zones, permeabilities of each grid cell are estimated. The simplicity of the example chosen allows one to easily visualize the quality of estimates. Some of the pertinent information is summarized in Table 2.2 where the first row gives the zone and the layer number. Throughout, Zone 1 refers to the upper half of the system depicted in Figs. 2.4 and 2.5, Zone 2 refers to the lower right quadrant and Zone 3 refers to the lower left quadrant. Var $\ln(k)$ denotes the variance and $E[\ln(k)]$ denotes its mean used to generate the prior model.

Capillary effects are not included. Initial reservoir pressure at the depth corresponding to the center of the top layer is specified as $p_i = 4500$ psi. Initial bubble point pressure is set equal to 4515 psi. Initial water saturation is equal to irreducible water saturation which is equal to 0.2. Initial oil saturation is $S_{o,i} = 0.8$. As mentioned previously, Stone's second model is used to calculate the relative permeability to oil from the two sets of two-phase relative permeability curves. For the two-phase oil-water system, residual oil saturation is 0.2. For the two-phase gas-oil system, residual oil saturation is 0.3 and critical gas saturation is equal to 0.05.

The reservoir contains four producing wells which are completed only in the top layer. The white gridblocks in Fig. 2.4 show the location of the gridblocks that contain well completions. Four producing wells, referred to as Wells 1 through 4, are located near the four corners of the reservoir. Well 1 is completed in gridblock $(3, 3, 1)$, well 2 is completed is gridblock $(13, 3, 1)$, well 3 is completed in gridblock $(13, 13, 1)$ and well 4 is completed in gridblock $(3, 13, 1)$. Each of the wells is produced at total flow rate of 350 RB/D. A single water injection well (well 5) completed in gridblock $(8, 8, 2)$ is used to inject water into the bottom layer at a rate of 1100 STB/D. At initial reservoir pressure, this is equivalent to an injection rate of 1107 RB/D. Note that the injection well (well 5) is located in the lower right quadrant of the bottom layer, which corresponds to the highest permeability zone ($\ln(k) = 4.3$) of the bottom layer. The true skin factors at wells 1 through 5, respectively, are specified as 3.0, 4.0, 5.0, 2.0 and 0.0. By running the simulator using data from the truth case as input, we obtained synthetic pressure, GOR and WOR data at the producing wells and synthetic wellbore pressure data at the injection well.

For the example considered, wells produce from only one gridblock so the water-oil ratio and gas-oil ratio, respectively, are given by

$$WOR = \frac{k_{rw}\mu_o B_o}{k_{ro}\mu_w B_w}, \tag{2.83}$$

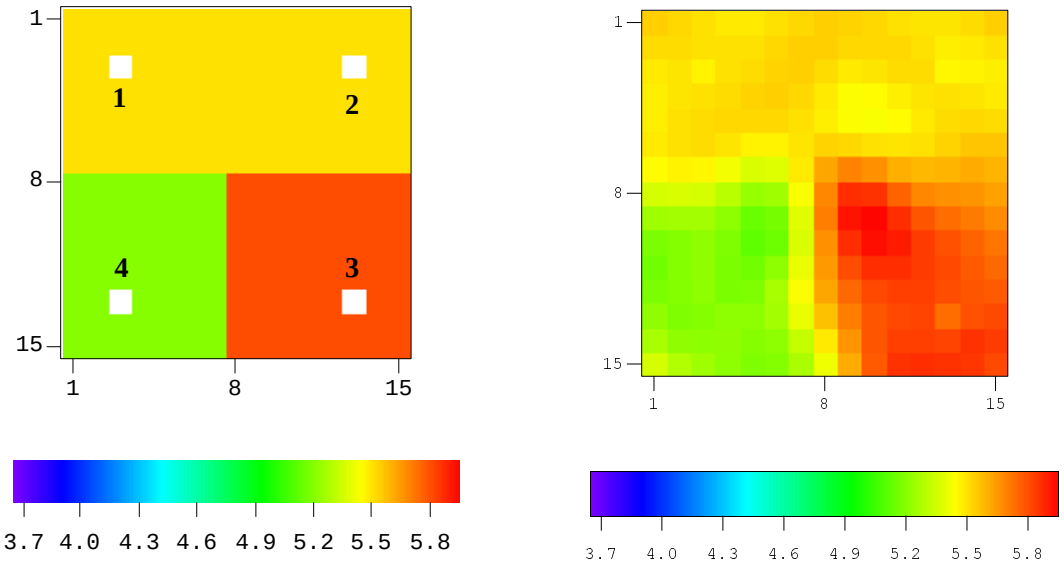
and

$$GOR = R_s + \frac{k_{rg}\mu_o B_o}{k_{ro}\mu_g B_g}. \tag{2.84}$$

These equations are applied at each producing well and are evaluated using the pressure and saturations of the gridblock containing the well.

Zone/layer	1/1	2/1	3/1	1/2	2/2	3/2
true $ln(k)$	5.5	5.8	5.2	4.0	4.3	3.7
true k, md	245	330	181	55	74	40
true $ln(k_z)$	5.5	5.8	5.2	4.0	4.3	3.7
true k_z, md	24.5	33	18	11	15	8
Var $[ln(k)]$	0.5	0.5	0.5	0.5	0.5	0.5
$E[ln(k)]$	5.5	5.5	5.5	4.0	4.0	4.0
Var $[k_z]$, md	0.5	0.5	0.5	0.5	0.5	0.5
$E[ln(k_z)]$	3.2	3.2	3.2	2.4	2.4	2.4

Table 2.2: True log-permeabilities and statistical parameters for example problem.



(a) True, layer 1.

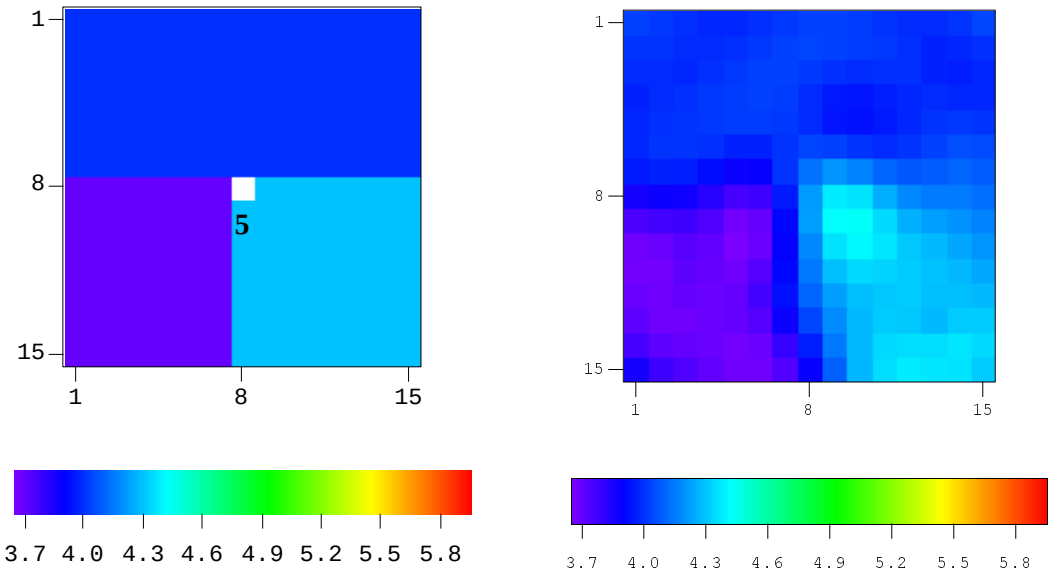
(b) MAP, layer 1.

Figure 2.6: MAP estimate conditioned to p_{wf}, WOR and GOR data versus true horizontal log-permeability, layer 1.

At each producing well, we selected 10 WOR data, 10 GOR data and 10 pressure data to use as conditioning data. At the injection well, we selected 10 pressure data to use as conditioning data. The data are uniformly distributed throughout the 300 days time period with the earliest time conditioning data corresponding to $t = 30$ days. We assumed pressure measurement errors to be independent, identically-distributed, normal random variables with mean zero and variance equal to 1 psi^2. GOR measurement errors were modeled similarly except the variance was set equal to 100 (scf/STB)^2. The variances of WOR measurement errors were specified by Eq. 2.6 with $\sigma_{w,\min} = 2.0$ STB/STB, $\epsilon_o = 0.01$ and $\epsilon_w = 0.02$.

For the top layer, the prior means for $\ln(k)$ and $\ln(k_z)$, respectively, were specified as 5.5 and 3.2 with the variances of both random variable equal to 0.5. For the second layer, the prior means for $\ln(k)$ and $\ln(k_z)$, respectively, were specified as 4.0 and 2.4 with the variances of both random variable equal to 0.5. This information as well as permeability values for the true model was presented previously in Table 2.2.

The same semivariogram was used for each of the four log-permeability fields. The semivariogram is an isotropic spherical semivariogram with range equal to 160 ft and sill equal to 0.5. As the areal dimensions of simulation gridblocks are $\Delta x = \Delta y = 40$ feet, if



(a) True, layer 2.

(b) MAP, layer 2.

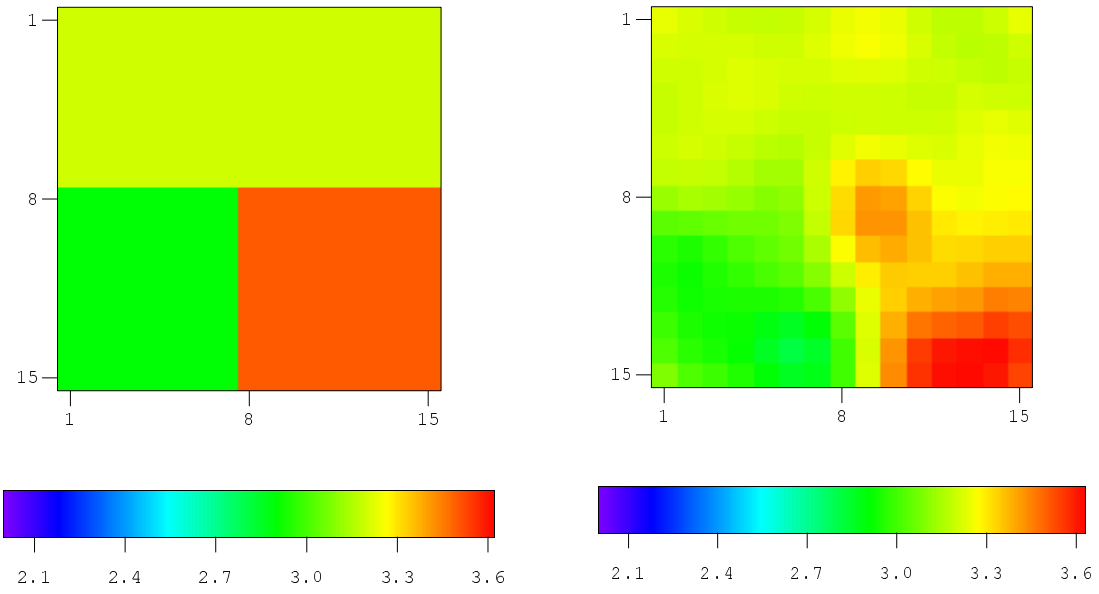
Figure 2.7: MAP estimate conditioned to p_{wf}, WOR and GOR data versus true horizontal log-permeability, layer 2.

the distance between the centers of two gridblocks at the same elevation is greater than or equal to $4\Delta x$, the two gridblock permeabilities are uncorrelated. There is no correlation between layer 1 permeabilities and layer 2 permeabilities. In each layer, the correlation coefficient between $\ln(k)$ and $\ln(k_z)$ is set equal to 0.7. Each well skin factor is modeled as an independent random variable with mean equal to 2.0 and variance equal to 25.

2.4.2 The MAP Estimate

As discussed previously, the MAP estimate was generated using the Levenberg-Marquardt algorithm to minimize the objective function of Eq. 2.10. The vector of prior means was used as the initial guess. Note even though each layer actually consists of three zones, horizontal and vertical log-permeability are estimated at each gridblock. This simple model is used only because it makes it easy to visualize the quality of the estimate. We consider results obtained by history matching all data, pressure, WOR and GOR.

The right plot in Fig. 2.6 shows the MAP estimate of layer 1 horizontal log-permeability field obtained by history-matching and the left plot shows the true horizontal log-permeability

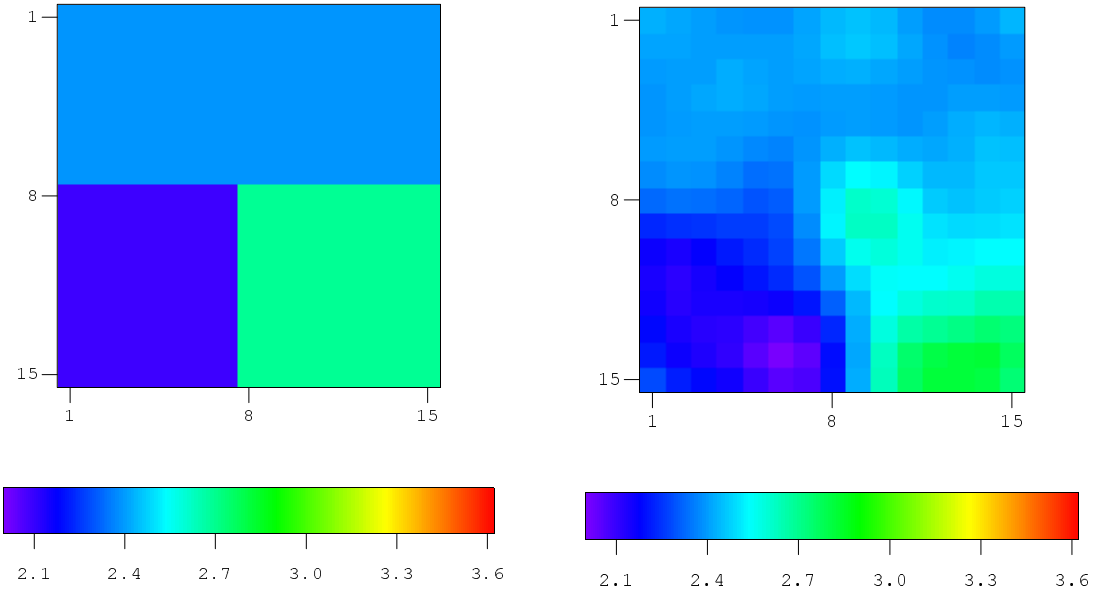


(a) True, layer 1.

(b) MAP, layer 1.

Figure 2.8: MAP estimate conditioned to p_{wf}, WOR and GOR data versus true vertical log-permeability, layer 1.

field. As the true field is discontinuous across the boundary between zones but the prior model assumes a correlation length of about 4 gridblocks, the history matched model which attempts to honor both the assumed prior and the production data is expected to be somewhat smoother across boundaries between zones than the true horizontal log-permeability field. Despite the inconsistency between the prior geostatistical model and the truth, the history matched model for $\ln(k)$ agrees remarkably well with the true horizontal log-permeability field. Fig. 2.7 shows similarly good agreement between the truth and the MAP estimate for the layer 2 log-permeability field. As shown in Figs. 2.8 and 2.9, we get slightly poorer agreement between the true and history-matched vertical log-permeability fields. As production data generated with the simulator depends only on the harmonic averaged vertical permeabilities at the boundary between layers 1 and 2 and not at individual layer vertical permeabilities, in truth, production data can only resolve these harmonic averages. However, as the prior model incorporates the prior mean, the tendency of the history matching procedure is to generate MAP estimates that are as close as possible to the prior means but still yield a history match. Because of this, MAP estimate of vertical log-permeability is a reasonable approximation to the true vertical log-permeability field.



(a) True, layer 2.

(b) MAP, layer 2.

Figure 2.9: MAP estimate conditioned to p_{wf}, WOR and GOR data versus true vertical log-permeability, layer 2.

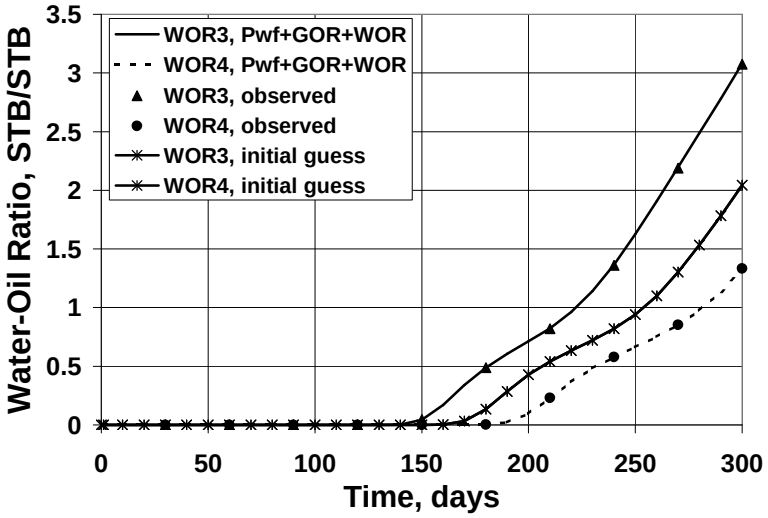


Figure 2.10: History match of WOR data at the producers.

Estimates of well skin factors obtained by matching various combinations of production data are shown in Table 2.3. Note the best estimates are obtained by history matching

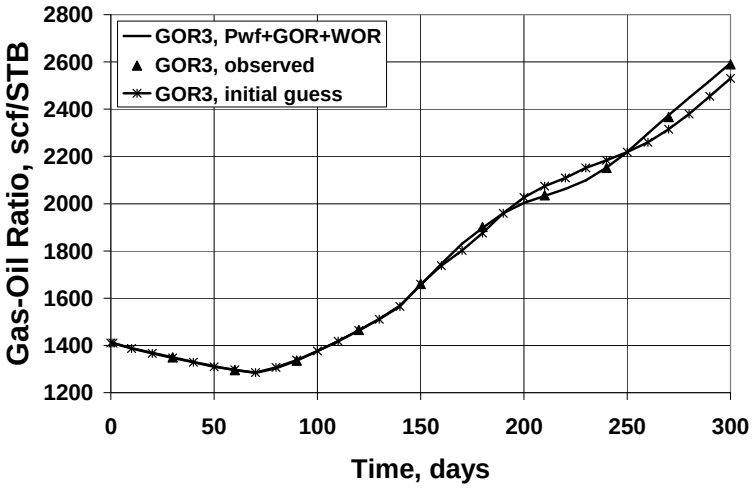


Figure 2.11: History match of GOR data at a producer.

pressure, WOR and GOR data, as was done to obtain the MAP estimates of log-permeability fields shown in Figs. 2.6(b), 2.7(b), 2.8(b) and 2.9(b).

Well No.	1	2	3	4	5
True skin factors	3.00	4.00	5.00	2.00	0.00
Initial guess	2.00	2.00	2.00	2.00	2.00
p_{wf}	2.60	3.30	2.97	2.61	-0.19
p_{wf}+WOR	2.67	3.39	3.18	2.22	0.24
p_{wf}+GOR	2.73	3.39	3.54	2.07	0.00
p_{wf}+GOR+WOR	2.83	3.65	4.37	1.96	-0.27

Table 2.3: The true and estimated skin factors.

The history match of the observed WOR data at two of the producing wells (Wells 3 and 4) is shown in Fig. 2.10. Here the solid triangular points denote observed data for well 3 and the solid curve that closely matches these data represents the WOR predicted with the MAP estimate of the permeability, Figs. 2.6 through 2.9. The solid circular data points and the dashed curve depict corresponding results for well 4. (Note we have WOR data during the period when the water-oil ratio is equal to zero.) The solid curve through asterisks represents the WOR predicted for all wells based on the initial guess for the permeability fields. In the initial guess, each layer is homogenous with all gridblock permeabilities set equal to the

appropriate prior mean so the initial guess gives the same WOR at all producers.

Solid circular and triangular data points represent the observed data used as conditioning data. Curves through the cross data points indicate the data predicted from the initial guess (m_{prior}) of the model parameters. (Some initial pressure mismatches exceeded 400 psi.) The predicted data shown is based on the model obtained by simultaneously matching pressure, WOR and GOR data. Matches of similar quality were obtained for all wells for all pressure, WOR and GOR observed data.

Chapter 3

RESULTS: ESTIMATION OF RELATIVE PERMEABILITY CURVES

3.1 Generation of Estimates

For the synthetic examples presented here, the reservoir is assumed to be a rectangular parallelepiped and the permeability field is assumed to be isotropic. The forward model used to calculate production data is a fully-implicit finite-difference simulator based on a block centered grid. Additional information on the simulator can be found in Li et al. (2001a). The model parameters to be estimated include the gridblock log-permeabilities and the parameters involved in the power law expressions for the relative permeability curves. All other rock and fluid properties are assumed to be known. Log-permeabilities, instead of absolute permeabilities, are used as parameters because it is assumed that absolute permeability is log-normal.

Throughout, m denotes the vector of model parameters (gridblock log-permeabilities and parameters defining relative permeability curves). In the Bayesian approach used here, m is considered to be random vector. We assume that m has a prior multivariate Gaussian distribution with covariance matrix C_M and prior mean m_{prior}. It is assumed that the model parameters defining the relative permeability functions are not correlated with each other or with the gridblock log-permeabilities. In practice, the prior means and covariance function for log-permeability will be obtained from a geostatistical model constructed from static data, e.g. well log, core and seismic data but not production data. The prior covariance

matrix and prior means for relative permeability parameters could be constructed from laboratory measurements of relative permeability curves, from lab data for analogue reservoirs, or relative permeability correlations.

As discussed in Chapter 2, the maximum a posteriori estimate of model parameters can be obtained by minimizing the following objective function, Tarantola (1987); Wu et al. (1999); Li et al. (2001a)

$$O(m) = \frac{1}{2}\Big[\big(m - m_{\text{prior}}\big)^T C_M^{-1}\big(m - m_{\text{prior}}\big) + \big(g(m) - d_{\text{obs}}\big)^T C_D^{-1}\big(g(m) - d_{\text{obs}}\big)\Big]. \qquad (3.1)$$

In the examples, considered here, the modified Levenberg-Marquardt algorithm originally presented in Bi (1999) is applied to minimize $O(m)$ with the Li et al. (2001b) implementation of the adjoint method used to calculate sensitivity coefficients.

3.2 Relative Permeability Models

Throughout, S_{wc} denotes irreducible water saturation, S_{gc} denotes critical gas saturation, S_{org} denotes the residual oil saturation for a two-phase, gas-oil system and S_{orw} denotes residual oil saturation for a two-phase, water-oil system. As is common, we define oil relative permeability under three-phase flow conditions as a combination of the two sets of two-phase relative permeability functions. In the simulator, and hence in the examples presented, the historically popular Stone's (Stone, 1973) Model II is applied. The form of this model applied here is the modification that is described in Aziz and Settari (1979). Our results, however, indicate this model has a peculiar characteristic that is troublesome.

As the examples presented are synthetic, simple analytical formulas are used for the relative permeability functions. Letting

$$S_{w,\max} = 1 - S_{orw}, \qquad (3.2)$$

the water relative permeability function is given by

$$k_{rw} = \begin{cases} 0 & \text{if } S_w \le S_{wc}, \\ k_{rwcw}\left(\frac{S_w - S_{wc}}{1 - S_{orw} - S_{wc}}\right)^{n_{rw}} & \text{if } S_{wc} \le S_w \le S_{w,\max} \\ k_{rwcw} & \text{if } S_w \ge S_{w,\max}. \end{cases} \qquad (3.3)$$

The endpoint relative permeability, k_{rwcw}, represents the maximum value of water relative permeability that can be obtained. The corresponding oil relative permeability function for

a two-phase oil-water system is given by

$$k_{row} = \begin{cases} k_{rocw} & \text{if } S_w \le S_{wc}, \\ k_{rocw}\left(\frac{1-S_{orw}-S_w}{1-S_{orw}-S_{wc}}\right)^{n_{row}} & \text{if } S_{wc} \le S_w \le S_{w,\max} \\ 0 & \text{if } S_w \ge S_{w,\max}. \end{cases} \tag{3.4}$$

We let k_{rog} denote the oil relative permeability for a two-phase flow oil-gas system. In the Stone Model II, it is assumed that the two-phase oil-gas relative permeabilities depend only on gas saturation and are measured in the presence of irreducible water saturation. Thus the endpoint values of oil relative permeability for a two-phase oil-water system and a two-phase oil-gas system are identical, i.e.,

$$k_{rog}(S_g = 0) = k_{row}(S_w = S_{wc}) = k_{rocw}. \tag{3.5}$$

The gas relative permeability function used in this work is given by

$$k_{rg} = \begin{cases} 0 & \text{if } S_g \le S_{gc}, \\ k_{rgcw}\left(\frac{S_g-S_{gc}}{1-S_{org}-S_{wc}-S_{gc}}\right)^{n_{rg}} & \text{if } S_{gc} \le S_g \le S_{g,\max} \\ k_{rgcw} & \text{if } S_g \ge S_{g,\max}, \end{cases} \tag{3.6}$$

where

$$S_{g,max} = 1 - S_{org} - S_{wc}. \tag{3.7}$$

Note k_{rgcw} denotes the maximum value of gas relative permeability and occurs when $S_w = S_{wc}$ and $S_o = S_{org}$. The oil relative permeability function for a two-phase oil-gas system is given by

$$k_{rog} = \begin{cases} k_{rocw} & \text{if } S_g \le S_{gc}, \\ k_{rocw}\left(\frac{1-S_{org}-S_{wc}-S_g}{1-S_{org}-S_{wc}-S_{gc}}\right)^{n_{rog}} & \text{if } S_{gc} \le S_g \le S_{g,\max} \\ 0 & \text{if } S_g \ge S_{g,\max}. \end{cases} \tag{3.8}$$

Equations 3.6 and 3.8 indicate that oil and gas relative permeabilities are a function of only S_g for a two-phase oil-gas system, whereas, the two-phase oil and water relative permeabilities depend only on water saturation.

The water and gas relative permeabilities of Eqs. 3.3 and 3.6 apply regardless of the number of phases present. If all three phases are mobile, the oil relative permeability function is given by

$$k_{ro} = k_{rocw}\Big(\Big[\frac{k_{row}}{k_{rocw}} + k_{rw}\Big]\Big[\frac{k_{rog}}{k_{rocw}} + k_{rg}\Big] - \Big[k_{rw} + k_{rg}\Big]\Big). \tag{3.9}$$

Equation 3.9 represents the modification of Stone's Model II as presented in Aziz and Settari (1979). This equation gives oil relative permeability as a function of S_w and S_g and applies for any values of saturations, subject to the proviso that k_{ro} is set equal to zero whenever the formula predicts a negative value. If $S_g = 0$, it is easy to show that Eq. 3.9 reduces to the oil relative permeability function of the two-phase oil-water system and if $S_w = S_{wc}$, then Eq. 3.9 reduces to the oil relative permeability function for the two-phase oil-gas system.

We also assume residual oil saturations, critical gas saturation and irreducible water saturation are known. Thus, in the general three-phase flow case, estimation of relative permeability curves requires estimation of the components of the vector

$$m_{\rm rel} = \left[k_{rwcw}, k_{rocw}, n_{rw}, n_{row}, k_{rgcw}, n_{rg}, n_{rog}\right]^T. \tag{3.10}$$

In general the components of the model m include gridblock log-pemeabilities and the components of $m_{\rm rel}$. If absolute permeability is assumed to be known, then $m = m_{\rm rel}$.

3.3 Comments on Estimation of Relative Permeabilities

The standard black-oil equations for three-phase flow involve only the product of absolute and relative permeability, i.e., kk_{rm} for $m = o, w, g$. Suppose we are given the true permeability field and true relative permeabilities as one plausible set of model parameters which match production data. If we generate a second set of model parameters by multiplying the absolute permeability field by any positive constant a and dividing the true relative permeabilities by a, then effective permeabilities as functions of saturations for the resulting set of model parameters will be identical to the true model. Moreover, the two models will predict identical production data when input into a reservoir simulator. Thus, it is clear that there is an infinite set of model parameters that will result in the same predictions of reservoir performance. In terms of the inverse problem of interest here, this means that we can not expect to generate accurate estimates of absolute and relative permeability by history matching observed production data. Any such estimate will have an infinite uncertainty attached to them unless the uncertainty is constrained by a prior model or by imposing a constraints on the permissible values of absolute permeability and relative permeabilities.

For the two sets of two-phase relative permeability curves, the effective permeabilities as functions of saturations do not change as long as kk_{rocw}, kk_{rgcw} and kk_{rwcw} are held constant. Thus, under two-phase flow conditions, production data will not allow one to construct a reliable estimate of k and the endpoint relative permeabilities. Quite interestingly, the

modified Stone's Model II three-phase oil relative permeability function has the strange characteristic that it is not possible to keep all three effective permeabilities constant if k is varied. To see this, we multiply Eq. 3.9 by absolute permeability and rearrange the resulting equation to obtain

$$kk_{ro} = kk_{rocw}\left(\left[\frac{kk_{row}}{kk_{rocw}} + \frac{kk_{rw}}{k}\right]\left[\frac{kk_{rog}}{kk_{rocw}} + \frac{kk_{rg}}{k}\right] - \left[\frac{kk_{rw}}{k} + \frac{kk_{rg}}{k}\right]\right). \tag{3.11}$$

If kk_{rocw}, kk_{rwcw} and kk_{rgcw} are held constant as k is varied, then all two-phase effective permeability functions remain unchanged, but Eq. 3.11 indicates that the three phase effective oil permeability function kk_{ro} varies with k. Thus, changes in the absolute permeability field will result in changes in predicted production data if three phases are mobile. Also note that if kk_{rocw} is held constant as k is varied, then both two phase flow effective oil permeabilities, kk_{row} and kk_{rog} will be unchanged, but the three-phase flow effective permeability function kk_{ro} will change unless effective gas and/or effective water permeability are altered. Thus, under three-phase flow conditioned it is not possible to alter absolute permeability while keeping all effective permeabilities constant. This means that changes in absolute permeability will result in changes in predicted reservoir performance. Thus, unlike the two-phase flow case, it may be possible to obtain reasonable estimates of both absolute permeability and relative permeability functions from three-phase flow production data. This result, however, is simply a peculiarity of Stone's Model II. For example, if one used the Dietrich and Bonder (1976) modification of Stone's model, then it is possible to obtain identical effective permeabilities under three-phase flow conditions for two different absolute permeability fields. The Dietrich-Bonder model, however, has other disadvantages; see Aziz and Settari (1979).

3.4 History Matching Examples

We consider two synthetic, two-dimensional three-phase flow problems. In the first example, the reservoir consists of three zones of uniform absolute permeability. In this case, we consider the estimation of relative permeability model parameters when absolute permeability is known and then consider the simultaneous estimation of absolute and relative permeabilities. In the second example, the true permeability field is an unconditional realization generated from a prior geostatistical model. In this case we generate a realization, rather than an estimate, of absolute and relative permeabilities. The information given in the following three paragraphs applies to all examples presented.

The reservoir simulation grid is $15 \times 15 \times 1$. The grid is uniform with areal dimensions given by $\Delta x = \Delta y = 12.2$ m (40 ft) and $\Delta z = 9.1$ m (30 ft). Throughout, gridblock (i, j) refers to the gridblock centered at (x_i, y_j, z_1). A water injection well (referred to as Well 5) is completed in gridblock $(8, 8)$. Water is injected through this well at a constant rate of 87.4 std m^3/d (550 STB/D) for all times. Producing wells are located in gridblocks $(3, 3)$ (Well 1), $(13, 3)$ (Well 2), $(13, 13)$ (Well 3) and $(3, 13)$ (Well 4). Each of these wells is produced at a constant total fluid rate of 34.98 m^3/d (220 RB/D). The synthetic production data used to generate estimates is generated by running the reservoir simulator with the true model. Observed production data corresponds to the pressure, water-oil ratio and producing gas-oil ratio output from the simulator at thirty days intervals, i.e., at $t = 30, 60, \ldots, 300$ days at each producing well. At the water injection well, the flowing wellbore pressure at these times is also recorded. Thus, 130 data points are available for estimating the model parameters by automatic history matching. In the examples, we consider the effect of using only one type of data. Initial reservoir pressure is $31,026$ kPa (4500 psi) and initial bubble-point pressure is $30,454$ kPa (4417 psi) so pressure in gridblocks containing producing wells drop below bubble-point pressure very shortly after the beginning of production. For all examples considered water breaks through at all the producing wells prior to the end of the 300 day production period. After water breaks through at a well, all three phases are mobile in the well's gridblock.

Throughout, the natural logarithm of absolute permeability is simply referred to as log-permeability and denoted by $\ln(k)$. To generate the prior model, we assumed that log-permeability is a stationary Gaussian random function with the associated covariance function defined from an isotropic spherical variogram (Journel and Huijbregts, 1978). The range of the variogram is equal to 73.2 m (240 feet) and its sill is equal to 0.5. Thus the correlation range of $\ln(k)$ is 73.2 m and the variance is 0.5. In all examples, the prior mean for $\ln(k)$ is set equal to 4.0, so all elements of the vector m_{prior} (see Eq. 2.2) which correspond to gridblock log-permeabilities are set equal to 4.0. Relative permeabilities are represented by power law expressions. In all cases $S_{wc} = 0.2$, $S_{gc} = 0.05$, $S_{org} = 0.1$ and $S_{orw} = 0.2$. Throughout, porosity is assumed to be homogeneous with $\phi = 0.22$.

For the examples considered here, the variance of the pressure measurement error is set equal to 47.5 kPa^2 (1 psi^2), the variance of the GOR measurement error is set equal to 0.79 (std m^3)/(std m^3) (25 (scf/STB)^2). The WOR measurement error is defined by Eq. 2.6 with $\epsilon_o = 0.01$, $\epsilon_w = 0.02$ and $\sigma_{w,\min} = 0.32$ std m^3/d. These values were used to construct the data covariance matrix C_D. In the examples presented, however, no noise was added to the data generated from the simulation runs, that is d_{obs} was set equal to the vector of

production data output from the simulator.

3.4.1 Example 1a

In this case, the true reservoir model is not generated from an unconditional realization of the prior model, but is prescribed as a three zone reservoir as shown in Fig. 3.1. In all gridblocks contained within the lower left zone, $\ln(k) = 3.7$, at all gridblocks in the lower right zone, $\ln(k) = 4.3$, and in all gridblocks in the upper half, $\ln(k) = 3.9$. The prior variances and prior means for the parameters which characterize the power law relative permeability curves are defined in Table 3.1.

	True	Mean	Var
n_{rw}	1.90	2.17	1.0
k_{rwcw}	0.40	0.58	0.04
n_{rg}	2.40	2.14	1.0
k_{rgcw}	0.90	0.49	0.04
n_{row}	2.60	2.05	1.0
n_{rog}	1.70	1.74	1.0
k_{rocw}	0.80	0.49	0.04

Table 3.1: Prior means and variances of relative permeability model parameters.

First we consider the case where the true log-permeability field is fixed at its true value and attempt only to estimate the model parameters defining the power law relative permeability curves. We consider the three cases where we construct the estimate of model parameters by history matching only one type of data (pressure, gas-oil-ratio, or water-oil ratio) and the case where we history match all 130 data to construct the estimate. The results of these four history matches are summarized in Table 3.2. In this and similar tables presented later, a column labeled true contains the true values of the parameters, a column labeled p_{wf} contains estimates of parameters obtained by history matching only pressure data, a column labeled GOR contains estimates of parameters obtained by history matching only producing gas-oil ratio data, a column labeled WOR contains estimates obtained by history matching only water-oil ratio data, and a column labeled "All" contains estimates obtained by history matching all observed data, pressure, GOR and WOR.

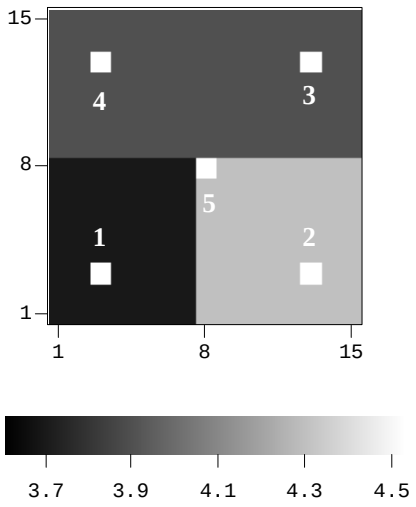


Figure 3.1: Three-zone reservoir, true model; Examples 1a and 1b.

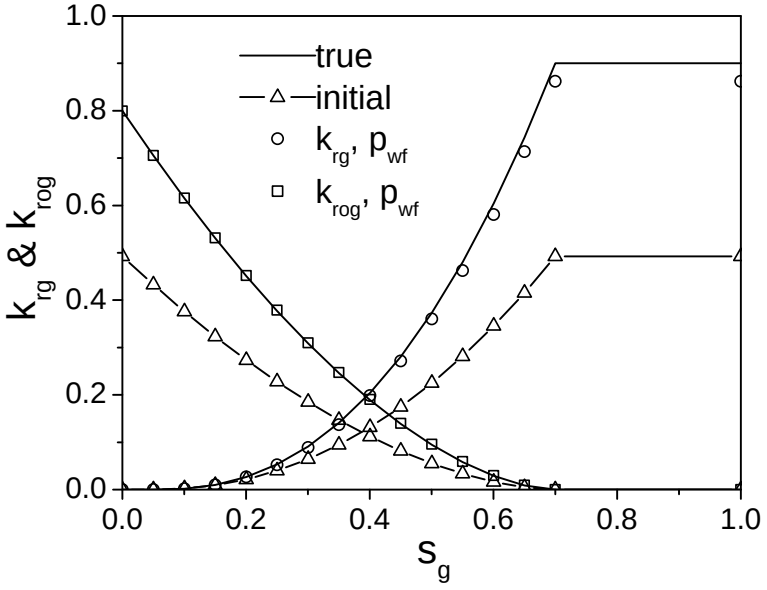


Figure 3.2: Prior, true and estimated gas-oil relative permeabilities, $\ln(k)$ known; Ex. 1a.

	True	p_{wf}	GOR	WOR	All
n_{rw}	1.90	1.91	1.91	1.95	1.90
k_{rwcw}	0.40	0.40	0.39	0.34	0.40
n_{rg}	2.40	2.38	2.40	2.32	2.40
k_{rgcw}	0.90	0.86	0.87	0.54	0.90
n_{row}	2.60	2.61	2.61	2.66	2.60
n_{rog}	1.70	1.69	1.73	1.75	1.70
k_{rocw}	0.80	0.80	0.79	0.68	0.80

Table 3.2: True and estimated relative permeability parameters; absolute permeability known.

The results of Table 3.2 indicate that the estimates of the relative permeability parameters are in extremely close agreement with the true values when we history match all available data, only pressure data, or only GOR data. The estimates obtained by history matching only the WOR data are significantly less accurate. Fig. 3.2 shows the estimated two-phase gas and oil relative curves obtained by history matching only pressure data compared with the true relative permeability curves and the initial guesses for the relative permeability curves. In this and similar figures, the true relative permeability curves are shown as solid curves with no data points on them and the initial guesses for relative permeability curves used in the automatic history matching procedure are shown as curves through triangular data points. The estimated relative permeability curves are always represented in our results as open circular data points and open square data points. We did not draw the complete relative permeability curves through these open data points as it makes the figures too cluttered. The initial guesses for relative permeability curves are generated using the prior values of m_r, the model parameters that characterize the relative permeability curves. Note the initial guesses for relative permeability curves are far from the truth. As can be deduced from the results of Table 3.2, the relative permeability curves estimated by history matching only GOR data are also very close to the true case. Factors that contribute to the excellent estimates obtained for this example include the following: (i) the observed and predicted data are generated from the same simulator with identical spatial grids and time steps, so no modeling error (Tarantola, 1987) occurs; (ii) the observed production data that are history matched are not corrupted by measurement error, i.e. the observed production data are identical to data obtained from the simulator run based on the true model; (iii) all rock and fluid properties except relative permeability parameters are known, so only the seven relative permeability model parameters are adjusted during the matching process (fewer

model parameters tend to yield less uncertain estimates); (iv) the saturation ranges that occur during the 300 day time frame spanned by the production data vary significantly; gas saturation varies from 0.0 to almost 0.25 at producing well gridblocks and water saturation varies from about 0.2 to over 0.5 at producing well gridblocks and increases up to 0.8 at the injection well gridblock. Thus, production data is influenced by the values of relative permeabilities over large saturation ranges.

The results of Table 3.2 indicate that history matching either wellbore pressure data or producing GOR data yields good estimates of the parameters k_{rwcw} and n_{rw} which define the water relative permeability function. As pressure data at producing wells are quite sensitive to changes in total mobility (Thompson and Reynolds, 1997), it is not completely surprising that one can obtain reasonable estimates of water relative permeability model parameters even if we history match only pressure data corresponding to times prior to water breakthrough. It is not clear why the information content of the GOR data is sufficient to resolve the parameters defining the oil and water relative permeability curves for the two-phase water-oil system. But, as shown next, an improved understanding of these results can be obtained by considering dimensionless sensitivity coefficients.

As discussed in Zhang et al. (2002), dimensionless sensitivity coefficients provide a relative measure of how different data affect estimates of model parameters, and the uncertainty in these estimates. Here, the dimensional sensitivity of data d_i to model parameter m_j is given by

$$s_{i,j} = \frac{\partial d_i}{\partial m_j} \frac{\sigma_{m_j}}{\sigma_{d_i}}, \tag{3.12}$$

where $\sigma^2_{m_j}$ denotes the prior variance for model parameter m_j and $\sigma^2_{d_i}$ denotes the variance of the measurement error for the ith observed data. In essence, data corresponding to higher values of dimensionless sensitivity are expected to result in better estimates of model parameters, i.e., if $s_{i,j} > s_{k,j}$, then history matching data $d_{obs,i}$ is expected to give a better estimate of model parameter m_j than is obtained by history matching data $d_{obs,j}$; see Zhang et al. (2002) for additional discussion.

Fig. 3.3 shows the dimensionless sensitivity of the flowing bottom hole pressure at well 1 to the seven parameters defining the two sets of relative permeability curves as a function of time, and Fig. 3.4 presents a similar plot for the GOR dimensionless sensitivity coefficients. The results of Fig. 3.3 show that pressure is slightly sensitive to the water relative permeability parameters (n_{rw}, k_{rwcw}), even at times prior to breakthrough which occurs at about 200 days. In fact up to 180 days, the dimensionless sensitivity of pressure to k_{rwcw} is roughly equal to the dimensionless sensitivity of pressure to the corresponding gas relative permeability parameter, k_{rgcw}. On the other hand, the results of Fig. 3.4 indicates that prior

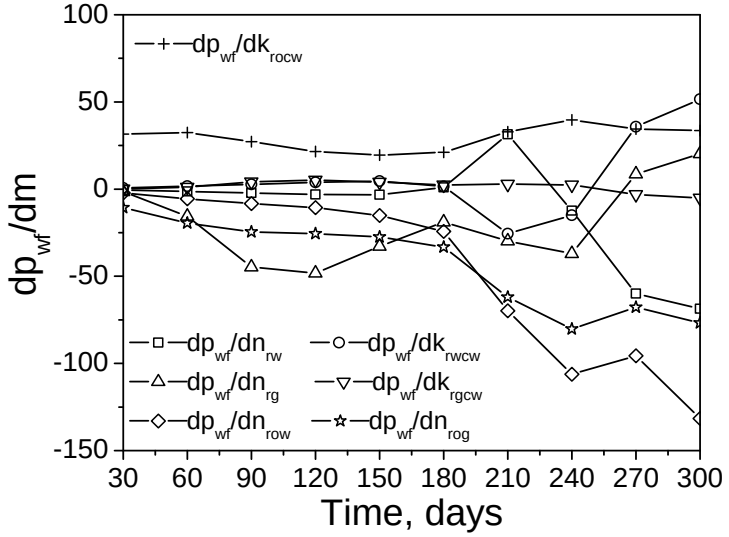


Figure 3.3: Dimensionless sensitivity of well 1 pressure to relative permeability model parameters; three-zone reservoir.

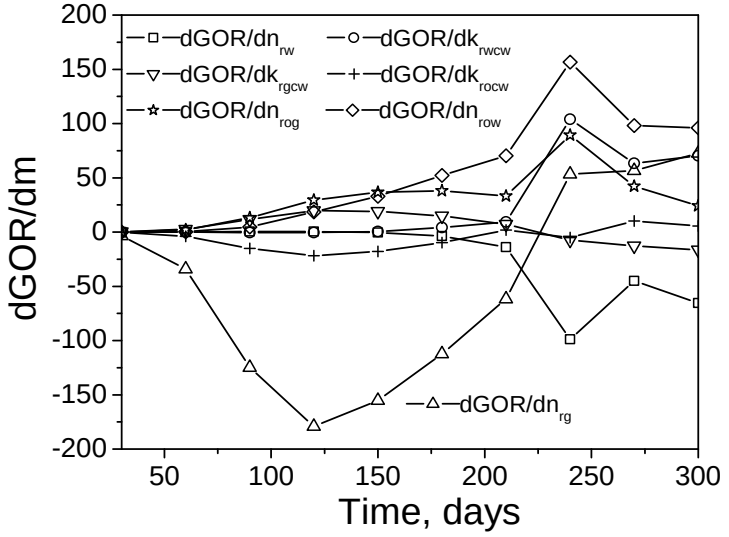


Figure 3.4: Dimensionless sensitivity of well 1 GOR to relative permeability model parameters; three-zone reservoir.

to 180 days, the GOR is almost insensitive to the water relative permeability parameters, but becomes very sensitive to these parameters after breakthrough. This indicates that the GOR data obtained after water breakthrough is primarily responsible for the good estimate of the water relative permeability curve (see Table 3.2) obtained by history matching GOR data.

To check the preceding supposition, we repeated the history match using only data up to 150 days. The results are shown in Table 3.3. Note in this case, history matching only GOR data gives relatively poor estimates of the parameters n_{rw} and k_{nrwc}, which define the water relative permeability curve. In fact, as expected, the results indicate that the estimates of these two parameters are essentially equal to their prior means. Matching only pressure data, however, gives good estimates of all relative permeability model parameters, although the estimates are not as accurate as those obtained by matching pressure data up to 300 days; see Table 3.2. Also note that the results of Table 3.3 indicate that the best estimates are obtained by matching both GOR and pressure data. In particular, the estimate of k_{rgcw} obtained by matching both pressure and GOR data is superior to that obtained by matching only one type of data. The dimensionless sensitivity of both GOR and pressure to this endpoint gas relative permeability are both fairly small, but the information content of the two sets of data is different because pressure is strongly influenced by the total mobility, but the GOR involves gas mobility divided by oil mobility. Because the information content for the two data types is different, history matching both pressure and GOR data yields an improved estimate of k_{rwcw}.

	True	p_{wf}	GOR	p_{wf} + GOR
n_{rw}	1.90	1.96	2.17	1.91
k_{rwcw}	0.40	0.41	0.55	0.40
n_{rg}	2.40	2.34	2.34	2.39
k_{rgcw}	0.90	0.79	0.60	0.89
n_{row}	2.60	2.63	2.78	2.61
n_{rog}	1.70	1.66	1.93	1.69
k_{rocw}	0.80	0.80	0.65	0.80

Table 3.3: True and estimated relative permeability parameters; absolute permeability known; match of data up to 150 days.

From the results of Figs. 3.3 and 3.4, one should note that both the dimensionless sensitivity of pressure to n_{rg} and the dimensionless sensitivity of GOR to n_{rg} are relatively large

in magnitude. Thus, these dimensionless sensitivities suggest that n_{rg} should be accurately resolved by matching either pressure or GOR data. The results of Table 3.3 confirm this expectation. Note, however, that the GOR is much less sensitive to the gas relative permeability endpoint value k_{rgcw}, and prior to water breakthrough, the sensitivity of GOR to k_{rgcw} is very roughly equal to the negative of the sensitivity of GOR to the k_{rocw}, the end-point oil relative permeability in the two-phase gas-oil system. (The difference in sign of these sensitivities is expected because the GOR involves k_{rg}/k_{rog} at times prior to water breakthrough.) Thus, even though the GOR data resolves n_{rg} accurately, the estimates of the other parameters in the two-phase oil relatively permeability curves are not very accurate even though the corresponding estimates of k_{rg}/k_{rog} are quite close to the true ratio at values of gas saturations that occur during the first 150 days of production. On the other hand, the results of Fig. 3.3 indicate that pressure is highly sensitive to k_{rocw}, n_{rg} and n_{rog}, three out of the four parameters that define the two- phase oil-gas relative permeability curves. Thus, history matching pressure should resolve these three parameters well. Pressure is somewhat less sensitive to the gas relative permeability endpoint k_{rgcw}, and the results of Table 3.3 indicate that a poorer estimate of this parameter is obtained by history matching pressure.

3.4.2 Example 1b

Next, the assumption that the true $\ln(k)$ field was known was removed. In this case, $\ln(k)$ at each gridblock and the relative permeability parameters were estimated simultaneously by history matching various combinations of production data. The estimated log-permeability field obtained by history matching only pressure data is shown in Fig. 3.5. The corresponding results obtained by history matching wellbore pressure, gas-oil ratio and water-oil ratio are shown in Fig. 3.6.

Matching all the data results in a more accurate estimate of the absolute permeability field than is obtained by matching only pressure data. In particular, matching only pressure results in several values of gridblock $\ln(k)$'s in the lower right quadrant and the upper center which are higher than the true values. Matching all data yields values somewhat closer to the truth. The over estimates of log-permeabilities are balanced by under estimates of the endpoint relative permeabilities, which are shown in Table 3.4. In this case, it looks like pressure data is effectively resolving effective permeabilities but not absolute permeability and relative permeabilities individually, which might cause one to question our discussion of Stone's model for three-phase relative permeabilities. Note, however, that when we condition to all data, we obtain good estimates of both log-permeability and the relative permeability parameters; see Fig. 3.6, Table 3.4, and Figs. 3.7 and 3.8. Fig. 3.7 shows the two-phase oil-gas

relative permeability curves obtained by matching only pressure data, and Fig. 3.8 shows the improved results obtained by matching all data (wellbore pressures, producing WOR

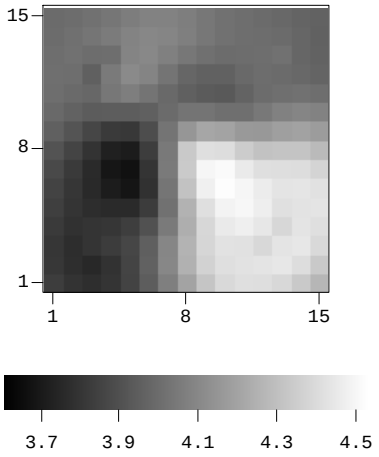


Figure 3.5: Log-permeability estimated from history match of pressure; Ex. 1b.

