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. But while the volume of data that can potentially provide information on reservoir architecture and fluid distributions has increased enormously in the past decade, it is not yet possible to make use of 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. The goal of this project is to develop computationally efficient automatic history matching techniques for generating geologically plausible reservoir models which honor both static and dynamic data. Solution of this problem is necessary for the quantification of uncertainty in future reservoir performance predictions and for the optimization of reservoir management.

Facies (defined here as regions of relatively uniform petrophysical properties) are common features of all reservoirs. Because the flow properties of the various facies can vary greatly, knowledge of the location of facies boundaries is of utmost importance for the prediction of reservoir performance and for the optimization of reservoir management. When the boundaries between facies are fairly well known, but flow properties are poorly known, the average properties for all facies can be determined using traditional techniques. Traditional history matching honors dynamic data by adjusting petrophysical properties in large areas, but in the process of adjusting the reservoir model ignores the static data and often results in implausible reservoir models. In general, boundary locations, average permeability and porosity, relative permeability curves, and local flow properties may all need to be adjusted to achieve a plausible reservoir model that honors all data. In this project, we will characterize the distribution of geologic facies as an indicator random field, making use of the tools of geostatistics as well as the tools of inverse and probability theory for data integration.

EXECUTIVE SUMMARY

Bayesian statistics provides a framework for the automatic history matching of multiphase flow production data to construct estimates or realizations of reservoir properties that are consistent with time-lapse seismic data, production data and static data obtained from logs, cores and geologic and geophysical interpretation. The automatic history matching procedure developed and implemented in our work requires the minimization of an objective function which consists of the sum of a regularization term and production data mismatch terms. The regularization term represents a geostatistical model constructed from static data. If the number of production data is small or the number of reservoir variables to be estimated is small, we showed in our previous DOE annual report on this project that a modified Levenberg-Marquardt algorithm or a Gauss-Newton method can be applied to minimize the appropriate objective function. The Levenberg-Marquardt algorithm and Gauss-Newton method, however, require the computation of individual sensitivity coefficients and this is not computationally feasible for large scale problems where the number of production data to be matched is greater than a few hundred and the number of reservoir variables is on the order of a few thousand to tens of thousands. In this report, we discuss our implementation and testing of a scaled limited memory Broyden-Fletcher-Goldfard-Shanno algorithm. Experiments done to date suggest that this is a viable algorithm for large scale history matching problems. To date, we have tested this algorithm only for the estimation of stochastic simulation of rock property fields where rock property fields refer to reservoir simulator gridblock porosities or log-permeabilities. Gridblock log-permeabilities are converted to permeabilities when running the simulator. The reason for constructing estimates of log-permeabilities is that the theoretical justification of our approach to automatic history matching is based on Bayesian statistics and in this setting the permeability fields are assumed to be log-normal.

We have applied the limited memory BFGS algorithm to history match production data for two realistic problems. The first example is a field case which pertains to the Tengiz reservoir. This is an undersaturated reservoir produced by forty four wells. In this example, we estimated permeability fields on a $59 \times 49 \times 9$ upscaled reservoir simulation grid by

conditioning the permeability field generated from a geostatistical model to pressure buildup data by automatic history matching of buildup data. The second example considered is a pseudo-field example based on a current model for an actual North Sea reservoir. For the case, we simulated only one-half of the reservoir using a $39 \times 25 \times 10$ grid. The steeply-dipping reservoir has a significant gas cap and two gas injection wells are completed in the gas cap. The reservoir has five producing wells. The aquifer is separated from the oil zone by a low permeability tar mat which prevents water coning. We estimated gridblock horizontal and vertical log-permeabilities by history matching wellbore pressure data and producing GOR data. The two results suggest that estimation or stochastic simulation of rock property fields by automatic history matching of production data within the Bayesian framework is feasible for field problems.

As the limited memory BFGS optimization algorithm can deal with large amounts of data and large models, it should prove useful for the incorporation of time-lapse seismic data into reservoir characterization. In Chapter 5, we discuss the status of our work on incorporating time-lapse seismic data. As time-lapse seismic data covers the areal extent of the reservoir and is related to vertically averaged pressure and fluid saturations, it is expected that matching these seismic data simultaneously with production data will reduce the uncertainty in our estimates of the rock property fields. Applying a gradient based optimization algorithm to condition rock property fields to seismic data requires the calculation of sensitivity of the seismic data to the rock property fields. Here, two synthetic two-dimensional reservoir examples are considered to investigate our proposed approach for integrating seismic data; one example, pertains to waterflooding an oil reservoir and one case pertains to a solution-gas drive reservoir. In both cases, seismic impedance data is available at two or more times where the distribution of phases saturations in the reservoir are quite different so useful time-lapse impedance data can be constructed. Our results show that the change in seismic impedance due to errors in static geologic properties such as shaliness, clay modulus and sand modulus are relatively small compared to the change due to saturation changes. Thus, uncertainties in mineralogy will not have a significant error on time-lapse impedance data and we should be able to integrate time-lapse impedance data to reduce the uncertainty in rock property fields even when the mineralogy is not accurately known. We have implemented the adjoint procedure to calculate the sensitivity of seismic impedance to the permeability and porosity fields. For the waterflooding example and the solution-gas drive example mentioned above, it has been shown that the adjoint method gives accurate sensitivity coefficients. The next step is to use this same adjoint algorithm to compute the gradient of the objective function so that limited memory BFGS algorithm can be used to estimate rock property fields by

history matching production data and seismic data simultaneously.

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. One should recognize, however, that the primary control on the distribution of permeability and porosity is the facies; typically the variation of rock properties between facies is much larger than the variation within a facies. Because of this, it may be critical 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 generate a rich variety of textures and shapes, but the new approach should make it easier to generate approximations of the sensitivity coefficients needed to condition reservoir models to facies distributions.

Chapter 1

INTRODUCTION

Automatic history matching is based on minimizing an objective function which includes a sum of squared production data mismatch terms. Typically, minimization is done based on a derivative based optimization routine, such as the Gauss-Newton and Levenberg-Marquardt algorithms, because algorithms which do not use derivative information converge too slowly for practical applications. We refer to the reservoir variables to be estimated as model parameters. The model parameters typically represent reservoir simulator gridblock porosities and permeabilities (or log-permeabilities), may also represent well skin factors, transmissibility multipliers, parameters describing relative permeability curves or parameters describing facies boundaries. Standard implementations of the Gauss-Newton method or Levenberg-Marquardt algorithm require calculation of sensitivity coefficients, which formally represent the derivative of predicted production data with respect to the model parameters.

For automatic history matching problems of interest to us, the number of model parameters is greater than the number of independent production data and thus the history matching problem does not have a unique solution. If the Gauss-Newton procedure is applied to minimize an objective function consisting of only the sum of squared production data misfit terms, the Hessian matrix will be singular and the optimization algorithm will be unstable. This instability problem can be avoided by adding a regularization term to the objective function to be minimized; see Tikhonov (1963) and Parker (1994). With a proper regularization, the Hessian matrix in the Gauss-Newton method will be real symmetric positive definite and hence nonsingular. In this work, we use a prior geostatistical model to provide regularization. With this approach, the history matching problem is equivalent to a Bayesian estimation problem (Gavalas et al., 1976; Tarantola, 1987; He et al., 1997; Wu et al., 1999).

The Gauss-Newton method is popular because it converges quadratically in the neigh-

borhood of a minimum; see, for example, Fletcher (1987). Sometimes, however, if the initial guess in the Gauss-Newton method results in a large initial data mismatch, the Gauss-Newton will converge to a reservoir model which represents a local minimum and does not give an acceptable match of production data; see, Wu et al. (1999)and Li et al. (2001). For this reason, we often apply a Levenberg-Marquardt algorithm instead of the Gauss-Newton method if the production data misfit based on the initial reservoir model is very large.

Automatic history matching traces its roots to research conducted in the 1960's by Jacquard (1964), Jacquard and Jain (1965) and Jahns (1966). To the best of our knowledge, Jacquard and Jain (1965) presented the first procedure for numerically computing sensitivity coefficients for history matching purposes. They applied their method to the estimation of permeability in a two-dimensional reservoir from pressure data obtained under single-phase flow conditions. They used a combination of zonation (less than twenty distinct values) and an algorithm conceptually similar to the Levenberg-Marquardt algorithm to provide regularization. Jahns estimated transmissibility (kh/μ) values and storativity $(\phi c_t h)$ simultaneously by history matching single-phase flow pressure data. He used the finite difference method to compute sensitivity coefficients and applied the Gauss-Newton method with an exact line search to estimate the rock property fields by minimizing an objective function consisting only of the sum of squared pressure mismatch terms. Zonation was used to provide regularization. Jahns actually used a sequence of minimization steps where the number of zones, and hence the number of parameters was increased at each minimization step. The maximum number of parameters estimated was nine, eight zonal transmissibilities (or permeabilities) and total storativity. The finite difference method used to compute sensitivities requires $N_m + 1$ runs of the simulator where N_m is the number of model parameters estimated. This procedure would not be feasible when thousands of model parameters are estimated.

Jacquard and Jain (1965) based their procedure for computing sensitivity coefficients on an electric-circuit analogue. Later, motivated by Jacquard and Jain's ideas, Carter et al. (1974) presented an elegant derivation of a method to compute sensitivity coefficients for two-dimensional single-phase flow problems. As originally presented, the Carter et al. procedure can be applied to compute the sensitivity of simulator gridblock pressures to all gridblock permeabilities and porosities. If each well penetrates only a single gridblock, one can compute the sensitivity of the wellbore pressure to model parameters from the well's gridblock pressure sensitivities, provided the simulator uses a formula like the one of Peaceman (1978) to relate wellbore pressure and gridblock pressure. For two-dimensional single-phase flow problems with pressure measurements at N_w wells, this procedure requires $N_w + 1$ reservoir simulation runs to compute all sensitivity coefficients regardless of the number of model parameters and

regardless of the number of pressure data. For three-dimensional problems, the number of simulation runs required would be equal to one plus the number of gridblocks penetrated by wells. If the number of such gridblocks is large, the Carter et al. procedure becomes less computationally attractive. However, He et al. (1996) have developed an approximate three-dimensional version of the Carter et al. method which is computationally efficient. Regardless of the number of gridblocks penetrated by wells, the He et al. method requires only $N_w + 1$ reservoir simulation runs to compute the sensitivity of all well pressure data to all gridblock permeabilities and porosities. The method is only approximate and does not always yield accurate results if vertical flow is significant in gridblocks penetrated by wells. It is not clear, however, whether the fact that the sensitivity coefficients are approximate leads to significant errors in the history matching process.

For nonlinear problems, e.g., multiphase flow problems, the derivations of Carter et al. (1974) and He et al. (1996) do not apply. Thus, we are forced to seek other alternatives. One possible choice is the adjoint or optimal control method, introduced independently for the single-phase history matching problem by Chen et al. (1974) and Chavent et al. (1975). (For single-phase flow problems, Carter et al. (1982) have shown that their method is equivalent to the adjoint method.)

Unlike the Carter et al. (1982) method, however, the adjoint method can be applied to compute sensitivity coefficients in multiphase flow problems. Unfortunately, the procedure requires N_d adjoint solutions where N_d is the number of production data to be history matched. The sensitivities can be calculated easily once the adjoint variables have been computed. Solving an adjoint problem is similar to solving the simulation finite difference equations with two distinct differences: (i) to find the adjoint variables needed to computed the sensitivity of a particular production data at the time t_l, the appropriate adjoint problem is solved backward in time, from time t_l to time zero; (ii) unlike the forward problem (simulator problem), the adjoint problem is linear. At each time step in the adjoint solution, a matrix problem is solved. The coefficient matrix is independent of the production data but the right hand side of the matrix problem is determined directly from the specific production data. If N_d production data are uniformly spaced in time and the final time at which we have measured production data is t_L, then computing all adjoint solutions needed to compute the sensitivities of all production data effectively requires solving a sequence of matrix problems related to solving the adjoint problem backward in time from t_L. At each time step, the matrix problem is solved with an average of $N_d/2$ right-hand side vectors; see Wu et al. (1999) for additional discussion. Even if one uses a procedure based on solving a matrix problem with multiple right hand sides, the solution of the adjoint systems needed

6

to compute sensitivities for N_d production data will not be feasible when N_d is large. If one assumes that solving the adjoint matrix systems with an average of $N_d/2$ right hand side vectors is equivalent to at least $(0.1N_d)/2$ simulation runs, the number of equivalent simulation runs required is prohibitive if there are several hundred production data to be matched. Because of this, the adjoint method traditionally has been used only in conjunction with optimization methods which require only the gradient of the objective function, e.g., conjugate gradient or variable metric methods; see, for example, Wasserman et al. (1975), Lee and Seinfeld (1987a,b), Yang and Watson (1988), Makhlouf et al. (1993). Computation of the gradient of the objective function requires only the solution of a single adjoint system and thus requires no more computational time than one reservoir simulation run. Unfortunately, the implementations of these methods have resulted in slow convergence. For example, Makhlouf et al. (1993) reported that history matching a two-phase flow 450 cell reservoir model required 6400 CPU seconds on a CRAY X-MP/48. In their work, a conjugate gradient method was used as the optimization algorithm. For one three-phase flow problem with 450 grid blocks, 222 iterations of the conjugate gradient algorithm were required to obtain convergence.