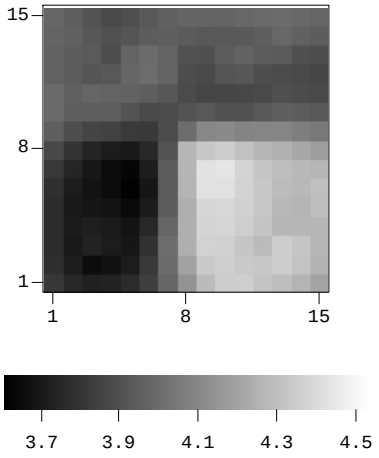


Figure 3.6: Log-permeability estimated from history match of pressure, GOR and WOR; Ex. 1b.

and producing GOR). As can be deduced from the results of Table 3.4, results of similar accuracy were obtained for the two-phase water-oil relative permeability curves. Although we did not show data matches for the examples considered, in all cases, we obtained excellent data matches. In fact, the matches were of similar quality to the ones that will be shown for the next example.

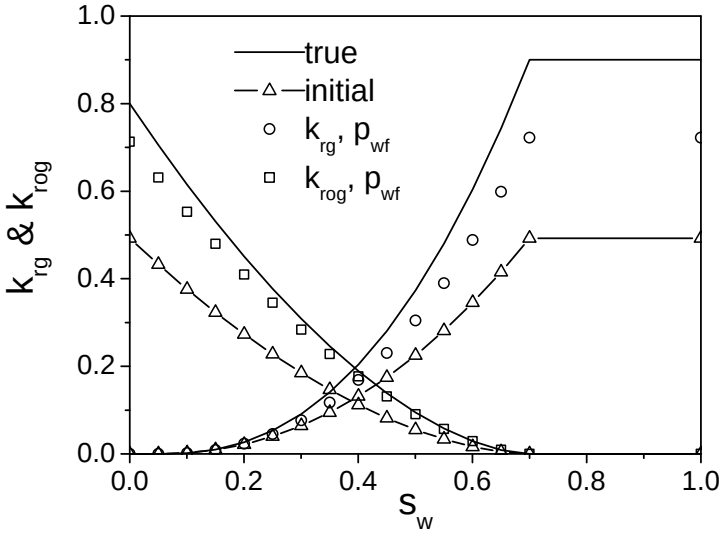


Figure 3.7: Prior, true and estimated gas-oil relative permeabilities, history match p_{wf}; Ex. 1b.

	True	p_{wf}	All
n_{rw}	1.90	1.91	1.91
k_{rwcw}	0.40	0.35	0.39
n_{rg}	2.40	2.36	2.40
k_{rgcw}	0.90	0.72	0.87
n_{row}	2.60	2.68	2.62
n_{rog}	1.70	1.64	1.70
k_{rocw}	0.80	0.71	0.78

Table 3.4: True and estimated relative permeability parameters; $\ln(k)$ estimated simultaneously; Example 1b.

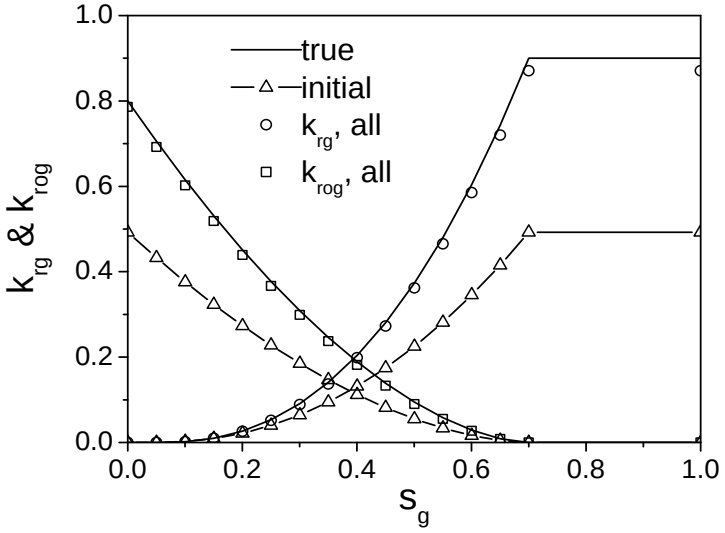


Figure 3.8: Prior, true and estimated gas-oil relative permeabilities, history match p_{wf}, GOR and WOR; Ex. 1b.

3.4.3 Example 2

In this example, an unconditional realization of log-permeability generated from the prior model is used as the true log-permeability field. The prior mean for $\ln(k)$ is still 4.0, and the prior means for relative permeability model parameters are shown in Table 3.5. In this case, we generate a realization of the model instead of the maximum a posteriori estimate. The randomized maximum likelihood method (Kitanidis, 1995; Oliver et al., 1996) is applied to generate a realization. To generate a realization with this procedure, we replace m_{prior} by m_{uc} (an unconditional realization of m generated from the prior model) and replace d_{obs} by a realization of the data d_{uc} (i.e., add noise) in Eq. 2.2 and then minimize the modified objective function. The unconditional realization m_{uc} is used as the initial guess when history matching d_{uc}.

Table 3.5 shows the realization of relative permeability model parameters obtained by history matching pressure, GOR and WOR. Figs. 3.9 and 3.10 present plots of the resulting realization of the relative permeability curves compared with the true curves and the initial guesses for the relative permeability curves. Fig. 3.11 shows the unconditional realization of the log-permeability field which was used as the initial guess in the randomized maximum likelihood method and Fig. 3.12 shows the realization of the log-permeability field estimated by history matching production data. Note the history matching process has radically altered

the unconditional realization in the interwell region. This alteration is necessary to obtain a good match to the production data. Fig. 3.13 shows the producing gas-oil ratio predicted at well 1 based on the initial guess for model parameters (solid diamond data points) and the GOR data (solid triangular data points). The continuous curve represents the GOR predicted from the model obtained by history-matching. Note the initial model results in a GOR much lower than the observed GOR at late times. GOR history matches of the same quality were obtained at other wells. Fig. 3.14 shows the water-oil-ratio data and the data predicted from the model obtained by history matching. Note that good history matches of WOR were obtained at all four producing wells.

	True	Mean	All
n_{rw}	1.90	2.3	1.84
k_{rwcw}	0.40	0.5	0.40
n_{rg}	2.40	2.1	2.40
k_{rgcw}	0.90	0.8	0.96
n_{row}	2.60	2.3	2.59
n_{rog}	1.70	2.1	1.69
k_{rocw}	0.80	0.6	0.84

Table 3.5: True and estimated relative permeability parameters; $\ln(k)$ estimated simultaneously; heterogeneous reservoir example 2.

3.5 Remarks

It is important to note that data can directly provide information only on the part of the relative curves that correspond to phase saturations that exist within the reservoir. Since a power law functional form is used to represent all relative permeability curves, with each curve described by only two parameters, resolving relative permeabilities corresponding to low saturations accurately is essentially equivalent to resolving the complete curve accurately. This explains why accurate estimates of relative permeability curves were obtained even though maximum gas saturation was less than 0.25. If the gas relative permeability function were modeled as a B-spline, the portion of the gas relative permeability curve corresponding to high gas saturations would not be resolved accurately by the available production data. For example, if the reservoir pressure remains above bubble-point pressure, the estimate of

gas relative permeability model parameters would be determined by the prior model and would not be improved by matching production data.

Even though, arguments have been presented in favor of the Stone model (see, for example, Aziz and Settari (1979)), Stone's model II leads to unsettling results. Namely, under two-phase flow conditions, one can not estimate accurately both absolute permeability and the coefficients in the power law relative permeabilities. An infinite set of values of absolute permeabilities and relative permeability coefficients will produce exactly the same production data when input to the simulator. We have shown, however, that this does not occur under three-phase flow conditions when Stone's Model II is used. In this case, reasonable estimates of both absolute and relative permeability functions can sometimes be obtained. Although a mathematical argument supporting this conclusion has been presented, we can provide no cogent physical explanation of why this should be the case, and believe a critical re-examination of Stone's Model II may be warranted.

When the objective is to estimate absolute permeability fields by history matching of production data, Li et al. (2001a) have shown that the information content of pressure data is higher than the information content of GOR and both of these data types generally have more information content than WOR data. The results presented here suggest that this is also true when estimating absolute and relative permeabilities simultaneously.

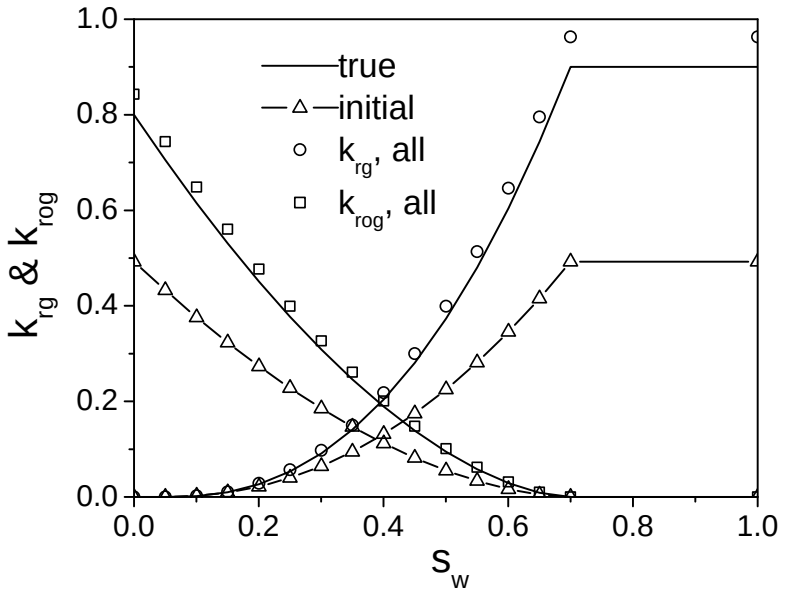


Figure 3.9: Prior, true and estimated gas-oil relative permeabilities, history match p_{wf}, GOR and WOR; Ex. 2.

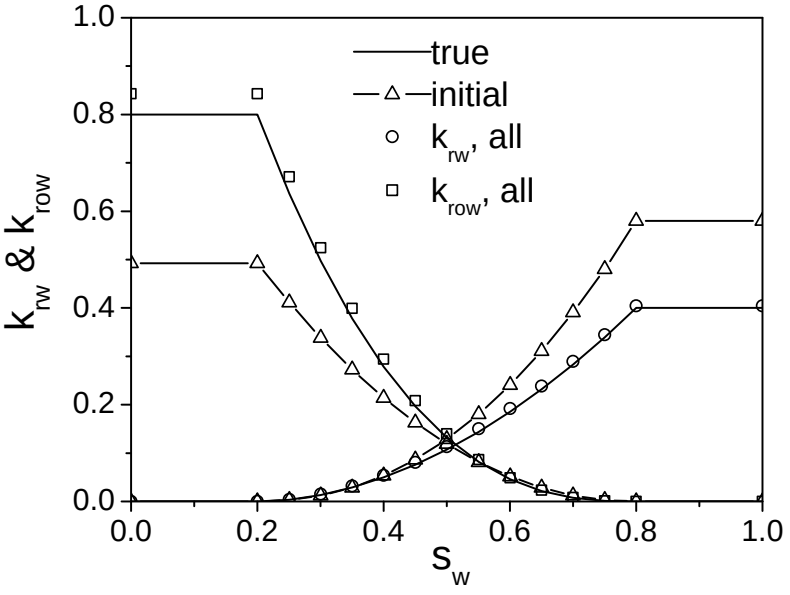


Figure 3.10: Prior, true and estimated water-oil relative permeabilities, history match p_{wf}, GOR and WOR; Ex. 2.

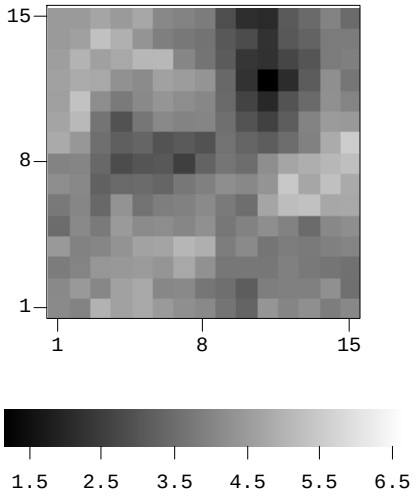


Figure 3.11: Unconditional realization of log-permeability.

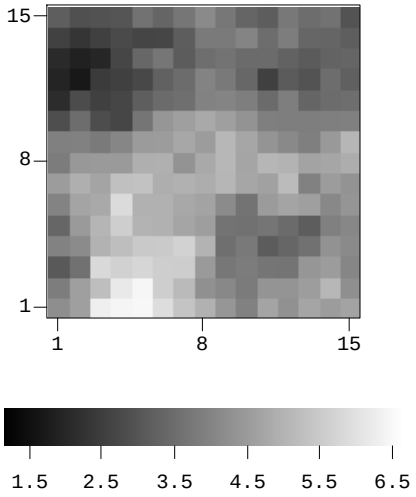


Figure 3.12: Conditional realization of log-permeability.

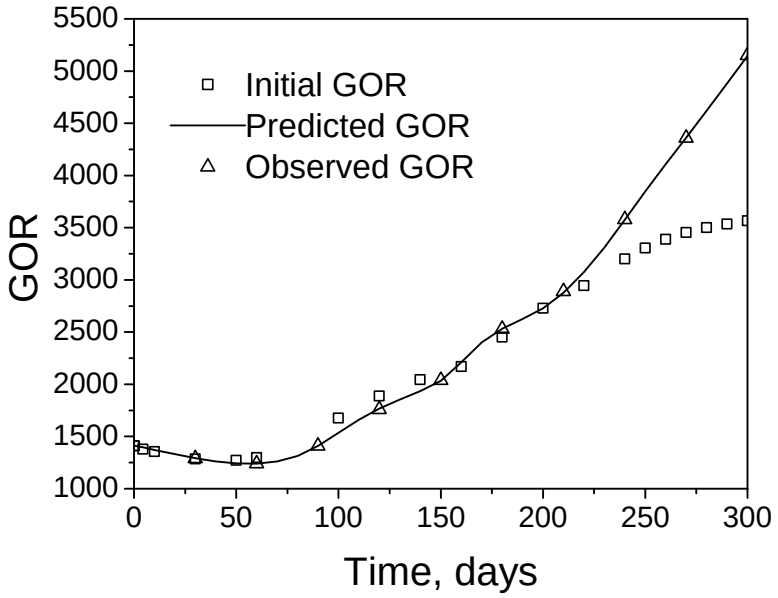


Figure 3.13: GOR data, GOR predicted with initial model and GOR predicted with history matched model.

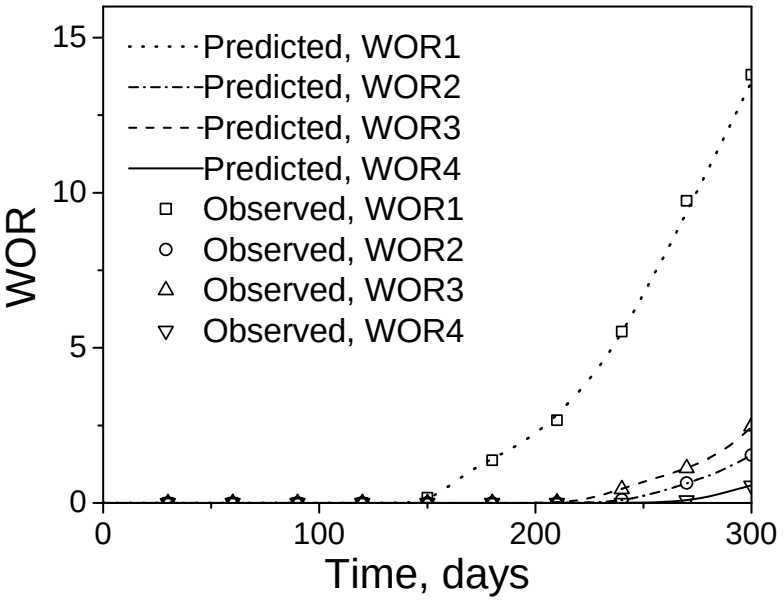


Figure 3.14: WOR data and history matched WOR.

Chapter 4

LBFGS and Nonlinear Conjugate Gradient

We wish to be able to history match several hundred production data to generate realizations of tens of thousands of model parameters. For such problems, it is not feasible to compute the individual sensitivity coefficients required by standard implementation of the Gauss-Newton and Levenberg-Marquardt algorithms. Thus, it is reasonable to implement algorithms that which require only the gradient of the objective function. As the steepest descent algorithm is very sensitive to scale and often exhibits poor convergence properties (see, for example, Fletcher (1987)), the only viable algorithms that use derivative information but ignore explicit computation of the Hessian appear to be quasi-Newton (variable metric) methods, preconditioned conjugate gradient (PCCG) methods and truncated Newton. The efficiency of PCCG methods largely rest on finding a good preconditioner. This, however, is not an easy task and as shown in the examples provided here, we have not found a preconditioner that yields a PCCG method that is as robust or as computationally efficient as our implementations of the quasi-Newton method known at the limited memory Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm. As discussed in section 4.1, our expectation is that truncated-Newton methods (Nash, 1985) will not be computationally efficient for problems where the number of production and/or time-lapse seismic data to be history matched is large; thus, we have not implemented a truncated Newton method.

As discussed in Chapter 2, the MAP estimate of the model is obtained by minimizing

$$O(m) = \frac{1}{2}(m - m_{\text{prior}})^T C_M^{-1}(m - m_{\text{prior}}) + \frac{1}{2}(g(m) - d_{\text{obs}})^T C_D^{-1}(g(m) - d_{\text{obs}}), \qquad (4.1)$$

and a realization of the model conditional to observed data with the covariance of data

measurement errors given by C_D is obtained by minimizing

$$O_r(m) = \frac{1}{2}(m - m_{uc})^T C_M^{-1}(m - m_{uc}) + \frac{1}{2}(g(m) - d_{uc})^T C_D^{-1}(g(m) - d_{uc}), \tag{4.2}$$

where m_{uc} is an unconditional realization generated from the prior pdf, and d_{uc} is obtained by adding noise to d_{obs}. Throughout, the reminder of this chapter, we simply let O denote the objective function that we wish to minimize.

4.1 Levenberg-Marquardt, Gauss-Newton and Truncated-Newton

In generating the MAP estimate, the search direction in the modified Levenberg-Marquardt (MLM) algorithm can be calculated from Eq. 2.19 which is rewritten here as

$$\Big[(1+\lambda_k)C_M^{-1} + G_k^T C_D^{-1} G_k\Big]\delta m_{k+1} = -\Big[C_M^{-1}(m_k - m_{\text{prior}}) + G_k^T C_D^{-1}(g(m_k) - d_{\text{obs}})\Big], \tag{4.3}$$

where k is the iteration index and m_k gives the approximation to the MAP estimate of the model at the kth iteration. Using matrix inversions lemmas, it is well known (Tarantola, 1987) that Eq. 4.3 is equivalent to

$$\delta m_{k+1} = \frac{m_k - m_{\text{prior}}}{1+\lambda_k} + C_M G_k^T \Big[(1+\lambda_k)C_D + G_k C_M G_k^T\Big]^{-1}\Big[\frac{G_k(m_k - m_{\text{prior}})}{1+\lambda_k} - \big(g(m_k) - d_{\text{obs}}\big)\Big]. \tag{4.4}$$

Choosing $\lambda_k = 0$ in Eqs. 4.3 and 4.4 gives the two corresponding formulas for the Gauss-Newton (GN) search direction. If the GN or MLM algorithm is applied to construct a realization by the randomized maximum liklihood method, i.e., by minimizing the objective function of Eq. 4.2, then d_{obs} and m_{prior}, respectively, should be replaced by d_{uc} and m_{uc} in Eqs. 4.3 and 4.4. Once δm_{k+1}, has been computed, we can compute $m_{k+1} = m_k + \mu_k \delta m_{k+1}$ where μ_k is the step size, which can be calculated by either a restricted step method or a line search algorithm when the Gauss-Newton method is used; see Fletcher (1987). When MLM is used, as noted in Chapter 2, we use a simple procedure in which we decrease λ_k by a factor of 10 if $O(m_{k+1}) < O(m_k)$; otherwise, we increase λ_k by a factor of 10 and redo the step. An initial value of $\lambda_0 = 1000$ to 10^5 works satisfactorily for most problems.

Applying Eq. 4.4 requires solving an $N_d \times N_d$ matrix problem, i.e., when $\lambda_k = 0$, the solution of

$$\Big[C_D + G_k C_M G_k^T\Big]x = \Big[G_k(m_k - m_{\text{prior}}) - \big(g(m_k) - d_{\text{obs}}\big)\Big], \tag{4.5}$$

for

$$x = \Big[C_D + G_k C_M G_k^T\Big]^{-1}\Big[G_k(m_k - m_{\text{prior}}) - \big(g(m_k) - d_{\text{obs}}\big)\Big] \tag{4.6}$$

If this matrix problem is solved iteratively by the conjugate gradient method, then similar to ideas of Paige and Saunders (1982), one does not need to explicitly compute G; one only needs to be able to calculate Gu and $G^T v$ for vectors u and v at each iteration of the conjugate gradient method. Moreover, there is no need to solve the matrix problem of Eq. 4.5 very accurately at early iterations when we are far from the minimum. The resulting algorithm is referred to as a truncated Newton method, actually truncated Gauss-Newton in our case. Mackie and Madden (1993) presented an implementation of this procedure in the geophysics literature. Chu et al. (2000) introduced a similar procedure into the petroleum engineering literature for single-phase flow problems. A somewhat different and clearer presentation of how one can compute Gu and $G^T v$ than presented by Chu et al. (2000) can be found in Abacioglu (2001). Computation of Gu requires a forward run of the simulation. Computation of $G^T v$ requires one solution of the adjoint system.

We have not implemented a truncated Newton procedure because our conjecture is that the method is not generally robust or efficient for large-scale automatic history matching problems. For example, if N_d is large and the variances of some data measurements are small, then the coefficient matrix of Eq. 4.5 may be very ill-conditioned and it may require a good non-diagonal preconditioning matrix to obtain a conjugate gradient method that converges sufficiently fast to render the method useful. But since, the coefficient matrix is not explicitly constructed, it is not clear how to obtain a good preconditioning matrix. Because of these considerations, we have not implemented a truncated Newton method, but we can not completely rule out that possibility that an implementation of truncated Newton that is feasible for large scale history matching problems may be found.

4.2 BFGS and LBFGS Algorithms

When $\lambda_k = 0$, Eq. 4.3 is equivalent to

$$d_k = \delta m_{k+1} = -H_k^{-1} g_k, \tag{4.7}$$

where the first equality defines d_k, g_k is the gradient of the objective function evaluated at m_k, i.e.,

$$g_k = \nabla O(m_k), \tag{4.8}$$

and H_k is the Hessian matrix evaluated at m_k. If the objective function is given by either Eq. 4.1 or Eq. 4.2, then

$$H_k = C_M^{-1} + G_k^T C_D^{-1} G_k, \tag{4.9}$$

where G_k represents the $N_d \times N_m$ sensitivity coefficient matrix evaluated at m_k. If N_d and N_m are both large, the evaluation of all entries of G_k by either the adjoint or gradient simulator method is not feasible.

In quasi-Newton methods, H_k^{-1} is approximated by a symmetric positive definite matrix $\tilde{H}_k^{-1}$, which is corrected or updated from iteration to iteration. All updating formulas involve the difference in gradients, $y_k = g_{k+1} - g_k$, and the difference in iterates, $s_k = m_{k+1} - m_k$. Here, only variants of the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm are considered because this quasi-Newton method has proved to be more robust in practice than other quasi-Newton algorithms; see Kolda et al. (1998). It is well known that scaling can improve the convergence attributes of quasi-Newton (QN) methods, and numerous suggestions have been made for calculating scaling factors; see, in particular, Oren and Luenberger (1974) and Shanno and Phua (1978). Here, we identify a scaling procedure that has worked well for the history matching problems that we have tried.

The BFGS formula for updating the approximation to the inverse Hessian is given by

$$\tilde{H}_{k+1}^{-1} = \gamma_k \Big(\tilde{H}_k^{-1} - \frac{\tilde{H}_k^{-1} y_k y_k^T \tilde{H}_k^{-1}}{y_k^T \tilde{H}_k^{-1} y_k} + v_k v_k^T \Big) + \frac{s_k s_k^T}{s_k^T y_k}, \tag{4.10}$$

where γ_k is the scaling factor and

$$v_k = (y_k^T \tilde{H}_k^{-1} y_k)^{1/2} \Big(\frac{s_k}{s_k^T y_k} - \frac{\tilde{H}_k^{-1} y_k}{y_k^T \tilde{H}_k^{-1} y_k} \Big). \tag{4.11}$$

If no scaling is done, $\gamma_k = 1$ for all k. If scaling is done only at the first update, then $\gamma_k = 1$ for $k > 0$. In our applications, $\tilde{H}_{k+1}^{-1}$ is an $N_m \times N_m$ matrix. Thus, explicit application of Eq. 4.10 requires the storage and multiplication of $N_m \times N_m$ matrices where for large scale history matching problems, the value of N_m may be significantly greater than ten thousand. In this case, it is not feasible to store $N_m \times N_m$ matrices unless they are sparse. The limited memory BFGS algorithm in the form implemented by Nocedal (1980) provides an alternate implementation of the BFGS and the LBFGS which avoids explicit computation and storage of the approximations to the inverse Hessian.

It is well known that the update formula of 4.10 can be rewritten as

$$\tilde{H}_{k+1}^{-1} = \gamma_k V_k^T \tilde{H}_k^{-1} V_k + \rho_k s_k s_k^T, \tag{4.12}$$

where $\rho_k = 1/y_k^T s_k$, and $V_k = I - \rho_k y_k s_k^T$. To motivate LBFGS, Nocedal rewrote Eq. 4.12 as

$$\begin{aligned}\tilde{H}_{k+1}^{-1} =& V_k^T V_{k-1}^T \cdots V_{k-p+1}^T (\gamma_k \tilde{H}_0^{-1}) V_{k-p+1} \cdots V_{k-1} V_k \\ &+ V_k^T \cdots V_{k-p+2}^T \rho_{k-p+1} s_{k-p+1} s_{k-p+1}^T V_1 \cdots V_k \\ &\vdots \\ &+ V_k^T \rho_{k-1} s_{k-1} s_{k-1}^T V_k \\ &+ \rho_k s_k s_k^T,\end{aligned} \tag{4.13}$$

where $p = \min\{L, k+1\}$ and the parameter L is an integer chosen by the user. Although we have used the same notation for the scaling parameter in Eqs. 4.12 and 4.13, the two equations are identical if and only if $\gamma_k = 1$ for all $k > 0$. Nocedal (1980) implemented Eq. 4.13 in a way that avoids storing any matrices except $\tilde{H}_0^{-1}$ and he suggests using a diagonal matrix for $\tilde{H}_0^{-1}$, so the storage required is minimal. His algorithm has the following advantages: (i) for $k > 0$, $\tilde{H}_k^{-1}$ is not computed explicitly or stored; instead, the vector $\tilde{H}_k^{-1} g_k$ is formed directly and the search direction δm_{k+1} is set equal to the negative of this vector; (ii) $\tilde{H}_k^{-1} g_k$ is calculated using only dot products of the vectors $\tilde{H}_0^{-1} g_0$, s_l and y_l, for $l = k-1, k-2, ..., \max\{0, k-L\}$. If L is greater than the total number of iterations allowed, then LBFGS becomes equivalent to BFGS if $\gamma_k = 1$ for $k > 0$. Otherwise, LBFGS requires less memory and less computational time per iteration than BFGS even if Nocedal's implementation is used for the BFGS algorithm. If L is too small, however, the number of iterations required for convergence is increased. For the history-matching problems we have considered to date, $L = 30$ has proved to be a good choice, but we often simply set $L = 100$ if the size of the problem is not too large. For the history-matching problems we have tried, values of L larger than 30 only resulted in a small improvement in the rate of convergence, but using values of L much smaller than 20 often resulted in a noticeable degradation in the rate of convergence and resulted in higher values of the objective function at convergence.

Nocedal suggested choosing $\tilde{H}_0^{-1}$ as a diagonal matrix. From Eq. 4.9, the obvious choice would then be the diagonal of C_M, but with this choice the smoothing effects of multiplication by C_M are lost and the rock property fields are much rougher than would be expected based on the prior model for reservoir variables. A better choice is $\tilde{H}_0^{-1} = C_M$, which then requires storage of C_M and multiplication of vectors by C_M using sparse matrix techniques. This is the approach we have adopted in all our history matching applications.

4.2.1 Line Search, Method 1

Each iteration of the algorithm requires an approximate line search to determine the step size in the search direction. The choice of the line search can affect both the convergence properties and the computational efficiency. An exact line search would determine the step size α_k so that $\alpha = \alpha_k$ minimizes $O(m_k + \alpha\delta m_{k+1})$ over α. Each new approximation for α_k requires at least one re-evaluation of the objective function which requires one forward simulation run. As is typical in practice, we do not do an exact line search; instead, we terminate the line search when the Wolfe conditions are satisfied; see Fletcher (1987). In our approximate line search, we (a) do one iteration of the Newton-Raphson algorithm (discussed below) starting with an initial guess equal to zero to obtain the approximation α_k^1; (b) determine a quadratic $q(\alpha)$, such that $q(0) = O(m_k)$, $q'(0) = \nabla O(m_k)$ and $q(\alpha_k^1) = O(m_k + \alpha_k^1 d_k)$ and set the new approximation of α_k equal to the minimum of $q(\alpha)$; (c) successively cut the step size by a factor of 10 until we find a model that gives a decrease in the objective function. We stop the line search whenever the Wolfe conditions are satisfied. In the most cases, this occurs after step (a) and the necessity to activate step (c) is extremely rare. When (c) occurs however, the algorithm becomes inefficient, and in some cases, the algorithm becomes stuck at this point. When this occurs, we are forced to restart the algorithm, and when this occurs more iterations are required for convergence, i.e., computational efficiency is degraded. Later in this chapter, we present an improved line search which somewhat improves computational efficiency. The Wolfe conditions (Dennis and Schnabel, 1996; Fletcher, 1987) will also be discussed in detail later.

The procedure for generating the first estimate of the step size α by the Newton-Raphson algorithm is given next. At the kth iteration, we wish to obtain the step size α such that

$$f(\alpha) = O(m_k + \alpha d_k), \tag{4.14}$$

is minimized along the search direction d_k. The minimizer can be found by setting the derivative of the function $f(\alpha)$ equal to zero, i.e.,

$$h(\alpha) \equiv f'(\alpha) = \frac{dO(m_k + \alpha d_k)}{d\alpha} = \big(\nabla O(m_k + \alpha d_k)\big)^T d_k = 0. \tag{4.15}$$

This equation can be solved by using the Newton-Raphson algorithm which is given by

$$\alpha_{j+1} = \alpha_j - \frac{h(\alpha_j)}{h'(\alpha_j)}, \tag{4.16}$$

where j denotes the index of the Newton-Raphson iteration. The first derivative of h is given by

$$h'(\alpha) = \frac{dh(\alpha)}{d\alpha} = d_k^T \nabla \Big[\big(\nabla O(m_k + \alpha d_k)\big)^T\Big] d_k = d_k^T H(m_k + \alpha d_k) d_k. \tag{4.17}$$

In an exact line search, the Newton-Raphson iteration is stopped when a convergence criterion is satisfied. In general, exact line searches are not done in practice as they lead to computational inefficiencies. In our case, an exact line search which required more than a few iterations would be prohibitively expensive if the Newton-Raphson algorithm were used because each iteration requires the evaluation of the term $d_k^T H(m_k + \alpha d_k) d_k$ and this evaluation requires at least one simulation run. In our procedure, as mentioned previously, we do only one Newton-Raphson iteration. To perform one Newton-Raphson iteration, we set $\alpha_0 = 0$ and then Eq. 4.16 gives

$$\alpha_1 = -\frac{\big(\nabla O(m_k)\big)^T d_k}{d_k^T H(m_k) d_k}. \tag{4.18}$$

Eq. 4.18 involves the Hessian matrix, $H_k = H(m_k)$, which is given by

$$H_k = G_k^T C_D^{-1} G_k + C_M^{-1}, \tag{4.19}$$

so

$$\begin{aligned} d_k^T H_k d_k &= d_k^T (G_k^T C_D^{-1} G_k + C_M^{-1}) d_k \\ &= d_k^T (G_k^T C_D^{-1} G_k) d_k + d_k^T C_M^{-1} d_k \\ &= (G_k d_k)^T C_D^{-1} (G_k d_k) + d_k^T C_M^{-1} d_k. \end{aligned} \tag{4.20}$$

In this equation, we do not need to compute the sensitivity coefficient matrix G directly. We only need to calculate $G d_k$ which can be done by using a finite-difference approximation as shown next. (It could also be calculated using one run of the gradient simulator method; see Abacioglu (2001)). The method given below was originally implemented by Kalita (2000). The elements of the sensitivity coefficient matrix can be written as

$$G_{i,j} = \frac{\partial g_i}{\partial m_j}, \tag{4.21}$$

where $i = 1, \cdots, N_d$ and $j = 1, \cdots, N_m$. The directional derivative is

$$\Big(\frac{dg}{d\alpha}\Big)_{\alpha=0} = \Big(\frac{dg(m + \alpha d_k)}{d\alpha}\Big)_{\alpha=0}. \tag{4.22}$$

Let $u = d_k / \parallel d_k \parallel$. So we have

$$\begin{aligned} \Big(\frac{dg_i}{d\alpha}\Big)_{\alpha=0} &= [\nabla g_i(m)]^T u \\ &= \frac{1}{\parallel d_k \parallel} [\nabla g_i(m)]^T d_k. \end{aligned} \tag{4.23}$$

The ith component of Gd_k is given by

$$\begin{aligned}\left[Gd_k\right]_i &= \sum_{j=1}^{N_m} \frac{\partial g_i}{\partial m_j} d_{k,j} \\ &= [\nabla g_i(m)]^T d_k, \end{aligned} \tag{4.24}$$

where $d_{k,j}$ denotes the jth component of the vector d_k. Substituting Eq. 4.23 into Eq. 4.24, we obtain

$$\begin{aligned} Gd_k =& \parallel d_k \parallel \left(\frac{dg}{d\alpha}\right)_{\alpha=0} \\ \approx& \parallel d_k \parallel \frac{g(m+\epsilon d_k) - g(m)}{\epsilon \parallel d_k \parallel} \\ =& \frac{g(m+\epsilon d_k) - g(m)}{\epsilon}, \end{aligned} \tag{4.25}$$

where ϵ is a small number. We choose ϵ based on the infinity norm of d_k such that ϵ satisfies $\epsilon \parallel d_k \parallel_\infty = 10^{-3}$. Note that calculating Gd_k needs one additional simulation run. Once we have Gd_k, it is straight forward to calculate $d_k^T H_k d_k$ using Eq. 4.20 and then Eq. 4.18 can be applied to calculate the step size. Application of Eq. 4.20 requires evaluating $C_M^{-1} d_k$. In our code, we provide two ways to calculate this term. One way is to solve a matrix problem

$$C_M x = d_k, \tag{4.26}$$

for $x = C_M^{-1} d_k$ using either LU decomposition or preconditioned conjugate gradient method (The other way is to approximate C_M^{-1} by using stencil method; see Skjervheim (2002) or Oliver (1998).)

4.2.2 Convergence Criteria

In our results, the following stopping criteria are used to terminate the algorithm:

1. $$\frac{\mid O_{k+1} - O_k \mid}{O_k + 10^{-14}} < \varepsilon_1 \tag{4.27}$$

   and

   $$\frac{\parallel m_{k+1} - m_k \parallel_2}{\parallel m_k \parallel_2 + 10^{-14}} < \varepsilon_2 \tag{4.28}$$

   where k denotes the iteration index and $\parallel \cdot \parallel_2$ denotes the l_2 norm of a vector. Both conditions must be satisfied to terminate the iteration. If we use only Eq. 4.27 as the convergence criterion, the algorithm may converge prematurely.

2. Specify a maximum allowable iteration number. If the number of iterations exceeds the specified number, we force the iteration to stop. In our examples, we usually specify the maximum number of iterations as 300.

4.2.3 Summary of LBFGS Algorithm

The basic LBFGS algorithm follows:

Step 1 Initialization
(a) Provide an initial guess, m_0, of the model, calculate the objective function corresponding to m_0 and evaluate the gradient of the objective function at m_0, i.e., compute g_0;
(b) provide an initial Hessian inverse approximation $\tilde{H}_0^{-1}$ (e.g., C_M in our examples), set the initial iteration index k=0.

Step 2 Calculate the search direction $\delta m^{k+1} = d_k = -\tilde{H}_k^{-1} g_k$ and check whether it is a downhill direction, i.e., check to see if $d_k^T g_k < 0$. If d_k is not a downhill search direction, set $d_k = -\tilde{H}_0^{-1} g_k$.

Step 3 Calculate the step size α_k by a line search procedure as discussed in subsection 4.2.1.

Step 4 Update the model to $m_c = m_k + \alpha_k d_k$.

Step 5 Calculate the objective function based on m_c.

Step 6 Determine if the Wolfe conditions are satisfied; if they are satisfied, then set $m_{k+1} = m_c$ and go to step 7, otherwise do
(a) fit a quadratic and find a step size by minimizing this quadratic, then go to step 4;
(b) if a quadratic fit has already been done, cut the step size by a specified factor (in our examples we cut the step size by a factor of 10) and go to step 4. All computations we have done suggest this case does not occur very often.

Step 7 Determine if the stopping criteria are satisfied. If satisfied, then stop; otherwise go to step 8.

Step 8 Calculate $s_k = m_{k+1} - m_k = \alpha_k d_k$ and $y_k = g_{k+1} - g_k$. Apply Eq. 4.12 or 4.13 to update the inverse Hessian approximation $\tilde{H}_{k+1}^{-1}$. Set $k = k + 1$ and then go to step 2.

Although the procedure presented above is convenient for discussion, we never explicitly update the inverse Hessian approximation as part of Step 8 after the first iteration. Instead

we use the algorithm presented by Nocedal (1980) to compute $d_k = -\tilde{H}_k^{-1} g_k$ in Step 2. This eliminates the need to explicitly compute or store H_k^{-1} for $k > 0$.

For BFGS and LBFGS, scaling can have a significant effect on the rate of convergence; Shanno (1970), Oren and Luenberger (1974), Oren (1974), Oren and Spedicato (1976). If scaling is used, Step 8 needs to be modified. As scaling was considered in detail in last year's annual report, here we only give the scaling algorithm we use. Although this scaling algorithm was shown to be robust for several examples Zhang (2002), there is no theoretical result that guarantees that it will always work well.

For the LBFGS algorithm, we have found a variant of the optimal switching rule given by Oren and Spedicato (1976) which works well. Specifically, we compute

$$\tilde{\tau}_k = \frac{s_k^T \tilde{H}_0 s_k}{s_k^T y_k}, \tag{4.29}$$

$$\tilde{\sigma}_k = \frac{s_k^T y_k}{y_k^T \tilde{H}_0^{-1} y_k}, \tag{4.30}$$

and then determine the scaling factor γ_k by the following rule:

$$\gamma_k = \begin{cases} \tilde{\tau}_k & \text{if} \quad \tilde{\tau}_k < 1.0 \\ \tilde{\sigma}_k. & \text{otherwise.} \end{cases} \tag{4.31}$$

In our applications, the prior covariance matrix is used as the initial approximation to the inverse Hessian, i.e., $\tilde{H}_0^{-1} = C_M$. Although it is not done in the examples presented here, we have found that in many examples it is satisfactory to compute $\tilde{\tau}_k$ by

$$\tilde{\tau}_k = \frac{s_k^T \tilde{D}^{-1} s_k}{s_k^T y_k}, \tag{4.32}$$

where $\tilde{D}$ is a diagonal matrix with diagonal entries equal to the diagonal of C_M. Even when Eq. 4.32 is used instead of the Eq. 4.29, one should still use $\tilde{H}_0^{-1} = C_M$ in Eq. 4.30. In section 4.5, we compare alternative scaling techniques.

For the LBFGS algorithm with initial scaling, we use Eq. 4.31 to compute γ_0 but set $\gamma_k = 1$ at all subsequent iterations. The efficient LBFGS method given by Nocedal (1980) avoids formation of $\tilde{H}_k^{-1}$ for $k \geq 1$, only $\tilde{H}_k^{-1} g_k$ is calculated at each iteration. However, $\tilde{H}_0^{-1}$ must be provided as the initial approximation to the inverse Hessian. In our implementation, we use $\tilde{H}_0^{-1} = C_M$, where C_M is the prior covariance matrix. In algorithmic form, the procedure given by Nocedal (1980) can be implemented by the following recursive formula which is reproduced from Zhang (2002):

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

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

3. For i=**bound**-1, ..., 0

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

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

For i=0, 1, ..., **bound-1**

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

where k is the iteration number; L is the user specified number of previous vectors used in the algorithm;. In this procedure, r_i is equal to the search direction vector given by $-\tilde{H}_i^{-1} \times g_i$.

Scaling can also be introduced into this algorithm (see Liu and Nocedal (1989)) by replacing $\tilde{H}_0^{-1}$ with $\gamma_k \tilde{H}_0^{-1}$ in step 3 of the recursion algorithm.

4.3 Nonlinear Conjugate Gradient

The nonlinear conjugate gradient method for minimization of non-quadratic function can be motivated by the linear conjugate gradient method which is normally used to solve a linear equation system. In this section, we focus on the application of the nonlinear conjugate gradient method to our history matching problem.

In the conjugate gradient method, the search direction is given by

$$d_{k+1} = -M_k^{-1} g_k + \beta_k d_k, \tag{4.33}$$

where k is the iteration index, g_k represents the gradient of the objective function, M_k is called the preconditioning matrix which is an approximation to the Hessian matrix H_k and β_k is obtained by the Polak-Ribière formula given by

$$\beta_k = \frac{r_{k+1}^T (M_{k+1}^{-1} r_{k+1} - M_k^{-1} r_k)}{r_k^T M_k^{-1} r_k}, \tag{4.34}$$

where $r_k = -g_k$. As discussed later, the step size can be obtained by a line search. If we choose the preconditioning matrix M_k to be identity matrix I, then Eq. 4.33 reduces to the standard conjugate gradient method without preconditioning.

It is well known that the nonlinear conjugate gradient method can be applied to minimize non-quadratic objective functions; see, Fletcher and Reeves (1964). Although the method has been applied for the history matching of production data (see, for example, Makhlouf et al. (1993)), its slow rate of convergence has precluded its use in large scale history matching problems. The success of the conjugate gradient method for nonlinear optimization depends on whether we are able to construct a good preconditioner. A good preconditioning matrix at the kth iteration is a matrix M_k which is a good approximation to the Hessian H_k so that

$$M_k^{-1} H_k \approx I. \tag{4.35}$$

For our problem, the Hessian at the kth iteration is given by

$$H_k = C_M^{-1} + G_k^T C_D^{-1} G_k, \tag{4.36}$$

An optional preconditioner for the conjugate gradient method would be

$$M_k = H_k, \tag{4.37}$$

but the conjugate gradient method requires solving the matrix problem

$$M_k \tilde{d}_k = -g_k, \tag{4.38}$$

to form search direction d_{k+1} using Eq. 4.33. If $M_k = H_k$, Eq. 4.38 requires the same computational effort as the direct application of Gauss-Newton method and does not improve computational efficiency. If we choose $M_k = C_M^{-1}$, however, then Eq. 4.38 becomes

$$\tilde{d}_k = -C_M g_k, \tag{4.39}$$

and the calculation of $\tilde{d}_k$ which is the first term in Eq. 4.33 requires only multiplication of g_k by the prior covariance matrix C_M. Kalita (2000) considered the problem of conditioning a gas reservoir model to well test pressure data by automatic history matching. Both the Gauss-Newton method and the conjugate gradient method with C_M^{-1} as the preconditioner were used to minimize the relevant objective function (Eq. 4.1 or Eq. 4.2). Kalita's results indicate that the conjugate gradient method was not always more efficient than the Gauss-Newton method. Moreover, in most cases, the conjugate gradient method converged to a value of the objective function which was significantly higher than the converged value of the objective function obtained by the Gauss-Newton method.

In the preconditioned conjugate gradient method, the preconditioning matrix M_k is used only in equations like Eq. 4.38. Thus, it is preferable to estimate M_k^{-1} directly instead of estimating M_k. We would like M_k^{-1} to be an approximation to the inverse Hessian. This suggests that $\tilde{H}_k^{-1}$ constructed from quasi-Newton might be a good candidate for a preconditioner. The difficulty with this procedure is that we can only approximate the quasi-Newton $\tilde{H}_k^{-1}$ using information in the conjugate gradient algorithm. Our work indicated that the preconditioner constructed by this scheme works better than C_M^{-1} for some cases, for example, in the gas reservoir examples shown by Zhang et al. (2001b); and works worse than C_M^{-1} for some cases, for example, in the three-phase example presented later. The reason is that the iterates generated by the quasi-Newton method are different from the iterates generated by the conjugate gradient method. The search direction for the conjugate gradient algorithm is given by Eq. 4.33 whereas it is given by Eq. 4.7 in the quasi-Newton method. Different search directions generate different iterates and in turn different y_k's and s_k's which are used to construct Hessian inverse approximation matrix $\tilde{H}_k^{-1}$. Therefore, the inverse Hessian approximation generated within the conjugate gradient algorithm will not be the same as the one generated in a quasi-Newton method. In particular, the "inverse Hessian approximation" generated with the conjugate gradient procedure may not be equal to the true inverse Hessian at the Nth iteration for a N-dimensional quadratic function even if the line search is exact; see Oren and Luenberger (1974) and Oren (1974).

4.4 Computational Requirements

Here we give a rough assessment of the computational efficiency of GN (Gauss-Newton), MLM (Modified Levenberg-Marquardt), PCG (preconditioned conjugate gradient), BFGS and LBFGS. Our assessment of the computational efficiency and memory requirements of the BFGS algorithm, and our implementation of the BFGS algorithm, are based on the update formula given in Eq. 4.12. In the evaluation of computational efficiency, we count only the number of adjoint solutions and the number of reservoir simulation runs required by each method. Moreover, we count one adjoint solution over the total time interval of a simulation run as one equivalent simulation run, even though in our implementation, an adjoint solution typically takes less than one half the time of a simulation run. Moreover, we do not account for computational savings that may be obtained by solving the adjoint system with multiple right-hand sides in cases where several sensitivity coefficients are calculated; see Wu et al. (1999). We also assume that only one iteration of the approximate line search algorithm is done. We provide only a summary of the results without details.

In GN and MLM, if the data are evenly distributed in the time domain, $N_d/2+1$ simulation runs are required at each GN or MLM iteration. In BFGS, LBFGS and PCG, the total computational cost of implementing one iteration is equivalent to 3 simulation runs. Thus, BFGS, LBFGS and PCG are $(N_d/2+1)/3$ times faster than GN and MLM for each iteration. For example, if we have 1000 data, one iteration of GN or MLM will require 167 times as much time as one iteration of BFGS, LBFGS and PCG.

Table 4.1 gives a rough estimate of the number of double precision real numbers used by each algorithm when applied to minimize the objective function of Eq. 4.1 or Eq. 4.2. (Recall that N_d is the number of production data, N_m is the number of model parameters, and L is the number of previous vectors used in the LBFGS algorithm.) Only the memory used by the algorithm itself is counted, e.g., the memory required to run the reservoir simulator is not included. For convenience, we use one memory unit to stand for the memory occupied by one double precision real number. From the results of Table 4.1, we see that the full-memory version of BFGS uses the most memory which is on the order of N_m^2; conjugate gradient uses the least memory which is on the order of N_m, and Gauss-Newton or Levenberg-Marquardt and limited memory BFGS have intermediate memory requirements. The memory used by limited memory BFGS depends on the number of previous vectors (denoted by L in Table 4.1) used to construct the update of the approximate inverse Hessian.

	No. of DP real numbers
GN/LM	$(2+2\times N_d)\times N_m$
CG	$10\times N_m$
PCG	$10\times N_m$ + memory for preconditioner
BFGS	$(12+N_m)\times N_m$
LBFGS	$(9+2L)\times N_m$

Table 4.1: Memory used by each algorithm.

4.5 Comparison of Scaling Methods in BFGS and LBFGS

At the end of subsection 4.2.3, we presented the procedure we normally use for scaling in BFGS and LBFGS. In this section, we present some of the computational examples we present a more detailed discussion of scaling procedures together with some computational results.

The self-scaling variable metric (SSVM) method developed by Oren and Luenberger (1974) and Oren (1974) is motivated by the desire to choose a γ_{k-1} so that the condition number of $R_k = H_k^{1/2}\tilde{H}_k^{-1}H_k^{1/2}$ is as close to one as possible. If $\tilde{H}_k^{-1}$ is identical to the inverse of the true Hessian, H_k, then this condition number is equal to one. For a quadratic objective function, these authors provide theoretical conditions and a method for computing γ_k that insure that (i) $\lambda_{\min} \leq 1 \leq \lambda_{\max}$ where $\lambda_{\min}$ and $\lambda_{\max}$, respectively, denote the minimum and maximum eigenvalues of R_k; and (ii) the condition number of R_{k+1} is less than or equal to the condition number of R_k. A quasi-Newton method which satisfies these two conditions is referred to as a self-scaling variable metric method. (Throughout, a quadratic objective function refers to a quadratic which has a Hessian matrix that is real symmetric positive definite.) For stability considerations, it is also desirable that the condition number of $\tilde{H}_k^{-1}$ not be too large, see Oren and Spedicato (1976). In particular, if $\tilde{H}_k^{-1}$ is a singular matrix, then for $l > k$, all m^l will be restricted to a subspace of N_m-dimensional Euclidean space; see Murray (1972). If the model which minimizes the objective function is not in this subspace, the algorithm can not converge to this model. Oren and Spedicato (1976) proposed an "optimal" conditioning of variable metric methods; specifically, they provide a procedure for calculating γ_k so that an upper bound for the condition numbers of $\tilde{H}_{k+1}^{-1}$ and $\tilde{H}_k\tilde{H}_{k+1}^{-1}$ is minimized. According to their results, one should actually consider switching between different update formulas from the general Broyden family from iteration to iteration. As for the results of Oren and Luenberger, these results assume that the quasi-Newton method is applied to a quadratic objective function and that an exact line search is performed at each iteration. From computational experiments, we have found that the general switching rule proposed by Oren and Spedicato exhibits poorer convergence properties than are obtained by applying the BFGS update at every iteration and then computing the optimal γ_k from the formula they provide. In a sense, however, our preferred method for computing γ_k for LBFGS discussed later represents a modification of the switching rule proposed by Oren and Spedicato.

The principal objection to SSVM algorithms that has been raised in the literature is that the sequence $\tilde{H}_{k+1}$ does not converge to the true inverse Hessian in n iterations in the case where the objective function is an n-dimensional quadratic, whereas the BFGS algorithm (in fact, all variable metric methods from the Broyden family) satisfy this quadratic termination property if $\gamma_k = 1$ for all $k > 0$. Motivated by this reasoning, Shanno and Phua (1978) suggested that one should only scale at the first iteration and then do no further scaling.

In our work, we have found the optimal condition of Oren and Spedicato (1976) is robust and efficient if the BFGS updating formula, Eq. 4.12, is applied at every iteration. For the

BFGS method, the Oren and Spedicato (1976) procedure for calculating γ_k reduces to

$$\gamma_k = \frac{s_k^T y_k}{y_k^T \tilde{H}_k^{-1} y_k}. \tag{4.40}$$

Shanno and Phua (1978) suggested scaling only $\tilde{H}_0^{-1}$ using γ_0 computed from Eq. 4.40 with $k = 0$ and then setting $\gamma_k = 1$ for all $k > 0$. This scheme is the one referred to as the self-scaling variable metric method in Yang and Watson (1988).

For the LBFGS algorithm, we have found an ad hoc variant of the optimal switching rule given by Oren and Spedicato (1976) which works well. Specifically, we compute

$$\tilde{\tau}_k = \frac{s_k^T \tilde{H}_0 s_k}{s_k^T y_k}, \tag{4.41}$$

$$\tilde{\sigma}_k = \frac{s_k^T y_k}{y_k^T \tilde{H}_0^{-1} y_k}, \tag{4.42}$$

and then determine the scaling factor γ_k by the following rule:

$$\gamma_k = \begin{cases} \tilde{\tau}_k & \text{if} \quad \tilde{\tau}_k < 1.0 \\ \tilde{\sigma}_k. & \text{otherwise.} \end{cases} \tag{4.43}$$

Again, we can scale only at the first iteration or scale at all iterations.

As discussed previously, in our applications, the prior covariance matrix is used as the initial approximation to the inverse Hessian, i.e., $\tilde{H}_0^{-1} = C_M$. A somewhat simpler procedure, which yields a less robust algorithm, is to set $\tilde{H}_0^{-1}$ equal to the diagonal matrix $\tilde{D}$ obtained from C_M by setting all off diagonal elements equal to zero. Even when $\tilde{H}_0^{-1} = C_M$, we have found that it is satisfactory to compute $\tilde{\tau}_k$ by

$$\tilde{\tau}_k = \frac{s_k^T \tilde{D}^{-1} s_k}{s_k^T y_k}, \tag{4.44}$$

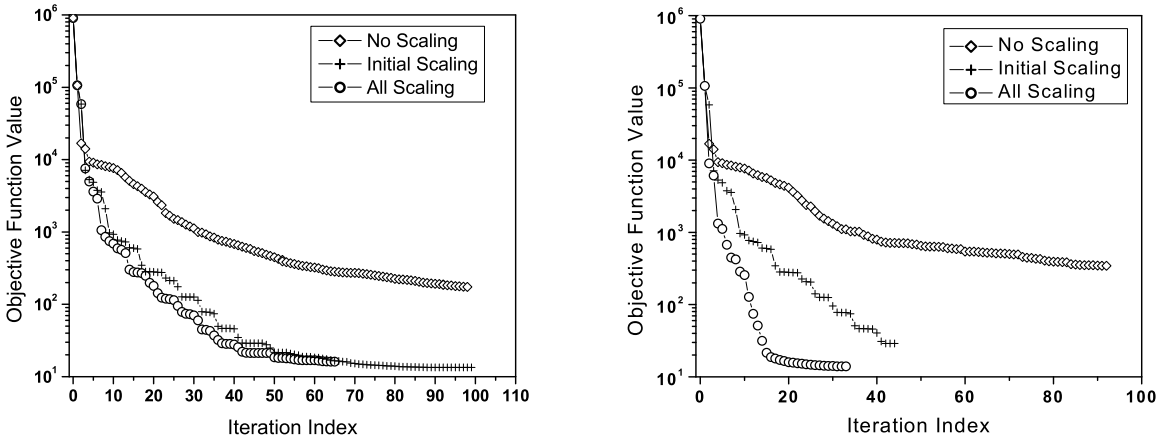
instead of using C_M^{-1} in place of $\tilde{D}^{-1}$ in Eq. 4.44. This method avoids solving the matrix problem $C_M x_k = s_k$ for x_k.

Figs. 4.1 (a) and (b) show the behavior of the objective function obtained by using BFGS and LBFGS, respectively, with and without scaling schemes. The results pertain to history matching WOR, GOR and pressure data for the 2D three-phase flow example considered later. In both figures, the diamonds represent the case where BFGS and LBFGS were applied without scaling.

In Fig. 4.1 (a), the scaling factor was computed from Eq. 4.40. The circular data point represent the case where scaling was done only at the first iteration and the plus signs

represent the case where scaling was done only at the first iteration. The behavior for the two scaling schemes is quite similar, although in the case where scaling was done only at the first iteration, several more iterations were done before the convergence criteria were satisfied. The most important thing to note about these results is that scaling does improve the convergence properties of the algorithm.

When the implementation of Nocedal (1980) is used in the LBFGS method, we can not use equations like Eqs. 4.40 to compute scaling parameters because $\tilde{H}_k^{-1}$ is never explicitly computed. Because of this, we use Eqs. 4.43 to compute γ_k. The results of Fig. 4.1 (b) are based on the application of Eq. 4.43 with Nocedal's algorithm for efficient implementation of the LBFGS algorithm. The diamond data points represent LBFGS results with no scaling, the circles represent results based on scaling at all iterations and plus signs represent results based on scaling only at the first iteration. Scaling greatly improves the efficiency of both the BFGS and the LBFGS algorithm. Here scaling at all iterations has a clear advantage over scaling at only the first iteration, and in the limited examples we have tried, we have found scaling at all iterations in the LBFGS algorithm is advantageous. It is important to note, however, that we have no rigorous theoretical basis for this result. Based on extensive computational experiments given in Zhang (2002), we have adopted the procedure represented by Eq. 4.43 to compute the scaling factors for the LBFGS algorithm.



(a) BFGS (b) LBFGS

Figure 4.1: Comparison of scaling procedures.

4.6 Results, LBFGS Versus Nonlinear Conjugate Gradient

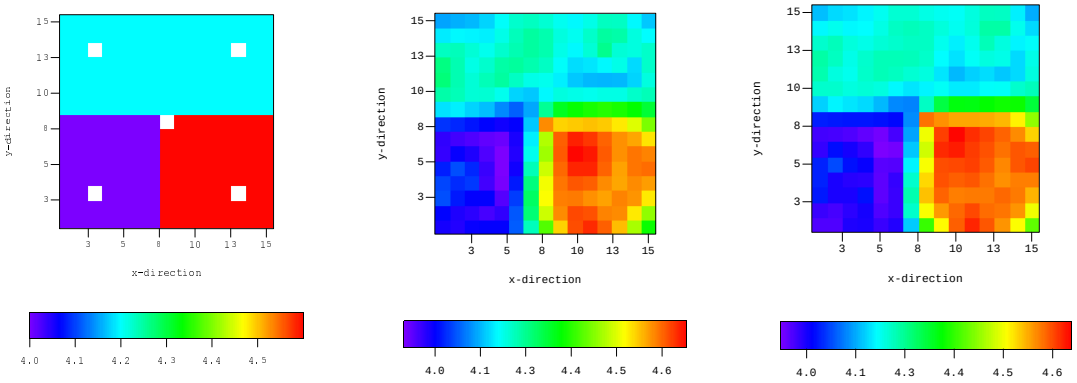
Here, two computational examples are presented to illustrate the behavior of the optimization algorithms.

4.6.1 3D Single Phase Gas Example

This example pertains to flow in a single-phase gas reservoir. Reservoir simulation was done on a $20 \times 20 \times 4$ grid. Two wells were completed in this reservoir. Well 1 was shut in for two days and then was produced at the rate of 4×10^4 Mscf/day for two days. Well 2 produced at the rate of 3.5×10^4 Mscf/day for the first two days and was then shut in for the following two days. We used 22 measured data from each well as conditioning data. Thus, a total of 44 data were history matched. The reservoir variables are the gridblock porosities, horizontal and vertical permeabilities and the well skin factors. The total number of reservoir variables is 4808. Stochastic simulation was done using the randomized maximum likelihood method. Ten realizations were generated using five different optimization algorithms: (i) MLM, (ii) preconditioned conjugate gradient (PCG) with C_M as the preconditioner (CM-PCG), (iii) PCG with the approximate inverse Hessian generated from the LBFGS equation used as the preconditioner (LBFGS-PCG), (iv) BFGS and (v) LBFGS. Table 4.2 gives the number of iterations required to obtain convergence and the value of the objective function at convergence for each minimization algorithm and for each realization. Results obtained by averaging the results for each set of ten conditional realizations are given in the last column of the table. From the results, we see that LBFGS and LBFGS-PCG behave similarly. The convergence properties of both of these algorithms are superior to CM-PCG and BFGS; i.e., LBFGS and LBFGS-PCG require fewer iterations to obtain convergence and yield a lower value of the objective function at convergence. Based on results presented in the computational requirements section, on average, LBFGS and LBFGS-PCG are about 7.7 times faster than MLM per iteration. As shown in Table 4.2, however, MLM requires about 2.3 times fewer iterations to converge. Therefore, on average, LBFGS and LBFGS-PCG are about 3.3 times faster than MLM for this particular example. For significantly larger problems, MLM would not even be feasible.

		R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	Average
MLM	Obj.	38	30	21	33	43	28	27	38	40	34	33.2
	No. Iter.	8	14	12	8	13	13	21	8	14	10	12
CM-PCG	Obj.	153	146	70	94	347	42	184	213	230	45	152
	No. Iter.	12	20	11	23	19	64	5	17	19	23	21.3
BFGS	Obj.	59	52	35	41	70	41	39	53	54	87	53
	No. Iter.	33	17	18	19	33	38	18	42	35	52	30.5
LBFGS-PCG	Obj.	44	51	29	38	57	33	36	47	54	36	42.5
	No. Iter.	23	17	23	31	23	38	21	35	20	30	26.1
LBFGS	Obj.	43	41	31	38	55	33	35	54	54	36	42
	No. Iter.	34	22	18	38	21	35	26	24	21	34	27.3

Table 4.2: Comparison between algorithms for a 3D single gas example.



(a) True (b) LBFGS-FCM (c) MLM

Figure 4.2: The log-permeability field.

4.6.2 2D Three Phase Example

This synthetic example pertains to a two-dimensional, three-phase flow problem simulated on $15 \times 15 \times 1$ grid. The gridblock porosities are fixed. The truth case, from which synthetic production data were generated, is shown in Fig. 4.2 (a). Note that there are three distinct zones with log-permeability uniform in each zone. This example has the advantage that the problem is small, so all methods require only modest computer resources. Moreover, because only three log-permeability values are involved, it is easy to visualize the quality

of the MAP estimate of log-permeability. The prior covariance matrix is generated from an isotropic spherical variogram with range equal to 6 gridblocks. Four producers are located near the four corners of the reservoir and one injector is located at the center. GOR, WOR and flowing bottomhole pressure (p_{wf}) data from the four producers and p_{wf} data from the injector are used as observed conditioning data. The total number of data is 364.

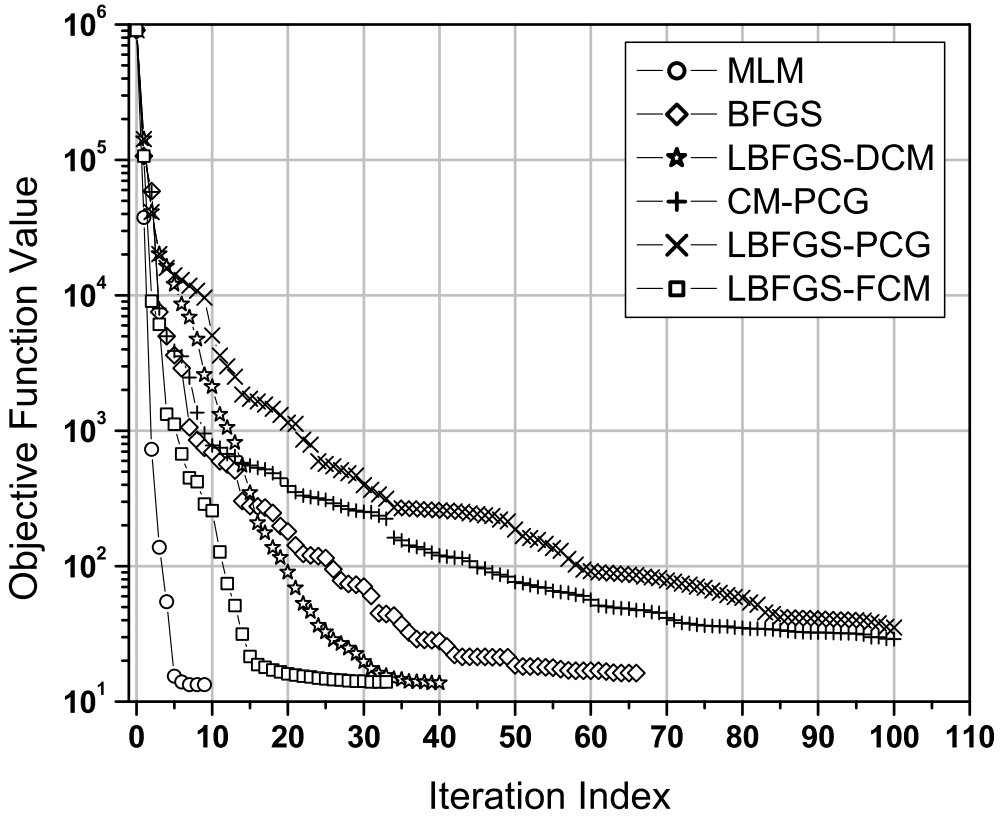


Figure 4.3: Behavior of the objective function.

MLM, CM-PCG, LBFGS-PCG, BFGS, LBFGS with the diagonal of C_M as the initial Hessian inverse approximation (LBFGS-DCM) and LBFGS with the full matrix of C_M as the initial Hessian inverse approximation (LBFGS-FCM) were tested for this problem. Fig. 4.3 shows the behavior of the objective function. As shown, MLM converged in 9 iterations to an estimate m such that $O(m) = 13.3$. CM-PCG and LBFGS-PCG, respectively, reduced the objective function from 906,670 to 28.9 and 35.2 in 100 iterations; at the hundredth iteration, CM-PCG and LBFGS-PCG had not converged based on the stopping criteria. The BFGS algorithm converged in 66 iterations to an estimate m such that $O(m) = 16.2$; LBFGS-DCM converged in 40 iterations to an estimate m such that $O(m) = 13.7$ and LBFGS-FCM

converged in 33 iterations to an estimate m such that $O(m) = 14.0$. Based on the discussion in the computational requirements section, for this example, LBFGS-FCM is expected to be roughly 61 times faster than MLM per iteration. However, MLM required about 4 times fewer iterations to converge. Therefore, LBFGS-FCM was expected to be about 16 times faster than MLM overall based on the approximate results of the computational requirements section. Table 4.3 shows the real CPU time used by the different algorithms. In terms of the real CPU time, optimization with the LBFGS-FCM algorithm was about 11 times faster than optimization with MLM. Fig. 4.2 (b) shows the log-permeability field obtained by LBFGS-FCM. We can see this model is very close to the log-permeability field obtained by the MLM method and is similar to the true model. The log-permeability field obtained with the LBFGS-DCM is not shown here but is much rougher than those shown because the smoothing effect of multiplication by C_M is lost when we use only the diagonal of C_M as the initial approximate inverse Hessian.

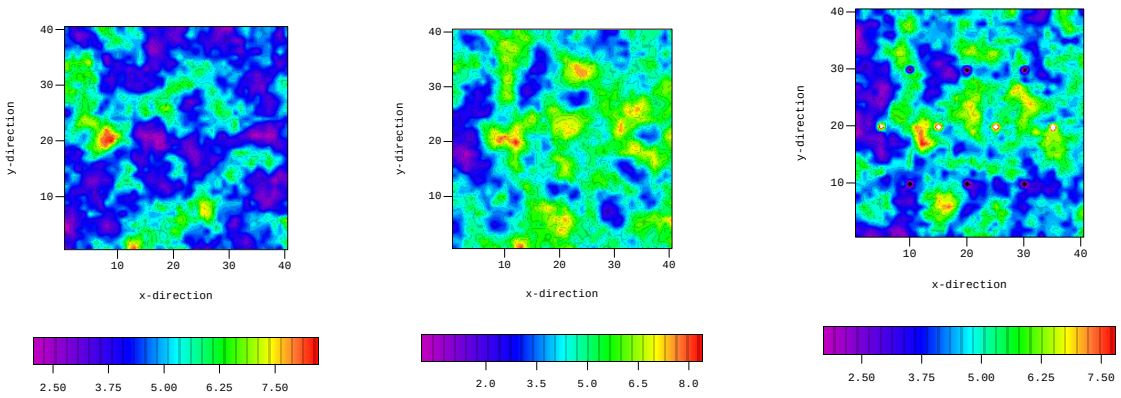
Also note that in this case, both preconditioned nonlinear conjugate gradient methods perform relatively poorly, and unlike in the gas reservoir example, generating an approximate inverse Hessian using LBFGS formulas gives a worse preconditioning matrix than simply using C_M as the preconditioner. We should note that when we apply the LBFGS formula in conjunction with the preconditioned CG algorithm, the y_k's and g_k's used in the formula are computed from the CG equations and are hence different than those that would be obtained with the actual LBFGS method.

Algorithms	Scaling Scheme	CPU time (seconds)
MLM	N/A	2930
CM-PCG	N/A	887
LBFGS-PCG	All Scaling	904
BFGS	Initial Scaling	923
LBFGS-DCM	All Scaling	279
LBFGS-FCM	All Scaling	263

Table 4.3: Comparison of the CPU time used by different minimization algorithms.

4.6.3 3D Three Phase Example

In this example, a 3D three-phase flow problem on a $40 \times 40 \times 6$ grid is considered. The true log-permeability field is an unconditional realization generated by Gaussian co-simulation.



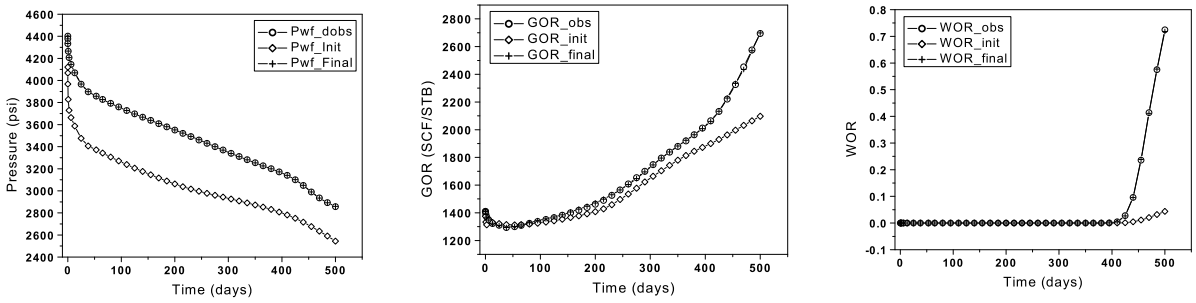
(a) Unconditional (b) Conditional (c) True

Figure 4.4: Unconditional, conditional and true realizations of the log-permeability field in the top layer.

The top layer of the true log-permeability field is shown in Fig. 4.4 (c). The porosity field is fixed. Six producers and four water injection wells are completed in the reservoir. Wellbore pressure (p_{wf}), GOR and WOR data from the producers and p_{wf} data from the injectors are used as conditioning data; the total number of data is 880. Fig. 4.4 (a) shows the top layer of the unconditional realization of the model which was used as the initial guess in the history-matching process. Fig. 4.4 (b) shows the corresponding layer of the model obtained by history matching the production data. Optimization was done with the scaled LBFGS algorithm. The objective function was reduced from a initial value of 3×10^8 to 675 in 70 iterations. Note that the conditional realization obtained by history matching captures the main features of the true model.

Figs. 4.5 (a) through (c) show the data match for the pressure, GOR and WOR from one producer. In all these figures, diamonds represent the data obtained from the unconditional realization, i.e., the initial model, circles represent the observed data and the plus signs represent the data obtained from the model which was obtained by history matching all observed data. Note that good matches were obtained. Matches of similar quality were obtained at all wells. It is important to note that the size of the problem precluded the application of MLM and the standard BFGS algorithm with the personal computer used for the study.

The rest of this chapter is devoted to a discussion of modifications we made to the LBFGS algorithm during the last year of the project. These modifications yield a more



(a) Pressure at producer (b) GOR (c) WOR

Figure 4.5: Data matches.

robust and computationally efficient procedure for history matching of large scale problems especially when the initial data mismatch is large. The same scaling procedure (Eq. 4.31 and stopping criteria (Eqs. 4.27 and 4.28) are used. The differences is that we use a modified line search and introduce damping and constraints to control overshooting and undershooting. Undershooting and overshooting refer to obtaining unreasonably large or small value of the model parameters.

4.7 Improved Line Search

After the LBFGS search direction d_k has been computed from

$$d_k = -\tilde{H}_k^{-1}\nabla O(m_k) = -\tilde{H}_k^{-1} g_k, \tag{4.45}$$

the step size is computed by a line search procedure. An exact line search would determine the step size α_k as the value of α that minimizes $\phi(\alpha)$ defined by

$$\phi(\alpha) = O(m_k + \alpha d_k), \tag{4.46}$$

i.e.,

$$\alpha_k = \operatorname{argmin} O(m_k + \alpha d_k). \tag{4.47}$$

4.7.1 Strong Wolfe Conditions

For the line search strategy, we should impose some conditions on the step size α_k to guarantee that it provides a sufficient decrease of the objective function. Moreover, to prove that

the LBFGS method converges, one must impose conditions to ensure that the line search is done accurately enough so that the new gradient and model update information will give an updated approximation to the inverse Hessian that is positive definite. One standard way to ensure a positive definite approximate inverse Hessian and guarantee convergence is to require that the line search satisfy the strong Wolfe conditions, (Dennis and Schnabel, 1996; Fletcher, 1987; Nocedal and Wright, 1999). The strong Wolfe conditions require the step size α_k to satisfy

$$O(m_k + \alpha_k d_k) \leq O(m_k) + c_1 \alpha_k [\nabla O(m_k)]^T d_k, \tag{4.48}$$

$$|[\nabla O(m_k + \alpha_k d_k)]^T d_k| \leq c_2 |[\nabla O(m_k)]^T d_k|, \tag{4.49}$$

where $0 < c_1 < 0.5$ and $c_1 < c_2 < 1$. With $\phi(\alpha)$ defined by Eq. 4.46, Eqs. 4.48 and 4.49 may be rewritten as

$$\phi(\alpha_k) \leq \phi(0) + c_1 \alpha_k \phi'(0), \tag{4.50}$$

$$|\phi'(\alpha_k)| \leq c_2 |\phi'(0)|. \tag{4.51}$$

The first strong Wolfe condition (Eq. 4.48 or 4.50) essentially stipulates that the step size α_k should give a sufficient decrease in the objective function but actually only guarantees this if the step size is bounded away from zero. In Fig. 4.6, the solid curve represents the objective function $\phi(\alpha)$; the dotted line connecting the solid squares represents the line $\phi_1(\alpha) = \phi(0) + c_1 \phi^{'}(0)\alpha$. The first strong Wolfe condition (Eq. 4.48 or 4.50) states that step size α is acceptable only if $\phi(\alpha) \leq \phi_1(\alpha)$. The first strong Wolfe condition is not enough by itself to ensure that the objective function decreases sufficiently, because, as shown in Fig. 4.6, it is satisfied for all sufficiently small step sizes. An unacceptably small step size is ruled out by applying the second strong Wolfe condition or curvature condition (Eq. 4.49 or 4.51). The two dashed lines connecting open circles in Fig. 4.6 illustrate the second strong Wolfe condition. These two lines are the tangent lines to the curve by $\phi(\alpha)$ at the points α_1 and α_2 where the points α_1 and α_2 are points such that $\phi^{'}(\alpha_1) = c_2 \phi^{'}(0)$ and $\phi^{'}(\alpha_2) = -c_2 \phi^{'}(0)$. The second strong Wolfe condition states that a step size α is acceptable only if $|\phi^{'}(\alpha)| \leq c_2 |\phi^{'}(0)|$, i.e., α is in the interval $[\alpha_1, \alpha_2]$. The smaller the value of c_2, the smaller the distance between α_1 and α_2. Thus, choosing a smaller value for c_2 makes the acceptable step size closer to the value of minimizer, however, we expect if we decrease c_2, it will require more iterations of the line search algorithm to obtain an α that satisfies the second strong Wolfe condition. The choice of c_2 is ad hoc. Nocedal and Wright (1999) recommend $c_2 = 0.9$ for quasi-Newton algorithms, and we use this value in the example presented later. In the first strong Wolfe condition, we use $c_1 = 10^{-4}$.

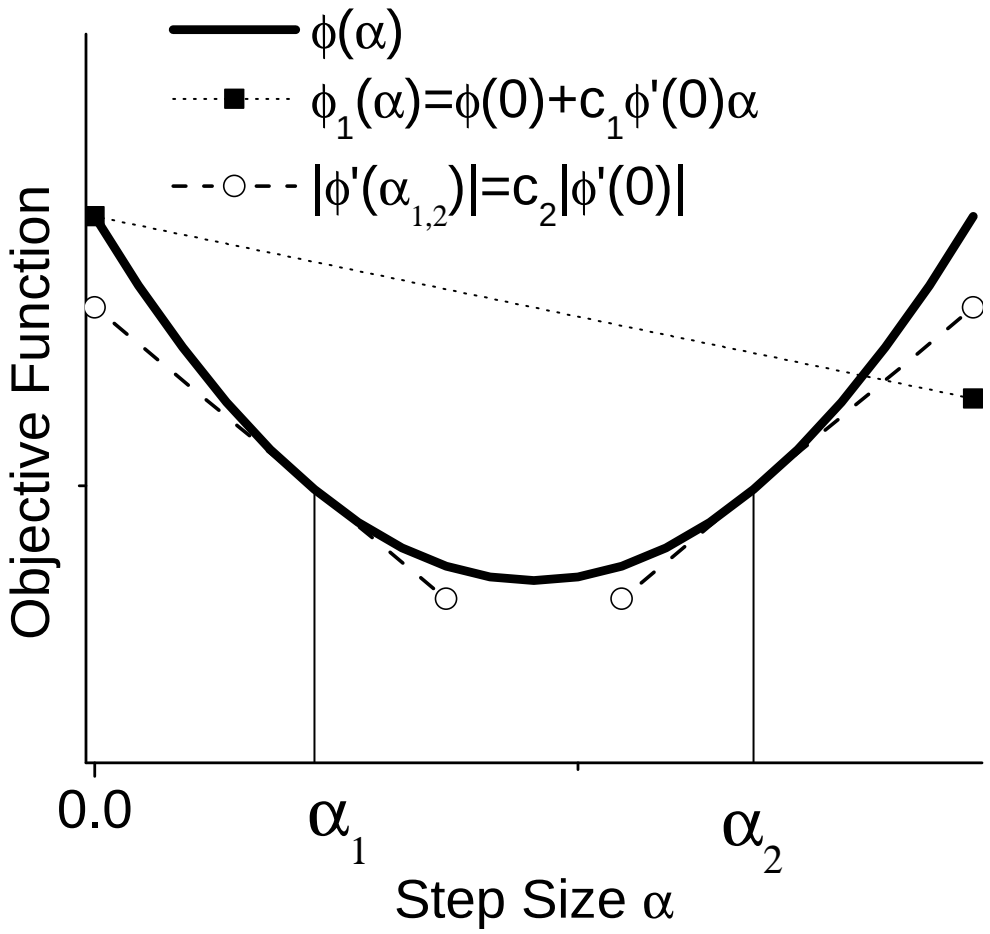


Figure 4.6: Behavior of the objective function.

With the search direction given by Eq. 4.45 and α_k determined by an approximate line search, the new estimate of the model that mimimizes $O(m)$ is given by

$$m_{k+1} = m_k + \alpha_k d_k. \tag{4.52}$$

With s_k defined by

$$s_k = m_{k+1} - m_k = \alpha_k d_k, \tag{4.53}$$

and y_k defined by

$$y_k = \nabla O_{k+1} - \nabla O_k, \tag{4.54}$$

the inverse Hessian generated by the LBFGS algorithm will be guaranteed to be positive

definite if

$$y_k^T s_k = s_k^T y_k > 0; \tag{4.55}$$

(Dennis and Schnabel, 1996; Fletcher, 1987; Nocedal and Wright, 1999). Thus, it is critical to ensure that the line search generate a stepsize α_k such that Eq. 4.55 is satisfied.

We now show that Eq. 4.55 is satisfied if the second strong Wolfe condition holds. Let $\nabla O_{k+1} = \nabla O(m_{k+1}) = \nabla O(m_k + \alpha_k d_k)$. From the second strong Wolfe condition, we know $[\nabla O_{k+1}]^T d_k \geq c_2 [\nabla O_k]^T d_k$ since $[\nabla O_k]^T d_k < 0$ when d_k is a downhill direction. It follows that

$$\begin{aligned} y_k^T s_k &= [\nabla O_{k+1} - \nabla O_k]^T \alpha_k d_k \\ &= \alpha_k [\nabla O_{k+1}]^T d_k - \alpha_k [\nabla O_k]^T d_k \\ &\geq (c_2 - 1)\alpha_k [\nabla O_k]^T d_k. \end{aligned} \tag{4.56}$$

Thus $y_k^T s_k > 0$ is satisfied if we choose $c_2 < 1$ and the second strong Wolfe condition holds. Thus, the second strong Wolfe condition not only guarantees a sufficient decrease of the objective function, but also guarantees that the updated Hessian inverse $\tilde{H}_{k+1}^{-1}$ is positive definite. It also indicates that if the second strong Wolfe condition fails at one iteration, the updated Hessian inverse, $\tilde{H}_{k+1}^{-1}$, may not be positive definite and may result in an uphill search direction at the next iteration.

In the our first line search method, which was described in 4.2.1, one iteration with Newton-Raphson is done and if the resulting value of α does not satisfy the strong Wolfe conditions, a quadratic fit is done. If $\phi(\alpha) = \nabla O(m_k + \alpha d_k)$ is a convex function, its minimizer can be approximated by applying the Newton-Raphson algorithm to find a zero of the derivative of $\phi(\alpha)$. With an initial guess of zero, one iteration of the Newton-Raphson algorithm for finding a zero of $h(\alpha) = \phi'(\alpha)$ is given by

$$\alpha^1 = -\frac{h(0)}{h'(0)}. \tag{4.57}$$

If the Newton-Raphson method converges to a value $\hat{\alpha}$ such that $\phi'(\hat{\alpha}) = 0$, there is no guarantee that $\hat{\alpha}$ minimizes $\phi(\alpha)$ unless we know $\phi(\alpha)$ is a convex function.

At any point in the line search procedure, a quadratic fit can be used to estimate a new step size. Letting α_k be the estimate of the stepsize from Newton-Raphson iteration(s), we fit a quadratic through $\phi(0)$, $h(0) = \phi'(0)$ and $\phi(\alpha_k)$. The value of α at which this quadratic is zero is given by

$$\alpha_k^{\text{new}} = -\frac{h(0)\alpha_k^2}{2[\phi(\alpha_k) - \phi(0) - h(0)\alpha_k]}, \tag{4.58}$$

where $\phi(\alpha_k) = O(m_k + \alpha_k d_k), \phi(0) = O(m_k),\ h_k(0) = [\nabla O(m_k)]^T d_k$. Similar to Newton-Raphson iteration, we can not guarantee that α_k^{new} corresponds to a minimum unless the quadratic is convex.

By definition, the objective function is strongly convex if

$$\phi(\alpha) > \phi(0) + \phi'(0)\alpha. \tag{4.59}$$

If $\phi(\alpha)$ is strongly convex, as is the case for the function $\phi_1(\alpha)$ shown in Fig. 4.7, then Eq. 4.57 would yield a positive step size as would Eq. 4.58. The curve through open circles is the derivative of the objective function $\phi_1(\alpha)$ and we see this line crosses the α axis at the point corresponding to the minimum of $\phi_1(\alpha)$. For such a function, the line search procedure given in subsection 4.2.1 is adequate. However, a negative or even an infinite step size may be obtained if the objective function is not strongly convex along the search direction. If the objective function $\phi(\alpha)$ is a linear function, as shown by the line with pluses in Fig. 4.7, then $[\phi(\alpha_k) - \phi(0) - h(0)\alpha_k] = 0$, and an infinite solution is obtained from Eq. 4.58. If the objective function is concave in a neighborhood of m_k, then $[\phi(\alpha_k) - \phi(0) - h(0)\alpha_k] < 0$ may occur for small values of α_k as is the case for the curve $\phi_2(\alpha)$ shown by solid triangles in Fig. 4.7. In this case, a negative step size $(\alpha_k, 0)$ can be obtained from Eq. 4.57 and 4.58. The curve through open triangles is the derivative of the objective function $\phi_2(\alpha)$.

For the history matching problems of interest to us, we have encountered two common situations where the objective function may be non-convex. The first case pertains to the situation where the current model, m_k, is close to the "critical" point at which gas or water breaks through in one or more of the production wells. In this case, if a step size moves the model across the critical point, then the objective function may decrease more quickly as the step size increases because the GOR or WOR mismatch term may decrease dramatically. A non-convex objective function may also occur when the well production condition changes from constant bottom hole pressure to a constant production rate. In this case, the pressure mismatch part of the objective function will decrease more rapidly as the step size increases.

As noted earlier, when the second strong Wolfe condition does not hold, the updated Hessian inverse approximation $\tilde{H}_{k+1}^{-1}$ may be not positive definite, and the search direction may become uphill. In this case, if the line search procedure of subsection 4.2.1 is used, the LBFGS algorithm is restarted with the first search direction given by the initial inverse Hessian approximation times the negative of the gradient. A restart leads to several iterations where only a small decrease in the objective function is obtained and this leads to a very inefficient procedure that requires more iterations than are desirable.

Fig. 4.8 shows one example of the consequence failing to guarantee the Wolfe conditions. The results presented pertain to the PUNQ history matching example discussed in the next

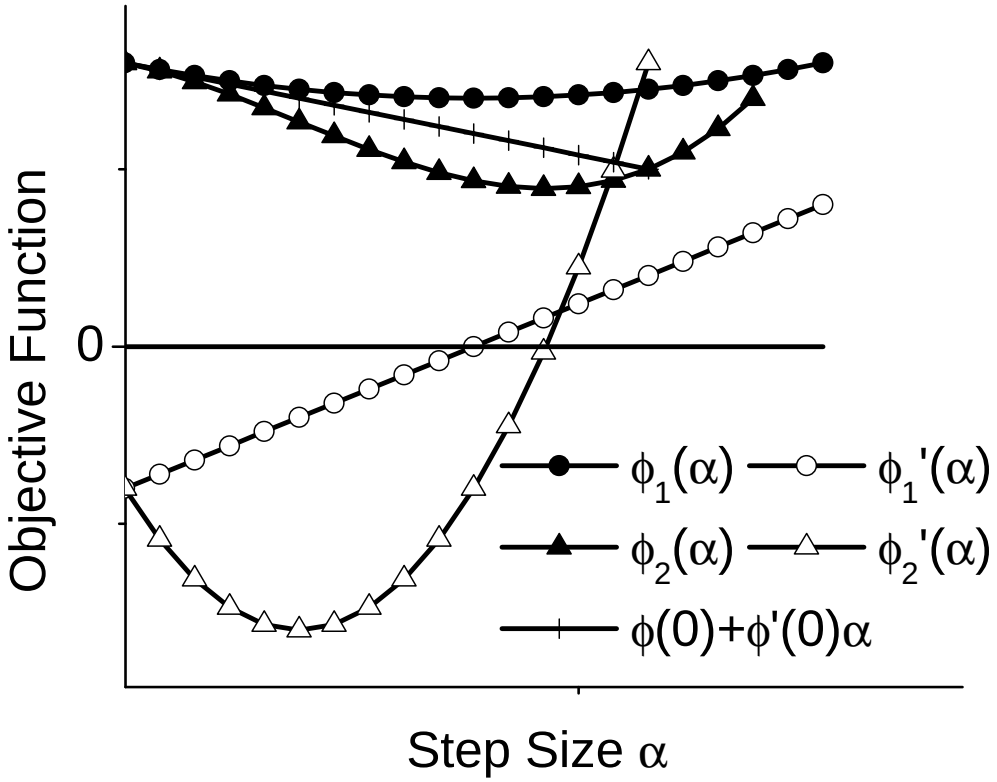


Figure 4.7: Behavior of the objective function.

chapter. Three curves are plotted in this figure. The curve through open triangles is the plot of the step size calculated at each iteration of the LBFGS algorithm with the old line search procedure. In this history-matching problem, the line search method of subsection 4.2.1 was used. The curves through solid circles and open circles, respectively, denote the infinity norm and 2-norm of the search directions vector. We can see that the step size, α_k, may change from 10^{-10} to 1. In this figure, the magnitude of $\|d_k\|$ and α_k show a big jump at the 38th iteration and at the 57th iteration. We checked the output data file and found that the Wolfe conditions failed to hold at one iteration before each of these two iterations. This failure to satisfy the Wolfe conditions resulted in a non-positive definite updated Hessian inverse, at iterations $k = 38$ and $k = 57$, and thus, the search direction became uphill. The program is then restarted by setting $k = 0$ in the LBFGS algorithm. Thus, all the useful information on the approximate inverse Hessian is lost, and the iterative process restarts with a search direction equal to the steepest descent direction preconditioned by $\tilde{H}_0^{-1} = C_M$. Then the

step size became very small, and the convergence criteria were satisfied very soon after 57th iterations.

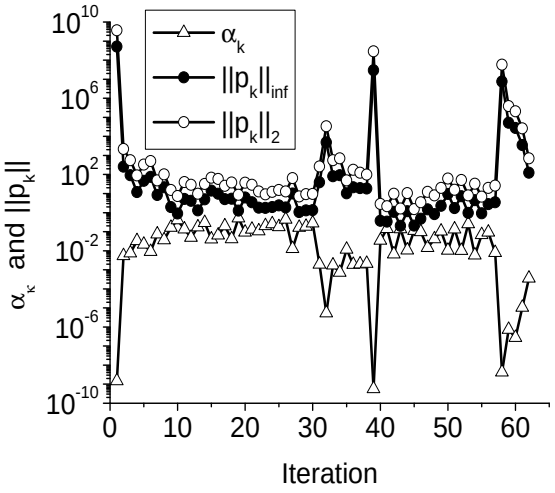


Figure 4.8: Step size and the norm of search direction.

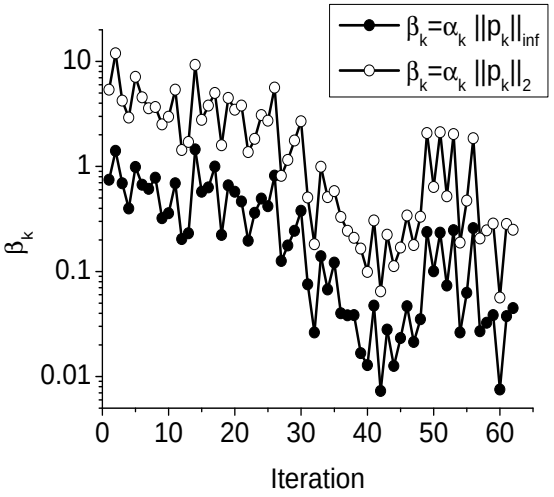


Figure 4.9: Normalized step size plot.

4.7.2 The New Line Search Algorithm

The improved line search procedure is developed in detail in this subsection. The procedure involves an method for generating an initial guess for the stepsize followed by a procedure to improve the approximation to the optimal step size.

Initial Step Size Algorithm.

The idea is based on the assumption that a reasonable estimate of an appropriate step size for the current iteration can be obtained from the information at the previous iteration. If the initial step size satisfies the strong Wolfe conditions, there is nothing left to do; the step size is accepted, the model is updated and we proceed to the next iteration of the LBFGS algorithm. If the initial step size does not satisfy the strong Wolfe conditions, the modified line search procedure discussed in the next sub-subsection is employed.

The motivation for generating an initial step size is based on the results of experiments shown in Fig. 4.8. From an examination of the results of Fig. 4.8, we find that the norm of model difference between two successive iterations usually does not change substantially,

i.e.,

$$\|m_{k+1} - m_k\| \approx \|m_k - m_{k-1}\|, \tag{4.60}$$

or, equivalently,

$$\alpha_k \|d_k\| \approx \alpha_{k-1} \|d_{k-1}\|. \tag{4.61}$$

We define β_k by

$$\beta_k = \alpha_k \|d_k\|, \tag{4.62}$$

and refer to β_k as the normalized step size. The normalized search direction is defined as

$$d_{N,k} = \frac{d_k}{\|d_k\|}. \tag{4.63}$$

Eq. 4.53 may be rewritten as

$$m_{k+1} - m_k = \beta_k d_{N,k} = \alpha_k d_k. \tag{4.64}$$

The modified line search algorithm is based on the normalized step size, β_k, and normalized search direction, $d_{N,k}$, instead of the original step size, α_k, and original search direction d_k. Because the normalized step size often does not change very much between two successive iterations, we can easily update it based on its value at the previous iteration, i.e., we can generate a good initial approximation for the step size for the line search from the step size used at the previous iteration.

Fig. 4.9 shows a plot of the normalized step size, β_k, versus the iteration number. In this figure, the data points with solid circles are the normalized step sizes based on the infinity-norm, and the plot with open circles represents results based on the 2-norm. Compared to the values of α_k in Fig. 4.8, the values of β_k vary much less from iteration to iteration. (In our procedure, the infinity norm is used to calculate the normalized search direction.) Although one could simply use the value of β_k from the preceding iteration as the initial step size for the current iteration, experiments indicate it is better to modify the value by the procedure discussed in the following paragraph.

Define h_k by

$$h_k = h_k(\beta_k) = [\nabla O(m_k + \beta_k d_{N,k})]^T d_{N,k}, \tag{4.65}$$

which represents the directional derivative of O at $m_k + \beta_k d_{N,k}$. In particular, $h_k(0) = [\nabla O(m_k)]^T d_{N,k}$ represents the directional derivative of $O(m_k)$ in the direction d_k. If $h_k(\beta_k) < 0$, it indicates that the normalized step size used at kth iteration, β_k, is not large enough and should be increased. On the other hand, if $h_k(\beta_k) > 0$, it implies that the normalized step size used at kth iteration should be decreased. Procedures for modifying or updating β_k are

discussed below. If $\phi(\beta) = O(m_k + \beta d_{N,k})$ is a quadratic function of β, then $h_k(\beta) = \phi'(\beta)$ is a linear function of β. Thus, it is reasonable to use the line through 0 and β_k as the equation for $h_k(\beta)$ and select the modified normalized step size, β_k^*, as the solution of $h_k(\beta) = 0$ which is given by

$$\beta_k^* = \beta_k \left(\frac{h_k(0)}{h_k(0) - h_k(\beta_k)} \right). \tag{4.66}$$

We define the step size modification factor by $\eta_k = \beta_k^*/\beta_k$. With β_k^* given by Eq. 4.66, it follows that

$$\eta_k = \frac{\beta_k^*}{\beta_k} = \frac{h_k(0)}{h_k(0) - h_k(\beta_k)}. \tag{4.67}$$

To satisfy the second strong Wolfe's condition, we must have $c_2 h_k(0) \leq h_k(\beta_k) \leq -c_2 h_k(0)$; it follows that the step size modification factor needs to be between $1/(1+c_2)$ and $1/(1-c_2)$. If c_2 is very close to 1, then η_k may become very large. To avoid a large value of η_k, an upper bound for η_k is specified; we use $\eta_{k,max} = 2$ in our algorithm. If $\eta_k > \eta_{k,max}$ we set $\eta_k = \eta_{k,max}$. Finally, $\eta_k \beta_k$ is chosen as the initial guess of the normalized step size for the next iteration, i.e.,

$$\beta_{k+1} = \beta_k \eta_k. \tag{4.68}$$

Updating the Step Size.

In order to obtain a step size that satisfies the strong Wolfe conditions, we introduce an improved step size fit method. We make two assumptions about the objective function to ensure along each downhill search direction, there exists a interval in which the strong Wolfe conditions hold. The assumptions are that (1) the objective function $O(m)$ and its gradient $\nabla O(m)$ exist and are continuous functions of the model parameters, and (2) along any downhill search direction d_k, the limit of the directional derivative as the step size goes to infinity is nonnegative, i.e.,

$$\lim_{\alpha \to \infty} [\nabla O(m_k + \alpha d_k)]^T d_k \geq 0. \tag{4.69}$$

The first assumption is required by the quasi-Newton algorithm. This assumption also guarantees that along any search direction, d_k, both $\phi(\alpha) = O(m_k + \alpha d_k)$ and $h(\alpha) = \phi'(\alpha)$ are continuous functions of the step size α. The second assumption (Eq. 4.69) eliminates the possibility that $[\nabla O(m_k + \alpha d_k)]^T d_k < 0$ as $\alpha \to \infty$, and prevents the objective function from approaching $-\infty$ as $\alpha \to \infty$. (Eq. 4.69 holds for history matching problems because the objective function is always positive. In fact, when a prior model is integrated as a regularization term, $[\nabla O(m_k + \alpha d_k)]^T d_k$ will approach positive infinite when $\alpha \to \infty$.)

In the following discussion, we use the normalized step size β, instead of the original step size α and let $\beta_{k,1}$ and $\beta_{k,2}$ be the two latest approximation to the step size at the current iteration where neither step size results in satisfaction of the strong Wolfe conditions. Thus, we need to calculate a new step size using information on the two values of the objective function along the search direction, $\phi(\beta_{k,1})$ and $\phi(\beta_{k,2})$, and the two corresponding values of the directional derivative, $h(\beta_{k,1})$ and $h(\beta_{k,2})$. Without loss of generality, we assume that $\beta_{k,2} > \beta_{k,1}$. Since the strong Wolfe conditions are not satisfied at $\beta_{k,2}$, we will update both $\beta_{k,1}$ and $\beta_{k,2}$ based on the ratio of the two directional derivatives and the values of the objective function at these two step sizes. In discussing the specific procedure, we let

$$\rho = \frac{h(\beta_{k,2})}{h(\beta_{k,1})}. \tag{4.70}$$

- Case 1: $\rho < 0$.

  In this case, $h(\beta_{k,2}) > 0$ and there is at least one minimum in the interval $[\beta_{k,1}, \beta_{k,2}]$, see Fig. 4.10. In Fig. 4.10, the solid curve represents the objective function $\phi(\beta)$, the dashed curve represents its derivative $h(\beta) = \phi'(\alpha)$. The equation of the line through the points $(\beta_{k,1}, h(\beta_{k,1}))$ and $(\beta_{k,2}, h(\beta_{k,2}))$ is shown by the dotted line in Fig. 4.10 and is given by

  $$\tilde{h}(\beta) = h_{\beta_{k,1}} + \left(\frac{h_{\beta_{k,2}} - h_{\beta_{k,1}}}{\beta_{k,2} - \beta_{k,1}} \right) (\beta - \beta_{k,1}). \tag{4.71}$$

  Solving $\tilde{h}(\beta) = 0$ for $\beta_{k,\text{new}}$ yields

  $$\beta_{k,\text{new}} = \beta_{k,2} + \frac{\rho}{1-\rho}(\beta_{k,2} - \beta_{k,1}). \tag{4.72}$$

  If $|\rho|$ is very small, $\beta_{k,\text{new}}$ will be very close to $\beta_{k,2}$. Similarly, if $|\rho|$ is very large, then, $\beta_{k,\text{new}}$ will be very close to $\beta_{k,1}$, and our new estimate does not yield much improvement. The step size changes very little for both cases. To avoid such problems, minimum and maximum values of ρ should be specified. In our implementation, we specify $\rho_{\min} = -4$, $\rho_{\max} = -0.25$. Then we set $\rho = \rho_{\min}$ when $\rho < \rho_{\min}$, and $\rho = \rho_{\max}$ when $\rho > \rho_{\max}$ when computing $\beta_{k,\text{new}}$ from Eq. 4.72.

  If the strong Wolfe conditions are not satisfied at the new step size $\beta_{k,\text{new}}$, set $\beta_{k,2} = \beta_{k,\text{new}}$ and replace $\phi(\beta_{k,2})$ and $h(\beta_{k,2})$, respectively, with $\phi(\beta_{k,\text{new}})$ and $h(\beta_{k,new})$, respectively, if $h(\beta_{k,new}) > 0$. If $h(\beta_{k,new}) < 0$, we set $\beta_{k,1} = \beta_{k,\text{new}}$ and replace $\phi(\beta_{k,1})$ and $h(\beta_{k,1})$, respectively with $\phi(\beta_{k,\text{new}})$ and $h(\beta_{k,\text{new}})$.

  We repeat this iterative procedure (Eq. 4.72) until the strong Wolfe conditions are satisfied.

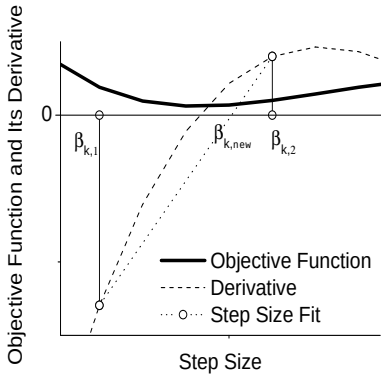


Figure 4.10: Step size fit for $\rho < 0$, case 1.

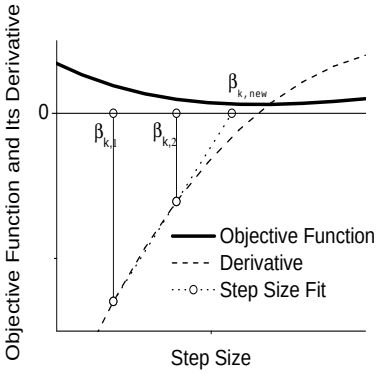


Figure 4.11: Step size fit for $0 < \rho < 1$, case 2.

- Case 2: $\rho > 0$ and $\phi(\beta_{k,2}) < \phi(\beta_{k,1}) \leq \phi(0)$.

  In this case, the objective function will decrease if the step size is increased. The linear fit equation Eq. 4.72 can still be used to calculate the new step size provided that $\rho < 1$, see Fig. 4.11. From Eq. 4.72, we see that $\beta_{k,\text{new}}$ will be very close to $\beta_{k,2}$ when ρ is very small, and $\beta_{k,\text{new}}$ becomes very large when $\rho \to 1$. Thus, similar to Case 1, we specify minimum and maximum allowable values for ρ with $0 < \rho_{\min} < \rho_{\max} < 1$. In our implementation, we specify $\rho_{\min} = 0.25$ and $\rho_{\max} = 0.75$. Then we set $\rho = \rho_{\max}$ when $\rho > \rho_{\max}$, and set $\rho = \rho_{\min}$ when $\rho < \rho_{\min}$. Also note Eq. 4.72 will generate a negative step size when $\rho > 1$. (Fig. 4.12 shows a case where $\rho > 1$.) However, we know that the objective function will decrease more as step size increases, and there must exist an interval beyond $\beta_{k,2}$ on which the strong Wolfe conditions are satisfied. Thus, if $\rho > 1 > \rho_{\max}$, the new step size is calculated with Eq. 4.72 by setting $\rho = \rho_{\max}$. In Fig. 4.12, $h_{\max} = \rho_{\max} h(\beta_{k,1})$.