Largely because of the results of Makhlouf et al. (1993), until recently, our work on automatic history matching has focused on using the Gauss-Newton and Levenberg-Marquardt algorithms instead of conjugate gradient or variable metric algorithms. Wu et al. (1999) were the first to use the adjoint method in conjunction with the Gauss-Newton method to perform history matching. They implemented the adjoint method to compute the sensitivity of all production data to gridblock permeabilities and porosities. In their work, they constructed estimates and realizations of permeability and porosity fields by conditioning a prior geostatistical model to pressure and water-oil ratio data. They considered only two-dimensional, two-phase flow (water-oil) systems. As noted in our previous DOE annual report on this process, we have extended the Gauss-Newton and Levenberg-Marquardt algorithms in conjunction with the adjoint method for sensitivity calculation to the problem of history matching production data for three-dimensional, three-phase flow problems. The resulting history matching process, however, is not practically feasible for problems of interest to us where both the number of data and the number of model parameters exceed a few hundred.

Perhaps because it is simple to implement, the so-called gradient method is frequently used to compute sensitivity coefficients needed for automatic history matching. This method was introduced into the petroleum engineering literature by Anterion et al. (1989), but was known earlier in the ground water hydrology literature as the sensitivity coefficient method;

7

see, for example, the review of parameter identification methods by Yeh (1986). In this procedure, the sensitivity of pressures and saturations to model parameters at the end of a simulator time-step can be obtained by solving a matrix problem obtained by differentiating the matrix form of the finite difference equations with respect to a model parameter, e.g., a gridblock value of permeability or porosity. From the pressure and saturation sensitivities, one can easily construct other sensitivity coefficients, e.g., the sensitivity of gas-oil ratio to model parameters. The advantage of the gradient simulator method is that the matrix problem solved to obtain these sensitivity coefficients involves the same coefficient matrix as the one used to solve for pressures and saturations at this time step. Moreover, the coefficient matrix does not depend on the model parameters; only the right hand side of the matrix problem depends on the model parameters. Thus, the problem reduces to solving a matrix problem with multiple right-hand side vectors, one right-hand side vector, for each model parameter. The difficulty is that if we wish to estimate (or construct realizations of) permeabilities and porosities at several thousand gridblocks, then we have several thousand right-hand sides. The number of right-hand sides is equal to the number of model parameters to be estimated. With the fast iterative solver developed by Killough et al. (1995), it appears that the computational time to compute a single sensitivity coefficient is on the order of 10% of a forward simulation. For the gradient simulator to be practical, the number of model parameters must be small. This means, if the underlying reservoir simulation problem involves tens of thousands of gridblocks, one must reduce the number of parameters estimated directly in the optimization algorithm by some form of reparameterization, e.g., zonation (Jacquard and Jain, 1965) or gradzones (Bissell et al., 1994; Bissell, 1994; Tan, 1995), pilot points (de Marsily et al., 1984; RamaRao et al., 1995; Bissell et al., 1997) or subspace methods (Kennett and Williamson, 1988; Oldenburg et al., 1993; Reynolds et al., 1996; Abacioglu et al., 2000).

When the number of model parameters and number of production data to be matched are both large and can not be reduced by some reparameterization technique without incurring a significant loss of information, one must seek an alternative to computing and storing the full sensitivity coefficient matrix, G. One can write the Gauss-Newton method such that each iteration requires the solution of an $N_d \times N_d$ matrix problem where N_d represents the number of production data to be matched. If this matrix problem is solved by a conjugate gradient method (Hestenes and Stiefel, 1952; Fletcher and Reeves, 1964; Shanno, 1978a,b), the explicit direct computation of G is not required. Each iteration requires only the product of G times a vector and the product of the transpose of G times a vector. A procedure for computing these matrix vector products without first computing G was introduced into the

petroleum engineering literature by Chu et al. (2000) although the basic idea appeared earlier in a somewhat simpler context in the geophysics literature; see Mackie and Madden (1993). Although computation of the matrix products is relatively efficient, the conjugate gradient method may require up to N_d iterations to obtain convergence if the matrix is poorly conditioned and no good preconditioning matrix is available (see Axelsson (1994)). If N_d is large, this would render the algorithm impractical. Although the convergence of the conjugate gradient method can be considerably accelerated by the choice of a good preconditioner, it is not clear that one can construct a good preconditioner since the coefficient matrix for the matrix problem that is solved is not explicitly computed. To compute this coefficient matrix would require the explicit computation of the full sensitivity coefficient matrix G.

One can also avoid explicit computation of all sensitivity coefficients if history matching is done using a nonlinear optimization method, that requires only the gradient of the objective function. As mentioned previously, Makhlouf et al. (1993) found that a nonlinear conjugate gradient algorithm could require over two hundred iterations to converge even for a small three-phase flow history matching problem. As each conjugate gradient iteration requires roughly the equivalent of three reservoir simulation runs, history matching a large problem using a nonlinear conjugate gradient method does not appear to be feasible based on the results of Makhlouf et al. (1993). However, Makhlouf et al. (1993) did not apply preconditioning. If a good preconditioning matrix can be found for nonlinear conjugate gradients, it is conceivable that convergence could be considerably accelerated.

Quasi-Newton or 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. The methods differ in how they correct or update the inverse Hessian approximation at each iteration. The rank one correction formula was first suggested by Broyden (1967). Another formula, now called the DFP algorithm, was first suggested by Davidon in 1959 and later presented by Fletcher and Powell (1963). The BFGS correction formula, suggested independently by Broyden (1970), Fletcher (1970), Goldfarb (1970) and Shanno (1970), and several variants of the BFGS formula (like the self-scaling variable metric (SSVM) by Oren (1973), limited memory BFGS by Nocedal (1980) and Liu and Nocedal (1989)) have also been advanced as useful variable metric methods.

The conjugate gradient method was originally proposed by Hestenes and Stiefel (1952) for solving linear systems and extended to nonlinear optimization by Fletcher and Reeves (1964) to obtain the Fletcher-Reeves algorithm. Later Polak (1971) proposed a different formula to calculate the coefficient involved in the search direction update equation. Powell (1977) presented some numerical results and theoretical reasons which indicate that the

9

Polak-Ribière algorithm is superior to the Fletcher-Reeves algorithm. The efficiency of the conjugate gradient method depends primarily on the preconditioner used.

The limited memory BFGS (LBFGS) was designed for the purpose of solving large scale problems which involve thousands of variables. Limited memory methods originated with the work of Shanno (1978a), and were subsequently developed and analyzed by Buckley (1978), Nazareth (1979), Nocedal (1980), Shanno (1978b), and Buckley and Lenir (1983). Liu and Nocedal (1989), and Nash and Nocedal (1991) tested LBFGS method with a set of problems. They concluded that LBFGS performs better than conjugate gradient in terms of computational efficiency, except in cases where the function evaluation is inexpensive. Nash and Nocedal (1991) also tested a truncated-Newton method in their work. From their comparison, none of the algorithms is clearly superior to the other.

The self-scaling variable metric (SSVM) method was used by Yang and Watson (1988) on hypothetical water floods of both 1D and 2D reservoir models. The 1D reservoir model consisted of 10 gridblocks with an injection well at one end and a producing well at the other end. Sixty data from each well were used for history matching. Four cases based on this 1D reservoir model were tested. The reservoir was characterized by different parameters in different cases. The number of model parameters varied from 9 to 19. Two other cases were based on a quarter of a five-spot 2D model which consisted of a 10×10 grid. Again sixty data from each well were history matched. The number of model parameters for these two cases were 4 and 11 respectively. In this paper, the authors tested four different algorithms, BFGS, SSVM, conjugate gradient and steepest descent. They concluded that (i) the self-scaling variable metric method is significantly more efficient than the BFGS method; (ii) the SSVM and BFGS methods are more efficient and robust than the conjugate gradient method, except in the case where the objective function is nearly quadratic; and (iii) both SSVM and BFGS methods perform significantly better than the steepest descent method.

Masumoto (2000) applied the SSVM method to a water-oil two phase fluid flow problem. The author considered a 1D reservoir model with 20 gridblocks. With a fixed porosity field, the author estimated the gridblock permeabilities. The objective function he minimized included a pressure mismatch part and the pressure derivative mismatch part. The author did not give any information about how many data he history matched or any assessment of the minimization algorithm. Savioli and Grattoni (1992) compared four different minimization algorithms: Davidon-Fletcher-Powell (DFP), Fletcher-Reeves (FR), BFGS and Levenberg-Marquardt (LM). The authors presented two examples. In the first example, they estimated one permeability value and one porosity value by applying these four algorithms. The second example they considered was an oil-water two phase water flooding problem. They estimated

the exponent used to define the relative permeability and capillary curves with a power law function (only one adjustable parameter for each curve). They concluded that among these four algorithms, BFGS performed best in terms of computational efficiency and stability. Given the small number of parameters estimated, it is difficult to know whether these results will extrapolate to large scale problems.

In this work, we explore the application of the nonlinear preconditioned conjugate gradient method and variable metric (quasi-Newton) methods as optimization algorithms for automatic history matching of production data. Our results confirm our earlier conjecture that the limited memory BFGS (LBFGS) procedure is the most viable optimization algorithm for automatic history matching of multiphase flow production data for problems where the number of reservoir variables to be estimated and the number of data are both large. We apply the LBFGS method to problems representative of field cases. The first example is a true field case, where we match static pressures from pressure buildup surveys for the Tengiz reservoir, which is an undersaturated oil reservoir. The second example is actually a pseudo-field case in which we generated a truth case based on a reservoir model for an actual North Sea reservoir. Using this truth case, we generated synthetic production data (pressures and gas-oil ratios) from a forward simulation run. In generating the synthetic data, we used well locations and wells constraints that were similar to the actual wells in the field. Then ignoring our knowledge of the truth case, we constructed estimates of the permeability fields by history matching the production data.

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

11

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. Advances in automatic or computer-assisted history matching have allowed researchers to consider the integration of time-lapse seismic data with production data. All quantitative approaches for doing this involve the minimization of an objective function that includes the mismatch between the synthetic changes in seismic data and the observed changes. Using optimization methods, a distribution of parameters that minimize the objective function is sought. The type of seismic data used in the objective function has varied among the researchers. Huang et al. (1997) used amplitude difference or other seismic attributes difference while Arenas et al. (2001) used velocity difference. Landa and Horne (1997) assumed that saturation changes could be obtained directly from time-lapse surveys.

While a number of geophysicists (Tura and D.Lumley, 1999; Landro, 2001; Meadows, 2001) have assumed that changes in saturation and pressure can be estimated directly from time-lapse seismic data (including offset data), it is clearly less restrictive to use all data (including production data) in the estimation of saturation and pressure. Thus we will use the seismic data in the objective function–not saturations and pressures.

Because the number of data and model parameters can be quite large in history matching problems which include time-lapse seismic data, it is common to reduce the number of model parameters by using "pilot points" (Arenas et al., 2001; van Ditzhuijzen et al., 2001) or to divide model into zones with similar properties (Huang et al., 2001).

In studies to date, the sensitivity of time-lapse seismic data to changes in model param-

eters 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. It will also be necessary to use more efficient optimization methods in the history matching than those used previously to get optimum model parameters. A practical method for doing this will be outlined in this document.

Because the variation of permeability and porosity between facies is normally much larger than the variation within a facies, it may be important to estimate facies boundaries in the history matching process. To do so requires a model for the generation of facies maps. The truncated plurigaussian random field model is fairly appealing as a method for generating facies distributions. Galli and his coworkers at the center for Geostatistics in the School of Mines in Fontainebleau have established that it can be used to generate a remarkably wide variety of facies textures and shapes. Besides the qualitative visual appeal of the images, the method uses the covariance in a consistent way — unlike most indicator models. The truncated plurigaussian model is not, however, widely applied outside of France. The reason seems to be that it is fairly difficult to select model parameters that are consistent with a particular image such as might be provided by a geologist. We use the term parameters to refer to quantities such as the range, the variance, the covariance type (Gaussian, Exponential, Spherical, etc.) and the thresholds for discrimination of facies. In this work, we discuss an implementation a truncated plurigaussian model for the generation of facies distributions with thresholds determined by three intersecting lines. We believe it will be possible to integrate this model into the history matching process so that the location of boundaries between facies can be estimated by matching production data. This aspect of our research is in its early stages.