  If the strong Wolfe conditions are not satisfied at the new step size $\beta_{k,new}$, we update $\beta_{k,1}$ and $\beta_{k,2}$ with the following procedure:

  If $h(\beta_{k,\text{new}}) > 0$, set $\beta_{k,1} = \beta_{k,2}$ and replace $\phi(\beta_{k,1})$ and $h(\beta_{k,1})$, respectively, with $\phi(\beta_{k,2})$ and $h(\beta_{k,2})$; also set $\beta_{k,2} = \beta_{k,\text{new}}$ and replace $\phi(\beta_{k,2})$ and $h(\beta_{k,2})$, respectively, with $\phi(\beta_{k,\text{new}})$ and $h(\beta_{k,\text{new}})$ and then apply the step size calculation procedure of Case 1 until the strong Wolfe conditions are satisfied, as shown in Fig. 4.12. In Fig. 4.12, $\beta_{k,\text{new1}}$ is obtained with Eq. 4.72 by setting $\rho = \rho_{\max}$, and $\beta_{k,\text{new2}}$ is obtained by replacing β_1 with β_2, and replacing β_2 with $\beta_{k,\text{new1}}$.

  If $h(\beta_{k,\text{new}}) < 0$ set $\beta_{k,1} = \beta_{k,2}$ and replace $\phi(\beta_{k,1})$ and $h(\beta_{k,1})$, respectively, with

$\phi(\beta_{k,2})$ and $h(\beta_{k,2})$; set $\beta_{k,2} = \beta_{k,\text{new}}$ and replace $\phi(\beta_{k,2})$ and $h(\beta_{k,2})$, respectively, with $\phi(\beta_{k,\text{new}})$ and $h(\beta_{k,\text{new}})$; and then repeat the step size calculation procedure of Case 2 until the strong Wolfe conditions are satisfied or the conditions of Case 1 or Case 3 are satisfied.

- Case 3: $\rho > 0$ and $\phi(\beta_{k,2}) \geq \phi(\beta_{k,1})$.

  In this case, the objective function has both a maximum and a minimum in the interval $[\beta_{k,1}, \beta_{k,2}]$, see Fig. 4.13. The new step size is calculated as

  $$\beta_{k,\text{new}} = \beta_{k,1} + 0.5(\beta_{k,2} - \beta_{k,1}). \tag{4.73}$$

  (a) If at the new step size $\beta_{k,\text{new}}$, $h(\beta_{k,\text{new}}) > 0$, as shown in Fig. 4.13, set $\beta_{k,2} = \beta_{k,new}$, replace $\phi(\beta_{k,2})$ and $h(\beta_{k,2})$, respectively, with $\phi(\beta_{k,\text{new}})$ and $h(\beta_{k,\text{new}})$; and then apply the step size calculation procedure of Case 1 until the strong Wolfe conditions are satisfied.

  (b) If at the new step size $\beta_{k,\text{new}}$, $h(\beta_{k,\text{new}}) < 0$ and $\phi(\beta_{k,\text{new}}) \leq \phi(\beta_{k,1})$, set $\beta_{k,1} = \beta_{k,\text{new}}$ and replace $\phi(\beta_{k,1})$ and $h(\beta_{k,1})$, respectively, with $\phi(\beta_{k,new})$ and $h(\beta_{k,\text{new}})$; and then repeat the step size calculation procedure of Case 3 until the strong Wolfe conditions or the conditions of Case 1 are satisfied.

  (c) If at the new step size $\beta_{k,\text{new}}$, $h(\beta_{k,\text{new}}) < 0$ and $\phi(\beta_{k,\text{new}}) > \phi(\beta_{k,1})$, set $\beta_{k,2} = \beta_{k,\text{new}}$, replace $\phi(\beta_{k,2})$ and $h(\beta_{k,2})$, respectively, with $\phi(\beta_{k,\text{new}})$ and $h(\beta_{k,\text{new}})$; and then repeat the step size calculation procedure of Case 3 until the strong Wolfe conditions or the conditions of Case 1 are satisfied.

  In most cases, wherever Case 3 occurs, the procedure for Case 3 presented above leads to Case 1 and we converge to a step size where the strong Wolfe conditions are satisfied. On extremely rare occasions, however, Case 3 yields a situation where the Wolfe conditions can not be satisfied. This happens when the search direction is almost perpendicular to the gradient of the objective function, and we suspect the difficulties in resolving this case is due to numerical round off errors. If the strong Wolfe conditions are not satisfied when $|\beta_{k,new} - \beta_2|$ is smaller than a specified small number ε (we normally use ε =0.001), then we recompute the search directions as $d_k = -\tilde{H}_{k-1}^{-1}\nabla O(m_k)$ and do the line search for the normalized step size β_k. Assumimg we find a step size which satisfies the strong Wolfe conditions, then we accept $m_{k+1} = m_k + \beta_k d_{N,k}$ as the updated approximation to the model.

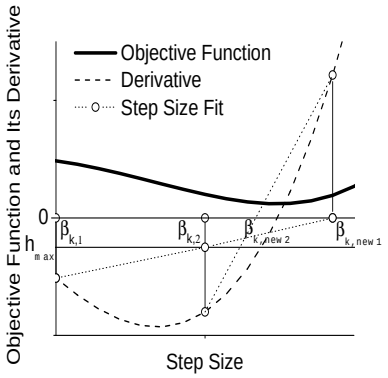


Figure 4.12: Step size fit for $\rho > 1$, case 2.

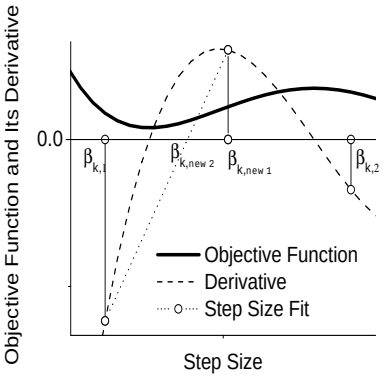


Figure 4.13: Step size fit for $\rho > 0$, case 3.

4.8 Rescaling, Damping and Constraints

In addition to the improved line search algorithm used above, we show that rescaling the variables prior to optimization can reduce the number of iterations required to obtain convergence with the LBFGS algorithm. In addition, damping and/or constraints can be used to prevent overshooting and undershooting.

4.8.1 Rescaling of Model Parameters

For the LBFGS algorithm, we found via experimentation that the convergence properties are often improved by rescaling the variables. Here, the original model parameters are shifted by their prior means $m_{\text{prior},i}$ and rescaled with their standard deviations, the $\sigma_{m,i}$'s. We define the new rescaled variables, x_i, $i = 1, 2, ..., N_m$, by

$$x_i = \frac{m_i - m_{\text{prior},i}}{\sigma_{m,i}}, \tag{4.74}$$

so

$$m_i = m_{\text{prior},i} + x_i \sigma_{m,i}. \tag{4.75}$$

Note the expectation of the rescaled vector of model parameters, $x = [x_1, x_2, \cdots x_{N_m}]^T$, is the zero vector and the covariance matrix for x is the correlation matrix for m. Although it is well known that a proper rescaling of model parameters can significantly improve the convergence properties of the steepest descent algorithm, there is no formal theoretical basis which proves that the rescaling given above will improve the convergence properties of the LBFGS algorithm. Moreover, it is possible for the condition number of correlation matrix to be greater than the condition number of the covariance matrix. Nevertheless, for the history

matching problems we have tried, the above rescaling usually results in faster convergence of the LBFGS algorithm.

4.8.2 Controlling Overshooting and Undershooting

In some history matching examples, one will find that an optimization algorithm exhibits undershooting or overshooting, i.e., extremely large or small values of model parameters, e.g., gridblock permeabilities and/or porosities, Wu et al. (1999); Li et al. (2003a). This problem occurs when the initial guess for the model gives predicted data far from the observed data, i.e., results in a large data mismatch. The problem is most severe when the initial data mismatches are unbalanced, i.e., at one or two wells, the initial production data mismatches are much larger than those in the other wells. In this case, the large data mismatches control the adjustments to the model during the early iterations of the optimization algorithm and often result in overshooting or undershooting problems in the first few iterations. In this situation, damping the production data with large initial mismatches is an effective method to control the problem of overshooting and undershooting. Applying a constrained optimization algorithm provides a method to keep model parameters within upper and lower bounds, but to fully control undershooting and overshooting with constrained optimization may require starting with very tight bounds on the model parameters at early iterations.

4.8.3 Damping the Data Mismatch Term

The data mismatch part of the objective function can be reweighted using a damping matrix Λ. Λ is a diagonal matrix with its diagonal element given by $\lambda_i > 0$. The objective function for constructing the MAP estimate with a damped data mismatch term is given by

$$\begin{aligned} O_{\text{damp}}(m) =& \frac{1}{2}(g(m) - d_{\text{obs}})^T \tilde{C_D}^{-1}(g(m) - d_{\text{obs}}) \\ &+ \frac{1}{2}(m - m_{\text{prior}})^T C_M^{-1}(m - m_{\text{prior}}). \end{aligned} \tag{4.76}$$

In the preceding equation, the matrix $\tilde{C}_D$ defined by

$$\tilde{C}_D = \Lambda C_D \Lambda, \tag{4.77}$$

has replaced the original data covariance matrix C_D of Eq. 4.1. A similar modification is made when constructed a realization using the randomized maximum likelihood method by minimizing Eq. 4.2. With $\sigma_{d,i}^2$ denoting the variance of the ith measurement error, i.e., the ith entry of C_D, the ith entry of $\tilde{C}_D$ is $\sigma_{d,i}^2 \lambda_i^2$. Thus if $\lambda_i > 1$, the standard derivation of

the ith measurement error has been increased by the factor λ_i and consequently, the relative weighting of the data mismatch term has been decreased in the objective function of Eq. 4.76 and the relative weighting of the model mismatch term has been increased. The effect is that, in the minimization process, the changes in the model will be smaller and the tendency to obtain abnormally high or low values of model parameters will be decreased, i.e., damping can be used to eliminate overshooting.

Different damping factors can be specified for different types of data and it is also possible to specify a damping factor for each datum. One way to specify the damping factor for ith datum, λ_i, is based on the difference between the observed value and the simulated value with the initial model. If the difference is larger than κ standard deviations of the noise for this datum, we damp this data. The damping factors are calculated by

$$\lambda_i = \max\left[1, \left|\frac{g_i(m_0) - d_{obs,i}}{\kappa\sigma_i}\right|\right], \tag{4.78}$$

where $g_i(m_0)$ is the ith predicted datum for the initial model m_0, and $\kappa > 1$. This is the procedure we normally use.

As we wish to use the correct C_D in constructing the final estimate or realization, we stop the program when the original normalized objective function $O_N(m) = 2O(m)/N_d$ is smaller than 3. Then we use the model obtained with damped production data as the initial model, and rerun the case without damping. This is the procedure used in PUNQS3 history matching example presented in the next chapter, but is also possible to use multistep procedure. In a multistep procedure, one can adjust the λ_i every 5 to 10 iterations where at each readjustment, one applies Eq. 4.78 with $g_i(m_0)$ replaced by $g_i(m_k)$ where m_k is the latest iterate. In this process, all damping is removed once we obtain a normalized objective function smaller than three. The number three is ad hoc; and it would be equally reasonable to use a value of five or more. The definition of κ is also somewhat ad hoc. If $\kappa = 3$ results in apparent undershooting/overshooting at the end of the first step, κ would be decreased to 1. We have never encountered this situation. If $\kappa = 3$ results in a negligible change in the objective function, we have over damped and κ should be increased to 10 or more. Fortunately, one can determine whether the value of κ is appropriate by examining the results obtained at the first iteration.

4.8.4 Applying Constraints

Conceptually, one can also control undershooting and overshooting by applying constraints. One has to be careful in doing so, however, and one should not be satisfies with a history

matched model has many of its model parameters equal to the upper or lower bound specified by the constraints.

Let m_l and m_u, respectively, denote vectors with jth entry given by $m_{l,j}$ and $m_{u,j}$ which respectively represent the the lower and upper bounds for the jth model parameter. One method for transforming the constrained optimization problem into an unconstrained optimization problem is to find a suitable transformation that can map the upper bounds to ∞ and the lower bounds to $-\infty$. Through this transformation, the boundaries are removed. In general, the transformation needs to be invertible (one-to-one) which is normally achieved by using a monotonic transformation. One choice for such a transformation is given by the logarithmic transformation discussed below.

We first apply a linear transformation to rescale the model parameters by their upper and lower bounds. Defining

$$m_{m,j} = \frac{1}{2}(m_{u,j} + m_{l,j}), \tag{4.79}$$

and

$$m_{r,j} = \frac{1}{2}(m_{u,j} - m_{l,j}), \tag{4.80}$$

the rescaled model parameters are defined by

$$x_j = \frac{m_j - m_{m,j}}{m_{r,j}}, \text{ for } j = 1, 2, \cdots N_m \tag{4.81}$$

When $m_j \to m_{u,j}$, $x_j \to 1$, and when $m_j \to m_{l,j}$, $x_j \to -1$. The final transformed parameters are defined by the following log-transformation:

$$s_j = \ln\Big(\frac{x_j + 1}{1 - x_j}\Big) \text{ for } j = 1, 2, \cdots N_m. \tag{4.82}$$

The log-transformation maps $x_j = 1$ to $s_j = \infty$, and $x_j = -1$ to $s_j = -\infty$. From Eqs. 4.81 and 4.82, we can show that the relationships between the original variable m_j and the transformed variable s_j is given by

$$m_j = m_{m,j} + m_{r,j}\left(\frac{\exp(s_j) - 1}{\exp(s_j) + 1}\right), \tag{4.83}$$

and

$$s_j = \ln\Big(\frac{m_j - m_{l,j}}{m_{u,j} - m_j}\Big), \tag{4.84}$$

for $j = 1, 2 \cdots N_m$. If $m_j \to m_{u,j}$, then $s_j \to \infty$ and if $m_j \to m_{l,j}$, then $s_j \to -\infty$. Thus, the boundaries due to the constraints are removed. The relationship between the derivatives of

any function $f(m)$ with respect to m_j and its derivative with respect to s_j can be obtained from the chain rule. From Eq. 4.83, it follows that

$$\frac{dm_j}{ds_j} = \frac{(m_{u,j} - m_j)(m_j - m_{l,j})}{m_{u,j} - m_{l,j}}, \tag{4.85}$$

and then the chain rule gives

$$\frac{\partial f}{\partial s_j} = \frac{\partial f}{\partial m_j}\frac{dm_j}{ds_j} = \frac{(m_{u,j} - m_j)(m_j - m_{l,j})}{m_{u,j} - m_{l,j}}\frac{\partial f}{\partial m_j}. \tag{4.86}$$

The second derivative of $f(m)$ with respect to s_i and s_j for $i \neq j$ is given by

$$\frac{\partial^2 f}{\partial s_j \partial s_i} = \frac{dm_i}{ds_i}\frac{\partial^2 f}{\partial m_j \partial m_i}\frac{dm_j}{ds_j}. \tag{4.87}$$

The second derivative of $f(m)$ with respect to s_j is given by

$$\frac{\partial^2 f}{\partial s_j^2} = \frac{\partial^2 f}{\partial m_j^2}\left[\frac{dm_j}{ds_j}\right]^2 + \frac{\partial f}{\partial m_j}\frac{dm_j}{ds_j}\frac{d}{dm_j}\left[\frac{dm_j}{ds_j}\right], \tag{4.88}$$

where

$$\frac{d}{dm_j}\left[\frac{dm_j}{ds_j}\right] = \frac{m_{u,j} + m_{l,j} - 2m_j}{m_{u,j} - m_{l,j}}. \tag{4.89}$$

For the objective function of Eq 4.1, the model mismatch part of the objective function is given by

$$O_m(m) = \frac{1}{2}[m - m_{\text{prior}}]^T C_M^{-1}[m - m_{\text{prior}}] = O_s(s), \tag{4.90}$$

where the explicit equation for O_s as a function of s can be obtained by replacing each component of m by the right side of Eq. 4.83. The gradient of the model mismatch part of the objective function $O_m(m)$ with respect to the original model vector m is given by

$$\nabla_m O_m = C_M^{-1}[m - m_{\text{prior}}]. \tag{4.91}$$

The gradient of the model mismatch part of the objective function $O_s(s)$ with respect to s can be obtained from the chain rule and is given by

$$\nabla_s O_s = \Lambda_1 C_M^{-1}[m - m_{\text{prior}}] = \Lambda_1 \nabla_m O_m(m), \tag{4.92}$$

where Λ_1 is a diagonal matrix with j'th diagonal entry given by

$$\lambda_{1j} = \frac{dm_j}{ds_j}. \tag{4.93}$$

The Hessian matrix of $O_m(m)$ with respect to m is given by

$$H_m = \nabla_m[\nabla_m O_m(m)]^T = C_M^{-1}. \tag{4.94}$$

Because of this equation, the initial approximation of the inverse Hessian is given by C_M if optimization is done in terms of the original model parameters. When the log-transformation is applied, however, we do optimization in terms of the transformed variables. Thus we need to find the Hessian based on the model mismatch part in terms of s evaluated at the initial guess. The Hessian matrix of $O_s(s)$ with respect to s is given by

$$H_s = \nabla_s[\nabla_s O_s(s)]^T, \tag{4.95}$$

and also can be obtained by the chain rule using Eqs. 4.87, 4.88 and 4.94. It follows that

$$H_s = \Lambda_1 C_M^{-1} \Lambda_1 + \Lambda_2, \tag{4.96}$$

where Λ_2 is a diagonal matrix with j'th diagonal entry given by

$$\lambda_{2j} = \frac{dm_j}{ds_j}\frac{d}{dm_j}\left[\frac{dm_j}{ds_j}\right]\frac{\partial O_m}{\partial m_j}. \tag{4.97}$$

As optimization is done in terms of s, we wish to use $[H_s]^{-1}$ evaluated at the initial guess for m as the initial guess of the approximate Hessian inverse in the LBFGS algorithm. However, in general, $[H_s]^{-1}$ is dense and cannot be easily computed for large scale problems. Note, however, that the diagonal matrix Λ_2 is a null matrix if $m = m_{\text{prior}}$ or if m equals the average of its lower and upper bounds. Then the initial guess for the Hessian inverse for the LBFGS algorithm can be chosen as

$$\tilde{H}_0^{-1} = \Lambda_{1,\text{mprior}}^{-1} C_M \Lambda_{1,\text{mprior}}^{-1}, \tag{4.98}$$

or

$$\tilde{H}_0^{-1} = \Lambda_{1,\text{ave}}^{-1} C_M \Lambda_{1,\text{ave}}^{-1}. \tag{4.99}$$

The subscript *mprior* denotes evaluation at $m = m_{\text{prior}}$ and the subscript *ave* denotes evaluation at the average of the upper and lower bounds.

Chapter 5

RESULTS, HISTORY MATCHING OF PRODUCTION DATA

In this chapter, we show the results obtained for two history matching examples with the LBFGS algorithm. The first is based on a reservoir model for the North Sea Reservoir. The second is based on the Tengiz reservoir, which is a over pressured oil reservoir operating above bubblepoint pressure located in Kazakstan.

5.1 PUNQS3

The PUNQS3 represents a synthetic model based on an actual North Sea reservoir (Floris et al., 2001).

5.1.1 Reservoir Model Description

The model consists of five layers. The simulation model contains $19 \times 28 \times 5$ grid blocks, of which 1761 blocks are active. The field is bounded to the east and south by a fault, and links to the north and west to a fairly strong aquifer. A small gas cap is located in the center of the dome shaped structure. The field initially contains 6 production wells located around the gas oil contact. We introduce 66 water injection wells around the WOC line to simulate the aquifer. The grid block dimensions in the x and y directions is uniform with $\triangle x = \triangle y = 590.55$ (ft).

There are six producers, PROD1(10,22,4/5), PROD4(9,17,4/5), PROD5(17,11,3/4), PROD11(11,24,3/4), PROD12(15,12,4/5) and PROD15(17,22,4). Here, the numbers in parentheses refer to the gridblock indices. For example, PROD1(10,22,4/5) means that

production well one is located in the gridblock centered at (x_{10}, y_{22}) (tenth gridblock in the x-direction, twenty second gridblock in the y-direction), and perforated in the fourth and fifth gridblocks in the z-direction.

The porosities in each layer are normally distributed. The horizontal and vertical permeabilities are log-normally distributed. The porosity, $\ln(k)$ and $\ln(k_z)$ within each layer are correlated to each other, but there is no correlation between properties in different layers. The correlation coefficients between porosity, $\ln(k)$ and $\ln(k_z)$ in each layer are all specified as 0.8.

The observed data are obtained by adding noise (measurement error) to the "true" production data generated by running the reservoir simulator with the true model. In generating noise, the standard deviations of measurement error for bottom hole pressure, GOR and WOR, respectively, were specified as $\sigma_p = 43.509$ (psi), $\sigma_{GOR,i} = 0.1 GOR_{\text{true},i}$ (scf/STB), and $\sigma_{WOR} = 0.01$ (STB/STB). Measurement errors were assumed to be independent Gaussian random variables with mean zero. By adding noise to the true data, we obtain the observed or noisy production data which we history match. In all cases, the production data matched pertain to the first 2920 days (8 years) of production. After history matching these production data, we generate predictions for another 8.5 years of production based on the estimated model obtained by history matching. In this example, there are a total of 2604 observed data in d_{obs}, 900 bottom hole pressure data, 852 GOR data and 852 WOR data. The correct geostatistical model (variogram) for each layer is used to define the prior covariance matrix.

5.1.2 History Matching Data from PUNQS3

We will discuss four different implementations of our history matching procedure. In all cases, optimization is done with the LBFGS algorithm and the improved line search procedure presented in subsection 4.7.2. Rescaling refers to changing variables according to Eq. 4.74 prior to applying the LBFGS algorithm. The four procedures we consider are (1) rescaling without applying damping or constraints; (2) rescaling with damping but no constraints; (3) damping without applying constraints or rescaling, (4) no damping with constraints. Because the constrained algorithm is based on the log-transformation which automatically does an automatic rescaling of variables, we do not apply Eq. 4.74 if constaints are imposed. Damping and constrained optimization refer to the procedures discussed in section 4.8

When damping is applied, we first damp the production data using the damping factors determined by Eq. 4.78 with $\kappa = 3$. This optimization step is terminated when the normalized objective function calculated without damping becomes smaller than 3 which normally

takes five to twenty iterations. Then we use the model obtained with damped production data as the initial model, and apply the LBFGS algorithm with no further damping.

When we apply the constrained optimization algorithm based on the log-transformation, the upper and lower bound of model parameters are specified by

$$\begin{aligned} m_{u,i} &= m_{\text{prior},i} + 1.5\sigma_{m,i} \\ m_{l,i} &= m_{\text{prior},i} - 1.5\sigma_{m,i} \end{aligned} \tag{5.1}$$

where $m_{\text{prior},i}$ is the prior mean of m_i, and $\sigma_{m,i}$ is the standard deviation of m_i. When the normalized objective function is smaller than 3, we restart the history matching program with the unconstrained optimization algorithm. The model parameters are automatically rescaled with the log-transformation. When we restart with an unconstrained optimization algorithm, we rescale the variables according to Eq. 4.74 prior to applying the LBFGS algorithm.

5.1.3 Convergence Behavior

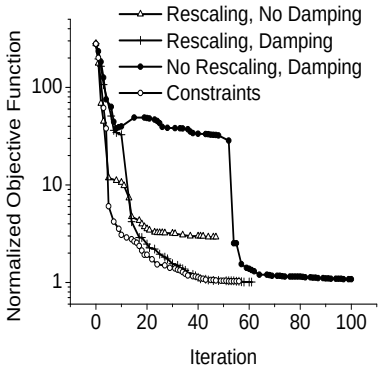


Figure 5.1: Normalized objective function for unconstrained optimization.

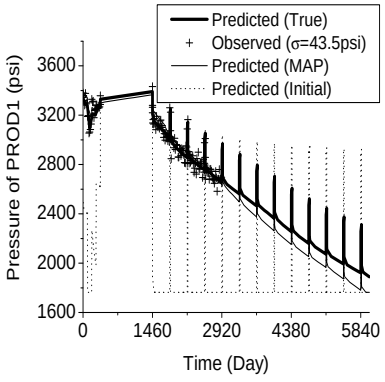


Figure 5.2: Pressure data match, PROD1, no damping, no constraints.

Fig. 5.1 illustrates the behavior of the normalized objective function for the four different implementations of the algorithm, In this figure, the curve with open triangles represents the normalized objective function for implementation 1 (rescaling, no damping, no constraints); the curve through pluses represents the behavior of the objective function for implementation 2 (rescaling with damping but no constraints); the curve through solid circles represents implementation 3 (no rescaling with damping but no constraints); and the curve with open circles is for implementation 4 (the constrained algorithm with no damping).

For implementation 1, the normalized objective function converged to a value of 3, which is larger than the expected value of 1. Although results not shown indicate that the convergence properties of the LBFGS algorithm are improved by rescaling, rescaling by itself does not result in optimum convergence performance. By comparing implementations 2 and 3, however, we see that rescaling is of value. When damping without rescaling was applied, we reached the maximum allowable value of 100 iterations without satisfying the convergence criteria, whereas, with rescaling and damping, we lowered the objective function to a slightly lower value and LBFGS converged in 60 iterations. For this example, implementations 2, 3 and 4 all resulted in an objective function close to 1 at termination of the algorithm. In terms of the number of iterations, the constrained implementation and LBFGS with rescaling and damping performed similarly in terms of the rate of convergence, however, as shown later, the model obtained at convergence with the rescaled damped algorithm appears to be somewhat more reasonable.

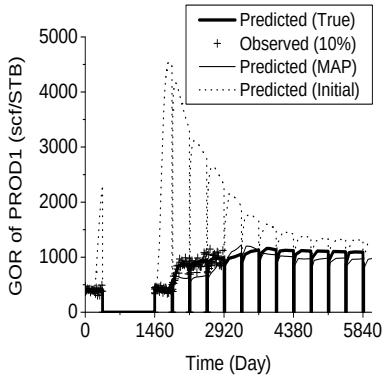


Figure 5.3: GOR data match, PROD1, no damping, no constraints.

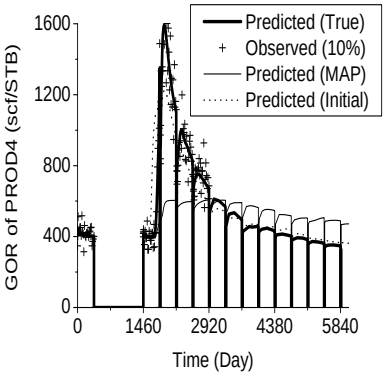


Figure 5.4: GOR data match, PROD4, no damping, no constraints.

5.1.4 Data Match

Fig. 5.2 shows the match of bottom hole pressure data for well PROD1 based on implementation 1 of the LBFGS algorithm. In this figure, the thick solid curve is the true pressure data generated with the true model; the plus marks are the noisy or 'observed' pressure data; the thin solid curve represents the predicted bottom hole pressure generated with the MAP estimate; and the dotted curve is the predicted bottom hole pressure with the initial model where all model parameters were set equal to their prior means. Similar to the results shown in Fig. 5.2, the initial pressure mismatches in all six production wells are very large,

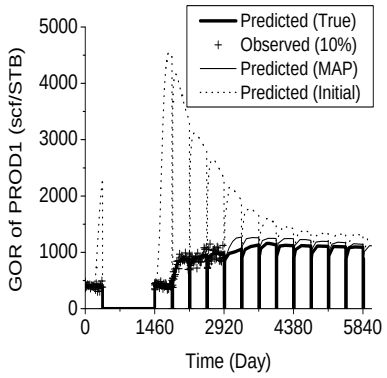


Figure 5.5: GOR data match, PROD1, damping, no constraints.

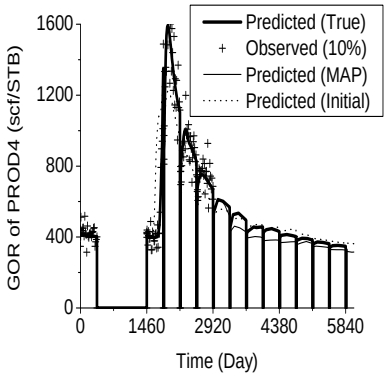


Figure 5.6: GOR data match, PROD4, damping, no constraints.

over 1000 psi for much of the time period, but after history matching the production data, the pressure data misfit is greatly decreased.

Fig. 5.3 shows the GOR data match for Well PROD1 obtained with implementation 1 of the LBFGS algorithm. (The notation in Fig. 5.3 is similar to that used in Fig. 5.2.) Well PROD1 is located at (10,22,4/5). The initial GOR data mismatch in well PROD1 is much larger than the GOR mismatch at the other wells. The maximum value of the initial GOR data mismatch for PROD1 reaches 4000 (SCF/STB), which is 100 times larger than the standard deviation (about 40 SCF/STB) of the GOR measurement error. The GOR data mismatch term in this well dominates the other production data in the early iterations of the LBFGS algorithm and leads to a poor GOR data match at other wells. As shown in Fig. 5.4, the GOR data mismatch in well PROD4 is worse than the initial GOR mismatch after history matching the production data. The large and unbalanced initial GOR data mismatch also results in changes in model parameters aimed primarily at reducing the GOR mismatch in PROD1. Specifically to decrease the GOR in well PROD1, the horizontal and vertical permeabilities around this well in the second layer are greatly decreased to form a barrier to flow from the gas cap, as illustrated by the results of Figs. 5.7(b) and 5.8(b). Because porosity is strongly correlated with the horizontal and vertical log-permeabilities, the porosity in the second layer is also decreased significantly; results for the estimated porosity fields are not shown here. It is important to note that Implementation 1, not only led to abnormally large decrease in properties in Layer 2, but also results in overshooting; as shown in Figs. 5.11(b) and 5.12(b), implementation 1 (case 1) results in $\ln(k)$ values in excess of 12 and $\ln(k_z)$ in excess of 10 in Layer 4. The damping procedure introduced earlier

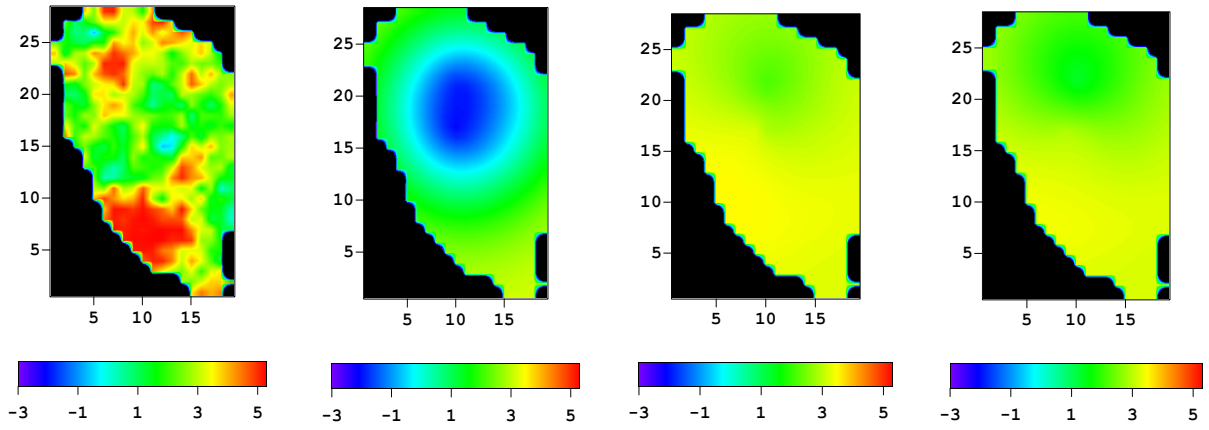
can be used to eliminate the undershooting/overshooting.

Figs. 5.5 and 5.6 illustrate GOR data matches in well PROD1 and PROD4 obtained when rescaling and damping (implementation 3) are used in the LBFGS optimization procedure. We see that GOR data in both wells (PROD1 and PROD4) are well matched with the observed data and are actually quite close to the true data. The notation used in these two figures is the same as in Fig. 5.2. We predicted the pressure and GOR performance for another 8.5 years of production. Figs. 5.5 and 5.6 show that the predicted GOR data for another 8.5 years of production are also close to the data predicted with the true model. Similar results were obtained using constrained optimization based on the log transformation.

5.1.5 Truth and MAP Estimates

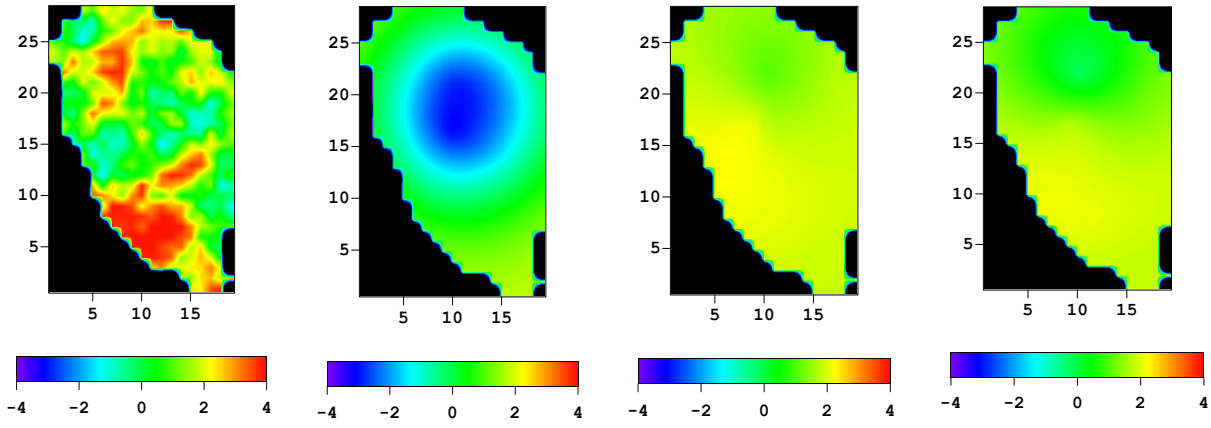
Figs. 5.7(a), 5.9(a) and 5.11(a), respectively, show the true $\ln(k)$ in layers 2, 3 and 4. Figs. 5.8(a), 5.10(a) and 5.12(a) show the true $\ln(k_z)$ in layers 2, 3 and 4. Figs. 5.8 through 5.12 also show the MAP estimates of $\ln(k)$ and $\ln(k_z)$ obtained with Implementations 1, 2 and 4 of the LBFGS algorithm. The MAP estimates obtained with Implementation 3 were fairly similar to those obtained with Implementation 2. Note, the estimated model obtained by history matching without damping (implementation 1) exhibits overshooting and undershooting. In particular, the maximum horizontal log-permeability in layer 4 increases from 3.68 to 14, and the minimum horizontal log-permeability in layer 1 decreases from 4.06 to -3. Figs. 5.7(b) and 5.8(b) illustrate the MAP estimates of $\ln(k)$ and $\ln(k_z)$ in layer 2 obtained with implementation 1. Note that both $\ln(k)$ and $\ln(k_z)$ in layer 2 are greatly decreased. Figs. 5.11(b) and 5.12(b) illustrate the MAP estimates of $\ln(k)$ and $\ln(k_z)$ in layer 4 for implementation 1. Note that overshooting occurs.

From Figs. 5.7(c), 5.7(d), 5.8(c), 5.8(d), 5.11(c), 5.11(d), 5.12(c), 5.12(d), we can see that the undershooting and overshooting problems are controlled by applying the damping procedure (implementation 2) or the constrained optimization algorithm (implementation 4). We obtain permeability fields with no evidence of undershooting or overshooting. Although not shown, a reasonable estimate of the porosity field was also obtained with implementations 2 and 4.



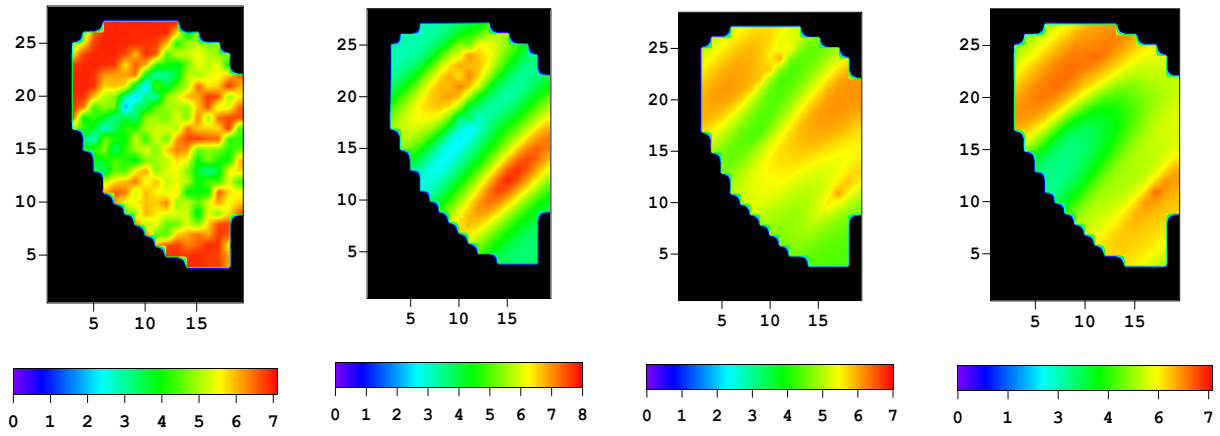
(a) True (b) Implementation 1 (c) Implementation 2 (d) Implementation 4

Figure 5.7: True and MAP estimates of $\ln(k)$ in layer 2, implementations 1, 2 and 4.



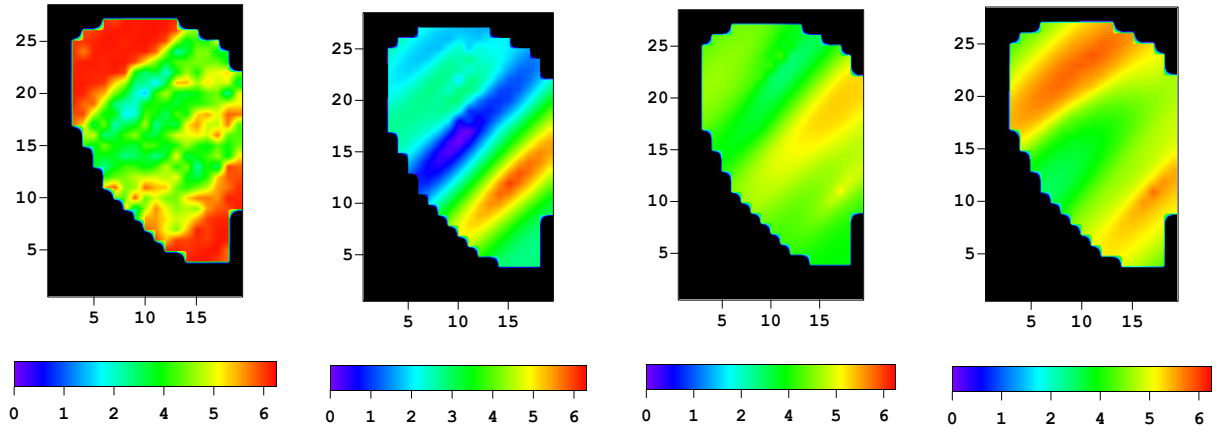
(a) True (b) Implementation 1 (c) Implementation 2 (d) Implementation 4

Figure 5.8: True and MAP estimates of $\ln(k_z)$ in layer 2, implementations 1, 2 and 4.



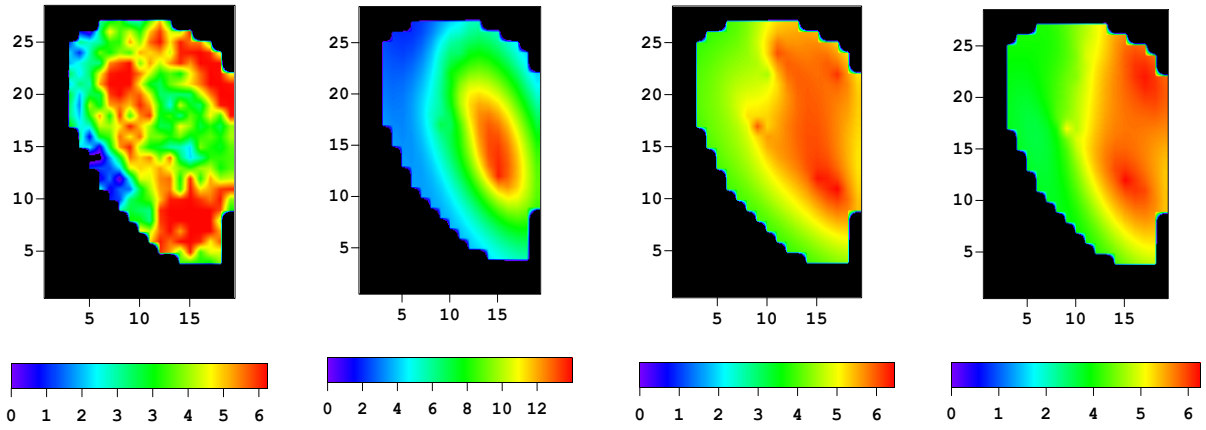
(a) True (b) Implementation 1 (c) Implementation 2 (d) Implementation 4

Figure 5.9: True and MAP estimates of $\ln(k)$ in layer 3, implementations 1, 2 and 4.



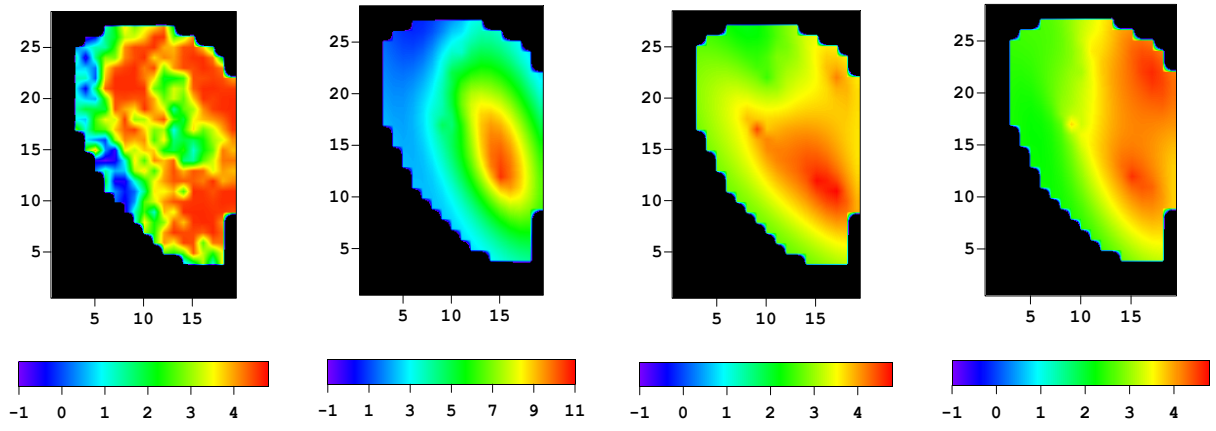
(a) True (b) Implementation 1 (c) Implementation 2 (d) Implementation 4

Figure 5.10: True and MAP estimates of $\ln(k_z)$ in layer 3, implementations 1, 2 and 4.



(a) True (b) Implementation 1 (c) Implementation 2 (d) Implementation 4

Figure 5.11: True and MAP estimates of $\ln(k)$ in layer 4, implementations 1, 2 and 4.



(a) True (b) Implementation 1 (c) Implementation 2 (d) Implementation 4

Figure 5.12: True and MAP estimates of $\ln(k_z)$ in layer 4, implementations 1, 2 and 4.

5.2 Tengiz Reservoir

In this section, we consider history matching buildup data from the Tengiz reservoir which is located in the Pri-Caspian Basin. Tengiz is a carbonate reservoir which was formed during Devonian and Carboniferous time. Figure 5.13 shows that the central or "platform" portion of the reservoir is relatively flat. As shown in Fig. 5.13, the platform is bounded by faults or lithologic breaks and surrounded by gently sloping "flanks" of carbonate debris; see Chambers et al. (1997) and He and Chambers (1999) for additional details on the geology. The index on the x and y axes is the number of gridblocks, i.e., the areal grid for the original reservoir model was 90×100.

Tengiz is an undersaturated oil reservoir produced by 44 wells. With very rare exception, all flowing bottomhole pressures have been maintained above bubble point pressure, which is equal to 3586 psi. Initial reservoir pressure is 11950 psi at a datum of 14765 ft subsea. Current average reservoir pressure remains more than twice the bubble point pressure, and consequently, the oil flows as a single phase in the reservoir.

Our history match of static pressure data from pressure buildup surveys is based on an upscaled reservoir model of Tengiz. The upscaled reservoir model of Tengiz was created by removing most of the sloping flanks near the outer edges of the reservoir and upscaling the remainder of the reservoir to a $59 \times 49 \times 9$ grid. In the upscaled model, the gridblock sizes in the x and y directions are almost uniform with values between 815 and 825 ft. Gridblock sizes in the z direction are non-uniform with values varying between 15 and 150 ft. Figure 5.14 shows a contour map of the top of the reservoir with well locations.

5.2.1 Geostatistical Model

The top boundary of the first simulator layer follows the top boundary of the Tengiz structure (Figs. 5.13 and 5.14) and other simulator layers are similarly curved. Thus, when the values of a rock property are shown for a layer, the reader should remember that the layer gridblocks are at different depths. In such figures, the greatest depths occur in the gridblocks near the lower-left. At these "flank" gridblocks, the values of rock properties are lower than those associated with the central platform or the reservoir.

We were supplied an initial reservoir model on the upscaled reservoir simulation grid. This model represents m_{uc} when history matching pressure data by minimizing the objective function of Eq. 4.2. Throughout, this initial model is referred to as the unconditional realization. Figs. 5.15 and 5.16, respectively, show the unconditional realization of porosity (left), horizontal log permeability (middle) and vertical log permeability (right) for layers 2

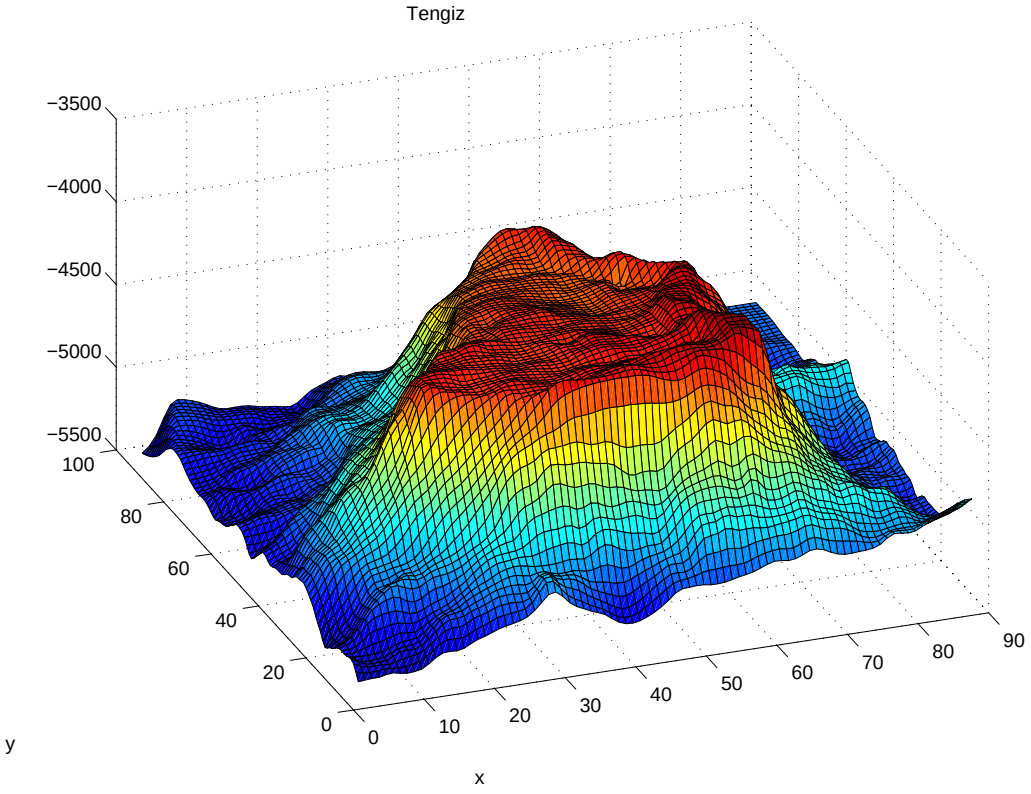


Figure 5.13: A 3D plot of the Tengiz field.

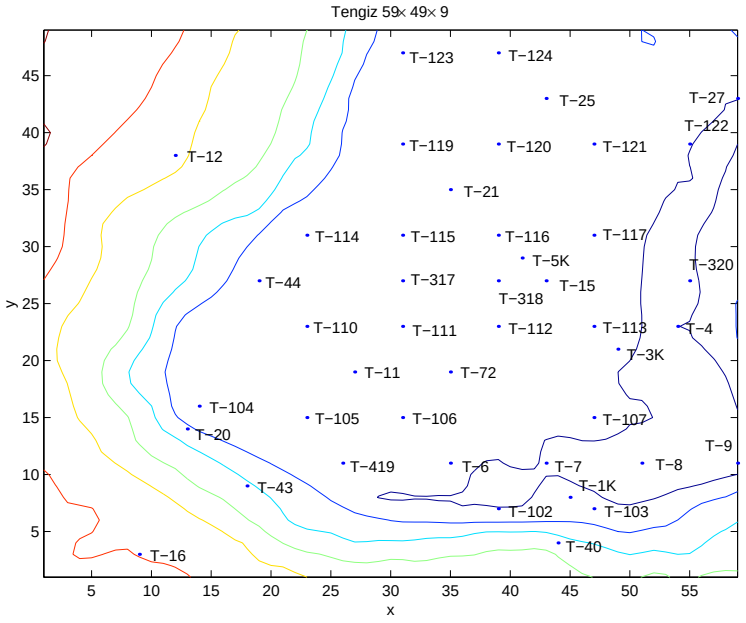


Figure 5.14: A contour plot of the upscaled model with the well locations.

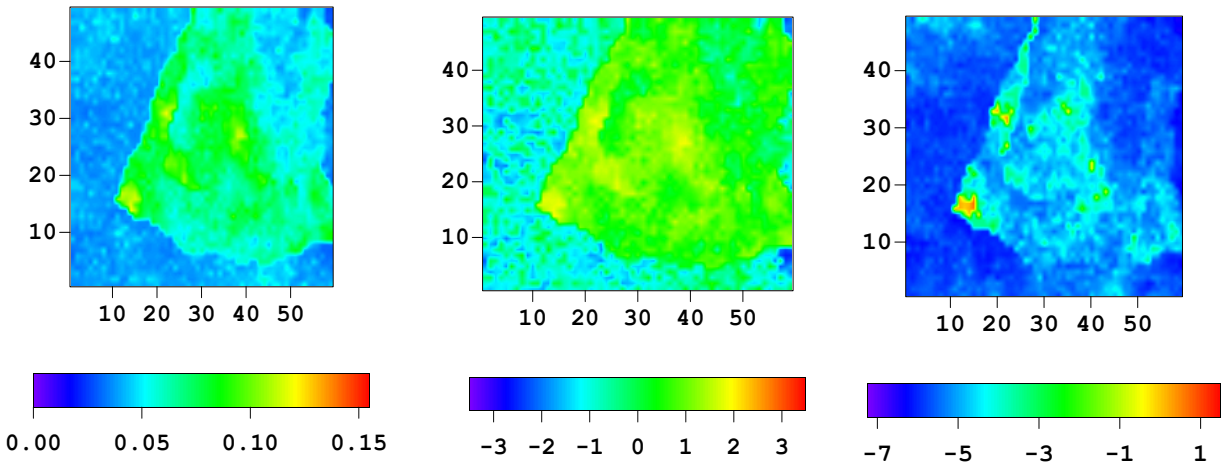


Figure 5.15: Initial porosity (left), $\ln(k)$ (middle) and $\ln(k_z)$ (right) of layer 2.

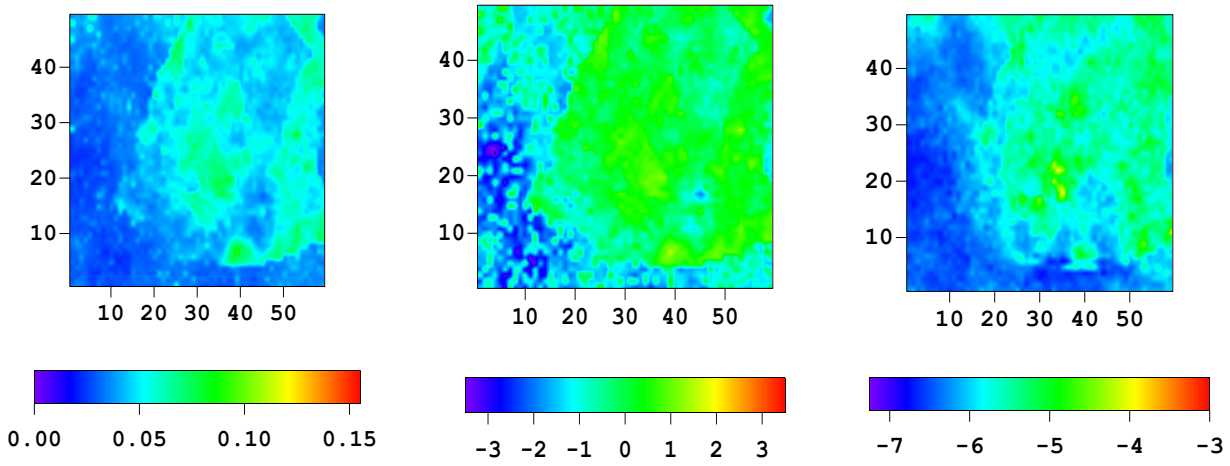


Figure 5.16: Initial porosity (left), $\ln(k)$ (middle) and $\ln(k_z)$ (right) of layer 4.

and 4.

Figs. 5.15 and 5.16 suggest that it would be appropriate to use a two-zone geostatistical model to define a prior geostatistical model for the purpose of history matching in a Bayesian setting. But since our history matching process will use m_{uc} as the initial guess and m_{uc} already reflects two zones, it may not be necessary to define distinct geostatistical models for the two zones and we do not do so.

The prior covariance matrix, C_M in Eq. 4.2 is generated by specifying a variogram model. We specify the variances (sills) of ϕ, $\ln(k)$ and $\ln(k_z)$, respectively, as 2.6092×10^{-4}, 0.8073 and 0.9376. The correlation coefficients between ϕ and $\ln(k)$, ϕ and $\ln(k_z)$, and $\ln(k)$ and $\ln(k_z)$, respectively are given by 0.9096, 0.662 and 0.5. These values were determined from

the unconditional realization, m_{uc}, and differ somewhat from the variances used by He and Chambers (1999) who used variances of 3.2×10^{-4} and 0.225, respectively, for ϕ and $\ln(k)$, but did not specify a variance for $\ln(k_z)$ and did not specify values of the correlation coefficients. We use a anisotropic spherical variogram with principle direction coincident with the x axis. The correlation ranges in the x, y and z direction, respectively, are given by 6561.6 ft, 4921.3 ft and 164.04 ft. These ranges are same as those used by He and Chambers (1999). Cross covariances are generated using the screening hypothesis Xu et al. (1992); Chu et al. (1995b).

5.3 The History Matching Process

He and Chambers (1999) matched pressure transient data from a drawdown/buildup tests conducted on several wells. Rather than doing a full history matching procedure as done here, they matched individual well tests and extrapolated the results to the whole field. They used a Gauss-Newton method and thus were forced to reduce the numbers of model parameters in order to reduce computational time and memory requirements. They tried two methods to reduce model parameters, one is based on adjusting permeability only within the radius of investigation of the welltest and the other is based on determining a permeability multiplier for each well by matching pressure transient data and then interpolating the multipliers across the entire reservoir.

In our work, we have only one to three pressure data at each well and match all data simultaneously using the LBFGS algorithm. In addition to the pressure data, we were given monthly rate data. Although the measurements of pressure were obtained during buildup tests, the pressure data supplied were "equivalent flowing wellbore pressures," obtained by using the ideas of Peaceman (1978, 1983). To obtain a history match, the objective function of Eq. 4.2 will be minimized with m_{uc} given by the initial model supplied by Chevron; see Figs. 5.15 and 5.16.

The model parameters, or reservoir variables, to be history matched include all gridblock porosities, horizontal log-permeabilities and vertical log-permeabilities. The skin factors at the wells are fixed at values determined independently and the reservoir geometry is not varied during the history matching procedure. Thus, the number of model parameters is $N_m = 3N$ where N is the number of gridblocks. With the upscaled grid, $N = 26,019$ and the number of model parameters estimated by history matching is $78,057$. The vector of model parameters is given by

$$m = [m_\phi^T, m_k^T, m_{k_z}^T]^T, \tag{5.2}$$

where m_ϕ is an N-dimensional column with its jth entry equal to the porosity of gridblock

j, m_k is an N-dimensional column with its jth entry equal to the horizontal log-permeability for gridblock j, and m_{k_z} is an N-dimensional column with its jth entry equal to the vertical log-permeability for gridblock j.

5.3.1 Measurement/Modeling Errors

In history matching pressure data from the Tengiz reservoir, $d_{\rm obs}$ is a vector containing the 105 pressure measurements we wish to history match and C_D is the covariance matrix for measurement errors which are assumed to be Gaussian. However, if modeling errors also exist and are Gaussian, they can and should be included (Tarantola, 1987) by adjusting C_D. Unfortunately, the knowledge of the magnitude and structure of modeling errors is seldom if ever sufficient to determine the proper adjustment to C_D. In the Tengiz field example, we will match buildup pressures that, in some cases, have apparently been converted to equivalent flowing bottomhole pressures using ideas of Peaceman (1978, 1983). Modeling errors include any errors in converting measured buildup pressure data, errors due to using average monthly rates especially for any month long period that contains a measured buildup pressure that will be history matched and the inability of the simulator to predict correct bottomhole pressures even if the true reservoir geometry and rock and fluid properties are known and used as simulator input. In the Tengiz field example, gridblocks are extremely large. Measured field data does not have any relationship to a reservoir simulation grid, but because of local truncation errors, the grid structure can have a significant impact on the history matching results even for extremely simple problems, Zhang et al. (2003).

Because we do not have knowledge of C_D, we do not add noise to d_{obs} to generate associated data d_{uc}. Instead, we simply set $d_{uc} = d_{\rm obs}$ in Eq. 4.2 when history matching the Tengiz pressure data. Because we do not have direct knowledge of measurement/modeling errors, we simply assume measurement/modeling errors are uncorrelated so that C_D is a diagonal matrix with ith diagonal entry equal to $\sigma_{d,i}$.

Even though the buildup pressure data for Tengiz were presumably measured very accurately, our history matching exericise indicates that some of the $\sigma_{d,i}$'s must be set to very high values to account for modeling errors. Under the preceding assumptions about C_D, the data mismatch part of the objective function we minimize can be written as

$$\begin{aligned} O_d &= \frac{1}{2}(g(m) - d_{\rm obs})^T C_D^{-1}(g(m) - d_{\rm obs}) \\ &= \frac{1}{2}\sum_{i=1}^{N_d}\Big(\frac{g_i(m) - d_{{\rm obs},i}}{\sigma_{d,i}^2}\Big). \end{aligned} \tag{5.3}$$

5.3.2 History Matching Pressure Data

We match all measured pressure data simultaneously, and adjust gridblock porosity, horizontal log-permeability and vertical log-permeability during the history matching process by minimizing the objective function of Eq. 4.2 with $d_{uc} = d_{\rm obs}$, i.e., we minimize

$$O(m) = \frac{1}{2}\left[(m - m_{uc})^T C_M^{-1}(m - m_{uc}) + \sum_{i=1}^{N_d}\left(\frac{g_i(m) - d_{{\rm obs},i}}{\sigma_{d,i}^2}\right)\right] \tag{5.4}$$

using the LBFGS algorithm presented in the previous chapter. The initial guess is set equal to the initial model m_{uc}. A total of 105 observed pressure data from 44 wells were history matched. These pressure data span the historical period from April 1991 to January 1998. We specified the oil production rate for each well based on monthly historical production data, and also set a minimum bottom hole pressure of 3000 psi for all field simulation runs.

Figs. 5.17 and 5.18 show the monthly oil rate (solid black curve) and the bottom hole pressure (solid red curve) as a function of time at wells T-16 and T-320 predicted with the initial model, m_{uc}. Observed data that are history matched are shown as open circles and the flow rate at the time of this pressure measurement is shown by a black cross. Note Well T-320 has only 2 observed pressure data. The red stars represent predicted bottomhole pressures corresponding to the observed data where the predicted data are generated from the simulator with the initial model as input.

Fig. 5.19 shows all 105 pressure data that are history matched plotted against their index number. Also shown as black crosses are the monthly oil production rates corresponding to the pressure data.

We will try different approaches for obtaining a reasonable history match. When constraints are imposed using the log-transformation, Eq. 4.84, no rescaling of model parameters is done. When constraints are not imposed model parameters are rescaled by using 4.74. Although we apply rescaling when constraints are not imposed, for the Tengiz example considered here, rescaling of model parameters has a negligible influence on the results.

Method 1, No Constraints, No Damping.

This is the basic case. The pressure data are matched with the unconstrained LBFGS optimization algorithm without damping the pressure data. The variances of pressure measurement errors, the $\sigma_{d,i}$'s, are all set equal to 100 (psi^2). (Originally, we tried simply setting the variances equal to 1 psi^2 but obtained even more severe undershooting/overshooting problems than are obtained with all variances equal to 100.)

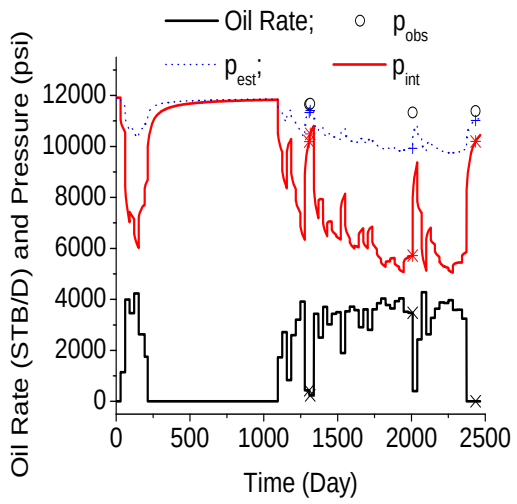


Figure 5.17: Production data of well T-16.

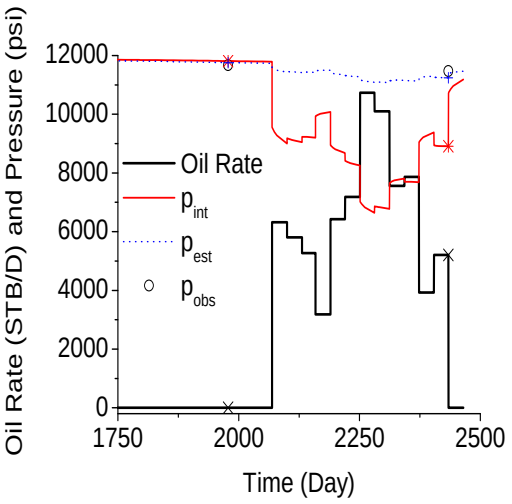


Figure 5.18: Production data of well T-320.

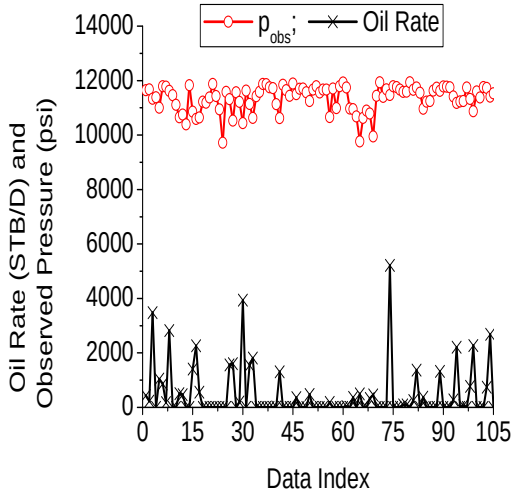


Figure 5.19: Production data of all wells.

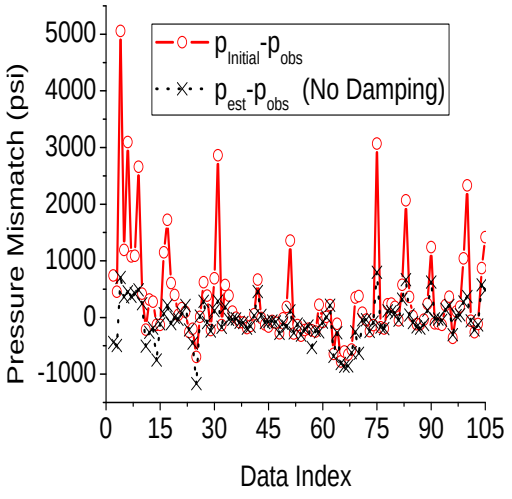


Figure 5.20: Pressure mismatch for all data, method 1.

In Figs. 5.17 and 5.18, the blue dotted curves with pluses represent wellbore pressure predicted from the model obtained by history matching all 105 pressure data using Method 1. Note the variation of the predicted bottomhole pressure with time is much less than that exhibited by the pressure response predicted with the initial model; this is largely a result

of the fact that the horizontal log-permeabilities around the well are greatly increased in the history matching process. The objective function (Eq. 4.2) (with all diagonal entries of C_D equal to 100 psi^2) decreases from 4.688×10^5 to 6.626×10^4 after 11 iterations of the LBFGS algorithm. Additional iterations do not result in any significant decrease.

In Fig. 5.20, the open red circles represent the pressure data mismatches generated with the initial model, i.e., $p_{\text{initial},j} - p_{\text{obs},j}$, for $j = 1, 2, \cdots 105$, where $p_{\text{obs},j} = d_{\text{obs},j}$ denotes the jth observed pressure data and $p_{\text{initial},j}$ denotes the corresponding pressure data predicted with the initial model. In Fig. 5.20, the black crosses represent the pressure data mismatch obtained with the history matched model. There are 16 data with an initial mismatch larger than 1000 (psi); the maximum initial data mismatch is 5056 (psi). After history matching, pressure mismatches for these data are greatly decreased. However, for some other data, pressure mismatches actually increase somewhat. After the history matching, the maximum pressure mismatch is still high, 1168.9 (psi), and there are 12 data with a mismatch exceeding 500 (psi).

Most initial pressure mismatches are similar to those shown in Figs. 5.17 and 5.18, i.e., wellbore pressures predicted with the initial model are lower, and in many cases, significantly lower, than the corresponding observed pressures. This suggests that horizontal permeability will have to be increased, in some cases significantly, to obtain a match of observed pressure data. Figs. 5.21 and 5.22 show the estimated porosity (left), horizontal log-permeability (middle) and vertical log-permeability (right) in layers 2 and 4 obtained by history matching the pressure data. The maximum value of $\ln(k)$ in layer 2 reaches 9.9, and the minimum value of $\ln(k)$ in layer 2 reaches -8.6. Given that the unconditional realization (starting model) has no $\ln(k)$ value above 2.8 or below -2.8 and the variance of $\ln(k)$ is 0.9, it is apparent that we have encountered extreme undershooting and overshooting in the history matching process. The log-permeability field for layer 4 also exhibits some abnormally high values (see Fig. 5.22), but the overshooting here is not quite so extreme.

5.3.3 Analysis of Overshooting and Undershooting

Based on the discussion in section 4.8 and the results of the PUNQS3 example presented earlier in this chapter, it seems possible that the undershooting and overshooting problems are caused by high and unbalanced initial pressure mismatches. Unbalanced means some initial pressure mismatches are much higher than others.

For example, well T-320 is located in gridblock (55,27). This well is perforated through layers 1 to 6. The initial pressure mismatch in this well at 2434 day is 3067.04 psi. At 2434 day, the observed pressure, 11470 psi, is much larger than the value of 8403 (psi) obtained

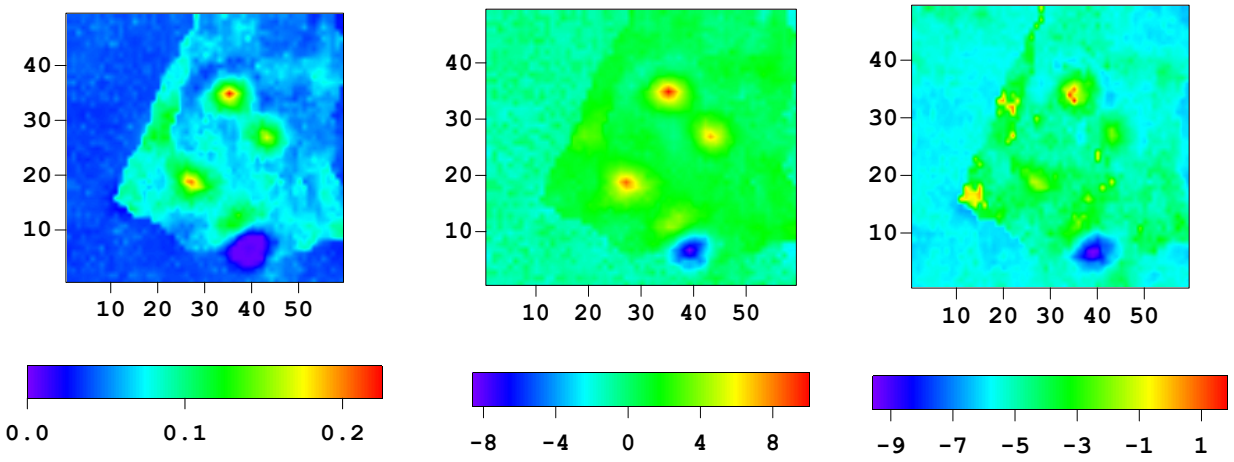


Figure 5.21: Estimated porosity (left), $\ln(k)$ (middle) and $\ln(k_z)$ (right) of layer 2, method 1.

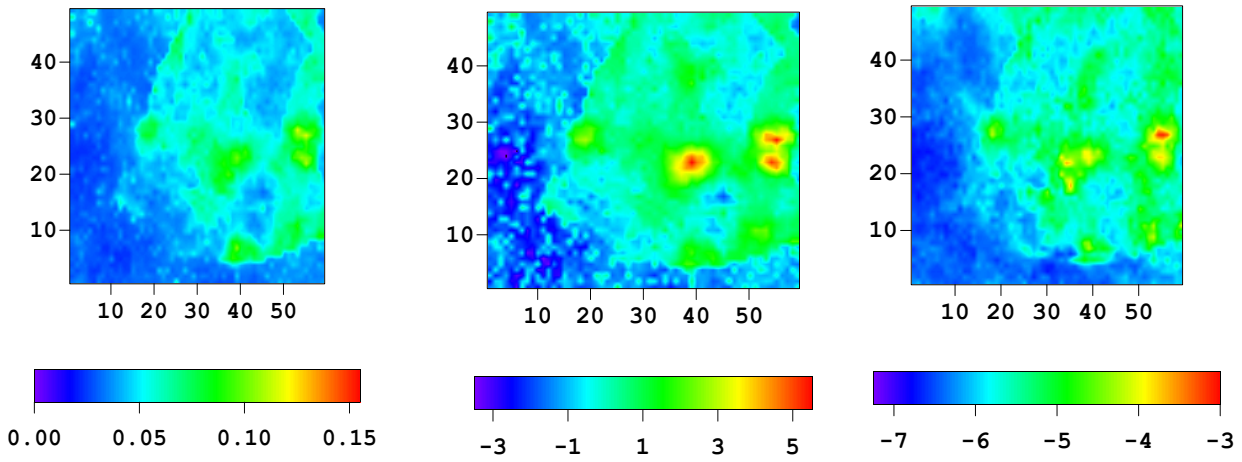


Figure 5.22: Estimated porosity (left), $\ln(k)$ (middle) and $\ln(k_z)$ (right) of layer 4, method 1.

with the initial model. To match the high observed pressure, the permeabilities around well T-320 are greatly increased. In layer 4, the horizontal log-permeability around this well increased from 0.28 to 5.5.

Large initial pressure mismatches tend to cause large changes in the model at early iterations and for problems like the one considered here, one can sometimes obtain unrealistic property fields. If a well is completed in more than one layer, pressure data is not sufficient to resolve individual layer values of horizontal permeability; at best, only thickness averaged horizontal permeability is well resolved by pressure data, He et al. (1997). The thickness

averaged horizontal permeability for n_l homogeneous layers is denoted by $\bar{k}$ and defined by

$$\bar{k} = \frac{1}{h_t} \sum_{j=1}^{n_l} k_j h_j. \tag{5.5}$$

Here, k_j and h_j, respectively, represent the permeability and thickness of the j'th layer, and h_t is the total thickness of the n_l layers open to flow. One can obtain the same the value of $\bar{k}$ with an infinite number of combinations of values of the k_j's. One possibility is to set all $k_j = \bar{k}$, but another possibility is to set some k_j values far less than $\bar{k}$ and other k_j values far greater than $\bar{k}$.

For example, well T-102 is located in gridblock (39,7), i.e., the gridblock index in the x-direction is 39, and gridblock index in the y-direction is 7. This well is perforated through layers 1, 2 and 3. We have only one observed pressure data at this well, 11004 psi at 1521 days. This observed pressure is much larger than the value of 7907.6 psi calculated with the initial model. Thus, the initial pressure mismatch in well T-102 is 3096.4 (psi). Because the wellbore pressure predicted with the initial model differs significantly from the corresponding observed pressure, large changes in the values of rock properties in gridblocks penetrated by this well and nearby gridblocks are made during the history matching process. As shown in Fig. 5.21, the horizontal log-permeabilities around well T-102 in layer 2 are decreased to -8, but in layer 3, the horizontal log-permeabilities around well T-102 increased from 1.6 to 4, and overall, the thickness average permeability increases.

The preceding results are consistent with our conjecture that unbalanced initial pressure data mismatches causes big changes in model parameters, i.e., causes the overshooting/undershooting problems noted earlier.

Method 2, Damping, No Constraints.

In order to control the overshooting and undershooting problems that can result from large and unbalanced initial pressure data mismatches, we apply the damping procedure discussed earlier. The damped or modified variances are specified by Eq. 4.78. For this example, we found a value of $\kappa = 3$ results in overdamping. With this value, the LBFGS algorithm results in only very small changes in the initial model and very little improvement in the pressure match. Thus, we tried $\kappa = 10$. We applied a three step process where at the end of each step, we replaced m_0 in Eq. 4.78 by the latest estimate of the model.

After three steps consisting of 16 total iterations, we obtained the results shown in Figs. 5.23 and 5.24, which show the estimated porosity (left), horizontal log-permeability (middle) and vertical log-permeability (right) fields in layers 2 and 4 obtained using Method 2 when applying the LBFGS optimization procedure.

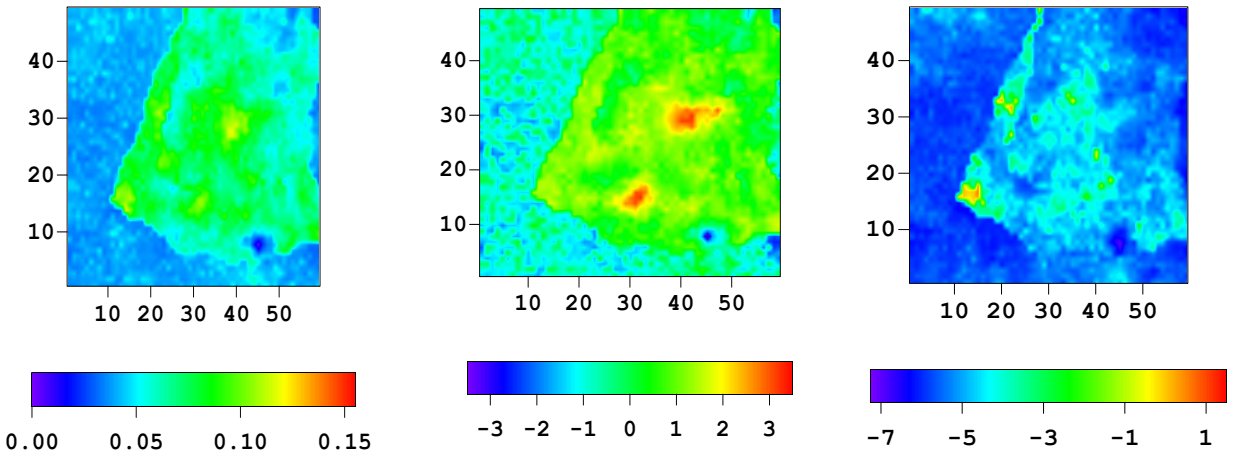


Figure 5.23: Estimated porosity (left), $\ln(k)$ (middle) and $\ln(k_z)$ (right) of layer 2, method 2.

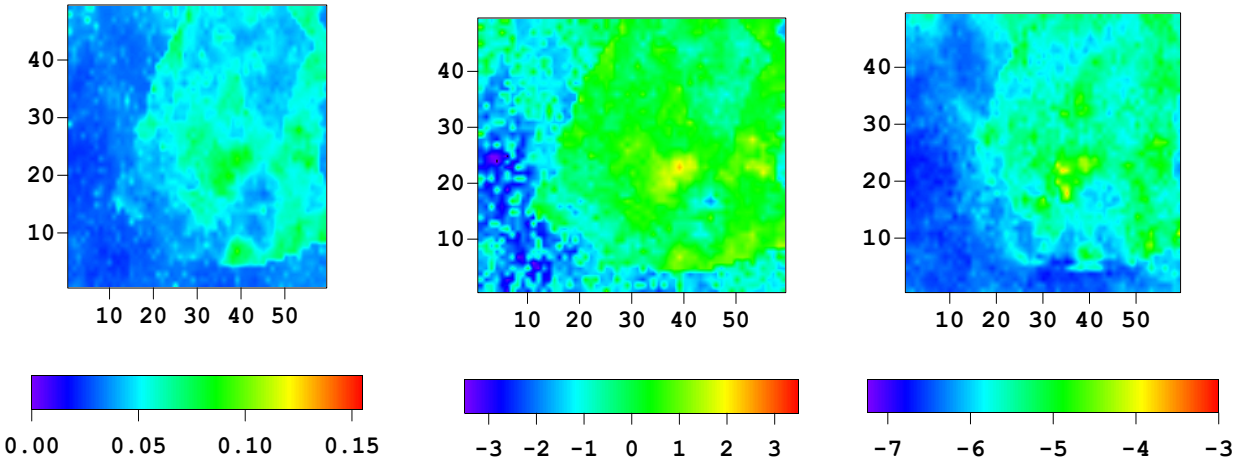


Figure 5.24: Estimated porosity (left), $\ln(k)$ (middle) and $\ln(k_z)$ (right) of layer 4, method 2.

The model parameters shown in Figs. 5.23 and 5.24 are more consistent with the geostatistical model than those in Figs. 5.21 and 5.22. Overshooting and undershooting has been eliminated by applying damping. However, the pressure match is not as good as the one obtained with Method 1. Fig. 5.25 shows the pressure mismatch based on the initial model (open red circles) compared with the pressure mismatch based on the history matched model (black crosses). Note the observed pressure matches are not as good as obtained with Method 1, Fig. 5.20.

In Fig. 5.26, the open triangles on the red dashed curve is the oil rate, and the black crosses

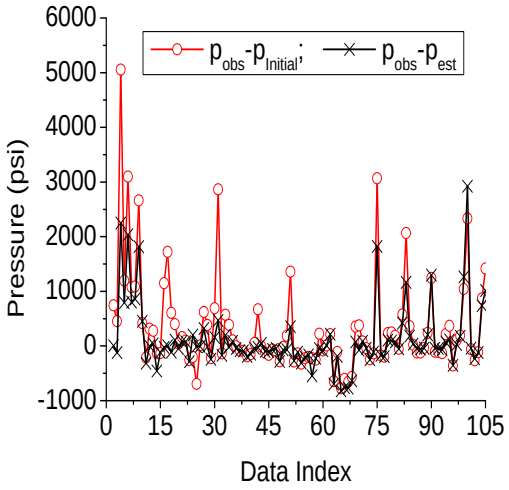


Figure 5.25: Pressure mismatch for all data, method 2.

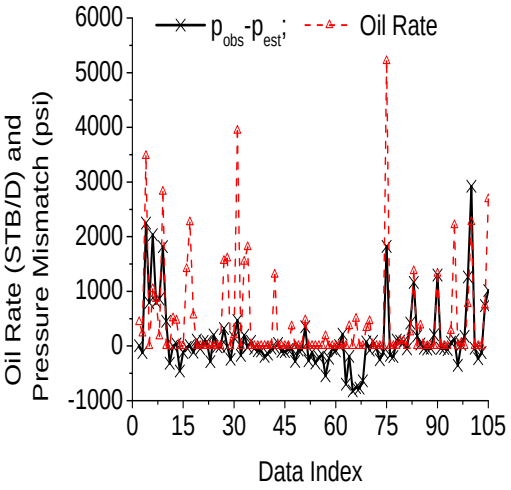


Figure 5.26: Oil rate and pressure mismatch, method 2.

on the solid black curve is the pressure mismatch after history matching using Method 2. We see that the high pressure mismatches are mostly associated with pressure data obtained in a time period with a large monthly oil rate. This suggests there may be some inconsistency between pressure and rate data. As noted earlier, pressure data corresponds to static buildup pressures converted to flowing wellbore pressure.

Method 3, Constraints and Damping.

The fact that Method 2 gave only a small decrease in the objective function suggests that we over damped data mismatch terms. Thus, we should increase κ. On the other hand, less damping tends to increase the possibility of overshooting and undershooting. Thus, Method 3 combines damping with constrained optimization based on the log-transformation introduced in Eq. 4.84. In Eq. 4.82, we set $\kappa = 30$. The upper bound and lower bound of $\ln(k)$ used as constraints are determined by

$$\ln(k)_{\text{upper}} = \ln(k)_{\text{uc,max}} + \sigma_{\ln(k)}, \tag{5.6}$$

and

$$\ln(k)_{\text{lower}} = \ln(k)_{\text{uc,min}} - \sigma_{\ln(k)}, \tag{5.7}$$

where $\ln(k)_{\text{uc,max}}$ and $\ln(k)_{\text{uc,min}}$, respectively, denote the maximum and minimum values of $\ln(k)$ displayed by the unconditional (initial) model, and $\sigma_{\ln(k)}$ is the standard deviation of

$\ln(k)$ in the prior model. The upper bounds and lower bounds of $\ln(k_z)$ and the upper bound of ϕ are determined with the same procedure, except that the lower bound of porosity is specified by

$$\phi_{\text{lower}} = \min[0.005, \phi_{\text{uc,min}} - \sigma_\phi], \tag{5.8}$$

because ϕ must be positive.

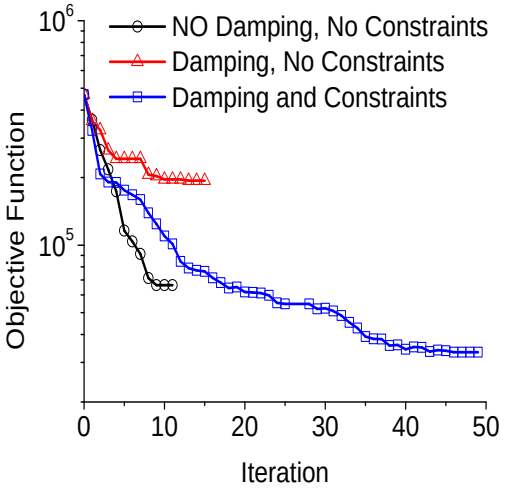


Figure 5.27: Objective function performance for Methods 1, 2, and 3.

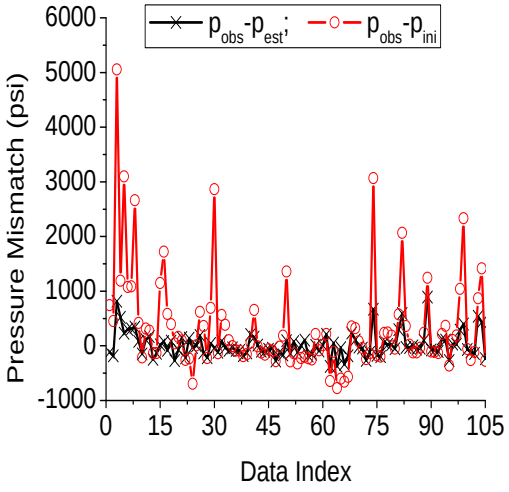


Figure 5.28: Pressure mismatch, method 3.

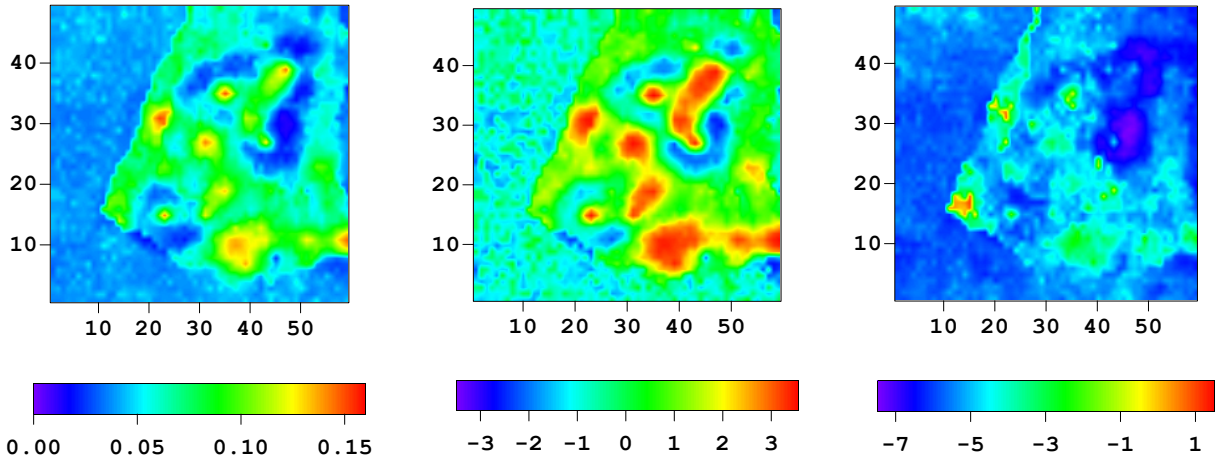


Figure 5.29: Estimated porosity (left), $\ln(k)$ (middle) and $\ln(k_z)$ (right) of layer 2, method 3.

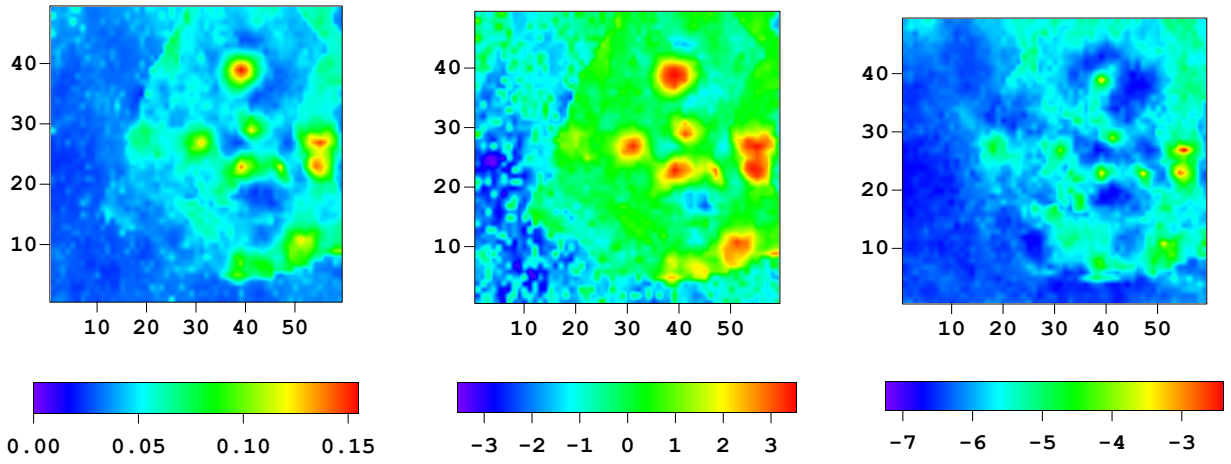


Figure 5.30: Estimated porosity (left), $\ln(k)$ (middle) and $\ln(k_z)$ (right) of layer 4, method 3.

Fig. 5.27 shows the behavior of the objective function (based on setting all measurement error variances to 100 psi^2) obtained by history matching pressure data by application of the LBFGS algorithm with Methods 1, 2 and 3. The objective function decreases from 4.688×10^5 to 6.626×10^4 after 11 iterations for Method 1 (no damping, no constraints). The objective function reduction is mainly caused by improving the match of of data that have very large initial misfits. When we only apply damping procedure (Method 2) with $\beta = 10$, the objective function does not decrease significantly. When we apply less damping ($\beta = 30$) with constraints, the objective function decreases from 4.688×10^5 to 3.33×10^4.

Fig. 5.28 illustrates the pressure mismatches obtained after history matching using the LBFGS algorithm with Method 3. All pressure mismatches are reduced to a value less than 1000 (psi) and most pressure mismatches are less than 500 (psi). Overall the pressure matches are as good as those obtained by Method 1, but with Method 3, undershooting and overshooting is controlled. Figs. 5.29 and 5.30, respectively, show the estimated porosity (left), horizontal log-permeability (middle) and vertical log-permeability (right) in layers 2 and 4 obtained by history matching pressure data using Method 3. The results obtained are qualitatively consistent with the results obtained by Chambers et al. (2001) with pressure-transient tests and production logs. Chambers et al. show that the production indices along the north side and the east side of the central platform are high. We obtained similar results. Figs. 5.29 and 5.30 show that in the north side (bottom side in these figures) and east side (right side in these figures) of layer 2 and layer 4, the estimated porosity, $\ln(k)$ and $\ln(k_z)$ have larger values.

Analysis of Data Consistency.

Part of the difficulty in obtaining a good match of all pressure data is undoubtedly due to the fact we do not have a reliable procedure to account for modeling error. In particular, it appears that some of the observed data are not consistent with the production schedule. With all methods large pressure mismatches that remain after history matching tend to be associated with pressure data from time periods where the corresponding monthly rate is high. (This is most apparent with the results obtained from Method 2; see Fig. 5.26. A more careful examination of the pressure rate data, indicates that most large pressure mismatches that remain after history matching correspond to a pressure observed at the final time of a large flow rate period where the rate for the next month is significantly lower, usually zero. As the actually pressure measurements were obtained from a buildup test, it seems possible that such pressure data actually correspond to the low rate period, i.e. the time record is wrong. As we do not have access to the raw data, we can not verify whether this conjecture is true.

Method 4 refers to the case where we essentially remove such "abnormal" data from the set of observed data to be history matched by setting $\sigma^2_{d,i} = 10^{10}$ psi^2, and setting the variances of all other pressure "measurement" errors to 100 psi^2. With Method 4, we obtained rock property fields fairly similar to those obtained for Method 3, but somewhat rougher and containing larger region of high and low horizontal low permeability values. The pressure matches for all but the abnormal data were also similar to those obtained using Method 3. We also tried Method 4 starting with the model obtained with Method 3 as the initial guess. In this case, the rock property fields were smoother than obtained with Method 4 and not radically different than those obtained with Method 3.

Comment.

When pressure data were history matched without imposing damping of data mismatch terms or constraints (Method 1), abnormally large increases in horizontal permeability were obtained. As noted above, we believe that the difficulty is due to inconsistencies between some pressure and rate data. We should note, however, that the reservoir is fractured and thus one might conjecture that data should be matched with a dual porosity model. Chambers et al. (2001), however, indicate that pressure transient tests show no dual porosity behavior.

5.3.4 Acknowledgements

We thank ChevronTexaco, ExxonMobil, Lukarco and the Republic of Kazakhstan for allowing us to use the Tengiz data.

Chapter 6

TIME LAPSE SEISMIC DATA

Although inverse problems in general, and automatic history matching problems in particular, are underdetermined, results derived from small amounts of data still provide estimates that are less than satisfying. It is clearly beneficial to make use of some type of "space-dense" data to improve the resolution of the estimate in the gridblocks far away from well locations. Of all usual data related to petroleum engineering, seismic data is the most promising candidate for improved spatial coverage. In a preliminary study (Dong and Oliver, 2002), we have showed that seismic impedance change data directly reflects the elastic properties of reservoir and are sensitive to permeability and porosity. We believe that seismic impedance change data will often provide useful constraints to our history matching procedure.

Since automatic history matching can ultimately be reduced to a minimization problem whose objective function includes both model mismatch and data mismatch parts, the choice of an efficient optimization method is very important, especially when processing large-scale problems. For our applications, we only consider gradient based optimization methods because non-gradient based optimization methods require far too many iterations to be of practical usefulness. Among those gradient based methods, some require the formation of the Hessian matrix as well as computation of the gradient of objective function. Formation of the Hessian is very expensive to compute in large-scale problems, so we have eliminated them from consideration. Instead, we use methods that only require the computation of the gradient of the objective function. According to Zhang and Reynolds (2002b), of the methods they evaluated, the limited memory Broyden-Fletcher-Goldfarb-Shanno (LBFGS) method is likely to be the most successful method for large-scale history matching problems. Thus, for our history-matching problem with both production data and seismic impedance change data, we have only used LBFGS method for the minimization.

It is usually possible to assume that the errors in production data are uncorrelated.

Because of the way the seismic impedance data are derived, it is not possible to make the same assumption of measurement error independence for the seismic impedance change data, which means that the part of the C_D matrix attributed to seismic impedance data will almost certainly not be diagonal. When we apply our method to field problems, such correlation among seismic impedance change data will have to be carefully evaluated (Aannonsen et al., 2002).

6.1 Automatic History Matching

The easiest way to include seismic impedance change data is to assume that one set of data is available before production begins, and the other set is available after a period of production. Then, for these two sets of seismic impedance data, it is necessary to compute their derivatives with respect to primary variables and model parameters, which depends on the relationships of impedance to saturations and pressure. In this work, we use Gassmann (1951) and Han (1986) equations. We will summarize the Gassmann and Han equations and their derivatives with respect to primary variables and model parameters in this report. The impedance Z is given by

$$Z = \rho V_p = \sqrt{\rho K + \frac{4}{3}\rho^2 V_s^2}\,, \tag{6.1}$$

where, K is the bulk modulus, ρ is bulk density and V_s is shear wave velocity. All of them can be computed by Gassman and Han equations and poro-elastic properties as

$$\rho = (\rho_o S_o + \rho_w S_w + \rho_g S_g)\phi + (1-\phi)\rho_{\text{solid}}\,, \tag{6.2}$$

$$K = K_{\text{grain}} \frac{K_{\text{frame}} + Q}{K_{\text{grain}} + Q}\,, \tag{6.3}$$

$$Q = \frac{K_{\text{fluid}}\,(K_{\text{grain}} - K_{\text{frame}})}{\phi\,(K_{\text{grain}} - K_{\text{fluid}})}\,, \tag{6.4}$$

$$\log_{10} K_{\text{frame}} = \log_{10} K_{\text{grain}} - 4.25\phi\,, \tag{6.5}$$

$$K_{\text{grain}} = \frac{1}{2}\left[\gamma K_c + (1-\gamma)K_s + \frac{K_s K_c}{K_s\gamma + K_c(1-\gamma)}\right]\,, \tag{6.6}$$

$$\frac{1}{K_{\text{fluid}}} = \frac{S_w}{K_w} + \frac{S_g}{K_g} + \frac{S_o}{K_o}\,, \tag{6.7}$$

$$V_s = 3520.0 - 4910.0\phi - 1890.0\gamma\,. \tag{6.8}$$

Using the chain rule, the derivatives of impedance with respect to pressure and saturation are

$$\frac{\partial Z}{\partial P} = \frac{1}{2}\left(\rho K + \frac{4}{3}\rho^2 V_s^2\right)^{-1/2} \times \left(K\frac{\partial \rho}{\partial P} + \frac{8}{3}V_s^2 \rho \frac{\partial \rho}{\partial P}\right)\,, \tag{6.9}$$

where

$$\frac{\partial \rho}{\partial P} = \phi \left(S_o \frac{\partial \rho_o}{\partial P} + S_w \frac{\partial \rho_w}{\partial P} + S_g \frac{\partial \rho_g}{\partial P} \right) , \tag{6.10}$$

and

$$\frac{\partial Z}{\partial S_w} = \frac{1}{2} \left(\rho K + \frac{4}{3} \rho^2 V_s^2 \right)^{-1/2} \times \left(K \frac{\partial \rho}{\partial S_w} + \rho \frac{\partial K}{\partial S_w} + \frac{8}{3} V_s^2 \rho \frac{\partial \rho}{\partial S_w} \right) , \tag{6.11}$$

where

$$\frac{\partial \rho}{\partial S_w} = \phi \left(\rho_w - \rho_o \right) , \tag{6.12}$$

$$\begin{aligned} \frac{\partial K}{\partial S_w} = & \frac{\left(\left(K_{\text{grain}}^2 - K_{\text{grain}} K_{\text{frame}} \phi - K_{\text{grain}} K_{\text{frame}} \right) \frac{\partial K_{\text{fluid}}}{\partial S_w} \right)}{\left(K_{\text{grain}}^2 \phi + K_{\text{fluid}} \left(K_{\text{grain}} - K_{\text{grain}} \phi - K_{\text{frame}} \right) \right)} \\ & - \frac{\left(\left(K_{\text{grain}} - K_{\text{grain}} \phi - K_{\text{frame}} \right) \frac{\partial K_{\text{fluid}}}{\partial S_w} \right)}{\left(K_{\text{grain}}^2 \phi + K_{\text{fluid}} \left(K_{\text{grain}} - K_{\text{grain}} \phi - K_{\text{frame}} \right) \right)} \\ & \times \frac{\left(K_{\text{grain}}^2 K_{\text{frame}} \phi + K_{\text{fluid}} \left(K_{\text{grain}}^2 - K_{\text{grain}} K_{\text{frame}} \phi - K_{\text{grain}} K_{\text{frame}} \right) \right)}{\left(K_{\text{grain}}^2 \phi + K_{\text{fluid}} \left(K_{\text{grain}} - K_{\text{grain}} \phi - K_{\text{frame}} \right) \right)} , \end{aligned} \tag{6.13}$$

and

$$\frac{\partial K_{\text{fluid}}}{\partial S_w} = \frac{\frac{1}{K_o} - \frac{1}{K_w}}{\left(\frac{S_w}{K_w} + \frac{1 - S_w - S_g}{K_o} + \frac{S_g}{K_g} \right)^2} , \tag{6.14}$$

$$\frac{\partial Z}{\partial S_g} = \frac{1}{2} \left(\rho K + \frac{4}{3} \rho^2 V_s^2 \right)^{-1/2} \times \left(K \frac{\partial \rho}{\partial S_g} + \rho \frac{\partial K}{\partial S_g} + \frac{8}{3} V_s^2 \rho \frac{\partial \rho}{\partial S_g} \right) , \tag{6.15}$$

where

$$\frac{\partial \rho}{\partial S_g} = \phi \left(\rho_g - \rho_o \right) , \tag{6.16}$$

$$\begin{aligned} \frac{\partial K}{\partial S_g} = & \frac{\left(\left(K_{\text{grain}}^2 - K_{\text{grain}} K_{\text{frame}} \phi - K_{\text{grain}} K_{\text{frame}} \right) \frac{\partial K_{\text{fluid}}}{\partial S_g} \right)}{\left(K_{\text{grain}}^2 \phi + K_{\text{fluid}} \left(K_{\text{grain}} - K_{\text{grain}} \phi - K_{\text{frame}} \right) \right)} \\ & - \frac{\left(\left(K_{\text{grain}} - K_{\text{grain}} \phi - K_{\text{frame}} \right) \frac{\partial K_{\text{fluid}}}{\partial S_g} \right)}{\left(K_{\text{grain}}^2 \phi + K_{\text{fluid}} \left(K_{\text{grain}} - K_{\text{grain}} \phi - K_{\text{frame}} \right) \right)} \\ & \times \frac{\left(K_{\text{grain}}^2 K_{\text{frame}} \phi + K_{\text{fluid}} \left(K_{\text{grain}}^2 - K_{\text{grain}} K_{\text{frame}} \phi - K_{\text{grain}} K_{\text{frame}} \right) \right)}{\left(K_{\text{grain}}^2 \phi + K_{\text{fluid}} \left(K_{\text{grain}} - K_{\text{grain}} \phi - K_{\text{frame}} \right) \right)} , \end{aligned} \tag{6.17}$$

and

$$\frac{\partial K_{\text{fluid}}}{\partial S_g} = \frac{\frac{1}{K_o} - \frac{1}{K_g}}{\left(\frac{S_w}{K_w} + \frac{1 - S_w - S_g}{K_o} + \frac{S_g}{K_g} \right)^2} , \tag{6.18}$$

$$\frac{\partial Z}{\partial \phi} = \frac{1}{2}\left(\rho K + \frac{4}{3}\rho^2 V_s^2\right)^{-1/2} \times \left(\left(K + \frac{8}{3}\rho V_s^2\right)\frac{\partial \rho}{\partial \phi} + \rho\frac{\partial K}{\partial \phi} + \frac{8}{3}\rho^2 V_s \frac{\partial V_s}{\partial \phi}\right), \tag{6.19}$$

where

$$\frac{\partial V_s}{\partial \phi} = -4910.0\,, \tag{6.20}$$

$$\frac{\partial \rho}{\partial \phi} = \rho_o\left(1 - S_w - S_g\right) + \rho_w S_w + \rho_g S_g - \rho_{\text{solid}}\,, \tag{6.21}$$

$$\begin{aligned}\frac{\partial K}{\partial \phi} = & \frac{\left(\left(K_{\text{grain}}^2 - K_{\text{fluid}}K_{\text{grain}}\right)\left(\phi\frac{\partial K_{\text{frame}}}{\partial \phi} + K_{\text{frame}}\right) - K_{\text{fluid}}K_{\text{grain}}\frac{\partial K_{\text{frame}}}{\partial \phi}\right)}{\phi\left(K_{\text{grain}}^2 - K_{\text{fluid}}K_{\text{grain}}\right) + K_{\text{fluid}}K_{\text{grain}} - K_{\text{fluid}}K_{\text{frame}}} \\ & - \frac{\left(K_{\text{frame}}\phi\left(K_{\text{grain}}^2 - K_{\text{fluid}}K_{\text{grain}}\right) + K_{\text{fluid}}K_{\text{grain}}^2 - K_{\text{fluid}}K_{\text{grain}}K_{\text{frame}}\right)}{\left(\phi\left(K_{\text{grain}}^2 - K_{\text{fluid}}K_{\text{grain}}\right) + K_{\text{fluid}}K_{\text{grain}} - K_{\text{fluid}}K_{\text{frame}}\right)} \\ & \times \frac{\left(K_{\text{grain}}^2 - K_{\text{fluid}}K_{\text{grain}} - K_{\text{fluid}}\frac{\partial K_{\text{frame}}}{\partial \phi}\right)}{\left(\phi\left(K_{\text{grain}}^2 - K_{\text{fluid}}K_{\text{grain}}\right) + K_{\text{fluid}}K_{\text{grain}} - K_{\text{fluid}}K_{\text{frame}}\right)}\,,\end{aligned} \tag{6.22}$$

and

$$\frac{\partial K_{\text{frame}}}{\partial \phi} = -4.25\ln(10)K_{\text{grain}}10^{-4.25}\,. \tag{6.23}$$

There is no direct dependance between seismic impedance and permeability k, so the sensitivity of seismic impedance with respect to permeability depends entirely on the indirect effect of permeability on pressure and saturation.