Chapter 2

EXPERIMENTAL

Experimental work is not applicable to the research tasks and goals of this project. Consequently, no experimental work has or will be done.

Chapter 3

HISTORY MATCHING OF PRODUCTION DATA

3.1 Model Estimation and Simulation

Here, we define the reservoir model parameters and the a posteriori probability density function (pdf) for these parameters. This pdf, which is conditional to production data, defines the set of plausible reservoir descriptions. We focus on 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. Methods for sampling this pdf to characterize the uncertainty in model parameters and the uncertainty in performance predictions are discussed only briefly.

3.1.1 The Prior Model.

For simplicity, the reservoir is assumed to be a rectangular parallelepiped which occupies the region

$$\Omega = \{(x, y, z) \mid 0 < x < L_x, 0 < y < L_y, 0 < z < L_z\}. \tag{3.1}$$

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).

In the current version of our inverse code, we can compute sensitivity of production data to gridblock porosities, horizontal permeabilities, vertical permeabilities and the skin factor at each well. However, for the specific examples considered here, we assume porosity is known and only estimate the horizontal and vertical permeability fields. Thus we present our equations for the case where the model parameters are simulator gridblock horizontal log-permeabilities, vertical log-permeabilities and well skin factors. Thus, if there are N simulator gridblocks and N_w wells, the total number of model parameters is equal to $N_m = 2N + N_w$. Specifically, the vector of model parameters is given by

$$m = [m_k^T, m_{k_z}^T, m_s^T]^T, \tag{3.2}$$

where m_k is an N-dimensional column with its jth entry equal to the horizontal permeability for gridblock j, m_{k_z} is an N-dimensional column with its jth entry equal to the vertical permeability for gridblock j, and m_s is an N_w dimensional column vector with its jth entry given by the skin factor at the jth well. 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 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 the recent work on automatic history matching by He et al. (1997) and Wu et al. (1999), we follow ideas that can be found in Tarantola (1987) and simply assume that a prior geostatistical model for $m_r = [m_k^T, m_{k_z}^T]^T$ can be constructed from static data. In our work, we assume this prior geostatistical model can be represented by a multivariate Gaussian distribution for m_r 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 horizontal, 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 vector of prior means is given by

$$m_{\text{prior}} = \begin{bmatrix} m_{k,\text{prior}} \\ m_{k_z,\text{prior}} \\ m_{s,\text{prior}} \end{bmatrix}. \tag{3.3}$$

We let C_k denote the prior covariance matrix for m_k, C_{k_z} denote the prior covariance for $\ln(k_z)$, C_{k,k_z} denote the cross covariance matrix between $\ln(k)$ and $\ln(k_z)$ and let C_s denote the $N_w \times N_w$ model covariance matrix for the vector of well skin factors. Then the prior model covariance matrix is given by

$$C_M = \begin{bmatrix} C_k & C_{k,k_z} & O \\ C_{k,k_z} & C_{k_z} & O \\ O & O & C_s \end{bmatrix}, \tag{3.4}$$

where the O's denote null submatrices of the appropriate size. 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

$$\pi_p(m) = a \exp\left\{ -\frac{1}{2}(m - m_{\text{prior}})^T C_M^{-1}(m - m_{\text{prior}}) \right\}, \tag{3.5}$$

where a is the normalizing constant. Note the model which has the highest probability based on Eq. 3.5 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.

3.1.2 The a Posteriori Probability Density Function.

We wish to determine the conditional pdf for m given observed production 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 production 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.

Pressure measurements errors are modeled as independent identically distributed Gaussian random variables with mean zero and variance σ_p^2. GOR measurement errors are modeled as independent identically distributed Gaussian random variables with mean zero and

variance σ_g^2. WOR ratio measurement errors are 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

$$\mathrm{Var}(e_{WOR}) = WOR_{\mathrm{obs}}^2 \epsilon_o + \frac{1}{q_{o,\mathrm{obs}}^2} \max\left[\epsilon_w^2 q_{w,\mathrm{obs}}^2, \, \sigma_{w,\mathrm{min}}^2\right], \tag{3.6}$$

where e_{WOR} denotes the error in the "measurement" of WOR constructed from the observed oil and water rates, $q_{o,\mathrm{obs}}$ and $q_{w,\mathrm{obs}}$. 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 $\sigma_{w,\mathrm{min}}$ 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 $\sigma_{w,\mathrm{min}}$. 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. If the total number of conditioning data is N_d, i.e., the dimension of d_{obs} is N_d, then the overall data covariance matrix is given by the following $N_d \times N_d$ diagonal matrix:

$$C_D = \begin{bmatrix} C_{D,p} & O & O \\ O & C_{D,w} & O \\ O & O & C_{D,g} \end{bmatrix}. \tag{3.7}$$

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 $d_{\mathrm{obs}} = d_{\mathrm{obs,g}}$ and $C_D = C_{D,g}$.

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

$$d = g(m), \tag{3.8}$$

to represent the operation of calculating d given m. In our work, Eq. 3.8 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

$$f(m|d_{\mathrm{obs}}) = a \exp\{-O(m)\}, \tag{3.9}$$

where a is the normalizing constant and

$$O(m) = \frac{1}{2}\left[(m - m_{\mathrm{prior}})^T C_M^{-1}(m - m_{\mathrm{prior}}) + (g(m) - d_{\mathrm{obs}})^T C_D^{-1}(g(m) - d_{\mathrm{obs}})\right]. \tag{3.10}$$

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. 3.9, or equivalently minimizes the objective function of Eq. 3.10. Although gradient based methods appear to be the only feasible way to construct a minimum of $O(m)$, there is no guarantee that Eq. 3.10 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.

If one wishes to generate multiple realizations of the model, it is necessary to sample the conditional pdf of Eq. 3.9. 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

$$ m_{\mathrm{uc}} = m_{\mathrm{prior}} + C_M^{1/2} z_M, \tag{3.11} $$

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

$$ C_M = LL^T, \tag{3.12} $$

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

$$ d_{\mathrm{uc}} = d_{\mathrm{obs}} + C_D^{1/2} z_D, \tag{3.13} $$

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

$$ O_r(m) = \frac{1}{2}(m - m_{\mathrm{uc}})^T C_M^{-1}(m - m_{\mathrm{uc}}) + \frac{1}{2}(d - d_{\mathrm{uc}})^T C_D^{-1}(d - d_{\mathrm{uc}}). \tag{3.14} $$

It can be argued (see 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. 3.14 gives a result m_c, we accept m_c as a legitimate realization if and only if

$$N_d - 5\sqrt{2N_d} \le O(m_c) \le N_d + 5\sqrt{2N_d}. \tag{3.15}$$

Although Eq. 3.15 has proved to be reliable for single-phase flow history matching problems and even some simple multiphase flow problems, in general we are unable to decrease the objective function value consistent with Eq. 3.15.

We wish to be able to history match several hundred production data to generate realizations of tens of thousands of model parameters. Thus computational efficiency is an extremely important consideration. 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, we are forced to focus on algorithms which require only the gradient of the objective function. As steepest descent often exhibits poor convergence properties (see, for example, Fletcher (1987)). The only viable algorithms in this category appear to be variable metric (quasi-Newton) methods and preconditioned conjugate gradient (PCCG) methods. The efficiency of PCCG methods largely rest on finding a good preconditioner. This, however, is not an easy task and despite significant effort we have not been able to find a preconditioner that yields a PCCG method that is as reliable as the quasi-Newton methods we have implemented.

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). Wu et al. (1999) overcame this problem by using an artificially high value for the variance of data measurement errors at early iterations. 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

$$\left[(1+\lambda_l)C_M^{-1} + G_l^T C_D^{-1} G_l\right]^{-1} \delta m^{l+1} = -\left[C_M^{-1}(m^l - m_{\text{prior}}) + G_l^T C_D^{-1}(g(m^l) - d_{\text{obs}})\right], \tag{3.16}$$

$$m^{l+1} = m^l + \alpha_l \delta m^{l+1} \tag{3.17}$$

where $\alpha_l = 1$. Here l, as either a subscript or superscript, refers to the iteration index. The matrix G_l denotes the $N_d \times M$ 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.

3.2 Calculation of the Gradient of the Objective Function

The equations that must be solved to compute sensitivity coefficient with the adjoint method were presented in the first annual report on this project. These results can also be found in Li et al. (2001). Here we discuss the procedure for computing the gradient of the objective function.

3.2.1 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. 3.1. 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 3.1 summarizes the different cases and the unknown 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.

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

Table 3.1: Equations and unknowns solved for in the simulator.

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	$S_g = 0$; $S_o = 1 - S_g$

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

$$f_{u,i}(y^{n+1}, y^n, m) = 0, \tag{3.18}$$

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

$$f_{wf,l}(y^{n+1}, y^n, m) = 0, \tag{3.19}$$

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

$$f_{u,i}^{n+1} = f_{u,i}(y^{n+1}, y^n, m), \tag{3.20}$$

and

$$f_{wf,l}^{n+1} = f_{wf,l}(y^{n+1}, y^n, m), \tag{3.21}$$

then Eqs. 3.18 and 3.19 can be rewritten as

$$f_{u,i}^{n+1} = 0, \tag{3.22}$$

and

$$f_{wf,l}^{n+1} = 0, \tag{3.23}$$

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. 3.21 is simplified to

$$f_{wf,l}^{n+1} = p_{wf,l}^{n+1} - p_{wf,l,0}^{n+1} = 0. \tag{3.24}$$

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

$$
\begin{align}
f_{1,i}^{n+1} &= f_{o,i}^{n+1} + f_{w,i}^{n+1} + f_{g,i}^{n+1} = 0 \tag{3.25} \\
f_{2,i}^{n+1} &= f_{o,i}^{n+1} = 0 \tag{3.26} \\
f_{3,i}^{n+1} &= f_{g,i}^{n+1} = 0 \tag{3.27}
\end{align}
$$

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

$$
\begin{align}
f_{1,i}^{n+1} &= f_{o,i} = 0 \tag{3.28} \\
f_{2,i}^{n+1} &= f_{w,i}^{n+1} = 0 \tag{3.29} \\
f_{3,i}^{n+1} &= f_{g,i}^{n+1} = 0, \tag{3.30}
\end{align}
$$

instead of Eqs. 3.25 through 3.27, to construct the Jacobian matrix, then, as discussed below, we can encounter numerical difficulties when generating the incomplete LU decomposition of the Jacobian matrix; this LU decomposition is needed when applying the iterative solver orthomin to solve the matrix problems involved in the Newton-Raphson procedure. As indicated in Table 3.1, we solve for pressure in all cases, i.e., the pressures are always primary variables. Hence, in the Jacobin matrix, the derivative of one of the three finite difference equations at ith gridblock with respect to the pressure of the ith gridblock is always a 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, the first diagonal entry and every third diagonal after that will be given by $df_{1,i}/dp_i$ 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 every individual term involved in $df_{1,i}/dp_i$ is related to oil saturation by either relative permeability or S_o itself, this derivative is equal to zero if oil saturation is equal to 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 $S_o = 0$ on gridblock i as occurs in the aquifer or gas cap, and we use individual component flow equations instead of Eqs. 3.25 through 3.27, the LU decomposition subroutine will terminate because of division by zero.