6.1.1 LBFGS Method

Previous studiess in TUPREP have showed that BFGS method is the most successful quasi-Newton method for history matching. However, its drawback is that it needs to store the Hessian matrix approximation, which will be impractical when large scale models are considered. The alternative is the limited memory BFGS (LBFGS) method of Nocedal (1980). LBFGS method only requires storage of a few vectors and uses these vectors to implicitly construct Hessian matrix approximation. A detailed discussion of the application of the LBFGS method to large scale history-matching problems can be found in Zhang and Reynolds (2002b).

6.2 Results

To test the effect of integration of both seismic impedance change data and production data, we used two models. One is a small synthetic model and the other one is a semi-synthetic

model created from a middle east oil field. In the following sections, we will discuss them in detail.

6.2.1 Synthetic Model

This small synthetic model has two layers. Each layer has 10×10 gridblocks with size equal to 40 ft. The vertical size of each gridblock is 30 ft. The only model parameter we adjust is log of horizontal permeability. For vertical permeability, we use a multiplier, here equal to 0.1, and for porosity, we fix it at the true value 0.2 in every gridblock. To compare matching results before integration of seismic impedance change data and after such integration, we divide each layer into three different zones. In each zone, horizontal permeability is homogeneous. However, in each zone, the horizontal permeability has a different value. Since there are very obvious edges among these three zones, it is easy to decide if one matching result is better than another. If a method is performing well, such edges would presumably be clearer. Here, we only use bottom hole pressure P_{wf} as production data.

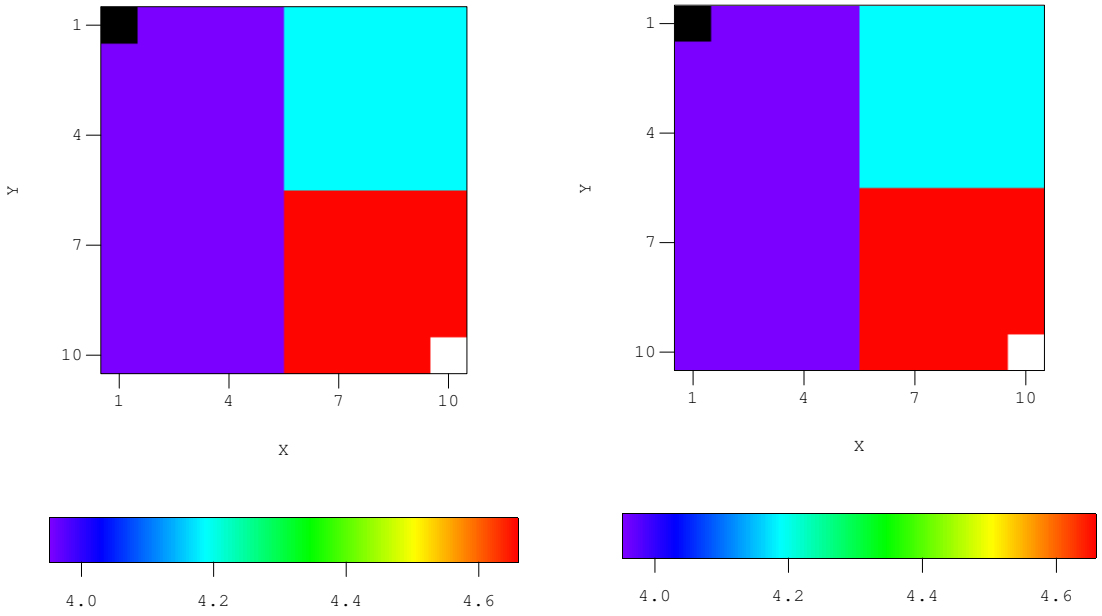
The true values of $\ln(k)$ in three zones are 4.0, 4.2 and 4.6, which can be seen from Fig. 6.1. The white point in lower-right corner denotes an injection well and the black one is a production well. Both of them perforate thoroughly in two layers. This is a three-phase problem.

Matching Production Data Only

In each well, we use 10 pressure data to do the history matching so that totally we have 20 pressure data to adjust 200 model parameters. We use a homogeneous value, 4.0, as the initial guess and the prior model. Final matching results are in Fig. 6.2 and Fig. 6.3. From Fig. 6.2 and Fig. 6.3, we can see that from production data alone, it is not possible to obtain a good estimate of the permeability field. The boundary between the blue area and the red area is more like arc than straight line, which simply reflects the symmetry of the problem and the area of water saturation change from the injection well in that corner.

Combination of Seismic and Production Data

In this section, we evaluate the improvement in the estimate after the integration of seismic impedance change data. Results for the same problem can be seen in Fig. 6.4 and Fig. 6.5. Compared to Fig. 6.2, Fig. 6.3 and Fig. 6.1, the permeability estimate from integration of both seismic impedance change data and production data is better than the estimate obtained from only using production data. Moreover, it is much closer to the true



(a) Top Layer (b) Bottom Layer

Figure 6.1: True log horizontal permeability ($\ln(k)$) field

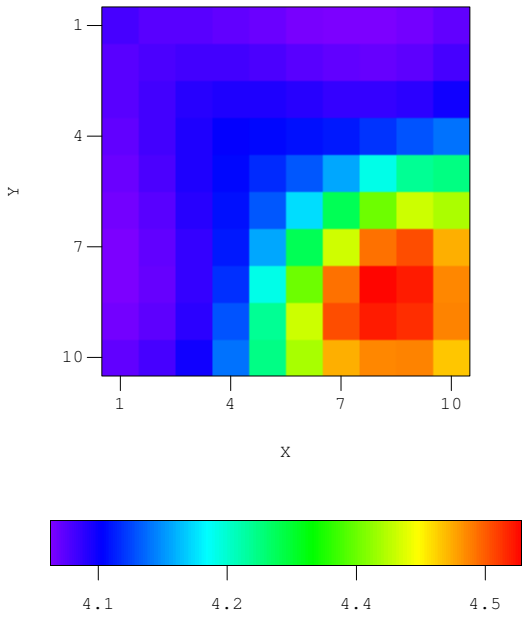


Figure 6.2: Estimate of log horizontal permeability field in top layer by integration of production data only

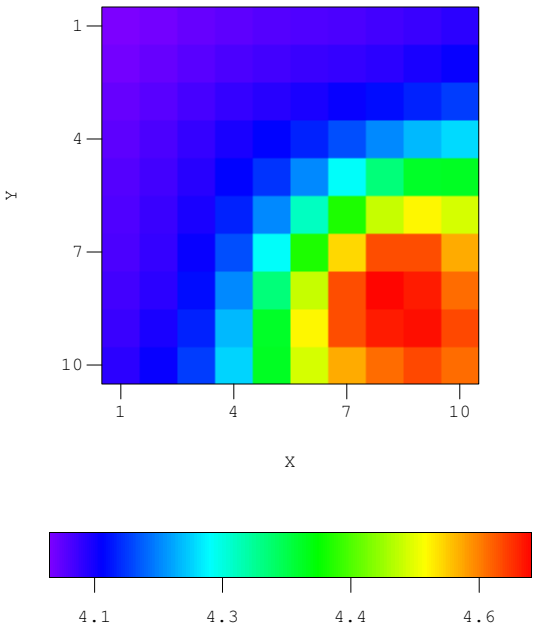


Figure 6.3: Estimate of log horizontal permeability field in bottom layer by integration of production data only

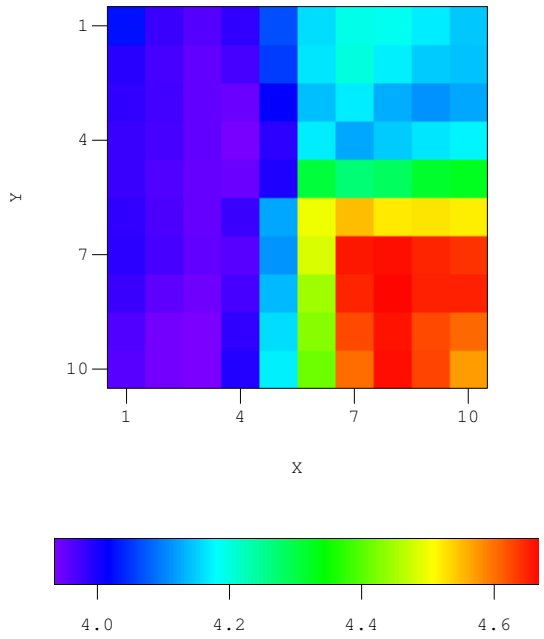


Figure 6.4: Estimate of log horizontal permeability field in top layer by integration of both production data and seismic impedance change data

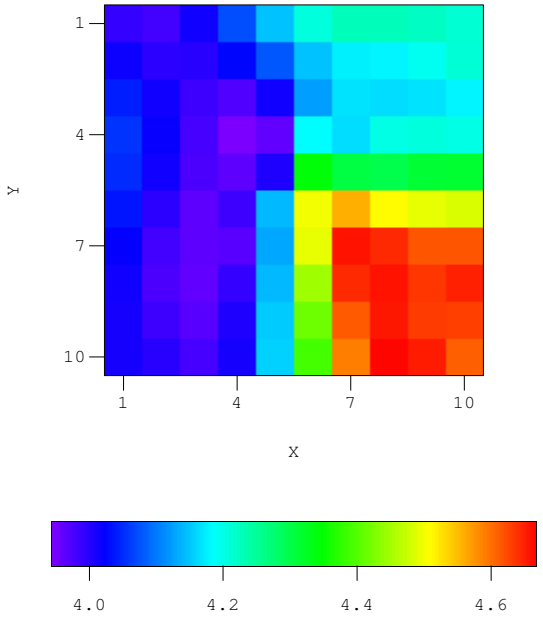


Figure 6.5: Estimate of log horizontal permeability field in bottom layer by integration of both production data and seismic impedance change data

field, especially its boundary of the homogeneous regions is much clearer. The reason for the improvement is that the spatial density of seismic impedance change data provides more constraints in gridblocks that are far from well locations. Obviously, seismic impedance change data makes the possible solution space narrower and can provide much better estimates of properties, which is what we predicted when we performed the preliminary investigation work. In summary, adding seismic impedance change data into automatic history matching is feasible and provides improved estimates.

6.2.2 Semi-Synthetic Model

Because our final goal is to apply the method to a field problem, it is important to test the method on real field data. As an intermediate step, before application to real field data, we are applying the method to a large "field-scale" problem rather than just using it on small synthetic model. One possible candidate is the simulation model from a field provided by an industry member to TUPREP. When TUPREP obtained that data, the initial aim was to test automatic history matching method on a large scale single-phase real field problem. At this time, we intend to investigate if seismic impedance change data can give a reasonable estimates of properties in history matching problems. This goal requires comparison with a true field. We used the first five layers of the reservoir created by the company geoscientists as the true geological model. It would presumably be unknown, except for observations at well locations. Using a covariance estimated from the model, we created a new synthetic model by Sequential Gaussian Simulation (SGS). A comparison of the supplied model with our model can be seen from Table 6.1. The real field has a very high initial reservoir pressure

Parameters	**True Field**	**Semi-Synthetic Model**
True Model	N/A	Synthetic model
Prior Model	Synthetic model with facies change	Generated by SGS
Well Completion	Partially perforated	Fully perforated
Initial Reservoir Pressure	11950 Psi	4000 Psi
Bubble Point Pressure	3586 Psi	3586 Psi
Number of Layers	9	5
Gridblocks in Each Layer	59×49	59×49

Table 6.1: Comparison between real field and semi-synthetic model

and relatively low bubble point pressure. Under such conditions, it remains single-phase even

after a long production period. A deep, single-phase reservoir, would not be a good candidate for 4D seismic. Thus, in order to create a more realistic example, we changed the initial reservoir pressure to be slightly above the bubble point pressure, to ensure that free gas will evolve soon after production begins.

Creation of Prior Model

Sequential Gaussian Simulation was used to create the prior porosity field, which also served as the initial guess. The prior horizontal permeability field, was generated directly from the porosity field using a functional relationship. That correlation equation is generated from cross plot of porosity and horizontal permeability in well locations, which can be seen in Fig. 6.6. Using regression, we have the relationship as

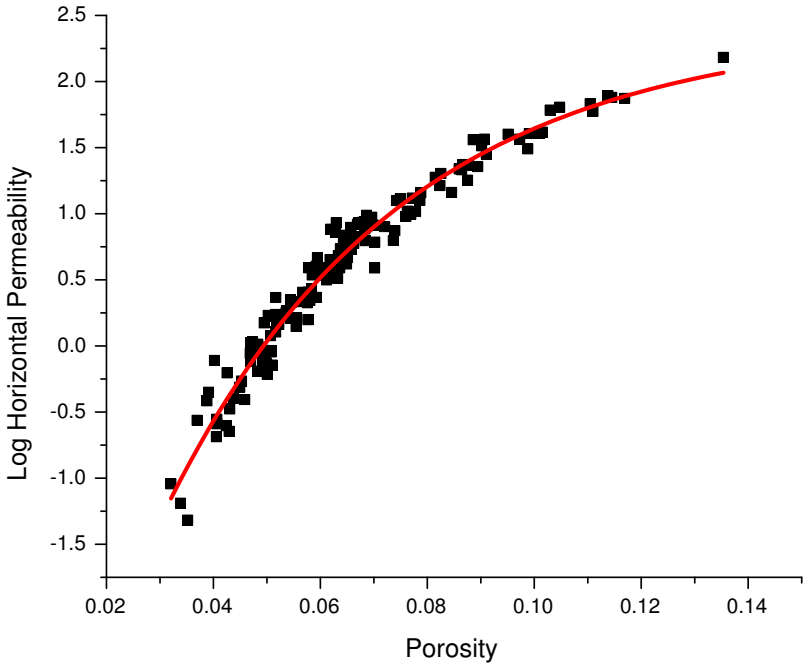


Figure 6.6: Correlation between porosity and log horizontal permeability in well locations

$$\ln k_h = 2.41073 - 7.3652 \times \exp\left(-\frac{\phi}{0.04419}\right) . \tag{6.24}$$

The prior horizontal permeability field was computed directly from Eq. 6.24 once we had simulated the porosity field. The relationship between vertical and horizontal permeability

was also estimated from a crossplot. A satisfactory relationship is provided by

$$k_v = a \times k_h \,, \tag{6.25}$$

where, a is a constant multiplier. In our semi-synthetic model, we assume that $a = 0.002$ to create the vertical permeability field.

To summarize, we use the true model to create observation data, here seismic impedance change data. Then, we adjust both the prior porosity field and horizontal permeability field to match the data from the true field. We intend to match our observation data as well as possible. In this example, we do not use any production data, in order to assess the constraint the seismic impedance change data provides in such large scale problems.

The true horizontal permeability field in the first four layers can be seen in Fig. 6.7. The prior permeability field for the first four layers is in Fig. 6.8. The fifth layer of both true permeability field and prior field can be seen in Fig. 6.9 and Fig. 6.10. Similarly, true and prior porosity field of each layer are in Figs. 6.11, 6.12, 6.13 and 6.14. From true permeability and porosity field, we can see that there is an obvious discontinuity in properties between the right lower part and the left upper part of the reservoir in each layer. This is a depth effect, higher porosities and permeabilities occur at shallower depths. Most of the wells are also located in this area, which gives more gas after production than in the low permeability and porosity area. This difference makes the seismic impedance change quite different in these two areas, which can be seen in Fig. 6.15 and Fig. 6.16. In the left region, because of lower gas saturation, the seismic impedance change value is low. The region on the right side has a higher value because of higher gas saturation. Moreover, with increase of depth, reservoir pressure becomes higher, which makes it more difficult for gas to come out, then seismic impedance change values become smaller with depth.

History Matching

In this report, we used only the seismic impedance change data—not the change in amplitude—and assumed that each gridblock had a seismic impedance change data so that the number of data was as same as the number of gridblocks. We adjusted both horizontal permeability and porosity, so the number of model parameters was twice the number of gridblocks. The minimization required 100 iteration loops, which is the maximum number of iterations allowed in our code. The square summation of data mismatch at the end of the iterations was lower than the total number of data, which implies that the resulting model was acceptable. The objective function behavior and data mismatch part decrease can be seen in Fig. 6.17(a)

and Fig. 6.17(b). The matching results of horizontal permeability field and porosity field can be seen in Fig. 6.18, Fig. 6.19, Fig. 6.20 and Fig. 6.21.

From these maps, we make the following observations,

1. Seismic impedance change data provides useful constraints in history matching problem. Especially in large scale models, the use of seismic impedance change data can decrease uncertainty. The results for both permeability and porosity include features which are similar to the true model.

2. Estimates of the porosity field are better than the estimates for permeability. The reason is almost certainly that seismic impedance change data is more sensitive to porosity than to permeability.

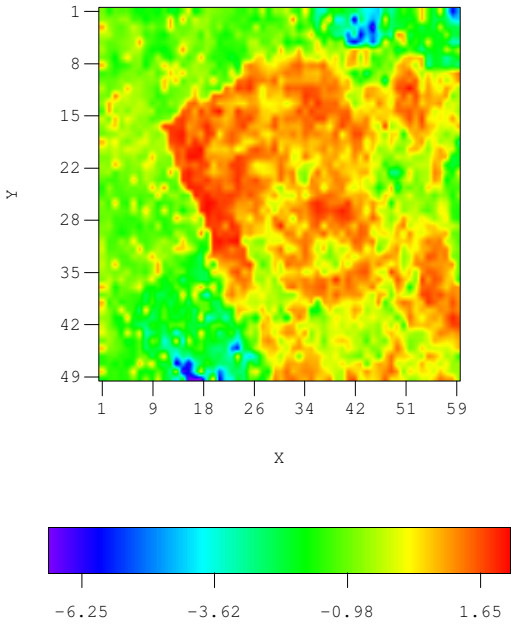
3. From the top layer to the bottom layer, reservoir pressure increases, which means that there less free gas evolves in the deeper layers. This decrease in gas saturation results in smaller changes in seismic impedance in deeper layers. Thus, the results in deeper layers were not as good as the results in top layers. This can be observed clearly from Fig. 6.18 and Fig. 6.19.

4. The properties in the upper left region of the simulation model do not change very much after integration of time-lapse seismic because there are almost no wells and the depth is greater, which makes it more difficult to have gas accumulated there. Thus, we do not have significant seismic impedance change in that region.

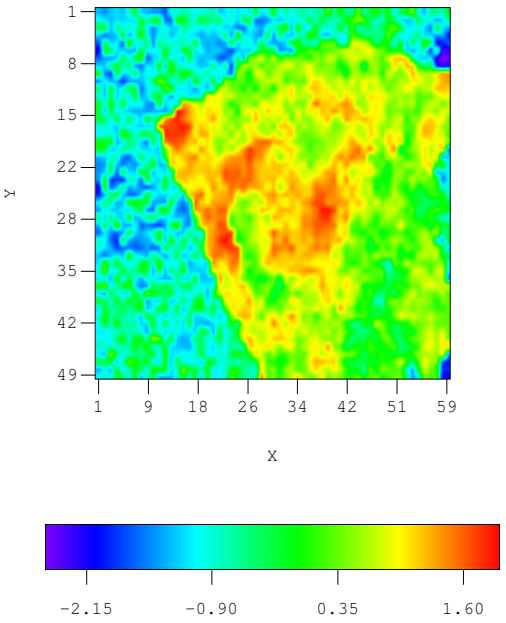
Finally, we note that integration of seismic impedance change data into automatic history matching seems to provide dramatically improved reservoir models, even when the data are noisy. The spatial density of the data appears to compensate for the sparsity of production data, especially in large scale models.

6.2.3 Real Case Study from Bay Marchand field, Gulf of Mexico

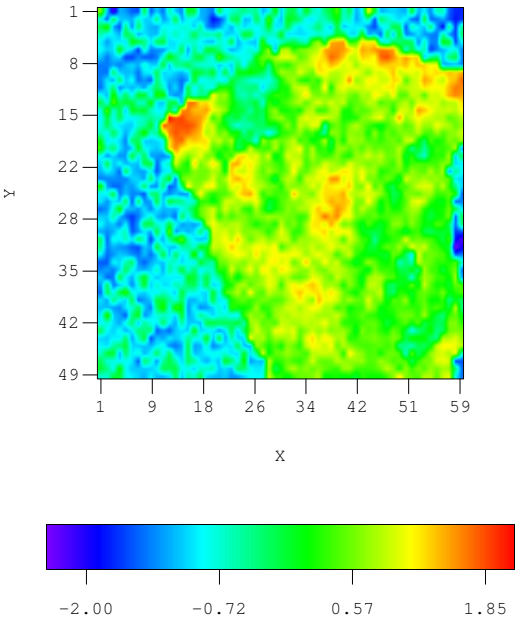
The basic principles and work flow of integration of both time-lapse seismic impedance change data and production data have been demonstrated clearly using the semi-synthetic and synthetic models discussed previously, which establishes the effectiveness of using seismic impedance change data in reservoir characterization work. In this section, we will apply the method to a real case, Bay Marchand field in the Gulf of Mexico. Since all data in this study were provided by Chevron-Texaco, the discussions shown here are limited to general descriptions, instead of detailed data and results analysis.



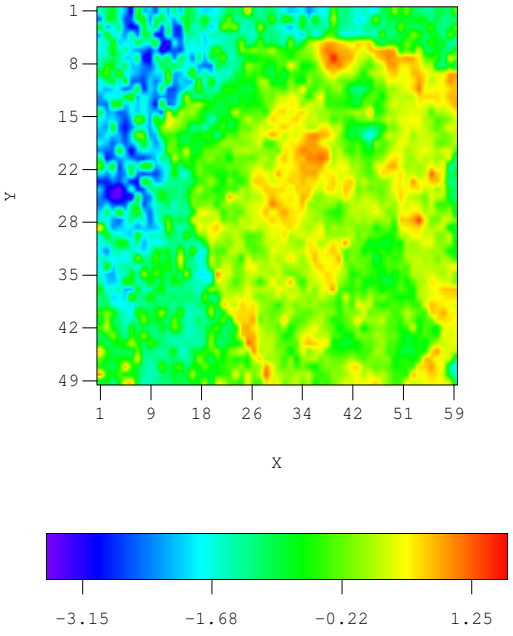
(a) Layer One



(b) Layer Two

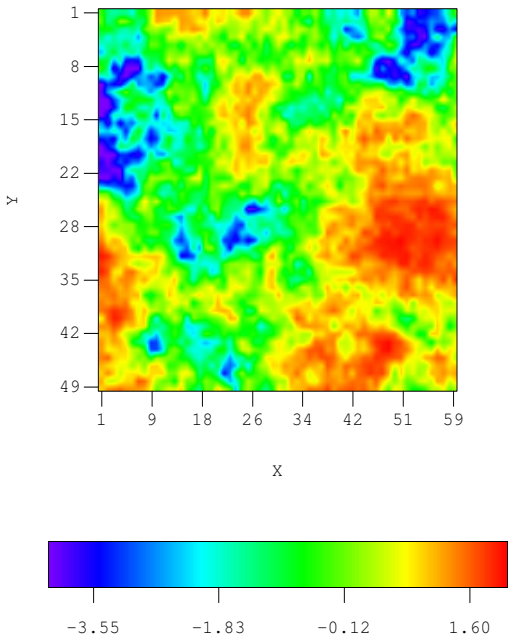


(c) Layer Three

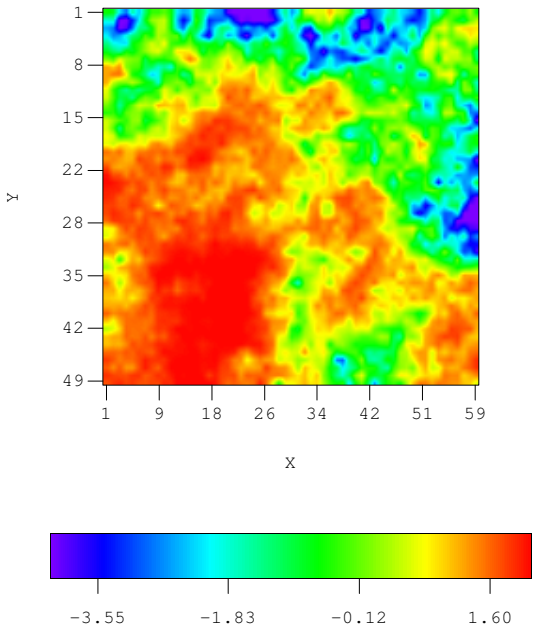


(d) Layer Four

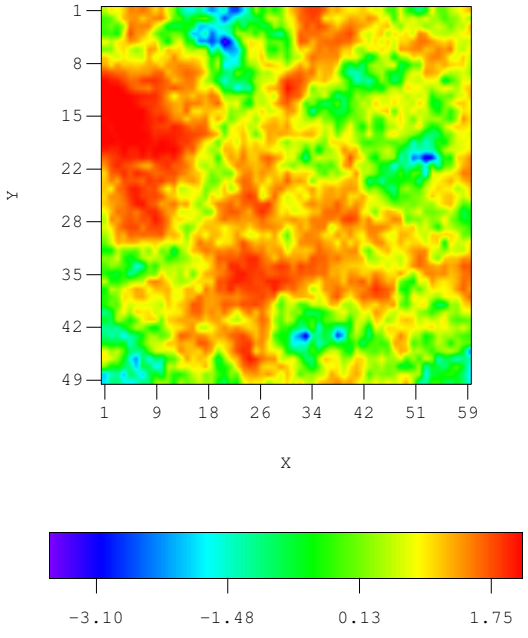
Figure 6.7: True log horizontal permeability field in first four layers



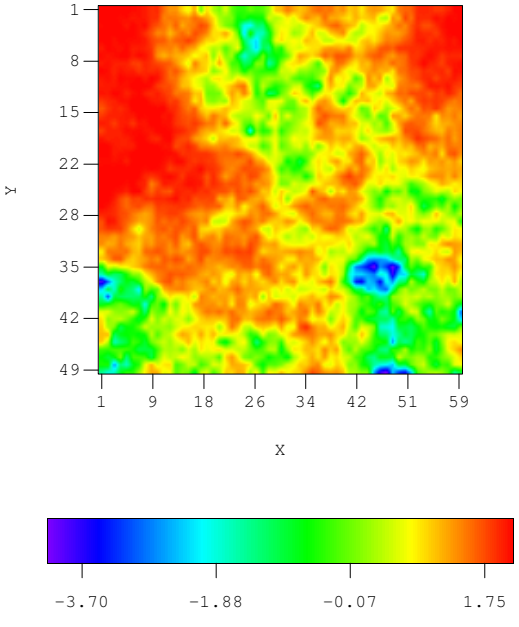
(a) Layer One



(b) Layer Two



(c) Layer Three



(d) Layer Four

Figure 6.8: Prior log horizontal permeability field in first four layers

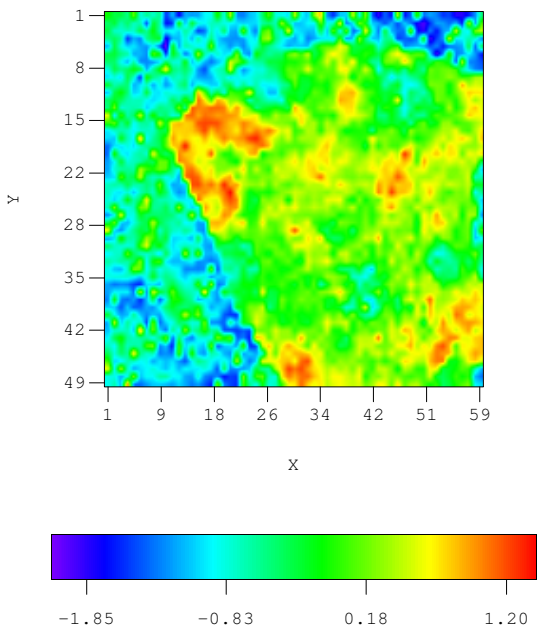


Figure 6.9: True log horizontal permeability field in the fifth layer

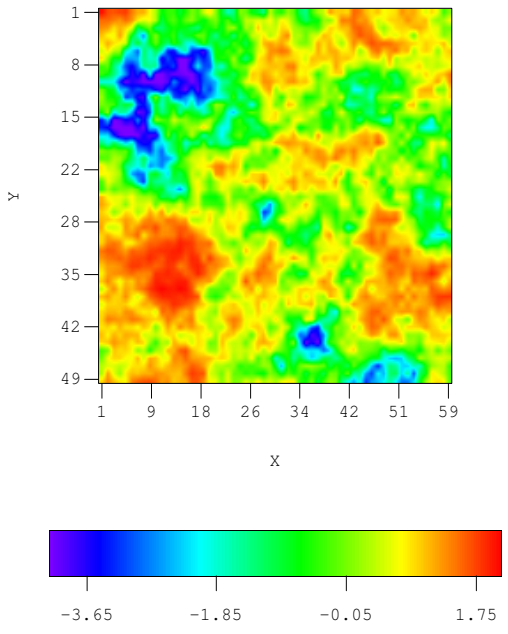
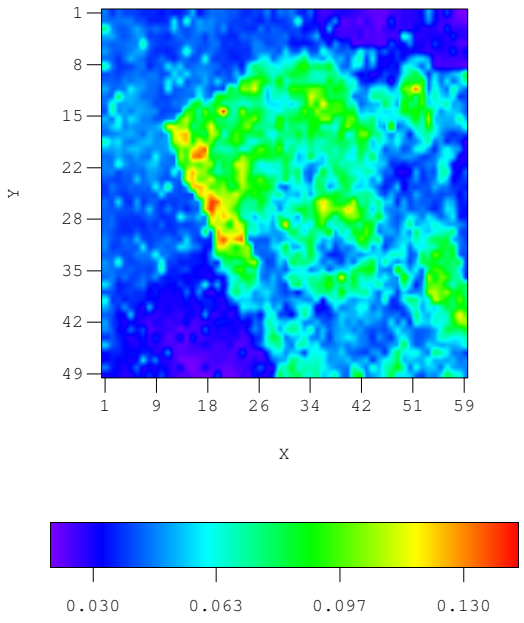
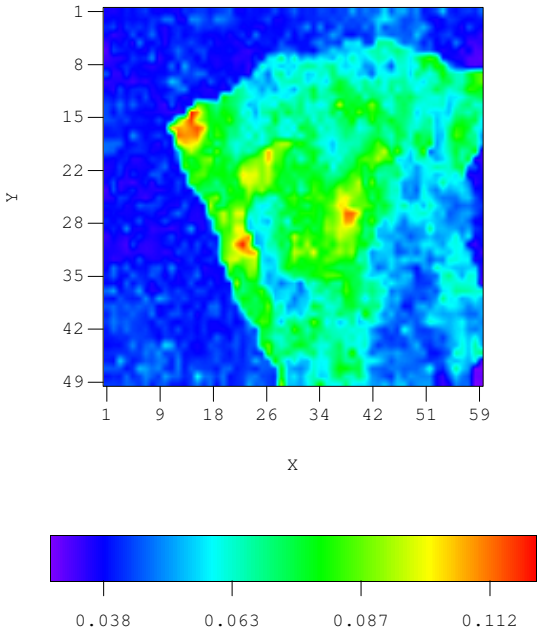


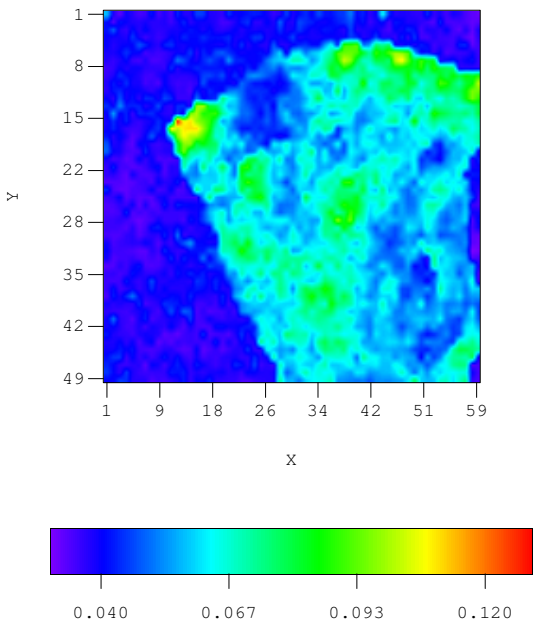
Figure 6.10: Prior log horizontal permeability field in the fifth layer



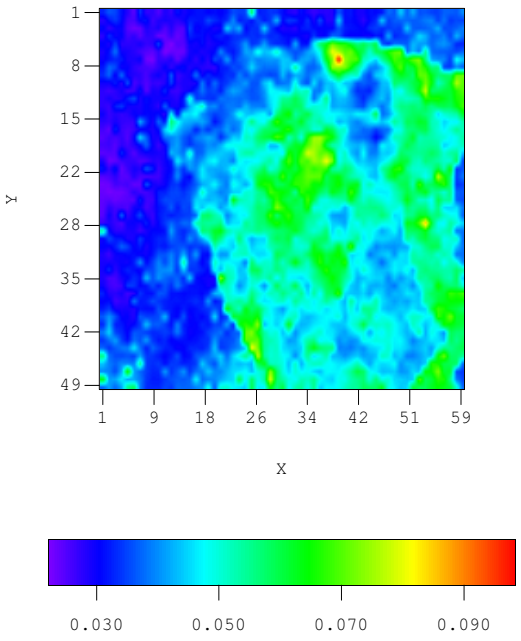
(a) Layer One



(b) Layer Two

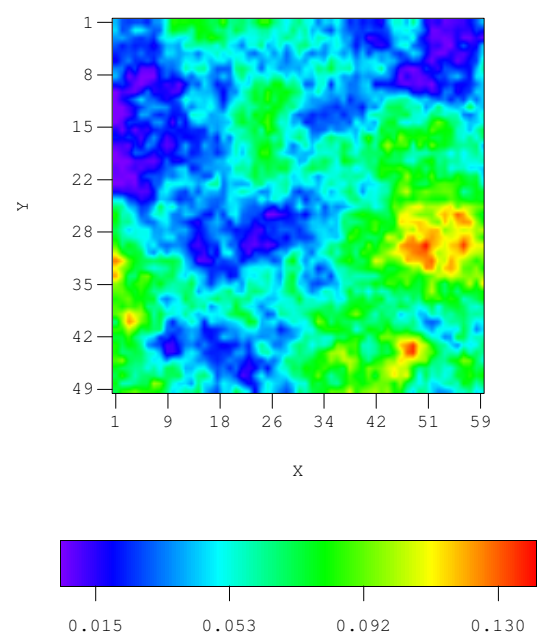


(c) Layer Three

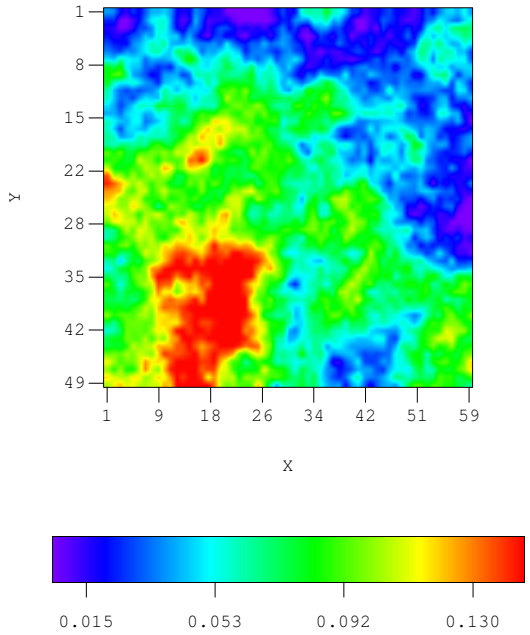


(d) Layer Four

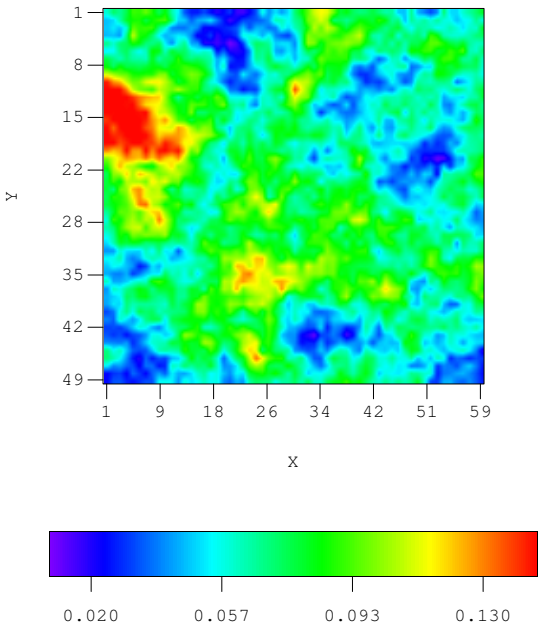
Figure 6.11: True porosity field in first four layers



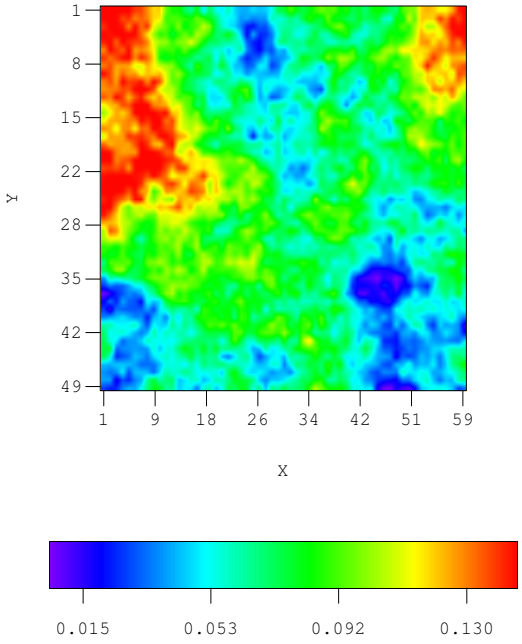
(a) Layer One



(b) Layer Two



(c) Layer Three



(d) Layer Four

Figure 6.12: Prior porosity field in first four layers

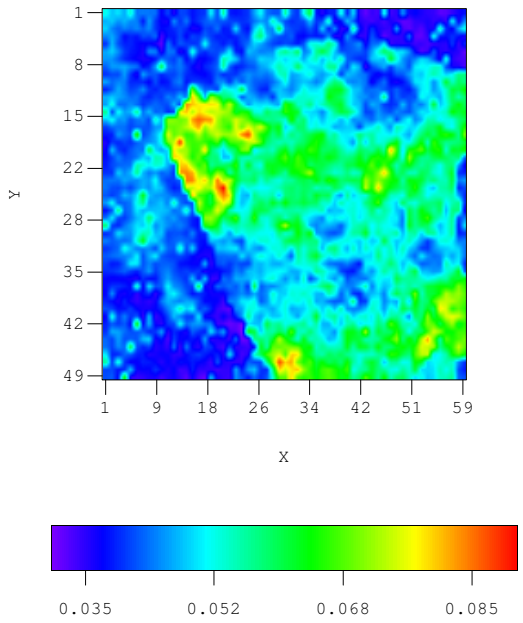


Figure 6.13: True porosity field in the fifth layer

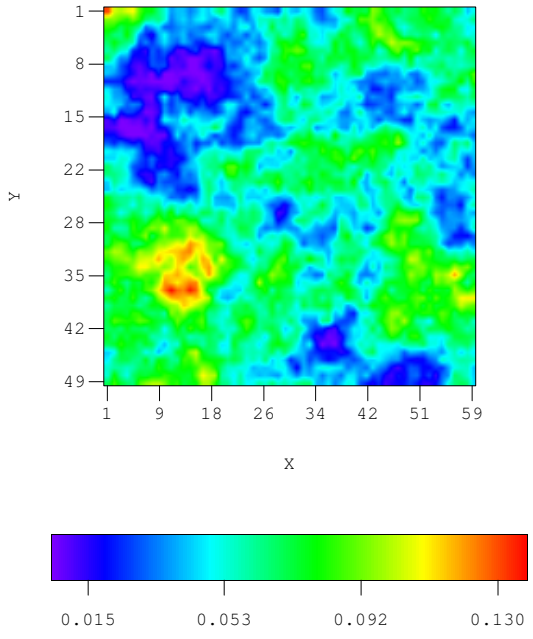
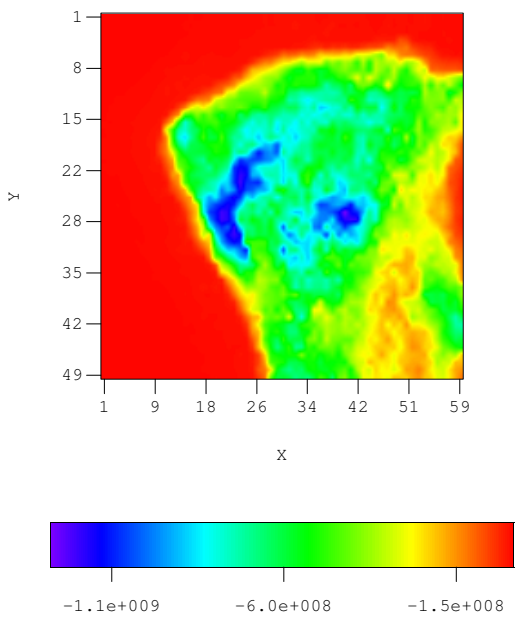
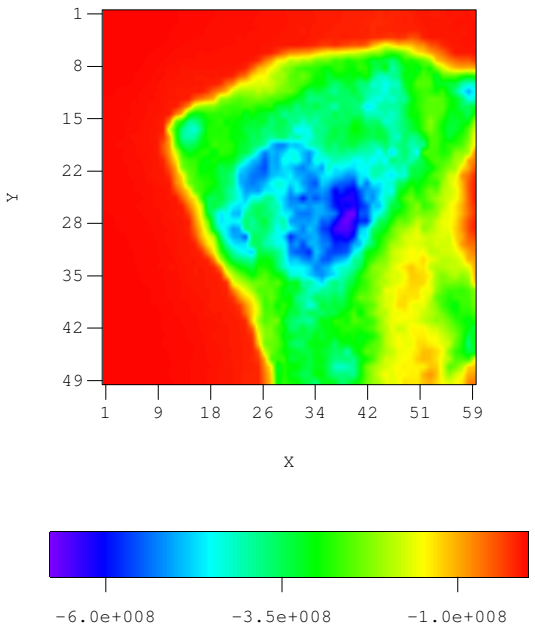


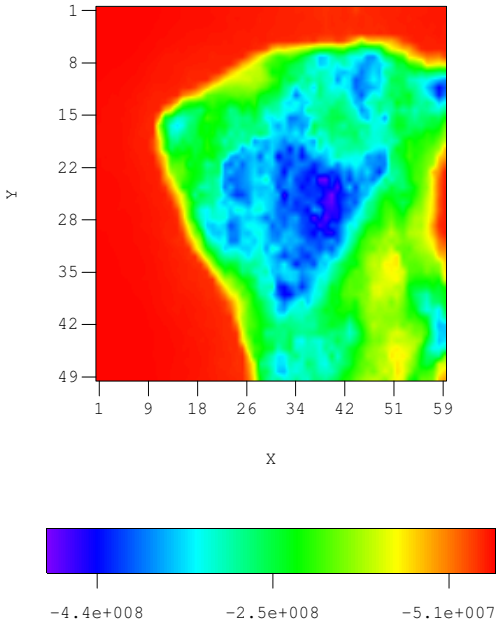
Figure 6.14: Prior porosity field in the fifth layer



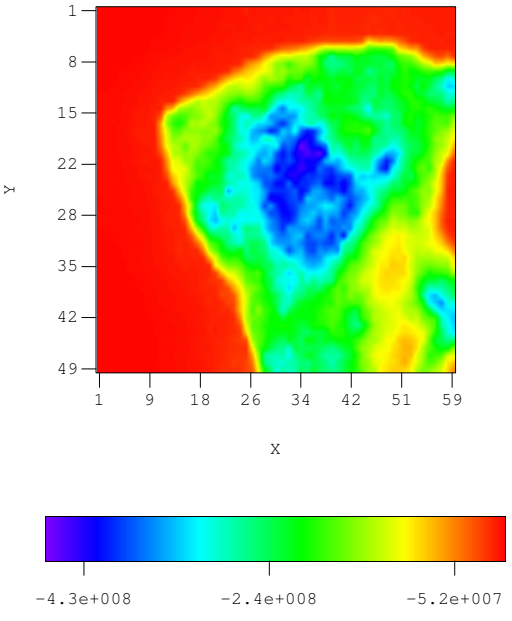
(a) Layer One



(b) Layer Two



(c) Layer Three



(d) Layer Four

Figure 6.15: Seismic impedance change in first four layers

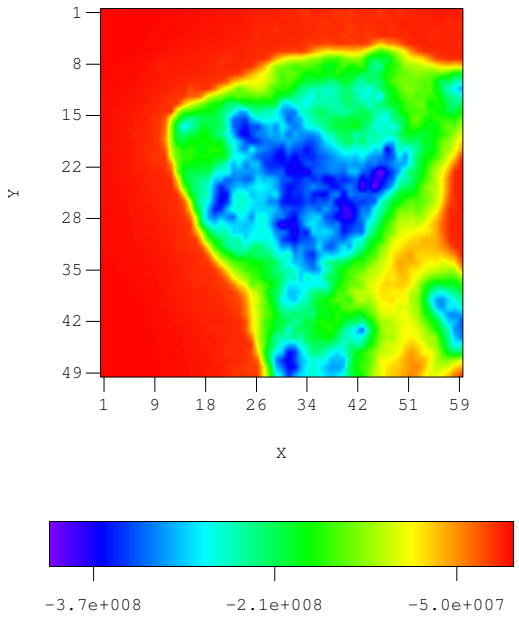
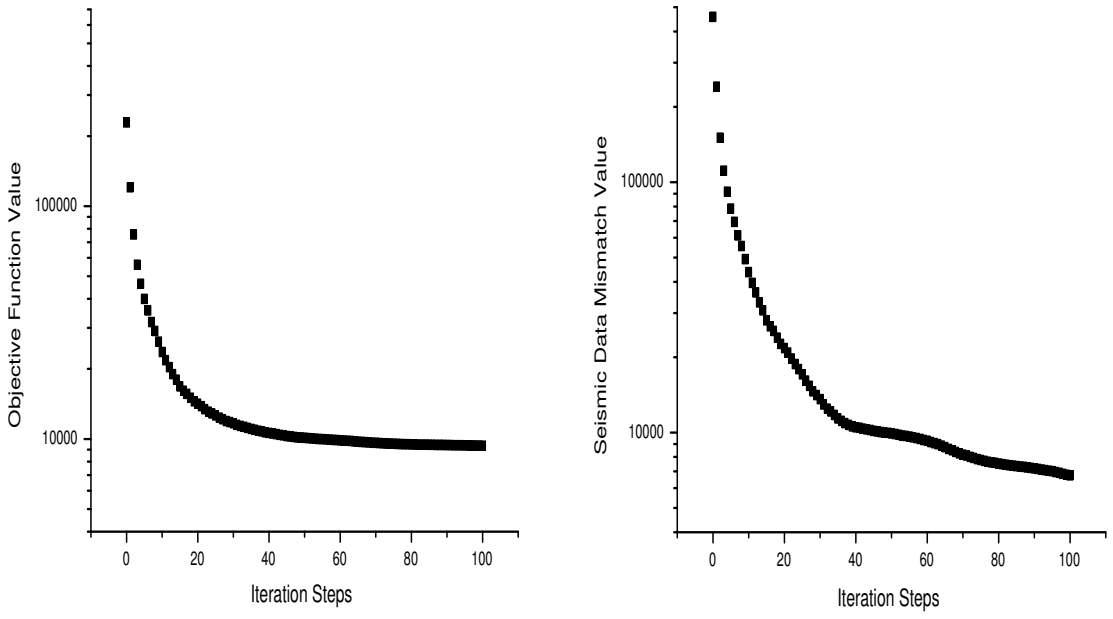


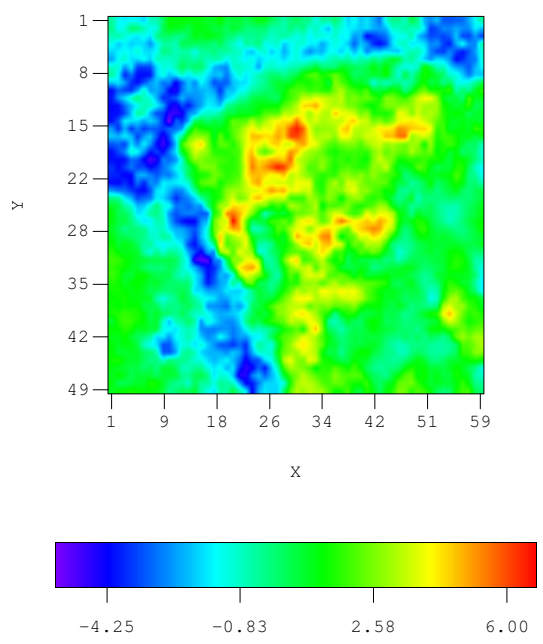
Figure 6.16: Seismic impedance change in the fifth layer