Eq. 3.23 and Eqs. 3.25 through 3.27 represent a system of N_e equations where

$$
N_e = 3N + N_w. \tag{3.31}
$$

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

$$q_{o,i,j,k}^{n+1} = WI_{i,j,k} \left(\frac{k_{ro}}{B_o \mu_o} \right)_{i,j,k}^{n+1} (p_{i,j,k}^{n+1} - p_{wf,l,k}^{n+1}), \tag{3.32}$$

$$q_{w,i,j,k}^{n+1} = WI_{i,j,k} \left(\frac{k_{rw}}{B_w \mu_w} \right)_{i,j,k}^{n+1} (p_{i,j,k}^{n+1} - p_{wf,l,k}^{n+1}), \tag{3.33}$$

and

$$\begin{aligned}
q_{g,i,j,k}^{n+1} &= WI_{i,j,k} \left(\frac{k_{rg}}{B_g \mu_g} \right)_{i,j,k}^{n+1} (p_{i,j,k}^{n+1} - p_{wf,l,k}^{n+1}) + R_{so,i,j,k}^{n+1} q_{o,i,j,k}^{n+1} \\
&= WI_{i,j,k} \left(\frac{k_{rg}}{B_g \mu_g} + R_s \frac{k_{ro}}{B_o \mu_o} \right)_{i,j,k}^{n+1} (p_{i,j,k}^{n+1} - p_{wf,l,k}^{n+1}).
\end{aligned} \tag{3.34}$$

The rates $q_{o,i,j,k}^{n+1}$ and $q_{w,i,j,k}^{n+1}$ are in units of STB/Day, and $q_{g,l,k}^{n+1}$ 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

$$WI_{i,j,k} = \frac{0.00708 \triangle z_k \sqrt{k_{x,i,j,k} k_{y,i,j,k}}}{\ln(r_{o,l,k} / r_{w,l,k}) + s_{l,k}}, \tag{3.35}$$

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

$$r_{o,l,k} = \frac{0.28073 \triangle x_i \sqrt{1 + \frac{k_{x,i,j,k}}{k_{y,i,j,k}} \left(\frac{\triangle y_j}{\triangle x_i} \right)^2}}{1 + \sqrt{k_{x,i,j,k} / k_{y,i,j,k}}}. \tag{3.36}$$

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

$$f^{n+1} = f(y^{n+1}, y^n, m) = \begin{bmatrix} f_{1,1}^{n+1} \\ f_{o,1}^{n+1} \\ f_{g,1}^{n+1} \\ f_{1,2}^{n+1} \\ \vdots \\ f_{g,N}^{n+1} \\ f_{wf,1}^{n+1} \\ \vdots \\ f_{wf,N_w}^{n+1} \end{bmatrix} = 0, \tag{3.37}$$

24

where
$$m = [m_1, m_2, \cdots , m_{N_m}]^T, \tag{3.38}$$

and

$$y^{n+1} = [p_1^{n+1}, S_{o,1}^{n+1}, x_1^{n+1}, p_2^{n+1}, \cdots , p_i^{n+1}, S_{o,i}^{n+1}, x_i^{n+1}, \cdots , x_N^{n+1}, p_{wf,1}^{n+1}, \cdots , p_{wf,N_w}^{n+1}]^T, \tag{3.39}$$

where

$$x_i^{n+1} = \begin{cases} S_{g,i}^{n+1} & \text{for} \quad S_{g,i} > 0 \\ R_{s,i}^{n+1} & \text{for} \quad S_{g,i} = 0 \end{cases} \tag{3.40}$$

Eq. 3.37 is solved by the Newton-Raphson method (Aziz and Settari, 1979) which can be written as

$$J^{n+1,k} \delta y^{n+1,k+1} = -f^{n+1,k} \tag{3.41}$$

$$y^{n+1,k+1} = y^{n+1,k} + \delta y^{n+1,k+1}, \tag{3.42}$$

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

$$J^{n+1,k} = \left[\nabla_{y^{n+1}} (f^{n+1})^T \right]^T_{y^{n+1,k}}, \tag{3.43}$$

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.,

$$y^{n+1,0} = y^n. \tag{3.44}$$

3.2.2 Adjoint Equations.

We define a general scalar function by

$$\beta = \beta(y^1, ..., y^L, m), \tag{3.45}$$

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. 3.37 to the function β:

$$J = \beta + \sum_{n=0}^{L} (\lambda^{n+1})^T f^{n+1}, \tag{3.46}$$

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

$$\lambda^{n+1} = \left[\lambda_1^{n+1}, \quad \lambda_2^{n+1}, \quad \ldots, \quad \lambda_{N_e}^{n+1} \right]^T. \tag{3.47}$$

25

Taking the total differential of Eq. 3.46, and doing some simple rearranging gives

$$dJ = d\beta + \sum_{n=0}^{L} \left\{ (\lambda^{n+1})^T [\nabla_{y^{n+1}}(f^{n+1})^T]^T dy^{n+1} + [\nabla_m(f^{n+1})^T]^T dm \right\}$$

$$+ \sum_{n=0}^{L} (\lambda^{n+1})^T [\nabla_{y^n}(f^{n+1})^T]^T dy^n$$

$$= d\beta + BT + \sum_{n=1}^{L} \{ [(\lambda^n)^T [\nabla_{y^n}(f^n)^T]^T$$

$$+ (\lambda^{n+1})^T [\nabla_{y^n}(f^{n+1})^T]^T] dy^n + (\lambda^n)^T [\nabla_m(f^n)^T]^T dm \}, \tag{3.48}$$

where

$$BT = (\lambda^{L+1})^T \left\{ [\nabla_{y^{L+1}}(f^{L+1})^T]^T dy^{L+1} + [\nabla_m(f^{L+1})^T]^T dm \right\} + (\lambda^1)^T [\nabla_{y^0}(f^1)^T]^T dy^0. \tag{3.49}$$

The total differential of β can be written as

$$d\beta = \sum_{n=1}^{L} [\nabla_{y^n}\beta]^T dy^n + [\nabla_m\beta]^T dm. \tag{3.50}$$

The initial conditions are fixed, so

$$dy^0 = 0. \tag{3.51}$$

Choosing

$$\lambda^{L+1} = 0, \tag{3.52}$$

it follows that $BT = 0$. Using this result and Eq. 3.50 in Eq. 3.48 and rearranging the resulting equation gives

$$dJ = \sum_{n=1}^{L} \left[\left\{ (\lambda^n)^T [\nabla_{y^n}(f^n)^T]^T + (\lambda^{n+1})^T [\nabla_{y^n}(f^{n+1})^T]^T \right. \right.$$

$$\left. + [\nabla_{y^n}\beta]^T \right\} dy^n \right] + \left\{ [\nabla_m\beta]^T + \sum_{n=1}^{N} (\lambda^n)^T [\nabla_m(f^n)^T]^T \right\} dm. \tag{3.53}$$

To obtain the adjoint system, the coefficients multiplying dy^n in Eq. 3.53 are set equal to zero; i.e., we require that the adjoint variables satisfy

$$(\lambda^n)^T [\nabla_{y^n}(f^n)^T]^T + (\lambda^{n+1})^T [\nabla_{y^n}(f^{n+1})^T]^T + [\nabla_{y^n}\beta]^T = 0. \tag{3.54}$$

Taking the transpose of Eq. 3.54, gives the adjoint system

$$\boxed{\left[\nabla_{y^n}(f^n)^T\right]\lambda^n = -\left[\nabla_{y^n}(f^{n+1})^T\right]\lambda^{n+1} - \nabla_{y^n}\beta.} \tag{3.55}$$

26

where

$$\nabla_{y^n}[f^n]^T = \begin{bmatrix} \frac{\partial f_{1,1}^n}{\partial p_1^n} & \frac{\partial f_{w,1}^n}{\partial p_1^n} & \cdots & \frac{\partial f_{g,N}^n}{\partial p_1^n} & \frac{\partial f_{wf,1}^n}{\partial p_1^n} & \cdots & \frac{\partial f_{wf,N_w}^n}{\partial p_1^n} \\ \frac{\partial f_{1,1}^n}{\partial S_{w,1}^n} & \frac{\partial f_{w,1}^n}{\partial S_{w,1}^n} & \cdots & \frac{\partial f_{g,N}^n}{\partial S_{w,1}^n} & \frac{\partial f_{wf,1}^n}{\partial S_{w,1}^n} & \cdots & \frac{\partial f_{wf,N_w}^n}{\partial S_{w,1}^n} \\ \frac{\partial f_{1,1}^n}{\partial S_{g,1}^n} & \frac{\partial f_{w,1}^n}{\partial S_{g,1}^n} & \cdots & \frac{\partial f_{g,N}^n}{\partial S_{g,1}^n} & \frac{\partial f_{wf,1}^n}{\partial S_{g,1}^n} & \cdots & \frac{\partial f_{wf,N_w}^n}{\partial S_{g,1}^n} \\ \frac{\partial f_{1,1}^n}{\partial p_2^n} & \frac{\partial f_{w,1}^n}{\partial p_2^n} & \cdots & \frac{\partial f_{g,N}^n}{\partial p_2^n} & \frac{\partial f_{wf,1}^n}{\partial p_2^n} & \cdots & \frac{\partial f_{wf,N_w}^n}{\partial p_2^n} \\ \vdots & \vdots & \cdots & \vdots & \vdots & \cdots & \vdots \\ \frac{\partial f_{1,1}^n}{\partial S_{g,N}^n} & \frac{\partial f_{w,1}^n}{\partial S_{g,N}^n} & \cdots & \frac{\partial f_{g,N}^n}{\partial S_{g,N}^n} & \frac{\partial f_{wf,1}^n}{\partial S_{g,N}^n} & \cdots & \frac{\partial f_{wf,N_w}^n}{\partial S_{g,N}^n} \\ \frac{\partial f_{1,1}^n}{\partial p_{wf,1}^n} & \frac{\partial f_{w,1}^n}{\partial p_{wf,1}^n} & \cdots & \frac{\partial f_{g,N}^n}{\partial p_{wf,1}^n} & \frac{\partial f_{wf,1}^n}{\partial p_{wf,1}^n} & \cdots & \frac{\partial f_{wf,N_w}^n}{\partial p_{wf,1}^n} \\ \vdots & \vdots & \cdots & \vdots & \vdots & \cdots & \vdots \\ \frac{\partial f_{1,1}^n}{\partial p_{wf,N_w}^n} & \frac{\partial f_{w,1}^n}{\partial p_{wf,N_w}^n} & \cdots & \frac{\partial f_{g,N}^n}{\partial p_{wf,N_w}^n} & \frac{\partial f_{wf,1}^n}{\partial p_{wf,N_w}^n} & \cdots & \frac{\partial f_{wf,N_w}^n}{\partial p_{wf,N_w}^n} \end{bmatrix}, \quad (3.56)$$

where $f_{1,i}$ is given by Eq. 3.25 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}, \quad (3.57)$$

and

$$\nabla_{y^n}\beta = \left[\frac{\partial \beta}{\partial p_1^n}, \frac{\partial \beta}{\partial S_{w,1}^n}, \frac{\partial \beta}{\partial S_{g,1}^n}, \frac{\partial \beta}{\partial p_2^n}, \cdots, \frac{\partial \beta}{\partial S_{g,N}^n}, \frac{\partial \beta}{\partial p_{wf,1}^n}, \cdots, \frac{\partial \beta}{\partial p_{wf,N_w}^n} \right]^T. \quad (3.58)$$

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.,

$$y = [p_1, S_{o,1}, S_{g,1}, p_2, \cdots, p_i, S_{o,i}, S_{g,i}, \cdots, S_{g,N}, p_{wf,1}, \cdots, p_{wf,N_w}], \quad (3.59)$$

whereas in the forward simulator, Eq. 3.39 is used. Our results indicate that this does not affect the accuracy of the adjoint solutions.

Eq. 3.55 with initial condition 3.52 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. 3.55 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. 3.57 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. 3.56) in the adjoint system is simply the transpose of the Jacobian matrix of Eq. 3.43 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. 3.56 and 3.57) 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. 3.55, 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

$$dJ = (\nabla_m J)^T dm. \tag{3.60}$$

By comparing Eq. 3.53 and Eq. 3.60, it follows that the desired sensitivity coefficients for J, or equivalently, β, are given by

$$\boxed{\nabla_m J = \nabla_m \beta + \sum_{n=1}^{L} [\nabla_m (f^n)^T](\lambda^n),} \tag{3.61}$$

28

where

$$\nabla_m[f^n]^T = \begin{bmatrix} \dfrac{\partial f_{1,1}^n}{\partial m_1} & \dfrac{\partial f_{w,1}^n}{\partial m_1} & \dfrac{\partial f_{g,1}^n}{\partial m_1} & \dfrac{\partial f_{1,2}^n}{\partial m_1} & \cdots & \dfrac{\partial f_{g,N}^n}{\partial m_1} & \dfrac{\partial f_{wf,1}^n}{\partial m_1} & \cdots & \dfrac{\partial f_{wf,N_w}^n}{\partial m_1} \\[2mm] \dfrac{\partial f_{1,1}^n}{\partial m_2} & \dfrac{\partial f_{w,1}^n}{\partial m_2} & \dfrac{\partial f_{g,1}^n}{\partial m_2} & \dfrac{\partial f_{1,2}^n}{\partial m_2} & \cdots & \dfrac{\partial f_{g,N}^n}{\partial m_2} & \dfrac{\partial f_{wf,1}^n}{\partial m_2} & \cdots & \dfrac{\partial f_{wf,N_w}^n}{\partial m_2} \\[2mm] \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\[2mm] \dfrac{\partial f_{1,1}^n}{\partial m_{N_m}} & \dfrac{\partial f_{w,1}^n}{\partial m_{N_m}} & \dfrac{\partial f_{g,1}^n}{\partial m_{N_m}} & \dfrac{\partial f_{1,2}^n}{\partial m_{N_m}} & \cdots & \dfrac{\partial f_{g,N}^n}{\partial m_{N_m}} & \dfrac{\partial f_{wf,1}^n}{\partial m_{N_m}} & \cdots & \dfrac{\partial f_{wf,N_w}^n}{\partial m_{N_m}} \end{bmatrix}, \quad (3.62)$$

and

$$\nabla_m\beta = \left[\frac{\partial\beta}{\partial m_1}, \frac{\partial\beta}{\partial m_2}, \cdots, \frac{\partial\beta}{\partial m_{N_m}}\right]^T. \tag{3.63}$$

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. 3.61, 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. 3.61.