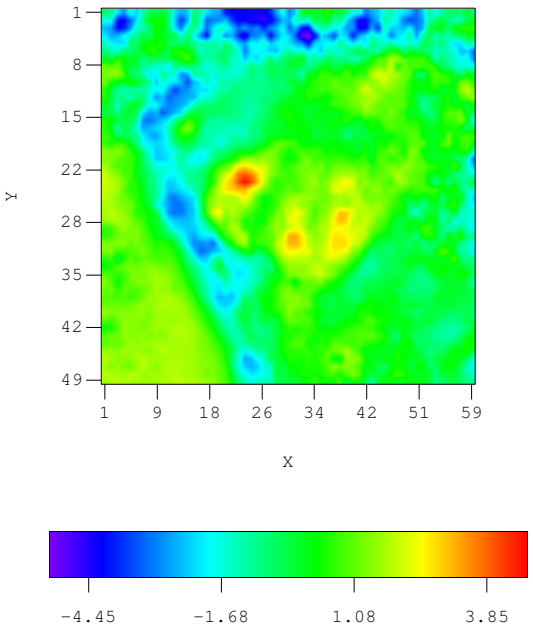
(a) Objective Function

(b) Seismic Data Mismatch

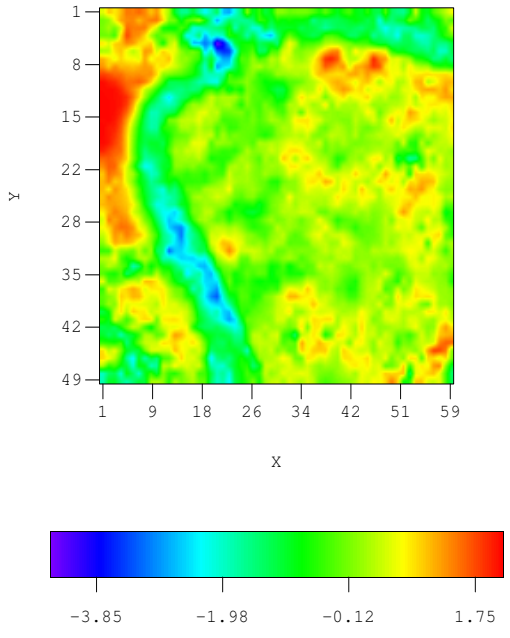
Figure 6.17: Objective function and seismic data mismatch decrease



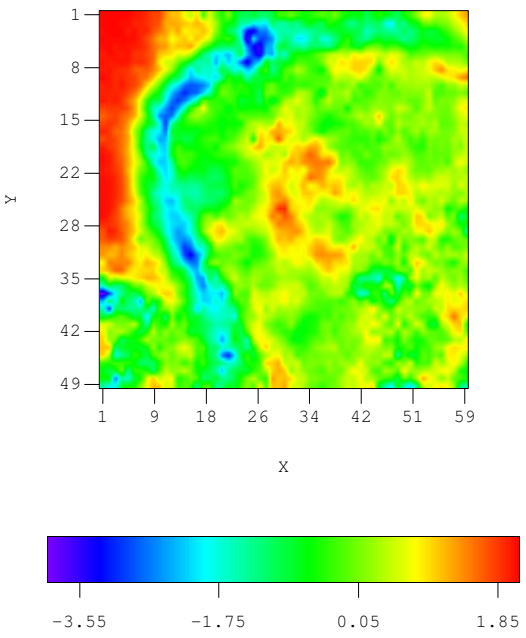
(a) Layer One



(b) Layer Two

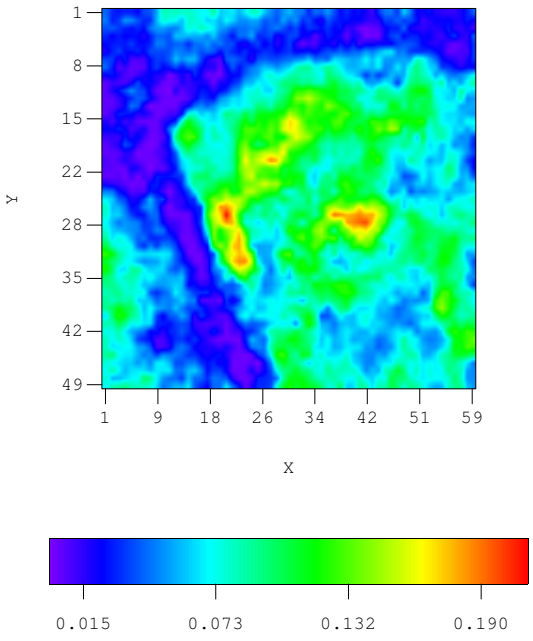


(c) Layer Three

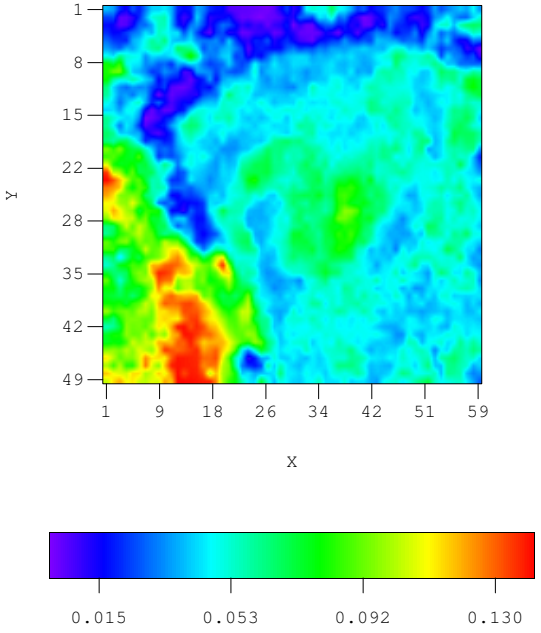


(d) Layer Four

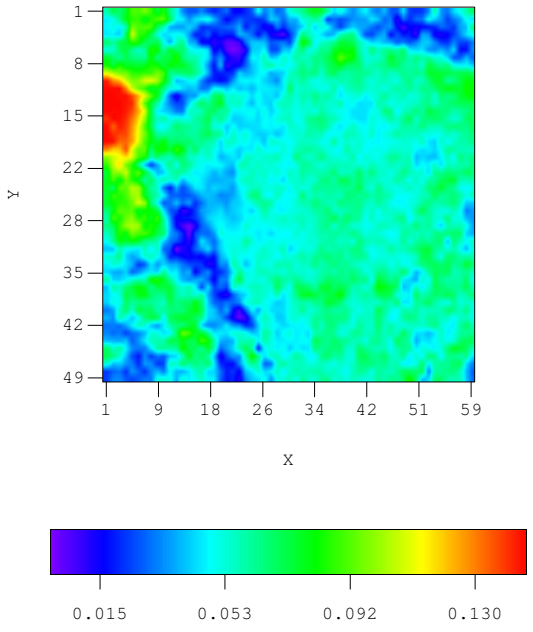
Figure 6.18: Estimate of log horizontal permeability field in first four layers



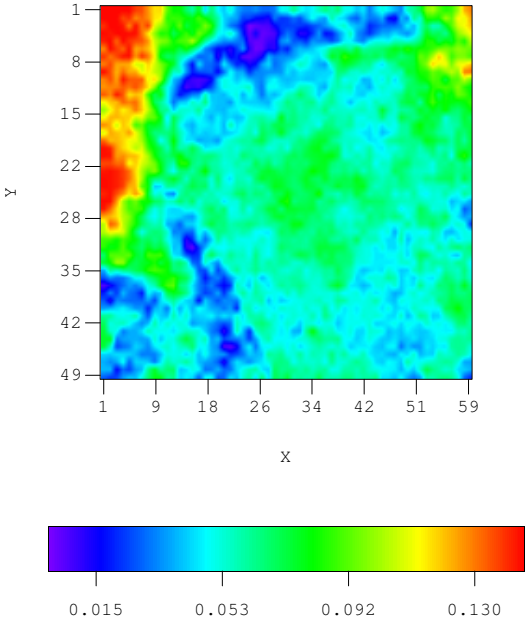
(a) Layer One



(b) Layer Two



(c) Layer Three



(d) Layer Four

Figure 6.19: Estimate of porosity field in first four layers

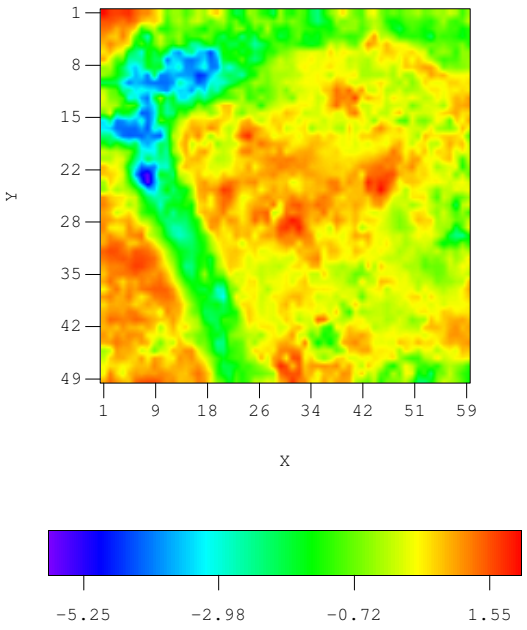


Figure 6.20: Estimate of log horizontal permeability field in the fifth layer

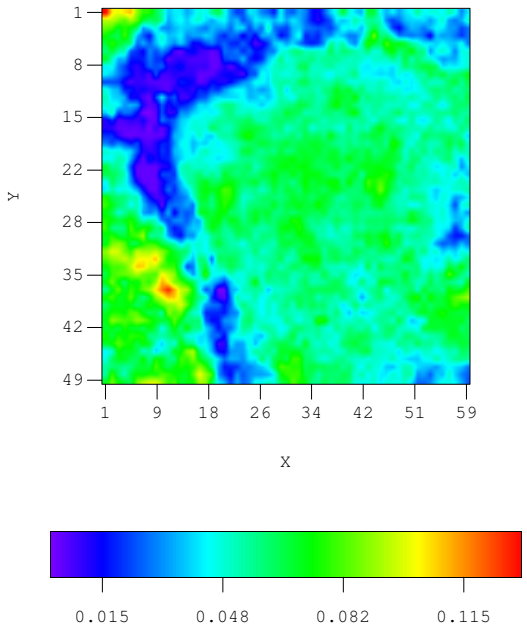


Figure 6.21: Estimate of porosity field in the fifth layer

Introduction

The Bay Marchand field is a mature field with production history over 40 years. The area in our history matching study is the 7100 sand, which has strong aquifer support. There are 7 producing wells in this area. Some of the 7 wells are sidetrack wells of the straight wells drilled from the same wellheads. Most of the wells are completed within the middle and lower zones, where both the porosity and the permeability have generally good values. Monthly oil, water and gas production rate were provided, from which the water oil ratio (WOR) and gas oil ratio (GOR) were calculated and used as observed production data in our history matching work.

A reservoir simulation model with $53\times18\times23$ gridblocks was also provided, along with the initial permeability and porosity distributions. Most of the gridblocks have sizes varying from 400 ft to 100 ft along X and Y directions, while the size along Z direction is more variable with some gridblocks less than 1 ft in thickness. There are three relative permeability zones in the simulation model, one of which, along with pore volume modifications, was used to simulate the aquifer support. Two 3D seismic surveys are available for Bay Marchand. The first was shot in 1987 by Chevron and the other one was acquired in 1998 by Geco-Prakla, which is now WesternGeco.

The field and the seismic surveys are described thoroughly by Behrens et al. (2002).

Rock physics model

To compute the predicted seismic impedance change from the output of the reservoir simulator, a rock physics model is required, which describes the relationship between changes in reservoir properties such as pressure and fluid saturations, and seismic properties such as velocity and impedance. In Bay Marchand field, the rock physics model is a combination of theoretical and empirical relationships.

If the shear modulus G instead of shear wave velocity V_s is used, the seismic impedance Z can be computed using Eq. 6.26,

$$Z = \sqrt{\rho\left(K + \frac{4}{3}G\right)}\,, \tag{6.26}$$

where, the bulk modulus K of fluid saturated rocks is still computed using the Gassmann equations (Gassmann, 1951), shown in Eq. 6.3 and Eq. 6.4. The bulk modulus of fluid is computed using Eq. 6.7. The bulk modulus of grain, K_{grain}, was assumed to be constant in all calculations. The bulk modulus of dry frame was computed using

$$K_{\text{frame}} = a_0 + a_1 P_{\text{diff}} + a_2\sqrt{P_{\text{diff}}} + \left(b_0 + b_1 P_{\text{diff}} + b_2\sqrt{P_{\text{diff}}}\right)\phi\,, \tag{6.27}$$

where ϕ is porosity, a through b are coefficients and $P_{\text{diff}} = P_{\text{overburden}} - P_{\text{pore}}$, which is the pressure difference between overburden pressure and pore pressure. The unit of pressure is Psia and the unit of K_{frame} is GPa. The shear modulus of dry frame G was computed using Eq. 6.28,

$$G = c_0 + c_1 P_{\text{diff}} + c_2 \sqrt{P_{\text{diff}}} + \left(d_0 + d_1 P_{\text{diff}} + d_2 \sqrt{P_{\text{diff}}} \right) \phi \,, \tag{6.28}$$

where c through d are empirically determined coefficients. Since the influence of fluid saturation on shear modulus is small, the shear modulus of dry frame is also used as shear modulus of fluid saturated rock.

Although the seismic impedance computations in Bay Marchand case study are different from the computations using the Gassmann and Han equations in synthetic cases, the derivatives can be easily calculated using the chain rule, which only changes the source terms in our history matching code. Therefore, only small modifications to the code were required.

Reservoir simulation model analysis

The 23-layer model provided by Chevron-Texaco was up-scaled from a 59-layer fine geological model. It was used as the start point for our history matching. The 23-layer model was unstable in the history matching mode so we decided to up-scale the 23-layer model to a 3-layer model, which was used as the initial model for our history matching. In addition, there are some non-active gridblocks, which were not subjected to adjustments during history matching.

After up-scaling, a variogram analysis of the natural log of the horizontal permeability values was performed. A spherical model was used to fit the experimental variogram in the two principle directions. The porosity field and the cross variograms between the porosity and horizontal permeability were computed using the same spherical variogram function, but with different sills respectively. The correlation coefficient between porosity and horizontal permeability used in the program was assumed to be 0.5 for history matching.

The adjustable parameters in our history matching procedure were porosity and horizontal permeability. The vertical permeability was calculated by multiplying a factor to the horizontal permeability, which is equal to 0.1.

Impedance change noise analysis

As mentioned previously, there are two seismic surveys available in Bay Marchand field. We used a set of time-lapse amplitude data to compute the impedance change data, which were calculated based on the reflection coefficient changes at the interface between the 7100 sand

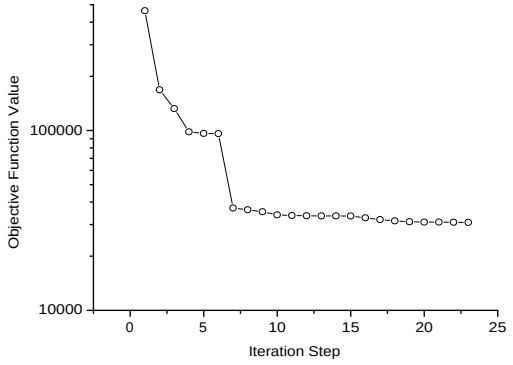


Figure 6.22: Objective function reduction in Bay Marchand field (Production data matching only)

and the overlying shale. The impedance change data calculated from the amplitude change were used as the observed seismic impedance changes in our history matching.

Since neither pressure nor saturation should change in the aquifer zone, the impedance changes should be close to zero. Therefore, any impedance changes in the aquifer can be assumed to be due to non-repeatable noise. Thus, the seismic impedance change data in the aquifer were used to estimate the magnitude and correlation of the noise. An analytical variogram model was chosen to fit the experimental variogram in the two principle directions. The variogram model was used to construct the data noise covariance matrix C_D required in the objective function. The variance of the impedance change noise is at the order of 10^9.

History matching

Although the simulation model provided by Chevron-Texaco is quite good, it was necessary to improve the starting model by first matching the production data. The production-matched model was then used as the start model for history matching seismic impedance change data. Because the primary effect is due to the advancement of water into the field, we focus on the water breakthrough time at all 7 wells. The gas oil ratio (GOR) changes at the 7 wells are less important compared to water cut changes although the GOR data were used as observations in history matching.

In Fig. 6.22, we show the objective function reduction for matching the WOR data and GOR data. It can be seen that the objective function reduces more than 1 order. After production data matching, all 7 wells have improvements in their water cut changes, in terms of closeness to the observed water cut curves. In the upper layers, the biggest changes

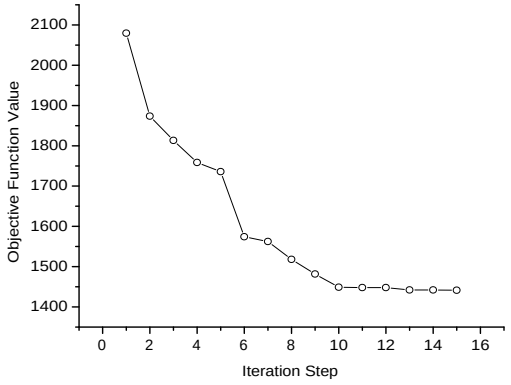


Figure 6.23: Objective function reduction in Bay Marchand field (Seismic impedance change data matching only)

happen around the zones between aquifer and the wells, where some obvious reduction in permeability can be observed, postponing water breakthrough in wells. In bottom layer, the changes are less systematic.

The production data matched model was used as the start point for matching seismic impedance change data. Seismic impedance changes in the aquifer were not included in history matching. The objective function behavior is shown in Fig. 6.23. Although some of the seismic data mismatch reductions resulted in further improvements in the matching of water cut at some wells, the match in the water cut got worse at others. The final level of the seismic objective function is not unreasonable considering the magnitude of the noise in the data and the quality of the initial match.

Chapter 7

HISTORY MATCHING OF FACIES DISTRIBUTIONS

7.1 Background

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 plurigaussian 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 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.

7.2 The Geostatistical Model

We consider a truncated bi-Gaussian field for which two independent Gaussian random fields, y_1 and y_2 are used to generate a facies map.

7.2.1 Generating Thresholds

The choice of the truncation method for the Gaussian variables is important in applying truncated Gaussian simulation in automatic history matching to generate reservoir models satisfying geological requirements. Our intention is to use three or more intersecting lines as thresholds. In this report, I will focus on introducing truncated Gaussian simulation using three 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 kinds of facies can be included in the same plane with appropriate relative percentage. This number of facies is generally enough for geology maps in petroleum reservoir study, but if not, another line could be added. The three lines are thresholds for different rock properties. Given an angle θ and a distance r, a threshold line could be described by the following equation:

$$y = \tan(\theta - \frac{\pi}{2})(x - \frac{r}{\cos\theta}), \tag{7.1}$$

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

$\{Z_{11}, Z_{12}, \ldots, Z_{1N_g}\}$	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 7.1: Continuous geostatistical variables.

Fig. 7.1 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. 7.1(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. 7.1(d)) that the gridblock $(20, 40)$ was assigned facies A. Calculation of the Gaussian fields y_1 and y_2 will be discussed in the later section.

7.2.2 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 on the grid are independent normal deviates with mean 0 and variance 1. These independent deviates must be converted to correlated random fields 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 this problem.

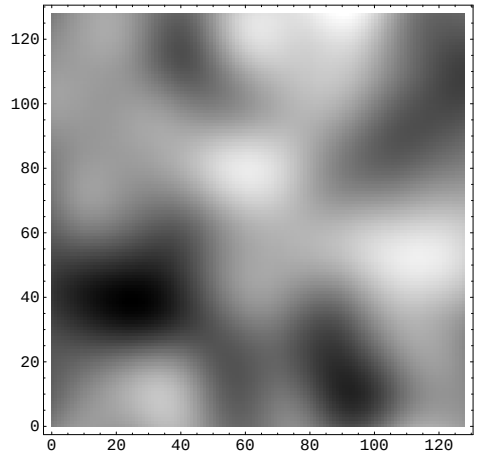
If we partition the truncation map based on truncation lines, then the locations of the lines are also continuous variables. We let the number of lines be denoted by N_l.

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

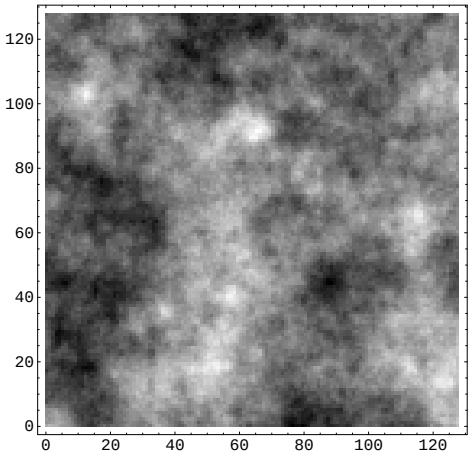
$$P(Z_1) \propto \exp\big(-\frac{1}{2} Z_1^T Z_1\big) \tag{7.2}$$

$$P(Z_2) \propto \exp\big(-\frac{1}{2} Z_2^T Z_2\big). \tag{7.3}$$

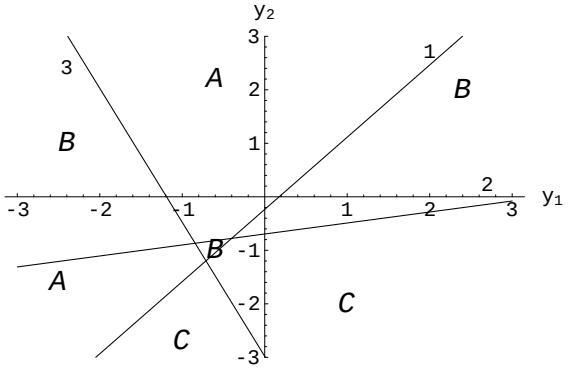
The prior pdfs for the ranges of the two variograms in the principle directions $(a_{11}, a_{12}, a_{21}, a_{22})$ might realistically be modelled as a χ-square distributions with 0 means and fairly large



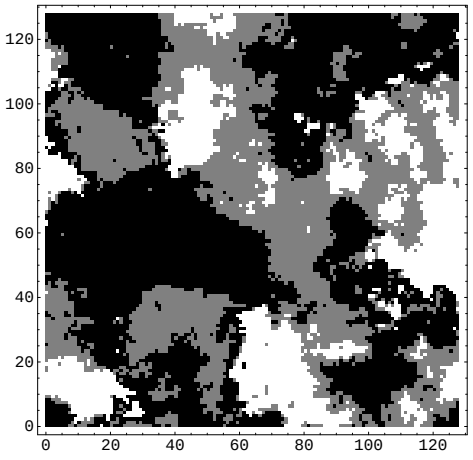
(a) The Gaussian type random field (y1).



(b) The Exponential type random field (y2).



(c) Threshold map.



(d) Calculated facies field map.

Figure 7.1: Simulation of lithofacies distribution in the field by truncation of random Gaussian fields y_1 and y_2 using intersecting line thresholds.

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 θ_{c1} and θ_{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 $\theta_1, \ldots, \theta_{N_l}$ 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 truncated plurigaussian problem is shown in Table 7.1. We can then define m_{cG} to be the vector of continuous variables whose prior distribution is Gaussian and m_{cU} to be the vector of continuous variables whose prior distribution is uniform, or

$$m_{cG} = \begin{bmatrix} Z_1 \\ Z_2 \\ a_{11} \\ a_{12} \\ a_{21} \\ a_{22} \\ r_1 \\ \vdots \\ r_{N_l} \end{bmatrix} \quad \text{and} \quad m_{cU} = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_{N_l} \end{bmatrix}. \tag{7.4}$$

7.2.3 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 7.2.

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

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 7.2: Discontinuous geostatistical variables.

and anisotropy angle. By adding a continuous variable as the weighting factor between the two covariance types, the covariance type becomes differentiable. 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.

7.2.4 Prior Probability Density

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

$$\begin{aligned} P(F) &= P(F|m_{cG}, m_{cU}, m_d)P(m_{cG}, m_{cU}, m_d) \\ &\propto P(m_{cG}, m_{cU}, m_d) \\ &= P(m_{cG})P(m_{cU})P(m_d) \\ &\propto P(m_{cG})P(m_d) \end{aligned} \tag{7.5}$$

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. We have also assumed independence of the variables in the prior distributions, which explains the third line of Eq. 7.5. 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

$$P(m_{cG}) \propto \exp\big(-\frac{1}{2} m_{cG}^T C_M^{-1} m_{cG}\big), \tag{7.6}$$

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

7.2.5 The Posterior Probability Density

Our goal is to generate samples from the posterior distribution, i.e., the distribution of F conditioned to observations, d_{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

$$P(m_{cG}, m_{cU}, m_d|d_{\text{obs}}) \propto P(d_{\text{obs}}|m_{cG}, m_{cU}, m_d)P(m_{cG}, m_{cU}, m_d). \tag{7.7}$$

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

$$P(d_{\text{obs}}|m_{cG}, m_{cU}, m_d) \approx A \exp\Big[-\frac{1}{2}(F - F_{\text{obs}})^T C_D^{-1}(F - F_{\text{obs}})\Big] \tag{7.8}$$

where the data or measurement error covariance matrix C_D simply reflects the possibility of error in the identification or modeling 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 $F - F_{\text{obs}} = 0$) or they are different (in which case $F - F_{\text{obs}} = 1$).

7.3 Minimization

At this stage, we assume homogeneity within facies, i.e. both the permeability and the porosity are constant for the same kind of facies type. However, the property fields are discontinuous at the facies boundaries. Define the difference between the facies type from optimization and the "true" facies type as:

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

where i indicates the ith gridblock. The general objective function for minimization is

$$O(m) = \frac{1}{2}(g(m) - d_{\text{obs}})^T C_D^{-1}(g(m) - d_{\text{obs}}) + \frac{1}{2}(m - m_{\text{pr}})^T C_M^{-1}(m - m_{\text{pr}}). \tag{7.9}$$

When the objective is to minimize the difference between a facies realization and a training image, the objective function takes the form:

$$\begin{aligned} O_F(m) &= \frac{1}{2}(F(m) - F_{\text{obs}})^T C_D^{-1}(F(m) - F_{\text{obs}}) + \frac{1}{2}(m - m_{\text{pr}})^T C_M^{-1}(m - m_{\text{pr}}) \\ &= \frac{1}{2} f(m)^T C_{DF}^{-1} f(m) + \frac{1}{2}(m - m_{\text{pr}})^T C_M^{-1}(m - m_{\text{pr}}), \end{aligned} \tag{7.10}$$

where f is a vector of f_i for $i = 1, \ldots, N_g$, if there were facies observation in each gridblock. Otherwise, when the number of facies observations N_f is less than N_g, the dimensions of f is $N_f \times 1$. C_{DF} is the covariance matrix of the facies observation. Assume the facies in each gridblock are independently observed with an unbiased Gaussian error $N(0, \sigma_F^2)$, the C_{DF} matrix is equivalent to an identity matrix multiplied by σ_F^2. The variance of the error in facies observation approaches to zero, therefore the weighting of the model mismatch is much smaller than the data mismatch, and can be ignored from the objective function. We would like the objective function to be as small as possible since in that case we should have a match between the observations and the model realization. The problem is that this function is not differentiable so we cannot use gradient-based methods to find a minimum. One solution is to redefine the function f so that it is differentiable (only for the purpose of computing the minimum). We do this by introducing an artificial transition region between facies.

7.4 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 θs and rs mostly controls the optimization process. So here I take the first step towards a complete optimization of all parameters – fix the two Gaussian random fields as the "true" and leave the threshold parameters as the only set of variables to be optimized. Another purpose of 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 LBFGS.

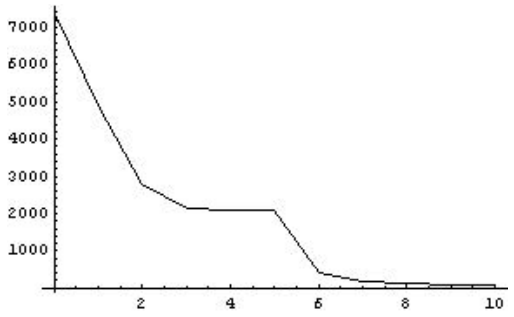


Figure 7.2: The reduction of the objective function with LM iterations.

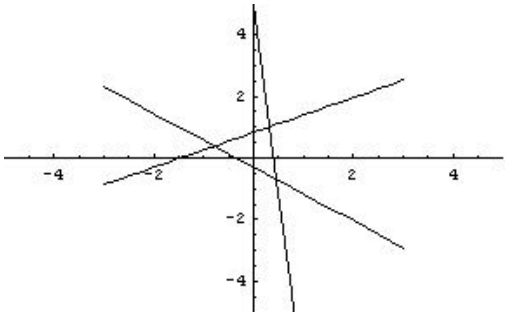
7.4.1 Case 1: using true Z_1 and Z_2

The optimization problem was applied on a fine grid field of 128×128. Prior experience with this research has proven that coarse grid such as 10×10 would make the optimization rather tough. 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, which was a pure guess and 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.

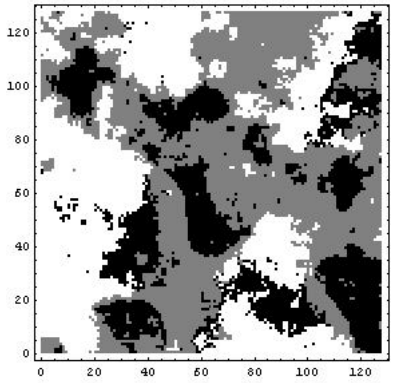
The reduction of the objective function with LM iterations is shown in Fig. 7.2. In Fig. 7.3, the threshold model after the tenth step and its corresponding facies distribution were compared with the truth case. The data mismatch of 68 is small enough that the difference between the calculated and the truth could be hardly recognized. Three set of threshold lines, the random prior, the posterior and the true are put together in Fig. 7.4. 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.

7.4.2 Case 2: unknown true Z_1 and Z_2

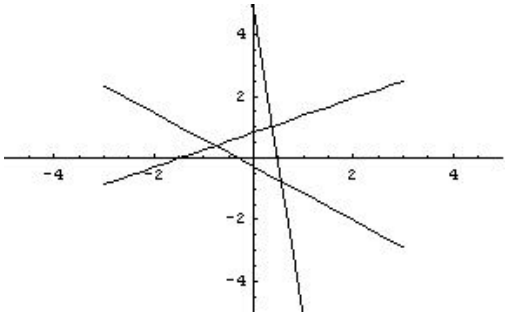
Given a map of lithofacies distribution, either from a geologist's experience, or from an outcrop, a set of geosatistical parameters which would generate a lithofacies realization with similar features and appearance can be estimated by the LBFGS optimization iterations. Unfortunately, a single estimate of the parameters is unlikely to provide a reasonable char-



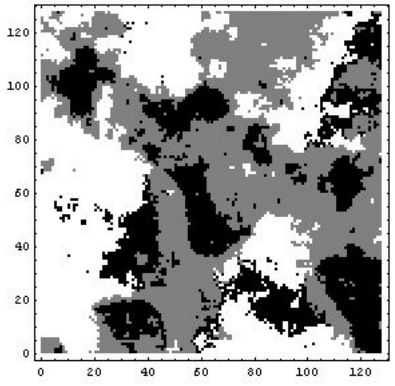
(a) The true threshold map.



(b) The true facies field.

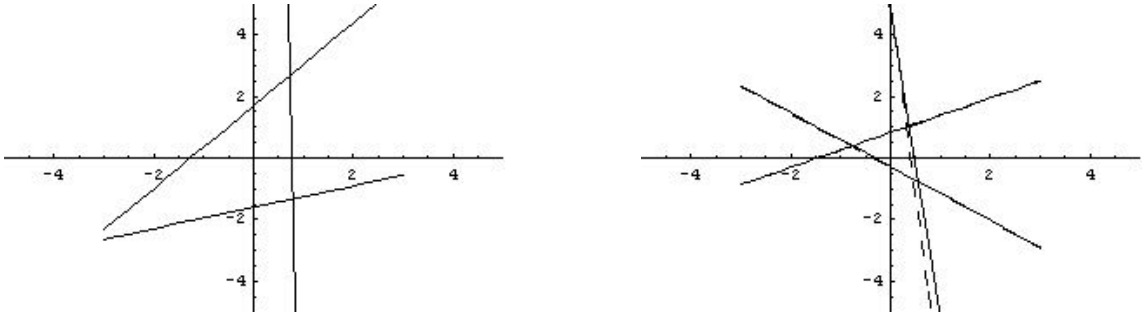


(c) Calculated threshold map.



(d) Calculated facies field.

Figure 7.3: 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 7.4: Comparison of the prior, the posterior and the true threshold map. Solid lines are stochastic model estimations and dashed lines are the truth.

acterization 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), we will characterize the pdf of the threshold line parameters using the method of randomized maximum likelihood. In this method, we generate unconditional realizations of model parameters Z_1, Z_2, R, Θ, then adjust the model parameters to minimize the difference between the true (or training) facies map and the predicted facies map from the current model. If we do this a large number of times, we can expect to develop an empirical estimate of the pdf for threshold line parameters conditional to the training image.

The objective function for this problem contains both the squared data mismatch term and the squared model mismatch term, as shown in Eq. 7.11. The model mismatch part was used to prevent the 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,

$$O(m_k) = \frac{1}{2}(m_k - m_{\rm pr})^T C_M^{-1}(m_k - m_{\rm pr}) + \frac{1}{2}(F(m_k) - F_{\rm obs})^T C_D^{-1}(F(m_k) - F_{\rm obs}). \quad (7.11)$$

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 $C_{\rm D} = 1$ and seemed to obtain reasonable answers.

As an example synthetic problem, the reservoir model is two dimensional and heterogeneous, with the size 300 ft $\times$300 ft. It is discretized into 128 $\times$ 128 gridblocks. Three

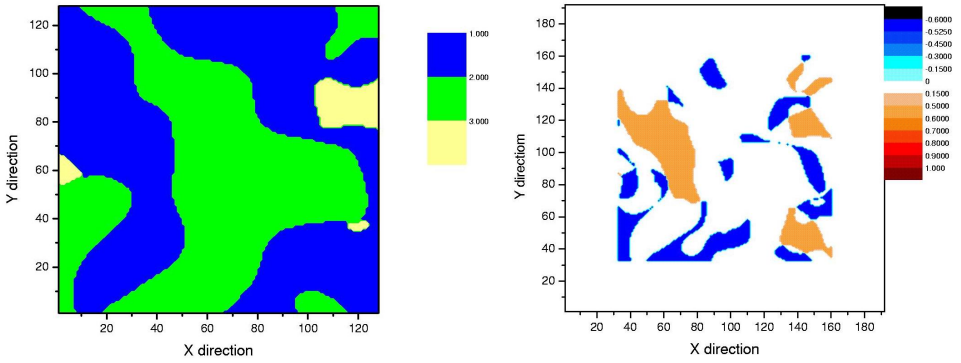


Figure 7.5: The initial facies map (left) and the gradient of the objective function with respect to the field evaluated at the initial map (right).

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 192×192. 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. Both Y_1 and Y_2 have isotropic Gaussian type covariance with range of 90 ft. Fig. 7.5 shows the facies distribution from the initial guess of model parameters. The initial facies mismatch is 12525, i.e. 76% of the gridblocks have wrong facies types compared to the training image [Fig. 7.8 (left)].

This problem is far too large for Levenberg-Marquardt, so the limited memory version of the BFGS (LBFGS) algorithm (Nocedal, 1980) was used for the minimization of the objective function because it requires storage of only a small set of vectors instead of the whole inverse Hessian matrix. The inverse Hessian is iteratively updated based on computations of the gradient, for which we used an adjoint method with automatic code generation (Corliss et al., 2001).

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 following.

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 $i = bound - 1, \ldots, 0$

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

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

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

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

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

where

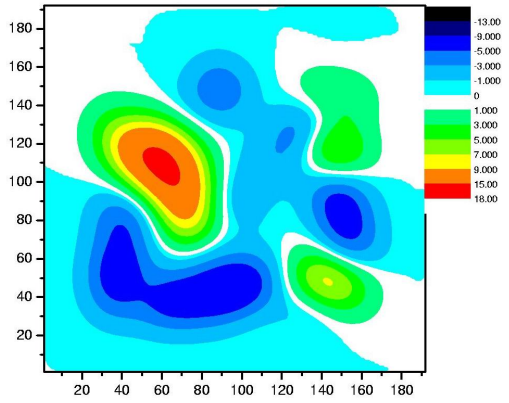
$$\begin{aligned} s_j &= m_{j+1} - m_j \\ y_j &= g_{j+1} - g_j \\ \rho_j &= 1/(y_j^T s_j) \end{aligned}$$

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

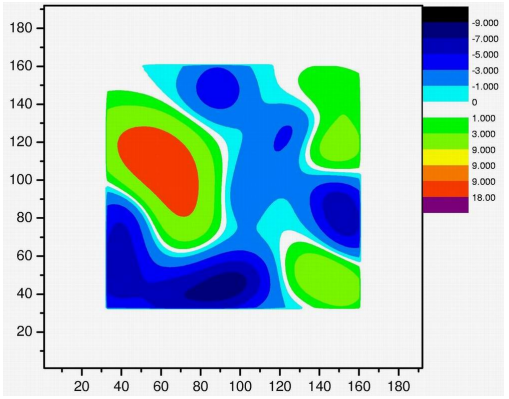
Fig. 7.5 (right) shows the gradient of the objective function with respect to Y_1. The gradient of the objective function to Y_1 clearly reflects the regions of high sensitivity. Comparing the right and left sub-figures in Fig. 7.5, we may conclude that the mismatch function is most sensitive to changes 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. 7.6(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. The gradient of 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. 7.5(right) was convolved with the Gaussian template and the result is plotted in Fig. 7.6(b). As expected, we found that Fig. 7.6(a) and Fig. 7.6(b) have very similar values. The difference of the two plots is in the frame areas, which are largely due to the 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 7.6: 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, we found that 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, we performed a simple investigation of the behavior of the objective function in the descent direction for the first iteration. The reduction of the objective function in first search direction is shown in Fig. 7.4.2. 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.

Finally the training image is shown together with the final facies map in Fig. 7.8. The facies in 95.6% of the gridblocks in the final result matched the facies in the training image.

We repeated the minimization procedure 200 times; each time we 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

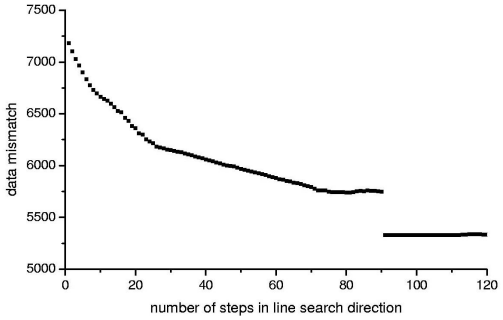


Figure 7.7: Shape of initial objective function in first search direction.

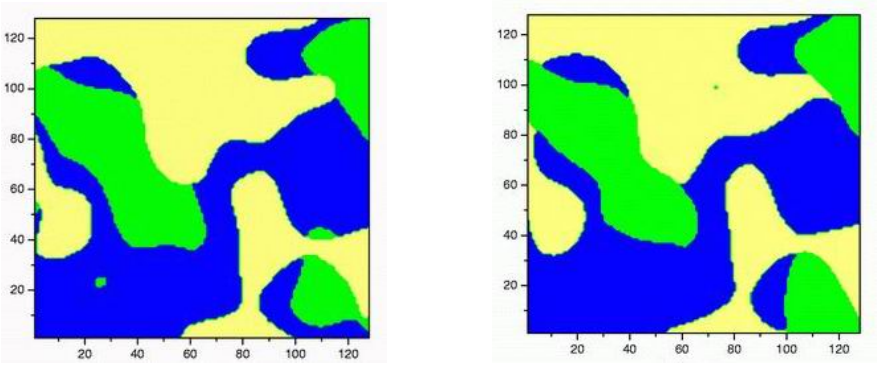


Figure 7.8: Comparison of the facies map generated from the minimization of the objective function (right) with the training image (left).

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, 2003). In approximately half of the cases the resulting value of the objective function seemed satisfactory (less than 2000). Fig. 7.9 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 (θ_1) to be close to π, and this is seen in Fig. 7.9. From the two plots, it is not apparent that the estimate of the posterior distribution for θ_1 or θ_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, we would have expected the orientations to cluster along lines in cross-plots of variables. Cross-plots of

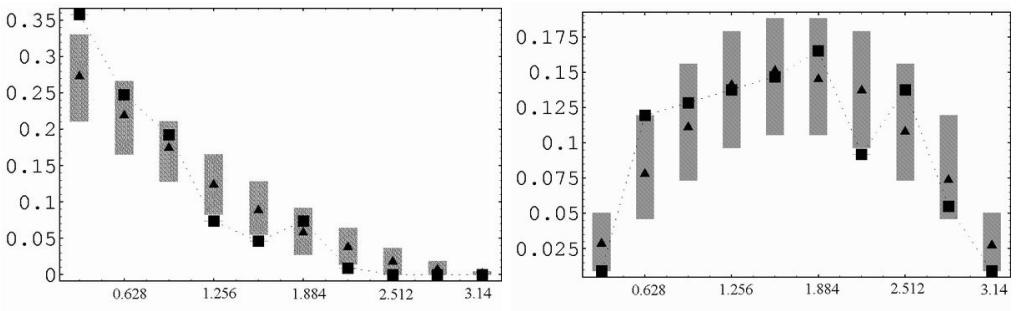


Figure 7.9: Comparison of the estimated pdf (squares) for θ_1 (left) and θ_2 (right) to the prior pdf (triangles). The width of the gray bars indicates the variability due to limited sample size of 109 ordered sets of 3 orientations from a uniform distribution. 10% are higher and 10% are lower.

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 relationships among the threshold line parameters, but they are not obvious from the conditional realizations. 2500 conditional threshold models are generated in addition to the 109 models, and the cross-plots are presented in Fig. 7.10. 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.

7.5 History Matching to Production Data

Practically the hard data of facies observations in a formation layer are only available at well bores with coring operations. The training image from geologists provides expected facies distribution pattern and features based on the sedimentary environment and the formation outcrops, but does not carry any local accuracy. In the previous study, PDFs of the geostatistical parameters, such as the ones deciding truncation schemes and the ones deciding the covariance of the Gaussian fields, have been estimated from a given training image. The geostatistical parameters are then proposed from the PDFs in a Bayesian scheme for simulation of facies conditional to logs and/or production data.

When the PDFs of the geostatistical parameters are known, realizations of the facies distribution map can be generated by minimizing the objective function with respect to the

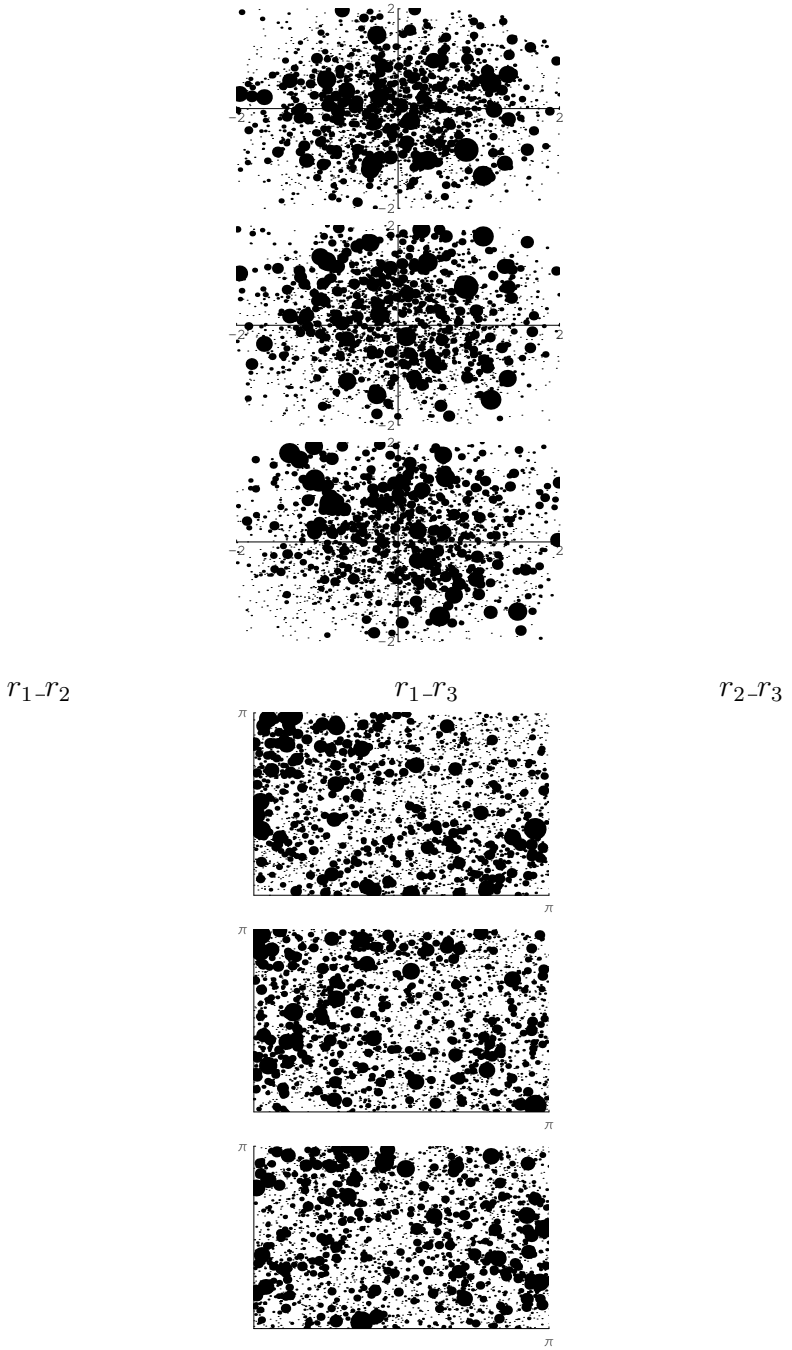


Figure 7.10: Cross-plots of the conditional threshold line models.

random field Z_1 and Z_2. The objective function in this case is:

$$\begin{aligned} O(m) &= \frac{1}{2}[P(m)-P_{\rm obs}]^T C_P^{-1}[P(m)-P_{\rm obs}] + \frac{1}{2}[F(m)-F_{\rm obs}]^T C_F^{-1}[F(m)-F_{\rm obs}] \\ &\quad + \frac{1}{2}(m-m_{\rm pr})^T C_M^{-1}(m-m_{\rm pr}) \\ &= \frac{1}{2}[d(m)-d_{\rm obs}]^T C_D^{-1}[d(m)-d_{\rm obs}] + \frac{1}{2}(m-m_{\rm pr})^T C_M^{-1}(m-m_{\rm pr}), \end{aligned} \tag{7.12}$$

where the prior model $m_{\rm pr} = \{Z_{1,pr}, Z_{2,pr}\}$ is composed of random deviates with mean 0 and variance 1, so the prior model covariance matrix C_M is an identity matrix. $P_{\rm obs}$ is the observed production data, such as the bottom whole pressures, the WORs, the GORs, and the production rates. $P(m)$ is the simulated production data. $F_{\rm obs}$ 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 hard data dominates the matching before all the facies observations are matched, so that the facies type at well locations are forced to honor the facies observations. The production data and the hard data are combined and denoted as $d_{\rm obs}$.

7.5.1 Gradient derivation

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

$$\begin{aligned} g &= \nabla_m O(m) \\ &= G_P(m)^T C_P^{-1}[P(m)-P_{\rm obs}] + G_F(m)^T C_F^{-1}[F(m)-F_{\rm obs}] + C_M^{-1}(m-m_{\rm pr}) \\ &= g_P + g_F + g_m \\ &= \nabla_m O_d(m) + \nabla_m O_m(m). \end{aligned} \tag{7.13}$$

When the geostatistical model has been decided, generating a facies realization that matches the production data and logging data is to optimize 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:

$$\begin{aligned}\nabla_{Z_1} O_d(m) &= \nabla_{Z_1}[O_P(m) + O_F(m)] \\ &= L_1 \cdot [\nabla_{Y_1} K(m) \cdot \nabla_K O_P(m) + \nabla_{Y_1}\phi(m) \cdot \nabla_\phi O_P(m) + \nabla_{Y_1} O_F(m)] \quad (7.14)\end{aligned}$$

$$\begin{aligned}\nabla_{Z_2} O_d(m) &= \nabla_{Z_2}[O_P(m) + O_F(m)] \\ &= L_2 \cdot [\nabla_{Y_2} K(m) \cdot \nabla_K O_P(m) + \nabla_{Y_2}\phi(m) \cdot \nabla_\phi O_P(m) \\ &\quad + \nabla_{Y_2} O_F(m)], \quad (7.15)\end{aligned}$$

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 ith gridblock, $Y_{1,i}$, is a vector of all zeros but the ith element. Eq. 7.16 shows the gradient of K about $Y_{1,1}$:

$$\nabla_{Y_{1,1}} K = \begin{bmatrix} \frac{\partial K_1}{\partial Y_{1,1}} \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad (7.16)$$

which indicates that a perturbation to the Gaussian random field Y_1 at the gridblock 1 only impacts the permeability at the gridblock 1.

So 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:

$$\frac{\partial K_i(m)}{\partial Y_{1,i}} = \frac{dK_i}{dd_l}\frac{\partial d_l}{\partial Y_{1,i}},$$
$$\frac{\partial \phi_i(m)}{\partial Y_{1,i}} = \frac{d\phi_i}{dd_l}\frac{\partial d_l}{\partial Y_{1,i}},$$
$$\frac{\partial K_i(m)}{\partial Y_{2,i}} = \frac{dK_i}{dd_l}\frac{\partial d_l}{\partial Y_{2,i}},$$
$$\frac{\partial \phi_i(m)}{\partial Y_{2,i}} = \frac{d\phi_i}{dd_l}\frac{\partial d_l}{\partial Y_{2,i}},$$

for $i = 1, \ldots, N_g$. d_l is the distance of $(Y_{1,i}, Y_{2,i})$ to the closest threshold line.

The property fields K and ϕ are regarded 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 corresponds to $(Y_{1,i}, Y_{2,i})$ is correct. Therefore the transition zone of the permeability and porosity values is made in both 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:

$$K(d_l) = \begin{cases} \frac{K_1+K_2}{2} + Sign(d_l)\frac{K_2-K_1}{2}\sqrt{\frac{|d_l|}{\epsilon}} & \text{for } |d_l| \le \epsilon \\ 0 & \text{for } |d_l| > \epsilon, \end{cases} \tag{7.17}$$

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 ϵ, the permeability along the direction perpendicular to a threshold line is:

$$K(d_l) = \begin{cases} K_1 - \frac{1}{2}(K_1 - K_2)(1 - \frac{|d_l|}{\epsilon}) & \text{for } |d_l| \le \epsilon \\ 0 & \text{for } |d_l| > \epsilon, \end{cases} \tag{7.18}$$

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. ϵ is the absolute distance to the threshold line. In Fig. 7.11, the permeability at point B can be computed by Eq. 7.18, and K_A should be computed by:

$$K_A = K_2 - \frac{1}{2}(K_2 - K_1)(1 - \frac{|d_{l,a}|}{\epsilon}), \tag{7.19}$$

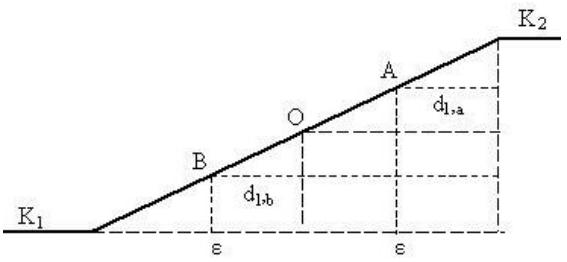


Figure 7.11: The gradient of the permeability is derived from this linear interpolation model. K_1 and K_2 are permeability values assigned to two adjacent regions in the threshold map. Point O is the cross-section with the threshold line, which is also the middle point of the transition zone in this 1-D plot.

as A is on the same side with K_2.