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. 3.55 once and substitute the resulting adjoint solutions to Eq. 3.61 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. 3.55 to obtain the adjoint variable λ, and then use Eq. 3.61 to calculate sensitivity coefficients. If we consider permeabilities (k_x, k_y and k_z) and porosities (ϕ) in each individual gridblock, i.e.,

$$m_{k_x} = k_x = [\ k_{x,1}, \quad k_{x,2}, \quad \cdots, \quad k_{x,N}\ ]^T, \tag{3.64}$$

$$m_{k_y} = k_y = [\ k_{y,1}, \quad k_{y,2}, \quad \cdots, \quad k_{y,N}\ ]^T, \tag{3.65}$$

$$m_{k_z} = k_z = [\ k_{z,1}, \quad k_{z,2}, \quad \cdots, \quad k_{z,N}\ ]^T, \tag{3.66}$$

and

$$m_\phi = \phi = [\ \phi_1, \quad \phi_2, \quad \cdots, \quad \phi_N\ ]^T, \tag{3.67}$$

then from Eq. 3.61, the equations to calculate the derivatives with respect to k_x, k_y, k_z and ϕ are given by

$$\nabla_{k_x} J = \nabla_{k_x}\beta + \sum_{n=1}^{L}[\nabla_{k_x}(f^n)^T](\lambda^n), \tag{3.68}$$

$$\nabla_{k_y} J = \nabla_{k_y}\beta + \sum_{n=1}^{L}[\nabla_{k_y}(f^n)^T](\lambda^n), \tag{3.69}$$

$$\nabla_{k_z} J = \nabla_{k_z} \beta + \sum_{n=1}^{L} [\nabla_{k_z} (f^n)^T](\lambda^n), \qquad (3.70)$$

and

$$\nabla_\phi J = \nabla_\phi \beta + \sum_{n=1}^{L} [\nabla_\phi (f^n)^T](\lambda^n), \qquad (3.71)$$

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.,

$$\beta = O_d(m) = \frac{1}{2}(g(m) - d_{\text{obs}})^T C_D^{-1}(g(m) - d_{\text{obs}}), \qquad (3.72)$$

or in the case of stochastic simulation of m,

$$\beta = O_d(m) = \frac{1}{2}(g(m) - d_{\text{uc}})^T C_D^{-1}(g(m) - d_{\text{uc}}). \qquad (3.73)$$

Thus, we have

$$\begin{aligned}
\nabla_{y^n} \beta &= \nabla_{y^n} \{\frac{1}{2}(g(m) - d_{\text{obs}})^T C_D^{-1}(g(m) - d_{\text{obs}})\} \\
&= \Big[\nabla_{y^n}(g(m) - d_{\text{obs}})^T\Big] C_D^{-1}(g(m) - d_{\text{obs}}) \\
&= \nabla_{y^n}[g(m)]^T C_D^{-1}(g(m) - d_{\text{obs}}).
\end{aligned} \qquad (3.74)$$

In the case of β given by Eq. 3.73, the d_{obs} in Eq. 3.74 should be replaced by d_{uc}. The matrix $\nabla_{y^n}[g(m)]^T$ is an $N_e \times N_d$ matrix and defined as

$$\nabla_{y^n}[g(m)]^T = \begin{bmatrix}
\frac{\partial g_1}{\partial p_1^n} & \frac{\partial g_2}{\partial p_1^n} & \cdots & \frac{\partial g_{N_d}}{\partial p_1^n} \\
\frac{\partial g_1}{\partial S_{w,1}^n} & \frac{\partial g_2}{\partial S_{w,1}^n} & \cdots & \frac{\partial g_{N_d}}{\partial S_{w,1}^n} \\
\frac{\partial g_1}{\partial S_{g,1}^n} & \frac{\partial g_2}{\partial S_{g,1}^n} & \cdots & \frac{\partial g_{N_d}}{\partial S_{g,N}^n} \\
\frac{\partial g_1}{\partial p_2^n} & \frac{\partial g_2}{\partial p_2^n} & \cdots & \frac{\partial g_{N_d}}{\partial p_2^n} \\
\vdots & \vdots & \cdots & \vdots \\
\frac{\partial g_1}{\partial S_{g,N}^n} & \frac{\partial g_2}{\partial S_{g,N}^n} & \cdots & \frac{\partial g_{N_d}}{\partial S_{g,N}^n} \\
\frac{\partial g_1}{\partial p_{wf,1}^n} & \frac{\partial g_2}{\partial p_{wf,1}^n} & \cdots & \frac{\partial g_{N_d}}{\partial p_{wf,1}^n} \\
\vdots & \vdots & \cdots & \vdots \\
\frac{\partial g_1}{\partial p_{wf,N_w}^n} & \frac{\partial g_2}{\partial p_{wf,N_w}^n} & \cdots & \frac{\partial g_{N_d}}{\partial p_{wf,N_w}^n}
\end{bmatrix}. \qquad (3.75)$$

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 $C_D^{-1}(g(m) - d_{\text{obs}})$ by this matrix to obtain $\nabla_{y^n}\beta$. Once we have $\nabla_{y^n}\beta$, we can apply Eq. 3.55 to compute the adjoint variables.

To apply Eq. 3.61 to compute the derivatives, we need to evaluate $\nabla_m\beta$ first. The vector $\nabla_m\beta$ is given by

$$
\begin{aligned}
\nabla_m\beta &= \nabla_m O_d(m) \\
&= \nabla_m\Big\{\frac{1}{2}(g(m) - d_{\text{obs}})^T C_D^{-1}(g(m) - d_{\text{obs}})\Big\} \\
&= \big[\nabla_m(g(m) - d_{\text{obs}})^T\big] C_D^{-1}(g(m) - d_{\text{obs}}) \\
&= \nabla_m[g(m)]^T C_D^{-1}(g(m) - d_{\text{obs}}).
\end{aligned}
\tag{3.76}
$$

In the case of β given by Eq. 3.73, the d_{obs} in Eq. 3.76 should be replaced by d_{uc}. The matrix $\nabla_m[g(m)]^T$ is an $N_m \times N_d$ matrix and defined as

$$
\nabla_m[g(m)]^T = \begin{bmatrix}
\frac{\partial g_1}{\partial m_1} & \frac{\partial g_2}{\partial m_1} & \cdots & \frac{\partial g_{N_d}}{\partial m_1} \\
\frac{\partial g_1}{\partial m_2} & \frac{\partial g_2}{\partial m_2} & \cdots & \frac{\partial g_{N_d}}{\partial m_2} \\
\vdots & \vdots & \cdots & \vdots \\
\frac{\partial g_1}{\partial m_{N_m}} & \frac{\partial g_2}{\partial m_{N_m}} & \cdots & \frac{\partial g_{N_d}}{\partial m_{N_m}}
\end{bmatrix}.
\tag{3.77}
$$

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 $\nabla_m[g(m)]^T$ can be found in Li (2001). After we compute $\nabla_m\beta$, we can use Eq. 3.61 to compute the derivatives of the objective function with respect to model parameters, i.e., the gradient of the objective function.

3.3 Application of Conjugate Gradient and Quasi-Newton Methods

Here, we explore the conjugate gradient and quasi-Newton (variable metric) methods. The advantage of these methods is that they require only the computation of the gradient of the objective function with respect to the model parameters. The sensitivity coefficient matrix G is not needed. Our results suggest that these methods can lead to considerable reductions in computer time and memory required for large scale history matching problems.

3.3.1 Nonlinear Conjugate Gradient Method

Nonlinear conjugate gradient method which is usually used to minimize non-quadratic function can "be evolved" from 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

$$d_{k+1} = -M_k^{-1}g_k + \beta_k d_k, \tag{3.78}$$

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

$$\beta_k = \frac{r_{k+1}^T(M_{k+1}^{-1}r_{k+1} - M_k^{-1}r_k)}{r_k^T M_k^{-1} r_k}, \tag{3.79}$$

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. 3.78 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

$$M_k^{-1}H_k \approx I. \tag{3.80}$$

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

$$H_k = C_M^{-1} + G_k^T C_D^{-1} G_k, \tag{3.81}$$

An optional preconditioner for the conjugate gradient method would be

$$M_k = H_k, \tag{3.82}$$

but the conjugate gradient method requires solving the matrix problem

$$M_k \tilde{d}_k = -g_k, \tag{3.83}$$

to form search direction d_{k+1} using Eq. 3.78. If $M_k = H_k$, Eq. 3.83 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. 3.83 becomes

$$\tilde{d}_k = -C_M g_k, \tag{3.84}$$

and the calculation of $\tilde{d}_k$ which is the first term in Eq. 3.78 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. 3.10 or Eq. 3.14). 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. 3.83. 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. Quasi-Newton method will be discussed in the next section. 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. 3.78 whereas it is given by Eq. 3.87 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 have the property that the inverse Hessian approximation will be equal to the true inverse Hessian at the Nth iteration for a N-dimensional quadratic function given that the line search is exact; see Oren and Luenberger (1974) and Oren (1974b).

3.3.2 Quasi-Newton Methods

The search direction in the Newton's method can be written as

$$d_{k+1} = -H_k^{-1} g_k, \tag{3.85}$$

where H_k and g_k, respectively, denote the second derivative (Hessian matrix) and the first derivative (gradient) of the objective function evaluated at m_k and k is the iteration index. With $O(m)$ given by either Eq. 3.10 or Eq. 3.14, the Gauss-Newton Hessian matrix is estimated by

$$H_k = C_M^{-1} + G_k^T C_D^{-1} G_k, \tag{3.86}$$

where G_k is the sensitivity matrix evaluated at m_k. As noted before, if both the number of model parameters and the number of data are large, the evaluation of G_k is computationally expensive. 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. With this Hessian inverse approximation matrix, the search direction can be written as

$$d_{k+1} = -\tilde{H}_k^{-1} g_k. \tag{3.87}$$

Because the matrix $\tilde{H}_k^{-1}$ takes the place of H_k^{-1} in Eq. 3.85, the method with search direction given by Eq. 3.87 is called a quasi-Newton method. This method is also called a variable metric method.

In a quasi-Newton method, the key issue is how to generate the approximation to the inverse Hessian matrix. Different quasi-Newton methods use different formulas to calculate $\tilde{H}_{k+1}^{-1}$ from $\tilde{H}_k^{-1}$. All updating formulas satisfy the quasi-Newton condition given by

$$\tilde{H}_{k+1}^{-1} y_k = s_k, \tag{3.88}$$

where

$$y_k = g_{k+1} - g_k, \tag{3.89}$$

and

$$s_k = m_{k+1} - m_k. \tag{3.90}$$

Various possible updating formulas honor this quasi-Newton condition. The Broyden family equation is given by

$$\tilde{H}_{k+1}^{-1} = \tilde{H}_k^{-1} + \frac{s_k s_k^T}{s_k^T y_k} - \frac{\tilde{H}_k^{-1} y_k y_k^T \tilde{H}_k^{-1}}{y_k^T \tilde{H}_k^{-1} y_k} + \theta_k v_k v_k^T, \tag{3.91}$$

where $\theta_k \in [0, 1]$ and

$$v_k = (y_k^T \tilde{H}_k^{-1} y_k)^{1/2} \left(\frac{s_k}{s_k^T y_k} - \frac{\tilde{H}_k^{-1} y_k}{y_k^T \tilde{H}_k^{-1} y_k} \right). \tag{3.92}$$

Given that the line search is exact and the initial Hessian inverse approximation is real symmetric positive definite, the Hessian inverse approximation generated by Eq. 3.91 is guaranteed to be symmetric positive definite; see details in Zhang (2002). In our procedure, we use the Broyden-Fletcher-Goldfarb-Shanno (BFGS) correction equation proposed by Broyden (1970), Fletcher (1970), Goldfarb (1970) and Shanno (1970) independently, which is a special case of Broyden family obtained by setting $\theta_k = 1$ in Eq. 3.91. The BFGS update equation is given by

$$\tilde{H}_{k+1}^{-1} = \tilde{H}_k^{-1} + \frac{s_k s_k^T}{s_k^T y_k} - \frac{\tilde{H}_k^{-1} y_k y_k^T \tilde{H}_k^{-1}}{y_k^T \tilde{H}_k^{-1} y_k} + v_k v_k^T. \tag{3.93}$$

The limited memory BFGS (LBFGS), which uses a limited number of previous vectors (y_k's and s_k's) to construct the inverse Hessian approximation at each iteration, is an appropriate method for large scale problems where it is not feasible to explicitly store and compute the full matrix $\tilde{H}_k^{-1}$. In our work, the algorithm proposed by Nocedal (1980) was implemented and applied. In order to derive the limited memory BFGS, the normal BFGS formula Eq. 3.93 can be written as

$$\tilde{H}_{k+1}^{-1} = V_k^T \tilde{H}_k^{-1} V_k + \rho_k s_k s_k^T, \tag{3.94}$$

where $\rho_k = 1/y_k^T s_k$ and $V_k = I - \rho_k y_k s_k^T$. Nocedal (1980) suggested a procedure where only the L previous vectors are used when constructing the new $\tilde{H}_{k+1}^{-1}$. When $k < L$, the update equation is still given by Eq. 3.94 which can be rewritten as

$$\begin{aligned} \tilde{H}_{k+1}^{-1} =& V_k^T V_{k-1}^T \cdots V_0^T \tilde{H}_0^{-1} V_0 \cdots V_{k-1} V_k \\ &+ V_k^T \cdots V_1^T \rho_0 s_0 s_0^T V_1 \cdots V_k \\ &\vdots \\ &+ V_k^T \rho_{k-1} s_{k-1} s_{k-1}^T V_k \\ &+ \rho_k s_k s_k^T. \end{aligned} \tag{3.95}$$

For $k + 1 > L$ the update equation is

$$\begin{aligned} \tilde{H}_{k+1}^{-1} =& V_k^T V_{k-1}^T \cdots V_{k-L+1}^T \tilde{H}_0^{-1} V_{k-L+1} \cdots V_{k-1} V_k \\ &+ V_k^T \cdots V_{k-L+2}^T \rho_{k-L+1} s_{k-L+1} s_{k-L+1}^T V_1 \cdots V_k \\ &\vdots \\ &+ V_k^T \rho_{k-1} s_{k-1} s_{k-1}^T V_k \\ &+ \rho_k s_k s_k^T. \end{aligned} \tag{3.96}$$

Unless the dimension of $\tilde{H}_k^{-1}$ is small, direct application of Eqs. 3.95 and 3.96, which involve matrix products, is inefficient. Instead, we form the product $\tilde{H}_k^{-1} g_k$, which is used to construct the search direction, directly by using the algorithm proposed by Nocedal (1980). The calculation of $\tilde{H}_k^{-1} g_k$ only involves vector products instead of matrix products. Because only the L most recent vectors from the set of s_k and y_k are used to construct $\tilde{H}_{k+1}^{-1}$, this algorithm is called the limited memory BFGS method.

The BFGS or LBFGS algorithm we used to minimize $O(m)$ (Eq. 3.10 or Eq. 3.14) is given below.

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 $d_k = -\tilde{H}_k^{-1} g_k$ 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 later.

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 (discussed later) 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. 3.93 or Eqs. 3.95 and 3.96 to update the inverse Hessian approximation $\tilde{H}_{k+1}^{-1}$. Set $k = k + 1$ and then go to step 2.

Scaling

The scaling is obtained by multiplying the old $\tilde{H}_k^{-1}$ by a factor γ_k and then substituting $\gamma_k \tilde{H}_k^{-1}$ instead of $\tilde{H}_k^{-1}$ itself into the update equation (e.g., Eq. 3.91) to calculate $\tilde{H}_{k+1}^{-1}$; see Oren (1973), Oren and Luenberger (1974), Oren (1974b) and Shanno (1970). If we do so, Eq. 3.91 becomes

$$\tilde{H}_{k+1}^{-1} = \gamma_k \tilde{H}_k^{-1} + \frac{s_k s_k^T}{s_k^T y_k} - \frac{\gamma_k \tilde{H}_k^{-1} y_k y_k^T \gamma_k \tilde{H}_k^{-1}}{y_k^T \gamma_k \tilde{H}_k^{-1} y_k} + \theta_k \gamma_k v_k v_k^T, \tag{3.97}$$

and this equation can be further simplified to

$$\tilde{H}_{k+1}^{-1} = \left(\tilde{H}_k^{-1} - \frac{\tilde{H}_k^{-1} y_k y_k^T \tilde{H}_k^{-1}}{y_k^T \tilde{H}_k^{-1} y_k} + \theta_k v_k v_k^T \right) \gamma_k + \frac{s_k s_k^T}{s_k^T y_k}, \tag{3.98}$$

where v_k is given by Eq. 3.92 and is the same one used in Eqs. 3.91 and 3.93.

For BFGS and LBFGS, scaling can have a significant effect on the rate of convergence. The self-scaling variable metric (SSVM) method developed by Oren and Luenberger (1974) and Oren (1974b) is motivated by the desire to choose a scalar 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.

Let

$$\tau_k = \frac{s_k^T \tilde{H}_k s_k}{s_k^T y_k}, \tag{3.99}$$

and

$$\sigma_k = \frac{s_k^T y_k}{y_k^T \tilde{H}_k^{-1} y_k}. \tag{3.100}$$

By applying the fact that

$$s_k = \alpha_k d_k = -\alpha_k \tilde{H}_k^{-1} g_k, \tag{3.101}$$

and using the fact that $\tilde{H}_k^{-1}$ is real symmetric, Eq. 3.99 can be rewritten as

$$\tau_k = \frac{s_k^T \tilde{H}_k s_k}{s_k^T y_k} \tag{3.102}$$

$$= -\alpha_k \frac{g_k^T s_k}{s_k^T y_k} \tag{3.103}$$

$$= \frac{s_k^T g_k}{g_k^T \tilde{H}_k^{-1} y_k}. \tag{3.104}$$

37

In general the motivation for using the last two formulae to calculate τ_k is to avoid calculating the inverse of $\tilde{H}_k^{-1}$.

There are many options we can choose to perform scaling. Oren and Spedicato (1976) proposed an optimal condition which minimizes the upper bound of the condition number of $\tilde{H}_{k+1}^{-1}$ by proper selection of θ_k and γ_k. This condition is given by

$$\theta_k = \frac{a_k b_k - a_k^2 \gamma_k}{b_k c_k \gamma_k - a_k^2 \gamma_k} \tag{3.105}$$

where θ_k is the parameter used in Broyden family update equation, Eq. 3.91, $a_k = s_k^T y_k$, $b_k = s_k^T \tilde{H}_k s_k$ and $c_k = y_k^T \tilde{H}_k^{-1} y_k$. With these definitions, there is still freedom to select scaling factors. In the same paper, Oren proposed four similar switching rules based on his earlier published paper (Oren (1974a)). One of the switching rules for choosing γ_k and θ_k in Eq. 3.98 is

if $\tau_k \leq 1$, choose $\gamma_k = \tau_k$ and $\theta_k = 0$;

if $\sigma_k \geq 1$, choose $\gamma_k = \sigma_k$ and $\theta_k = 1$;

if $\sigma_k \leq 1 \leq \tau_k$, choose $\gamma_k = 1$ and $\theta_k = \frac{1 - \sigma_k}{\tau_k - \sigma_k}$.

Some examples (see Zhang et al. (2001b)) show that when we use $\theta = \theta_k$ not equal to 1 in Eq. 3.98, the convergence rate is almost always slower than the case where we use $\theta_k = 1$. Therefore, we consider only the case where $\theta_k = 1$; this choice corresponds to the BFGS algorithm. Setting $\theta_k = 1$ in Eq. 3.105 and solving for $\gamma = \gamma_k$, we obtain the optimal scaling factor for BFGS which is given by

$$\gamma_k = \frac{a_k}{c_k} = \frac{s_k^T y_k}{y_k^T \tilde{H}_k^{-1} y_k} = \sigma_k, \tag{3.106}$$

where σ_k is given by Eq. 3.100. Shanno and Phua (1978) and Yang and Watson (1988) use this scaling factor and only scale the initial matrix, $\tilde{H}_0^{-1}$, in their work. In our implementation, we modify Oren's switching rule to

$$\begin{cases} \gamma_k = \tau_k, & \text{if} \quad \tau_k \leq 1; \\ \gamma_k = \sigma_k, & \text{otherwise} \end{cases} \tag{3.107}$$

and always use $\theta_k = 1$. Note that using switching rule Eq. 3.107 for choosing the scaling factor still guarantees the condition number of matrix $R_k = H_k^{1/2} \tilde{H}_k^{-1} H_k^{1/2}$ monotonously decreases at least for quadratic functions.

The scaled version BFGS algorithm is similar to the standard BFGS algorithm given previously. The only difference is in step 8. For the scaled BFGS algorithm, step 8 is replaced by

Step 8 Calculate $s_k = m_{k+1} - m_k = \alpha_k d_k$ and $y_k = g_{k+1} - g_k$. Calculate τ_k by Eq. 3.104 and determine whether τ_k is less than one. If it is, then set $\gamma_k = \tau_k$. Otherwise, calculate σ_k using Eq. 3.100 and set $\gamma_k = \sigma_k$. Apply Eq. 3.98 to update the inverse Hessian approximation $\tilde{H}_{k+1}^{-1}$. Set $k = k + 1$ and go to step 2.

This step 8 is given for the case where the inverse Hessian approximation is scaled at each iteration. For the case where only initial scaling is done, set $\gamma_k = 1$ for $k > 0$ in this step. For the LBFGS algorithm with initial scaling, we just replace $\tilde{H}_0^{-1}$ in Eqs. 3.95 and 3.96 by $\gamma_0 \tilde{H}_0^{-1}$ in computing $\tilde{H}_1^{-1}$ and use $\gamma_k = 1$ at all subsequent iterations. For the LBFGS with all scaling, we replace $\tilde{H}_0^{-1}$ in Eqs. 3.95 and 3.96 by $\gamma_k \tilde{H}_0^{-1}$ in computing $\tilde{H}_{k+1}^{-1}$. The efficient LBFGS method given by Nocedal (1980) avoid 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. Based on these considerations, we tried implementing Eq. 3.100 with $\tilde{H}_k^{-1}$ replaced by $\tilde{H}_0^{-1}$, then we have

$$\tilde{\sigma}_k = \frac{s_k^T y_k}{y_k^T \tilde{H}_0^{-1} y_k}. \tag{3.108}$$

The $\tilde{H}_k^{-1}$ and $\tilde{H}_k$ in Eqs. 3.102 and 3.104 are replaced by $\tilde{H}_0^{-1}$ and $\tilde{H}_0$ respectively. The resulting three equations for τ_k are no longer equivalent except at the first iteration. By using $\tilde{H}_0^{-1}$ in place of $\tilde{H}_k^{-1}$ and $\tilde{H}_0$ in place of $\tilde{H}_k$ in Eqs. 3.102 and 3.104, we obtain

$$\tilde{\tau}_{1k} = \frac{s_k^T \tilde{H}_0 s_k}{s_k^T y_k} \tag{3.109}$$

$$\tilde{\tau}_{3k} = \frac{s_k^T g_k}{g_k^T \tilde{H}_0^{-1} y_k} \tag{3.110}$$

respectively. However, Eq. 3.103 can be applied exactly in all cases. To simplify the notation, we let

$$\tilde{\tau}_{2k} \equiv \tau_k = -\alpha_k \frac{g_k^T s_k}{s_k^T y_k}. \tag{3.111}$$

More details about the scaling schemes we used and how they affect the convergence are given in the example sections.

3.3.3 Convergence Criteria

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

1.
$$\frac{|\, O_{k+1} - O_k \,|}{O_k + 10^{-14}} < \varepsilon_1 \qquad\qquad (3.112)$$

and

$$\frac{\|\, m_{k+1} - m_k \,\|_2}{\|\, m_k \,\|_2 + 10^{-14}} < \varepsilon_2 \qquad\qquad (3.113)$$

where k denotes the iteration index and $\|\cdot\|_2$ denotes the l_2 norm of a vector. Both conditions must be satisfied to terminate the iteration. If we use only Eq. 3.112 as the convergence criterion, the algorithm may converge prematurely especially when the objective function decreases very slowly at the early iterations. Because at the early iteration, the objective function is relatively big such that Eq. 3.112 becomes easier to be satisfied.

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 100. Note that reaching the maximum number of iterations does not imply that the algorithm has converged.