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

$$\frac{dK_i}{d|d_l|} = \begin{cases} \frac{K_1-K_2}{2\epsilon} & \text{for } |d_l| \le \epsilon \\ 0 & \text{for } |d_l| > \epsilon. \end{cases} \tag{7.20}$$

Substituting Eq. 7.20 into Eq. 7.5.1, the gradient of the permeability at the ith grid with respect to Y_1 at the ith grid is:

$$\begin{aligned} \frac{\partial K_i(m)}{\partial Y_{1,i}} &= \frac{dK_i}{d|d_l|}\frac{d|d_l|}{dd_l}\frac{\partial d_l}{\partial Y_{1,i}} \\ &= Sign(d_l)\frac{K_1-K_2}{2\epsilon}\sin(\theta_l - \frac{\pi}{2}). \end{aligned} \tag{7.21}$$

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

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

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

The third term of $\nabla_{Z_1} O_d(m)$ in the Eq. 7.14 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. Because the logical process of deciding the values of $\frac{df_i}{dY_{1,i}}$ is beyond the description capacity of equations, the following pseudo code is provided.

DO $l = 1$, N_l

$$\begin{aligned} d(l) &= \frac{\tan(\theta_l - \frac{\pi}{2})Y_{1,i} - Y_{2,i} - \tan(\theta_l - \frac{\pi}{2})\frac{r_l}{\cos\theta_l}}{\sqrt{1 + \tan^2(\theta_l - \frac{\pi}{2})}} \\ &= \sin(\theta_l - \frac{\pi}{2})Y_{1,i} - \cos(\theta_l - \frac{\pi}{2})Y_{2,i} + r_l \end{aligned}$$

END DO.
$d_l = \min(d)$
IF $F_i = F_{\text{obs,i}}$ THEN
$\frac{df_i}{dY_{1,i}} = 0$
ELSE
IF $|d_l| > \epsilon$ THEN
$\frac{df_i}{dY_{1,i}} = 0$
ELSE
IF $(F_{ab,i} = F_{\text{obs,i}})$ 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. $F_{ab,i} = F_{\text{obs,i}}$, 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 $\nabla_{Y_1} O_F(m) = G_{F,Y_1}(m)^T C_F^{-1} f$ is:

$$\frac{\partial O_F(m)}{\partial Y_{1,i}} = \frac{d_l}{|d_l|}\frac{1}{2\sigma_F^2\epsilon}\sin(\theta_l - \frac{\pi}{2}). \tag{7.22}$$

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:

$$\frac{\partial O_F(m)}{\partial Y_{2,i}} = \frac{d_l}{|d_l|}\frac{1}{2\sigma_F^2\epsilon}\sin(\theta_l - \frac{\pi}{2}). \tag{7.23}$$

7.5.2 Generate Initial Model

The initial model for history matching of production data needs to honor the facies observations, otherwise 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 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 facies mismatch, repeat step 3. If not, output the random fields Z_1, Z_2 as a set of initial model.

Twelve initial models are generated and the initial facies maps are shown in Fig. 7.12.

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. 7.13 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

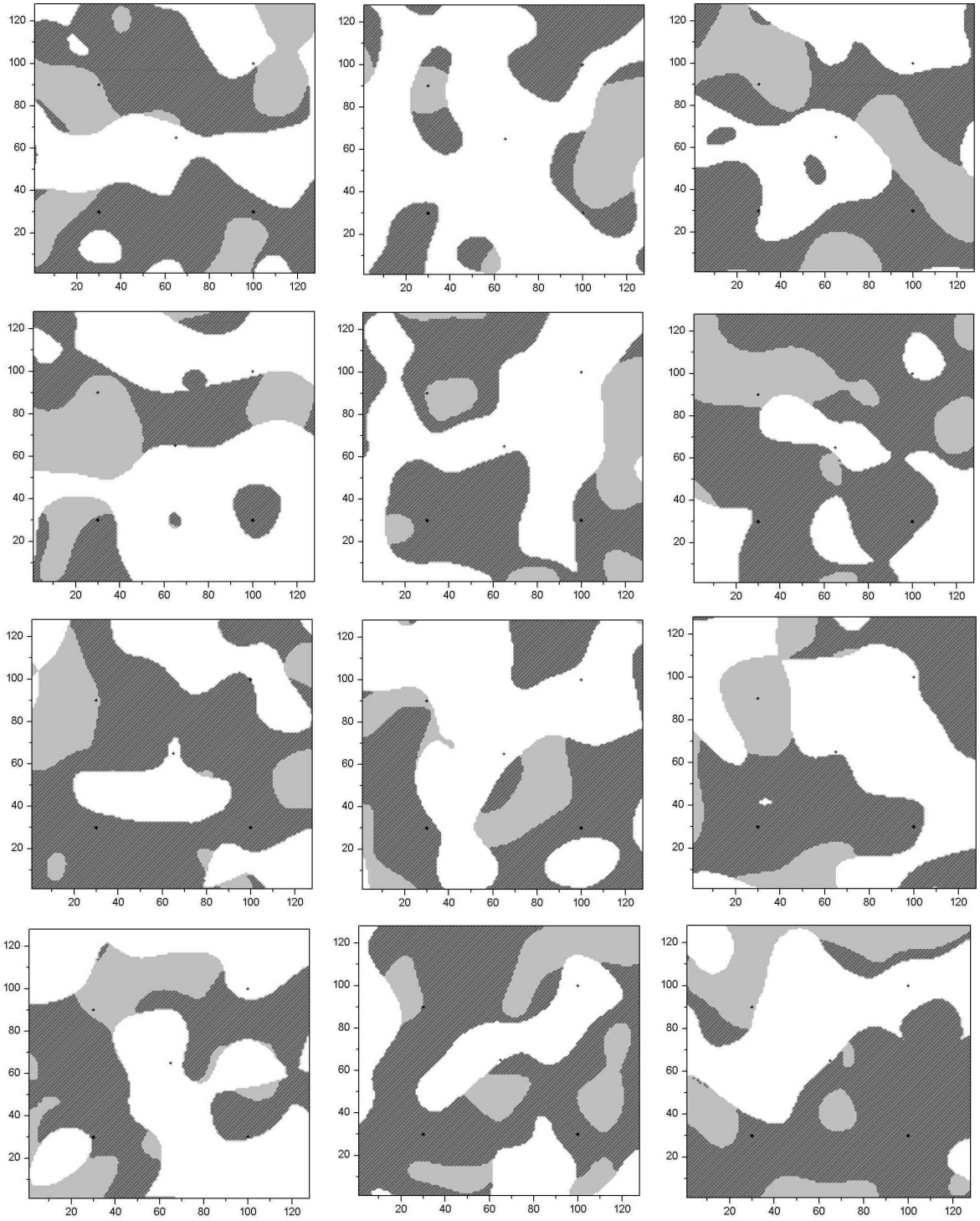


Figure 7.12: Twelve initial facies maps that honors the facies observations. The initial models are then used for matching the production data, while preserving the correct facies at observation locations.

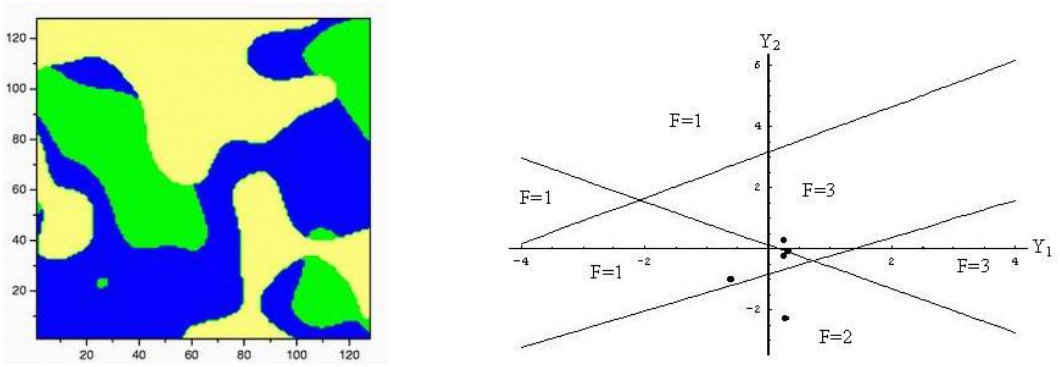


Figure 7.13: The true facies map and the true threshold map with the Gaussian variables (Y_1, Y_2) at each facies observation of the true facies map.

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 hard to be corrected.

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 $N(\mathbf{0}, I)$.

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.

7.5.3 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 the 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. 7.14 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 the five facies observations are at well locations, the facies mismatch caused great production data mismatch from the wells. Therefore the objective function jumped by a factor of 20.

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. To have a close look 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. 7.14. Comparing the two figures with the transition width of 0.2 and 1.0 respectively, the reduction of the objective function 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:

$$q(\alpha) = a\alpha^2 + b\alpha + c, \tag{7.24}$$

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

$$\begin{aligned} a &= \frac{O(m_k + \hat{\alpha}_k d_k) - \nabla O(m_k)\hat{\alpha}_k - O(m_k)}{\hat{\alpha}_k^2}, \\ b &= \nabla O(m_k), \\ c &= O(m_k). \end{aligned} \tag{7.25}$$

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

$$\alpha_k = -\frac{\nabla O(m_k)\hat{\alpha}_k^2}{2[O(\hat{\alpha}_k d_k) - \nabla O(m_k)\hat{\alpha}_k - O(m_k)]}, \tag{7.26}$$

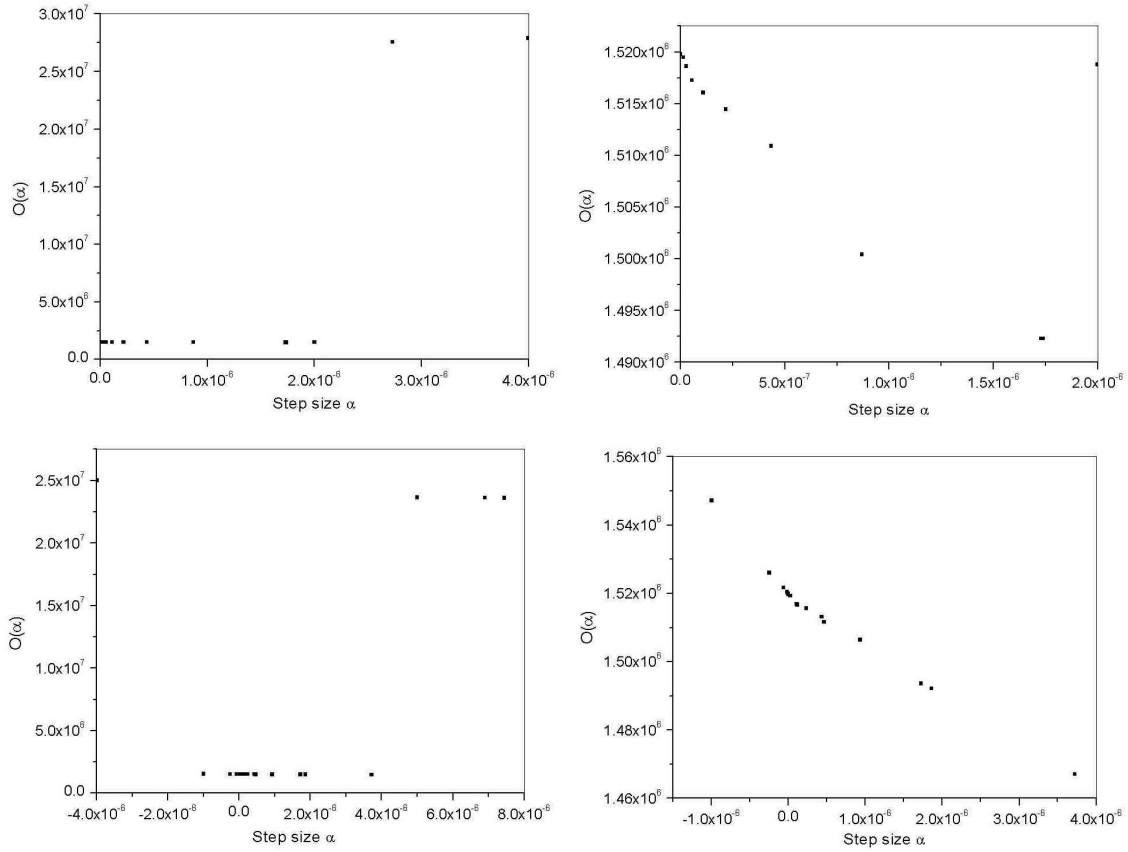


Figure 7.14: All the four figures are the objective function along the first search direction. The transition zone width in the first row is 0.2, in the second row is 1.0. The figures on the right column are amplifications of the flat region in the figures on the left column.

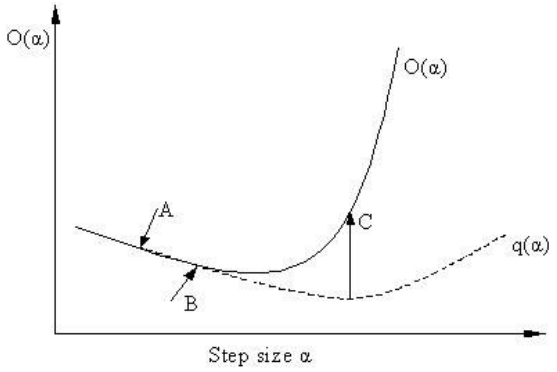


Figure 7.15: A schematic plot of the quadratic fit to an objective function curve with the typical shape for this minimization problem. The quadratic fit gives a higher objective function value than that from the Newton-Raphson iteration.

Both curves of the objective function with different transition zone width are nearly linear before the abrupt jump. Fig. 7.15 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 point 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 is provided to give a better illustration of the structure of the code developed along this study.

7.6 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 interpolations. The other type is the facies observation data from cores, which is regarded as hard data. The process of optimizing the model to match the production data is pretty much similar with common automatic history matching problems. The hard data in our problem, however, is not the same type as the model parameters. The hard data are the

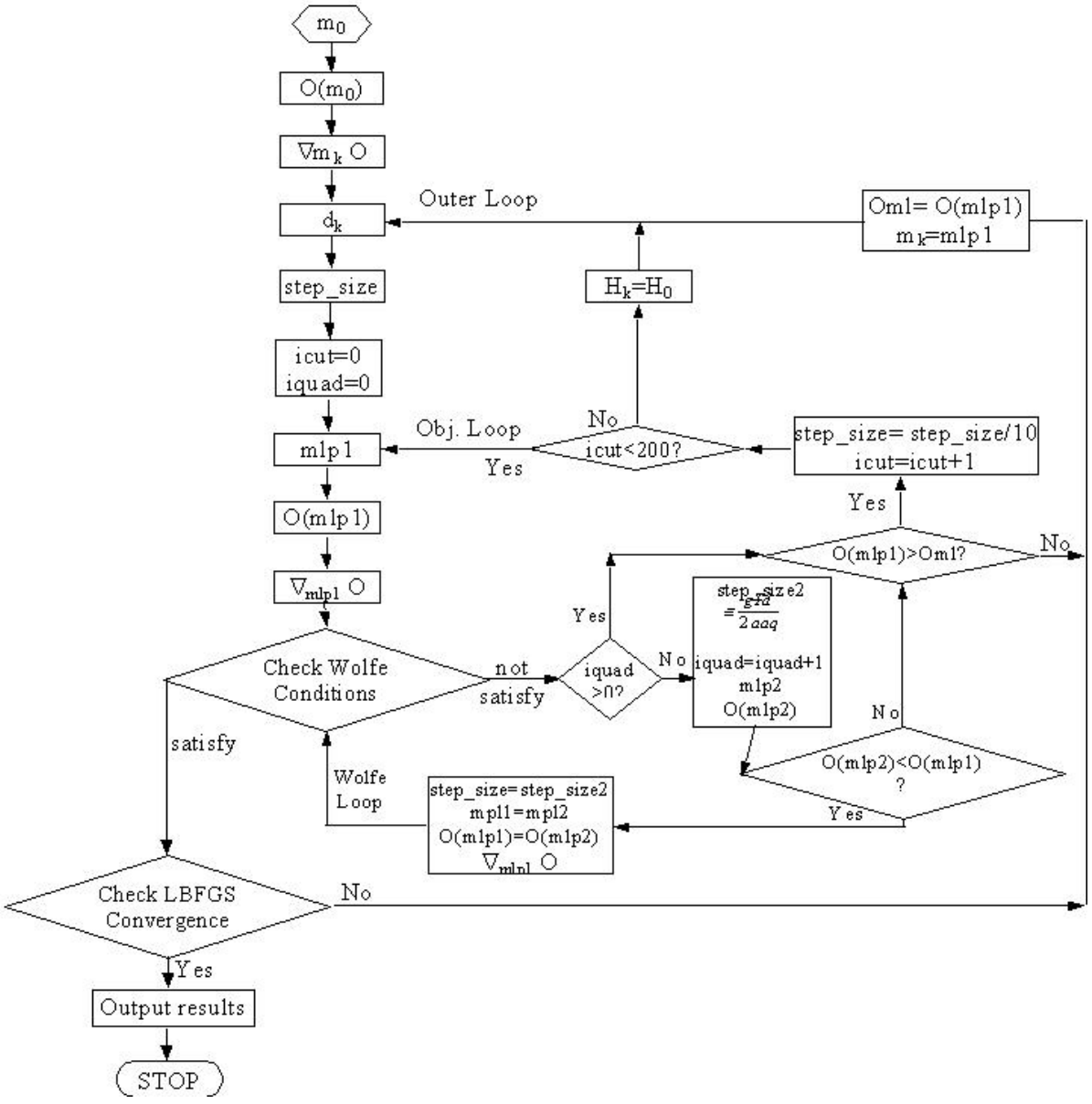


Figure 7.16: Flow chart for the automatic history matching process.

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 type at observation locations in the process of optimization.

In line search for the optimized step size, we take one Newton-Raphson iteration as:

$$\alpha_1 = -\frac{(\nabla O(m_k))^T d_k}{d_k^T H(m_k) d_k}, \tag{7.27}$$

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

$$\begin{aligned} Gd_k &= \|d_k\| \frac{dO(m_k)}{d\alpha} \\ &\approx \|d_k\| \frac{O(m_k + \epsilon d_k) - O(m_k)}{\epsilon \|d_k\|} \\ &= \frac{O(m_k + \epsilon d_k) - O(m_k)}{\epsilon}, \end{aligned} \tag{7.28}$$

where ϵ is a small constant chosen based on the infinite norm of d_k:

$$\epsilon = \frac{10^{-3}}{\|d_k\|_\infty}. \tag{7.29}$$

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 we first check the facies mismatch for the model $m_k + \epsilon d_k$, if it is non-zero, then the ϵ is cut back: $\epsilon = \epsilon/10$, 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 we check the facies mismatch for 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 efficient than running the reservoir simulation.

7.7 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 field width. Three

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 7.3: 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 7.4: Production conditions for all five wells in the field.

lithofacies, dolomite shale and sand, distribute in the field and have very distinct properties. Table 7.3 lists the permeability and porosity values for each of the facies types. Table 7.4 describes the production conditions for all five wells in this field. In Fig. 7.17, 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. The draw-down test at the four producing wells lasted three days and 20 bottom hole pressure data were collected from each well. Besides the bottom hole pressure data, we also assume the facies at well locations are observed.

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

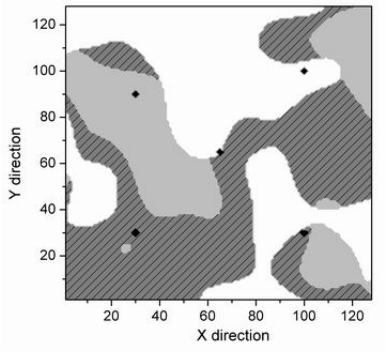


Figure 7.17: The "true" facies map with well locations.

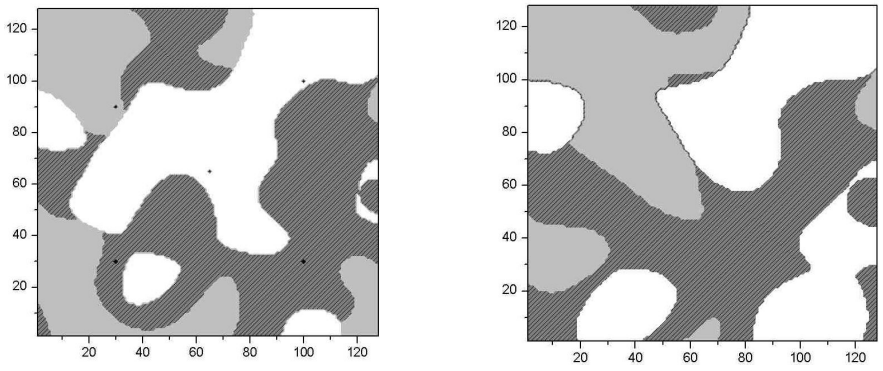


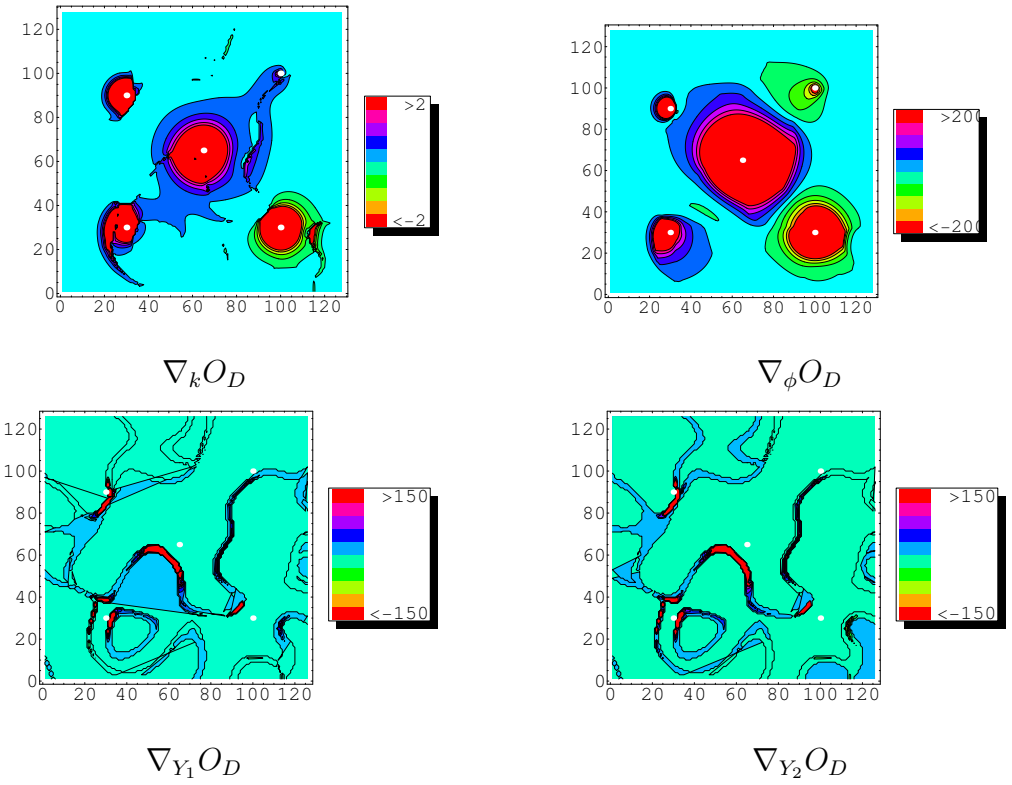
Figure 7.18: The initial facies map and the final facies map after convergence. The objective function reduced to 1% of the original objective function at the final model.

gridlocks because the moving average method required a margin of Z1 and Z2 fields outside of the facies grid. The width of the margins is determined by the range of the correlation of Y1 and Y2. 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) (LBFGS) is a quasi-Newton method used in computing the search direction. LBFGS 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, 2002a).

An initial model has been generated matching the regions of facies observations. The initial facies map is shown on the left of Fig. 7.18. After 13 LBFGS iterations, the objective function reduced to 1% of the initial value. The facies map for the converged model is shown on the right of Fig. 7.18. The production data is from the true case shown in Fig. 7.8.

The intermediate results for computing the gradient of the squared data mismatch O_d with respect to model parameters Z_1 and Z_2 are shown in Fig. 7.19. 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 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. 7.20. The production data for well 2 has the best matching, mostly because the gradient is large at that 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 gradient around well 4 is relatively small than those around other wells, thus the



$\nabla_k O_D$ $\nabla_\phi O_D$

$\nabla_{Y_1} O_D$ $\nabla_{Y_2} O_D$

Figure 7.19: 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.

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 can be improved around well 4 towards reductions of data mismatch of well 4.

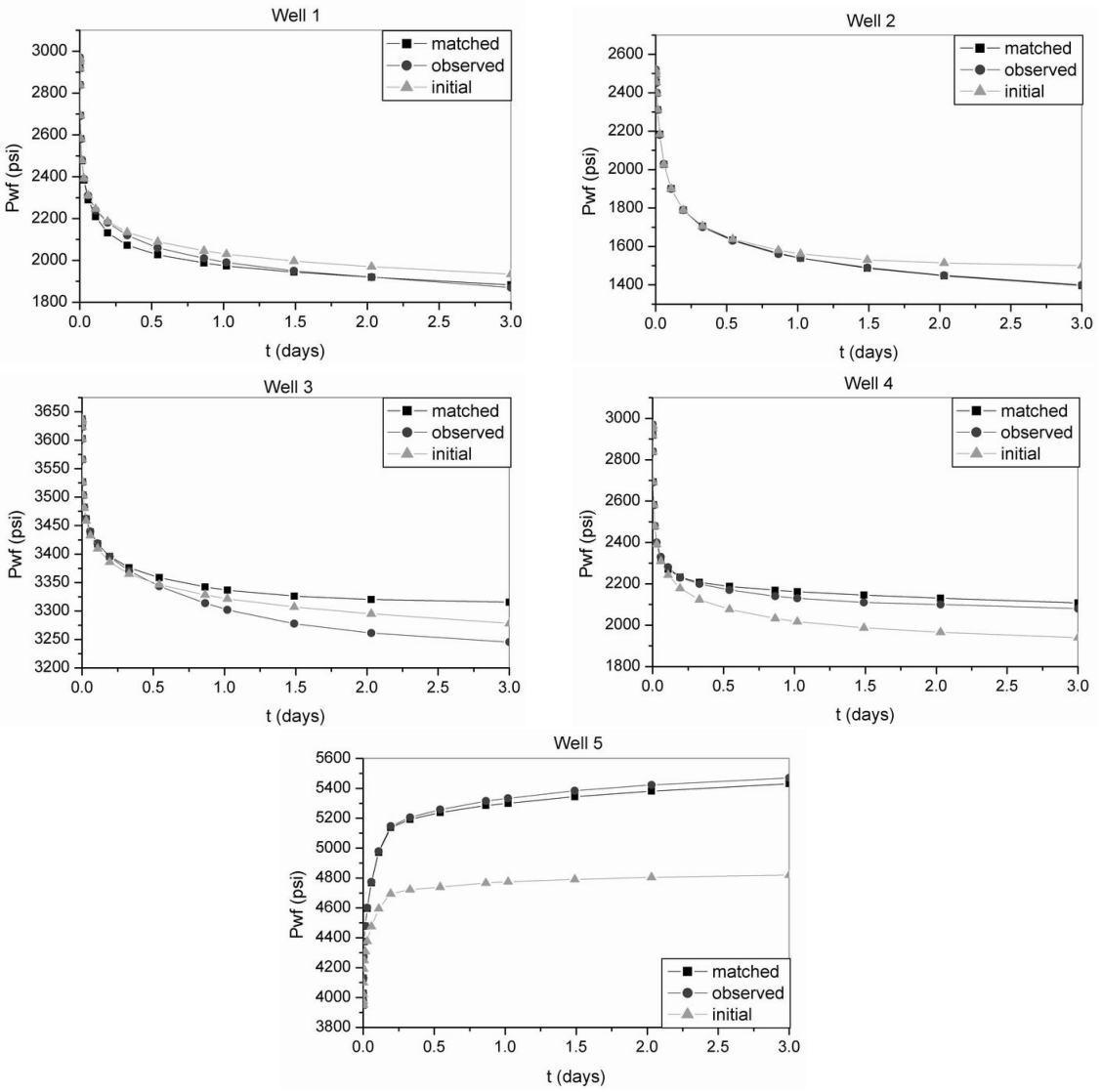


Figure 7.20: Comparison of the production data from the initial and the final model with the observation data.

Chapter 8

EXPERIMENTAL

Experimental work is not applicable to the research tasks and goals of this project. Consequently, no experimental work has been done.

Chapter 9

TECHNOLOGY TRANSFER

Our state of knowledge on the integration of seismic and production data has been considerably enhanced by the research results obtained during this project. Many of these achievements have been described in technical papers. The list of papers given below are based entirely or partially on results obtained from this research project.

9.1 Technical Papers

- Y. Abacioglu, D. S. Oliver, and A. C. Reynolds: "Efficient history-matching using subspace vectors," *Computational Geosciences*, 5(2)(2001), 151–172.
- F. Zhang and A. C. Reynolds: "Optimization algorithms for automatic history matching of production data," Proceedings of 8th European Conference on the Mathematics of Oil Recovery (2002), 10 pages.
- Ruijian Li, A. C. Reynolds, and D. S. Oliver: "Sensitivity coefficients for three-phase flow history matching," *J. Canadian Pet. Tech.*, 42(4) (2003) 70–77.
- F. Zhang , J. A. Skjervheim, A. C. Reynolds and D. S. Oliver: "Automatic history matching in a Bayesian framework: example applications," SPE-84461, Proceedings of the 2003 SPE Annual Technical Conference and Exhibition, 13 pages.
- N. Liu and D. S. Oliver: "Evaluation of uncertainty assessment methods, *Soc. Petrol. Eng. J.*, 8(2), 188–195, 2003.
- R. Li, A. C. Reynolds, and D. S. Oliver: "History matching of three-phase flow production data, *Soc. Petrol. Eng. J.*, 8(4), 328–340, 2003.

- A. C. Reynolds, Ruijian Li and D. S. Oliver, "Simultaneous estimation of absolute and relative permeability by automatic history matching of three-phase flow production data," *J. Canadian Pet. Tech.*, 43(3) (2004) 37–46.

- N. Liu and D. S. Oliver: "Automatic History Matching of Geologic Facies, *Soc. Petrol. Eng. J.*, 8(2), 188–195, 2004.

- N. Liu and D. S. Oliver: "Experimental assessment of gradual deformation method, *Mathematical Geology*, 36(1), 65–77, 2004.

- G. Gao and A. C. Reynolds: "An improved implementation of the LBFGS algorithm for automatic history matching," paper SPE-90058, Proceedings of the 2004 SPE Annual Technical Conference and Exhibition, 18 pages.

- G. Gao, M. Zafari, and A. C. Reynolds: "The Tengiz field history matching problem revisited," paper SPE-90896, Proceedings of the 2004 SPE Annual Technical Conference and Exhibition, 15 pages.

- G. Gao, M. Zafari, and A. C. Reynolds: "Quantifying uncertainty for the PUNQ-S3 problem in a Bayesian setting with RML and EnKF, SPE-93324," accepted for presentation at the 2005 SPE Reservoir Simulation Symposium.

- Yannong Dong and Dean S. Oliver: "Quantitative Use of 4D Seismic Data for Reservoir Description," *SPE Journal*, 10(1), March 2005.

9.2 Ph.D. Dissertations

Several graduate students have completed or are expected to complete their Ph.D. degrees based on research done under the auspices of this DOE project. Those who have completed together with their date of completion and current company affiliation follow:

- Yafes Abacioglu, "The Use of Subspace Methods for Efficient Conditioning of Reservoir Models to Production Data," Ph.D. dissertation, University of Tulsa (2001) BP.

- Ruijian Li, "Conditioning Geostatistical Models to Three-Dimensional Three-Phase Flow Production Data by Automatic History Matching," Ph.D. dissertation, University of Tulsa (2001) Shell International Exploration and Production Inc.

- Fengjun Zhang, "Automatic History Matching of Production Data for Large Scale Problems," Ph.D. dissertation, University of Tulsa (2002) ChevronTexaco.

The Ph.D. research of Ning Liu (University of Oklahoma), Yannong Dong (University of Oklahoma) and Guohua Gao (University of Tulsa) has also been partially supported by this project. All three are expected to receive their Ph.D. degrees in 2005.

Chapter 10

CONCLUSIONS

The history matching algorithms and software developed in this project were shown to be efficient at providing realistic maps of reservoir properties for simulation models. The reservoir properties honor both static and dynamic data, while maintaining geologic plausibility, even when the geologic model is for distinct facies.

As illustrated by our consideration of the Tengiz field example, the limited memory BFGS provides a feasible algorithm for realistic history matching problems. This example, illustrates, however, that when data are inconsistent, even a robust optimization algorithm can yield unrealistic estimates of rock property fields.

The constrained limited memory BFGS algorithm based on a log-transformation of model parameters can be used to alleviate undershooting and overshooting problems when estimating rock property fields by history matching production data.

The limited memory BFGS algorithm was effective for history matching changes in acoustic impedance data for a synthetic solution-gas drive reservoir based on an example from the middle east. Results obtained using a combination of seismic and production data were more accurate than those obtained from either seismic or production data alone. When applied to real field data, uncertainty in the rock physics model and in the magnitude of the noise in the data made the problem more difficult, but reasonable estimates were still obtainable.

In many reservoirs, knowledge of the location of geologic facies and their boundaries are critical to the problem of prediction of production. We have completed the early investigation of the problem of estimating geostatistical parameters for a truncated plurigaussian model, that will make it honor a training image and found that the geostatistical parameters that define a particular geologic traning image are poorly constrained by the image; in other words, many different sets of parameters such as varigram range and orientation will give the same training image. We have also demonstrated the feasibility of automatically adjusting

the locations of the boundaries of geologic facies to honor production data.

We have shown that it is feasible to construct reasonable estimates or realizations of the relative permeability curves and log-permeability fields by history matching production data obtained under three-phase flow conditions. The results assume that a prior model is available for absolute permeability and the parameters that characterize the relative permeability functions. Although the results indicate that reasonably good estimates of model parameters may be obtainable by history matching only pressure data, history matching pressure, gas-oil ratio and water-oil ratio data together gives better results.

Technical References

S. I. Aannonsen, A. Cominelli, O. Gosselin, I. Aavatsmark, and T. Barkve. Integration of 4d data in the history match loop by investigating scale dependent correlations in the acoustic impedance cube. In *8th European Conference on the Mathmatics of Oil Recovery*, 2002.

Y. Abacioglu, D. S. Oliver, and A. C. Reynolds. Efficient reservoir history matching using subspace vectors. *Computational Geosciences*, 5(2):151–172, 2001.

Yafes Abacioglu. *The Use of Subspace Methods for Efficient Conditioning of Reservoir Models to Production Data*. Ph.D. thesis, University of Tulsa, Tulsa, Oklahoma, 2001.

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

J. S. Archer and S. W. Wong. Use of a reservoir simulator to interpret laboratory waterflood data. *Soc. Petrol. Eng. J*, 12(6):343–347, 1973.

Eliana Arenas, Cor van Kruijsdijk, and Tanja Oldenziel. Semi-automatic history matching using the pilot point method including time-lapse seismic data. *SPE* 71634, 2001.

Khalid Aziz and A. Settari. *Petroleum Reservoir Simulation*. Elsevier Applied Science Publishers, London, 1979.

Ronald Behrens, Patrick Condon, William Haworth, Mark Bergeron, Zhijing Wang, and Christine Ecker. 4D seismic monitoring of water flux at Bay Marchand: the practical use of 4D in an imperfect world. *SPE Reservoir Evaluation and Engineering*, 2002.

Zhuoxin Bi. *Conditioning 3D Stochastic Channels to Well-Test Pressure Data*. Ph.D. thesis, University of Tulsa, Tulsa, Oklahoma, 1999.

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

Kevin T. Chambers, W. S. Hallager, C. S. Kabir, and R. A. Garber. Characterization of a carbonate reservoir using pressure-transient test and production logs: Tengiz field, kazakstan SPE 38657. In *Proceedings of the 1997 SPE Annual Technical Conference and Exhibition*, pages 155–170, 1997.

Kevin T. Chambers, W. S. Hallager, C. S. Kabir, and R. A. Garber. Characterization of a carbonate reservoir using pressure-transient test and production logs: Tengiz field, kazakstan. *SPE Reservoir Eval. & Eng.*, 4(4):250–259, 2001.

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

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

L. Chu and A. C. Reynolds. A general efficient method for generating sensitivity coefficients. In *Well Testing, Reservoir Characterization and Reservoir Simulation*, Petroleum Reservoir Exploitation Projects, pages 100–133. University of Tulsa, 1995.

Lifu Chu, M. Komara, and R. A. Schatzinger. An efficient technique for inversion of reservoir properties using iteration method. *SPE Journal*, 5(1):71–81, 2000.

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

Lifu Chu, Albert C. Reynolds, and Dean S. Oliver. Reservoir description from static and well-test data using efficient gradient methods. *SPE 29999 (1995 SPE International Meeting in Beijing)*, page 16 pages, 1995b.

M. Cooper, E. Thorogood, A. O'Donovan, P. Kristiansen, and P. Christie. Foinaven active reservoir management: The time-lapse signal. *SEG 1999 Expanded Abstracts*, 1999.

George Corliss, Christele Faure, Andreas Griewank, Laurent Hascoet, and Uwe Naumann, editors. *Automatic Differentiation of Algorithms, from Simulation to Optimization*. LNCSE. Springer, 2001.

J. E. Dennis and Robert B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. SIAM, Philidelphia, 1996.

J. K. Dietrich and P. L. Bonder. Three-phase oil relative permeability models, SPE 6044. In *Proceedings of the 1976 SPE Annual Fall Meeting*, 1976.

Yannong Dong and Dean S. Oliver. Sensitivity of seismic impedance change to permeability and porosity. Technical Report TUPREP 2002, University of Tulsa, 2002.

Yannong Dong and Dean S. Oliver. Quantitative use of 4D seismic data for reservoir characterization (SPE-84571). In *Proceedings of 2003 SPE Annual Technical Conference and Exhibition*, 2003.

R. Fletcher and C. M. Reeves. Function minimization by conjugate gradient. *Computer Journal*, 7:149–154, 1964.

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

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

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

Guohua Gao and A. C. Reynolds. An improved implementation of the lbfgs algorithm for automatic history matching (SPE-90058). In *2004 SPE Annual Technical Conference and Exhibition*, 2004.

F. Gassmann. Elastic waves through a packing of spheres. *Geophysics*, 16:673–685, 1951.

J. Jaime Gómez-Hernández and André G. Journel. Joint sequential simulation of multigaussian fields. In A. Soares, editor, *Geostatistic Troia 92*, pages 133–144. 1992.

A. A. Grimstad, T. Mannseth, G. Naevdal, and G. Urkedal. Scale splitting approach to reservoir characterization (SPE-66394). In *2001 SPE Annual Technical Conference and Exhibition*, 2001.

D. Han. *Effects of porosity and clay content on acoustic properties of sandstones and unconsolidated sediments*. PhD thesis, Stanford University, 1986.

N. He, A. C. Reynolds, and D. S. Oliver. Three-dimensional reservoir description from multiwell pressure data and prior information. *Soc. Pet. Eng. J.*, pages 312–327, 1997.

Nanqun He and Kevin T. Chambers. Calibrate flow models with well-test data to improve history matching, SPE 56681. In *Proceedings of the 1999 SPE Annual Technical Conference and Exhibition*, pages 1–12, 1999.

Xuri Huang, Laurence R. Bentley, and Claude Laflamme. Integration of production history and time-lapse seismic data guided by seismic attribute zonation. *SPE* 68819, 2001.

Xuri Huang, Laurent Meister, and Rick Workman. Improvement and sensitivity of reservoir characterization derived from time-lapse seismic data. *SPE* 49146, 1998.

Xuri Huang, Robert Will, Mashiur Khan, and Larry Stanley. Reservoir characterization by integration of time-lapse seismic and production data. *SPE* 38695, 1997.

John K. Hughes. Examination of seismic repeatability as a key element of time-lapse seismic monitoring. *SPE* 50627, 1998.

P. Jacquard and C. Jain. Permeability distribution from field pressure data. *Soc. Petrol. Eng. J.*, 5(4):281–294, 1965.

Hans O. Jahns. A rapid method for obtaining a two-dimensional reservoir description from well pressure response data. *Soc. Petrol. Eng. J.*, 6(12):315–327, 1966.

André Journel and C. J. Huijbregts. *Mining Geostatistics*. Academic Press, New York, 1978. 600 p.

Rintu Kalita. *Conditioning a Three Dimensional Reservoir Model to Gas Production Data*. M.S. thesis, University of Tulsa, Tulsa, Oklahoma, 2000.

B. L. N. Kennett and P. R. Williamson. Subspace methods for large-scale nonlinear inversion. In *Mathematical Geophysics*, pages 139–154. D. Reidel, 1988.

P. D. Kerig and A. T. Watson. Relative-permeability estimation from displacement experiments: An error analysis. *SPE Res. Eng.*, 1(1):175–182, 1986.

P. D. Kerig and A. T. Watson. A new algorithm for estimating relative permeabilities from displacement experiments. *SPE Res. Eng.*, 1(1):103–112, 1987.

Peter K. Kitanidis. Quasi-linear geostatistical theory for inversing. *Water Resour. Res.*, 31 (10):2411–2419, 1995.

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

Jorge L. Landa and Roland N. Horne. A procedure to integrate well test data, reservoir performance history and 4-D seismic information into a reservoir description (SPE-38653). In *1997 SPE Annual Technical Conference and Exhibition*, 1997.

Jorge L. Landa, Roland N. Horne, Medhat M. Kamal, and C. D. Jenkins. Reservoir characterization constrained to well test data: A field example. *SPE Reservoir Evaluation & Engineering*, 3(4):74–79, 2000.

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

G. Le Loc'h, H. Beucher, A. Galli, B. Doligez, and Heresim 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.

G. Le Loc'h and A. Galli. Truncated plurigaussian method: Theoretical and practical points of view. In *Geostatistical Simulations*, pages 211–222. Kluwer Academic, 1997.

Tai-Yong Lee and John H. Seinfeld. Estimation of absolute and relative permeabilities in petroleum reservoirs. *Inverse Problems*, 3(4):711–728, 1987.

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

Ruijian Li, A. C. Reynolds, and D. S. Oliver. History matching of three-phase flow production data, SPE 66351. In *Proceedings of the 2001 SPE Reservoir Simulation Symposium*, 2001a.

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

Ruijian Li, Albert C. Reynolds, and Dean S. Oliver. Sensitivity coefficients for three-phase flow history matching. In *2001 Canadian International Petroleum Meeting*, 2001b.

Ruijian Li, Albert C. Reynolds, and Dean S. Oliver. Sensitivity coefficients for three-phase flow history matching. *J. Canadian Pet. Tech.*, 42(4):70–77, 2003b.

D. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. *Math. Programming*, 45:503–528, 1989.

Ning Liu and Dean S. Oliver. Automatic history matching of geologic facies, SPE 84594. *Proceedings of the 2003 SPE Annual Technical Conference and Exhibition*, (History matching of facies boundaries using truncated pluri-Gaussian method.):1–15, 2003.

D.E. Lumley, A.G.Nunns, G.Delorme, and M.F.Bee. Meren field, nigeria: A 4d seismic case study. *SEG 1999 Expanded Abstracts*, 1999.

Randall L. Mackie and Theodore R. Madden. Three-dimensional magnetotelluric inversion using conjugate gradients. *Geophys. J. Int.*, 115:215–229, 1993.

Eliana M. Makhlouf, Wen H. Chen, Mel L. Wasserman, and John H. Seinfeld. A general history matching algorithm for three-phase, three-dimensional petroleum reservoirs. *SPE Advanced Technology Series*, 1(2):83–91, 1993.

Kiyoshi Masumoto. Pressure derivative matching method for two phase fluid flow in heterogeneous reservoir. *SPE-59462*, 2000.

W. Murray. *Numerical Methods for Unconstrained Optimization*. Academic Press, New York, 1972.

Stephen G. Nash. Preconditioning of truncated-Newton methods. *SIAM. J. SCI. STAT. COMPUT*, 6(3):599–616, 1985.

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

Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, New York, 1999.

D. W. Oldenburg, P. R. McGillivray, and R. G. Ellis. Generalized subspace methods for large-scale inverse problems. *Geophys. J. Int.*, 114(1):12–20, 1993.

Douglas W. Oldenburg and Yaoguo Li. Subspace linear inverse method. *Inverse Problems*, 10:915–935, 1994.

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

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

Dean S. Oliver. Calculation of the inverse of the covariance. *Mathematical Geology*, 30(7): 911–933, 1998.

Dean S. Oliver, Nanqun He, and Albert C. Reynolds. Conditioning permeability fields to pressure data. In *European Conference for the Mathematics of Oil Recovery, V*, pages 1–11, 1996.

Henning Omre, Håkon Tjelmeland, Yuanchang Qi, and Leif Hinderaker. Assessment of uncertainty in the production characteristics of a sand stone reservoir. In *Reservoir Characterization III*, pages 556–603. PennWell Books, Tulsa, OK, 1993.

S. S. Oren. Self-scaling variable metric (SSVM) algorithms II: Implementation and experiments. *Management Science*, 20:863–874, 1974.

S. S. Oren and D.G. Luenberger. Self-scaling variable metric (SSVM) algorithms I: Criteria and sufficient conditions for scaling a class of algorithms. *Management Science*, 20:845–862, 1974.

S. S. Oren and E. Spedicato. Optimal conditioning of self-scaling variable metric algorithms. *Mathematical Programming*, 10:70–90, 1976.

Christopher C. Paige and Michael A. Saunders. Algorithm 583, LSQR: Sparse linear equations and least squares problems. *ACM Transactions on Mathematical Software*, 8(2): 195–209, 1982.

D. W. Peaceman. Interpretation of well block pressures in numerical reservoir simulation. *Soc. Pet. Eng. J.*, 18(6):183–194, 1978.

D. W. Peaceman. Interpretation of well-block pressures in numerical reservoir simulation with non-square grid blocks and anisotropic permeability. *Soc. Pet. Eng. J.*, 23(6):531–543, 1983.

Daniel Rahon, Georges Blanc, and Dominique Guérillot. Gradients method constrained by geological bodies for history matching. In *Proceedings of the 5th European Conference on the Mathematics of Oil Recovery*, pages 283–293, 1996.

Daniel Rahon, 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*, pages 141–150, 1997.

Albert C. Reynolds, Nanqun He, Lifu Chu, and Dean S. Oliver. Reparameterization techniques for generating reservoir descriptions conditioned to variograms and well-test pressure data. *Soc. Petrol. Eng. J.*, 1(4):413–426, 1996.

Albert C. Reynolds, Ruijian Li, and Dean S. Oliver. Simultaneous estimation of absolute and relative permeability by automatic history matching of three-phase flow production data. *J. Canadian Pet. Tech.*, 43(3):37–46, 2004.

G. B. Savioli and C. A. Grattoni. On the inverse problem application to reservoir characterization. *SPE-025522*, 1992.

D. F. Shanno. Conditioning of quasi-Newton method for function minimization. *Maths. Comp.*, 24:647–656, 1970.

D. F. Shanno and Kang-Hoh Phua. Matrix conditioning and nonlinear optimization. *Mathematical Programming*, 14:149–160, 1978.

P. M. Sigmund and F. G. McCaffery. An improved unsteady-state procedure for determining the relative-permeability characteristics of heterogeneous porous media. *Soc. Petrol. Eng. J.*, 19(1):15–28, 1979.

Jan Arild Skjervheim. Gradual deformation. Class project, NTNU, Trondheim, Norway, 2002.

H. L. Stone. Estimation of three-phase relative permeability and residual oil data. *J. Can. Pet. Tech.*, 12(4):53–61, 1973.

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

L. G. Thompson and A. C. Reynolds. Well testing for radially heterogeneous reservoirs under single and multiphase flow conditions. *SPE Formation Evaluation*, 12(1):57–64, 1997.

A. N. Tikhonov. Regularization of incorrectly posed problems. *Soviet Math. Dokl.*, 4:1624–1627, 1963.

R. van Ditzhuijzen, T. Oldenziel, and van Kruijsdijk. Geological parameterization of a reservoir model for histroy matching incorporating time-lapse seismic based on a case study of the statfjord field. *SPE* 71318, 2001.

W. T. Watson, P. C. Richmond, P. D. Kerig, and T. M. Tao. A regression-based method for estimating relative relative permeabilities from displacement experiments. *SPE Res. Eng.*, 43(5):953–958, 1988.

Zhan Wu, A. C. Reynolds, and D. S. Oliver. Conditioning geostatistical models to two-phase production data. *Soc. Petrol. Eng. J.*, 3(2):142–155, 1999.

W. Xu, T. T. Tran, R. M. Srivastava, and A. G. Journel. Integrating seismic data in reservoir modeling: the collocated cokriging approach, (SPE-24742). In *1992 SPE Annual Technical Conference and Exhibition*, 1992.

Pin-Huel Yang and A. Ted Watson. Automatic history matching with variable-metric methods. *SPE Reservoir Engineering*, 3(3):995–1001, 1988.

Pin-Huel Yang and A. Ted Watson. A Bayesian methodology for estimating relative permeability curves. *SPE Reservoir Engineering*, 6(2):259–265, 1991.

William W-G Yeh. Review of parameter identification in groundwater hydrology: The inverse problem. *Water Resour. Res.*, 22(2):95–108, 1986.

F. Zhang and A. C. Reynolds. Optimization algorithms for automatic history matching of production data. *Proceedings of 8th European Conference on the Mathematics of Oil Recovery*, page 10 pages, 2002a.

F. Zhang and A. C. Reynolds. Optimization algorithms for history matching. In *TUPREP Research Report 19*, pages 14–105. The University of Tulsa, 2002b.

F. Zhang, A. C. Reynolds, and D. S. Oliver. Evaluation of the reduction in uncertainty obtained by conditioning a 3D channel to multiwell pressure data. TUPREP report, The University of Tulsa, 2001a.

F. Zhang, A. C. Reynolds, and D. S. Oliver. Optimization algorithms for history matching. In *TUPREP Research Report 18*, pages 144–171. The University of Tulsa, 2001b.

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

F. Zhang, A. C. Reynolds, and D. S. Oliver. The impact of upscaling errors on conditioning a stochastic channel to pressure data. *SPE J*, 8(1):13–21, 2003.

Fengjun Zhang. *Automatic History Matching of Production Data for Large Scale Problems*. Ph.D. thesis, University of Tulsa, Tulsa, Oklahoma, 2002.