3.3.4 Line Search

In our implementation of conjugate gradient and quasi-Newton methods, the line search is performed using one iteration of the Newton-Raphson method followed by a quadratic fit if necessary. We do not do an exact line search, but terminate the line search when the Wolfe conditions are satisfied; see, for example, Fletcher (1987). The Wolfe conditions are used to ensure that step sizes are not too small and that the reduction in the objective function is not negligible. In addition, the Wolfe conditions are side conditions for the exact line search; see Kolda et al. (1998). At each iteration, we perform one Newton-Raphson iteration to find a step size. Then we check whether this step satisfies the Wolfe conditions. If it does, we accept this step. Otherwise we find an optimum step size by fitting a quadratic, as discussed in the next section, and then check whether the new step satisfies the Wolfe conditions. If it does, we accept this new step. Otherwise, we check whether the objective function increases or decreases. If it increases, we cut the step size by a factor of 10. If it decreases, we accept this new step size no matter whether the Wolfe conditions are satisfied or not. Our experience shows that for most of the iterations, the step size generated by one Newton-Raphson iteration satisfies the Wolfe conditions and virtually all the step sizes satisfy the

Wolfe conditions after the quadratic fit. One may argue that we should perform a sequence of quadratic fits or a sequence of cubic fits after one quadratic fit instead of cutting the step size after one quadratic fit. Our limited experience shows that the Wolfe conditions may never be satisfied during the sequence of quadratic or cubic fits. If this situation happens, then we are in a "dead loop" and the iteration never terminates. Because of this, we implemented the simple scheme of reducing the step size by a factor of ten whenever the situation arise that the Wolfe conditions are not satisfied after the quadratic fit. Sometimes, however, this procedure leads to a false convergence. Thus, whenever we obtain convergence as a result of reducing the step size by a factor of 10, we check the objective function value. If the objective function value is still so big, then we restart the algorithm manually.

A line search is used to find the step size α at the kth iteration such that

$$f(\alpha) = O(m_k + \alpha d_k), \tag{3.114}$$

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.,

$$h(\alpha) \equiv f'(\alpha) = \frac{dO(m_k + \alpha d_k)}{d\alpha} = \left(\nabla O(m_k + \alpha d_k)\right)^T d_k = 0. \tag{3.115}$$

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

$$\alpha_{j+1} = \alpha_j - \frac{h(\alpha_j)}{h'(\alpha_j)}, \tag{3.116}$$

where j denotes the index of the Newton-Raphson iteration and the first derivative of h can be evaluated by

$$h'(\alpha) = \frac{dh(\alpha)}{d\alpha} = d_k^T \nabla \left[\left(\nabla O(m_k + \alpha d_k)\right)^T \right] d_k = d_k^T H(m_k + \alpha d_k) d_k. \tag{3.117}$$

In an exact line search, the Newton-Raphson iteration is stopped when a convergence criterion is satisfied. The exact line search is very expensive due to the evaluation of the term $d_k^T H(m_k + \alpha d_k) d_k$ which requires at least one simulation run. In our procedure, we use an inexact line search. Specifically, we do only one Newton-Raphson iteration as mentioned previously. To perform one Newton-Raphson iteration, we set $\alpha_0 = 0$ and then Eq. 3.116 gives

$$\alpha_1 = -\frac{\left(\nabla O(m_k)\right)^T d_k}{d_k^T H(m_k) d_k}. \tag{3.118}$$

Eq. 3.118 involves the Hessian matrix which can be approximated. The Hessian for the objective function given by Eq. 3.10 is given by

$$H_k = G_k^T C_D^{-1} G_k + C_M^{-1}. \tag{3.119}$$

So

$$d_k^T H_k d_k = d_k^T (G_k^T C_D^{-1} G_k + C_M^{-1}) d_k$$
$$= d_k^T (G_k^T C_D^{-1} G_k) d_k + d_k^T C_M^{-1} d_k \qquad (3.120)$$
$$= (G_k d_k)^T C_D^{-1} (G_k d_k) + d_k^T C_M^{-1} d_k.$$

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. The method given below was originally implemented by Kalita (2000). The elements of the sensitivity coefficient matrix can be written as

$$G_{i,j} = \frac{\partial g_i}{\partial m_j}, \qquad (3.121)$$

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

$$\left(\frac{dg}{d\alpha}\right)_{\alpha=0} = \left(\frac{dg(m + \alpha d_k)}{d\alpha}\right)_{\alpha=0}. \qquad (3.122)$$

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

$$\left(\frac{dg_i}{d\alpha}\right)_{\alpha=0} = [\nabla g_i(m)]^T u$$
$$= \frac{1}{\parallel d_k \parallel} [\nabla g_i(m)]^T d_k. \qquad (3.123)$$

The ith component of $G d_k$ is given by

$$[G d_k]_i = \sum_{j=1}^{N_m} \frac{\partial g_i}{\partial m_j} d_{k,j}$$
$$= [\nabla g_i(m)]^T d_k, \qquad (3.124)$$

where $d_{k,j}$ denotes the jth component of the vector d_k. Substituting Eq. 3.123 into Eq. 3.124, we obtain

$$G d_k = \parallel d_k \parallel \left(\frac{dg}{d\alpha}\right)_{\alpha=0}$$
$$\approx \parallel d_k \parallel \frac{g(m + \epsilon d_k) - g(m)}{\epsilon \parallel d_k \parallel} \qquad (3.125)$$
$$= \frac{g(m + \epsilon d_k) - g(m)}{\epsilon},$$

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 $G d_k$ 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. 3.120 and then Eq. 3.118 can be applied to calculate the step size. Application of Eq. 3.120 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

$$C_M x = d_k \tag{3.126}$$

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

Fig. 3.1 presents plots of $O(m_k + \alpha d_k)$ (circles) and $f'(\alpha) = [\nabla O(m_k + \alpha d_k)]^T d_k$ (diamonds) versus α. The cross shows $f'(\alpha_1)$ where α_1 was computed by Eq. 3.118. Note $f'(\alpha_1)$ is close to zero and α_1 is close to the point ($\alpha = 3.765$) where $O(m_k + \alpha d_k)$ is minimum. This result illustrates our observation that one iteration of the Newton-Raphson algorithm usually yields a sufficient accurate line search so that the Wolfe conditions are satisfied.
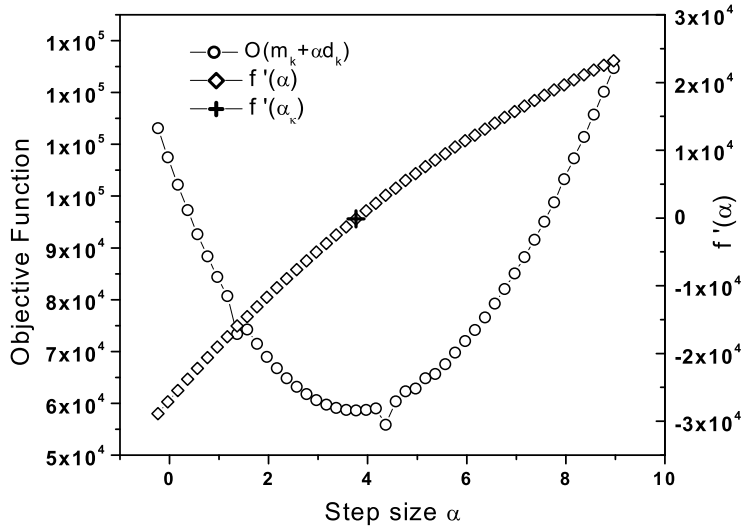
Fig. 3.1: Illustration of Newton-Raphson.

As mentioned above, instead of doing an exact line search, we generate a sequence of iterate approximations to α that minimizes $f(\alpha)$ and terminate the iterations when the Wolfe conditions are satisfied. As discussed below, the Wolfe conditions were formulated to ensure that step sizes are not too small and that there is a non-negligible reduction in the objective function at each iteration. In addition, the Wolfe conditions are the side conditions for quadratic termination for linear problems in practice; see Kolda et al. (1998).

3.3.5 Wolfe Conditions

Following Fletcher (1987), let $\bar{\alpha}_k$ denote the smallest positive value of α for which $O(m_k + \alpha d_k) = O(m_k)$. Negligible reductions in the value of the objective function can occur either if $\alpha_k \to \bar{\alpha}_k$ or $\alpha_k \to 0$; see Fig. 3.2. Goldstein (1965) conditions can be used to avoid the occurrence of these negligible reductions. Again we let $f(\alpha)$ denote $O(m_k + \alpha d_k)$, so $f(0) = O(m_k)$. The Goldstein conditions are

$$f(\alpha) \leq f(0) + \alpha \rho f'(0), \tag{3.127}$$

to exclude the right-hand side extreme of $[0, \bar{\alpha}_k]$, and

$$f(\alpha) \geq f(0) + \alpha(1 - \rho)f'(0) \tag{3.128}$$

to exclude the left-hand side extreme of $[0, \bar{\alpha}_k]$, where $\rho \in (0, \frac{1}{2})$ is a fixed parameter. In our examples, we choose $\rho = 0.0001$. Eq. 3.127 is often also referred to as a Wolfe condition. In Figs. 3.2 and 3.3, the line labeled $\rho f'(0)$ goes through $f(0)$ and its slope is equal to $\rho f'(0)$; the line labeled $f'(0)$ goes through $f(0)$ and has slope equal to $f'(0)$. Eq. 3.127 can be rewritten as

$$\frac{f(\alpha) - f(0)}{\alpha} \leq \rho f'(0). \tag{3.129}$$

From Eq. 3.129, we can see that if the step size satisfies this condition then the line through $(0, f(0))$ and $(\alpha, f(\alpha))$ in Fig. 3.2 must be below the line with slope of $\rho f'(0)$. Similarly, Eq. 3.128 can be rewritten as

$$\frac{f(\alpha) - f(0)}{\alpha} \geq (1 - \rho)f'(0). \tag{3.130}$$

If the step size satisfies this condition then the line through $(0, f(0))$ and $(\alpha, f(\alpha))$ must be above the line labeled $(1 - \rho)f'(0)$ in Fig. 3.3. Hence, if the step size satisfies both Eq. 3.127 and Eq. 3.128, then the line through $(0, f(0))$ and $(\alpha, f(\alpha))$ must fall between the line with slope of $\rho f'(0)$ and the line with slope of $(1 - \rho)f'(0)$.

Applying the fact that $f'(0) = g_k^T d_k$ where $g_k = \nabla O(m_k)$, Eq. 3.127 can be rewritten as

$$f_k - f_{k+1} \geq -\alpha \rho g_k^T d_k. \tag{3.131}$$

When $f(\alpha)$ is non-quadratic, the second Goldstein condition (Eq. 3.128) may exclude the minimum point of $f(\alpha)$; see Fig. 3.3. Wolfe (1969) replaced Eq. 3.128 by a new condition which is given by

$$f'(\alpha) \geq \eta f'(0), \tag{3.132}$$
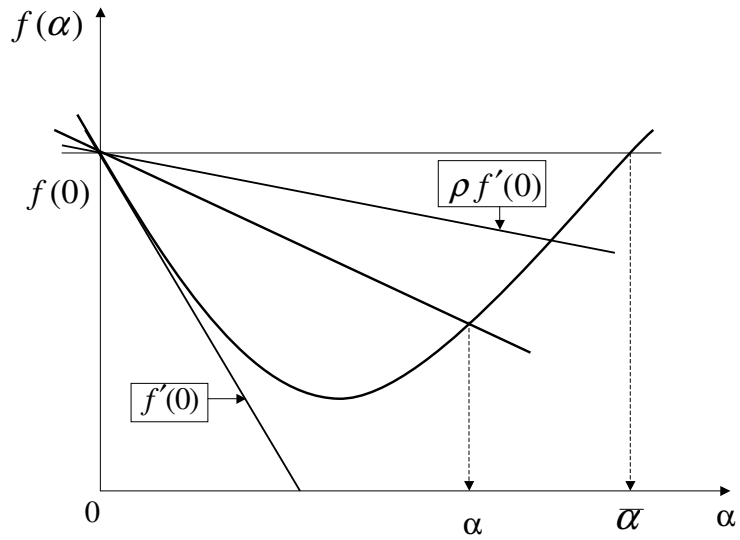
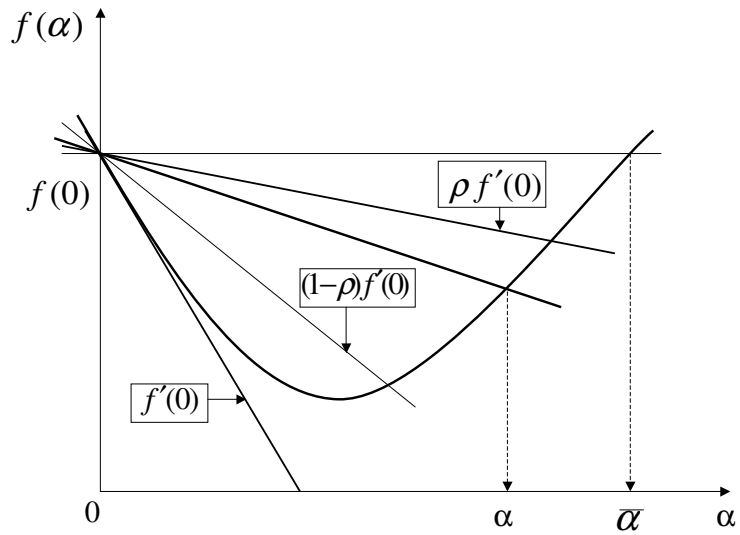Fig. 3.2: Illustration of the Goldstein or the first Wolfe condition.

Fig. 3.3: Illustration of the second Goldstein condition.

where $\eta < 1$. Eq. 3.132 can be rewritten as

$$g_{k+1}^T d_k \geq \eta g_k^T d_k, \tag{3.133}$$

which is called Wolfe's condition. In practice, Eq. 3.133 is often replaced by

$$\mid g_{k+1}^T d_k \mid \leq \eta \mid g_k^T d_k \mid . \tag{3.134}$$

45

which is called the strong Wolfe condition. The reason for using Eq. 3.134 instead of Eq. 3.133 is given below, also see Fletcher (1987) for details. In our examples we use $\eta = 0.25$. In Fig. 3.4, the top dashed line shows a situation where Eq. 3.133 is satisfied but the strong Wolfe condition Eq. 3.134 is not satisfied. This dashed line, which intersects the objective function curve at the point $(\tilde{\alpha}, f(\tilde{\alpha}))$, falls below the line with slope of $\rho f'(0)$. At the point $(\tilde{\alpha}, f(\tilde{\alpha}))$ where this dashed line intersects the objective function, the slope of the objective function which is $g_{k+1}^T d_k$ is greater than $-\eta f'(0)$. With $m_{k+1} = m_k + \tilde{\alpha} d_k$, Eq. 3.133 is satisfied, due to the fact that $f'(\tilde{\alpha})$ is positive whereas $f'(0)$ is negative. However, the strong Wolfe condition (Eq. 3.134) is not satisfied at $\alpha = \tilde{\alpha}$, because the value of $f'(\tilde{\alpha}) > \eta f'(0)$. To satisfy the strong Wolfe condition, we have to move the dashed line toward the minimum until it falls below the dot dashed line which intersects the objective function curve at the point at which the derivative is equal to $-\eta f'(0)$. We use $\eta = 0.25$ in our work. So the strong Wolfe condition is more restrictive. The first condition (Eq. 3.131) ensures that the objective function is reduced sufficiently, and the second condition (Eq. 3.134) prevents the steps from being too small. As is standard, we simply refer to Eqs. 3.131 and 3.134 as the Wolfe conditions.
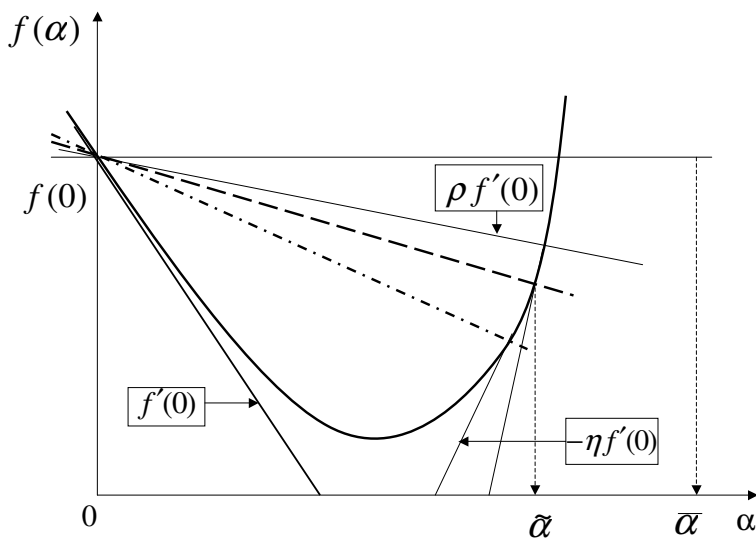

Fig. 3.4: Illustration of the strong Wolfe condition.

3.3.6 Quadratic Fit

Suppose at iteration k, we perform one Newton-Raphson iteration and find a step size $\hat{\alpha}_k$ for the current search direction d_k. With this step size, we calculate a new objective function

value $O(m_k + \hat{\alpha}_k d_k) = f(\hat{m}_{k+1})$ and check the Wolfe conditions. If the Wolfe conditions are not satisfied, we fit the function $f(\alpha) = O(m_k + \alpha d_k)$ with a quadratic function given by

$$q(\alpha) = a\alpha^2 + b\alpha + c. \tag{3.135}$$

With the known values of $q(0) = f(m_k)$, $q'(0) = f'(m_k)$ and $q(\hat{\alpha}_k) = f(m_k + \hat{\alpha}_k d_k)$, we find

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

Minimizing q gives

$$\alpha_k = -\frac{f'(m_k)\hat{\alpha}_k^2}{2[f(m_k + \hat{\alpha}_k d_k) - f'(m_k)\hat{\alpha}_k - f(m_k)]}, \tag{3.136}$$

which is used as the new step size. Based on our experience, the quadratic fit almost always results in a decrease in the objective function. Even though it rarely happens, however, a quadratic fit may yield a model at which the objective function value is bigger than the value of the objective function corresponding to the model obtained by one Newton-Raphson iteration. Such a situation is depicted in Fig. 3.5 where the quadratic approximation to the true objective function is inaccurate near the minimum. In this figure, point A corresponds to m_k; point B corresponds to $m_k + \tilde{\alpha} d_k$ obtained after the Newton-Raphson iteration and point C corresponds to $m_k + \alpha_k d_k$ which is obtained after quadratic fit. We see that the value of the objective function at point C obtained from the quadratic fit is larger than the value of the objective function at point B obtained by one Newton-Raphson iteration.

3.4   Evaluation of Computational Efficiency

Here, we assess the computational efficiency of GN (Gauss-Newton), LM (Levenberg-Marquardt), PCG (preconditioned conjugate gradient), BFGS and LBFGS. 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 although in our examples, one adjoint solution typically takes less than one half of the time of a simulation run. We do not keep track of the computational effort incurred when a proposed model update is rejected because it results in a violation of the Wolfe conditions.
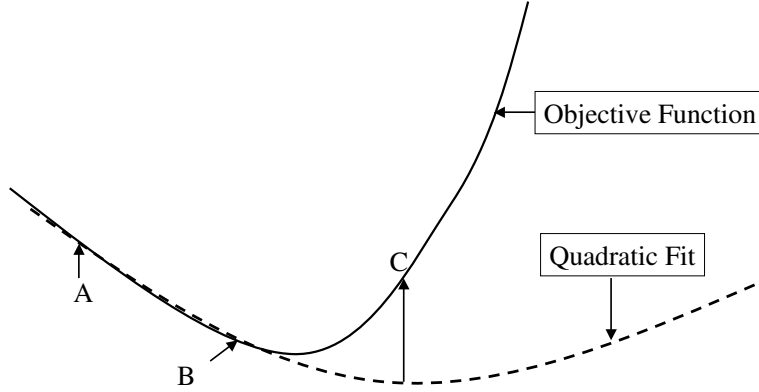
Fig. 3.5: Illustration of quadratic fit.

In GN and LM, if the data are evenly distributed in the time domain, the computational cost of calculating sensitivity of all data to all model parameters requires $N_d/2$ adjoint solutions which we count as being equivalent to $N_d/2$ simulation runs. GN and LM require one additional simulation run to calculate the new value of the objective function. So a total of $N_d/2 + 1$ simulation runs are needed to accomplish one GN or LM iteration.

In LBFGS and PCG, the total computational cost of implementing one iteration is equivalent to 3 simulation runs, which include one equivalent simulation run for calculating the gradient of the objective function by using the adjoint method, one simulation run for calculating the step size when using Newton-Raphson iteration (only one iteration of Newton-Raphson is done in our implementation) and another simulation run for calculating the objective function. Thus, LBFGS and PCG are $(N_d/2 + 1)/3$ times faster than GN and LM for each iteration. For example, if we have 1000 data, LBFGS and PCG will be roughly 167 times faster than GN or LM per iteration. In terms of the total time, if GN or LM require n_{GN} iterations to converge on average, while LBFGS or PCG need n_{BFGS} iterations to converge on average, then LBFGS or PCG will be l times faster than GN or LM where

$$l = \frac{n_{GN}}{\frac{n_{BFGS}}{(N_d/2 + 1)/3}} = \frac{n_{GN}}{n_{BFGS}} \times \frac{N_d/2 + 1}{3}. \tag{3.137}$$

Although BFGS requires more time than LBFGS and PCG to perform the matrix operations involved in the update equation, it is the memory requirement that makes the standard BFGS method inferior to LBFGS and PCG for large scale problems. Hence, the "standard" BFGS

refs to using Eq. 3.93 and storing $\tilde{H}_k^{-1}$. We could of course implement BFGS in exactly the same way as LBFGS in which case we do not explicitly compute or store $\tilde{H}_k^{-1}$.

3.5 Comparison of Memory Requirements

For large scale problems, the memory required by an optimization algorithm is a key issue that needs to be considered. Because we are only concerned with the difference between algorithms, we only consider the memory used by the optimization algorithm itself. Table 3.2 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. 3.10 or Eq. 3.14. 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. For convenience, we use one memory unit to stand for the memory occupied by one double precision real number. Recall the dimension of m_k, m_{k+1}, δm_{k+1}, s, d_k, g_k, g_{k+1}, m_{prior} and $\nabla_m O$ is N_m, the dimension of a sensitivity coefficient matrix G is $N_d \times N_m$ and the dimension of C_M is $N_m \times N_m$. In the GN or LM method, $(4 + 2N_d) \times N_m$ (m_k, m_{k+1}, δm, m_{prior}, sensitivity coefficient matrix G and $C_M G^T$) memory units are used. In CG, $8 \times N_m$ ($8 \times N_m$: m_k, m_{k+1}, δm_{k+1}, s_k, d_k, g_{k+1}, m_{prior}, $\nabla_m O$) memory units are used. For PCG, in addition to the memory required for the standard conjugate gradient, memory is required to store the preconditioner. In BFGS $(10 + N_m) \times N_m$ ($10 \times N_m$: m_k, m_{k+1}, δm_{k+1}, d_k, g_k, g_{k+1}, m_{prior}, v_k, $\tilde{H}_k^{-1} y_k$, y_k; $N_m \times N_m$: $\tilde{H}_k^{-1}$) memory units are used. In LBFGS, $(7 + 2 \times L) \times N_m$ ($7 \times N_m$: m_k, m_{k+1}, δm_{k+1}, g_k, d_k, m_{prior}, diagonal inverse Hessian approximation; $2 \times L \times N_m$: y_k and s_k for $k = 1, 2, \cdots, L$) memory units are used. From the results of Table 3.2, we see that the full-memory version of BFGS uses the most memory which is on the order of N_m^2, the standard conjugate gradient method 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. For large scale problems in which the number of data and the number of model parameters are both large, the memory used by limited memory BFGS depends on the number of previous vectors (denoted by L in Table 3.2) used to construct the Hessian inverse approximation update; L must be specified by the user. Fig. 3.6 shows a snapshot of the panel which monitors the memory usage history and the CPU usage history when the LBFGS algorithm was applied to do a history match for a 2D problem presented in the next section. In the right bottom black window, the curve shows the memory usage. The period of high memory usage corresponds to a simulation run and the lower memory period corresponds to solving an adjoint system.

Table 3.2: Memory used by each algorithm.

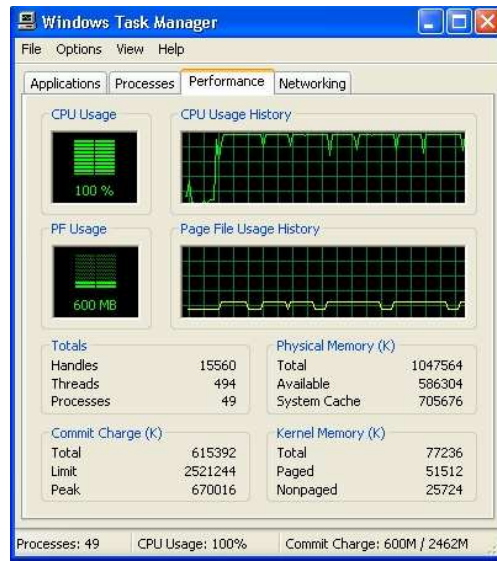|	No. of DP real numbers
GN/LM	$(4 + 2 \times N_d) \times N_m$
CG	$8 \times N_m$
PCG	$8 \times N_m$ + memory for preconditioner
BFGS	$(10 + N_m) \times N_m$
LBFGS	$(7 + 2L) \times N_m$

Fig. 3.6: Memory and CPU usage history of the computer.

3.6 Two-Dimensional Three-Phase Synthetic Example

A 2D three-phase history matching problem was considered in this example. We use a 15×15 grid with $\Delta x = \Delta y = 40$ ft and $\Delta z = 30$ ft. We consider a very small problem so we can easily apply all optimization algorithms. The porosity for the true model is homogeneous and equal to 0.22. Permeability is isotropic and uniform in three different zones; see Fig. 3.7. The values of $\ln(k)$ in the lower left zone, lower right zone and the upper half zone are equal to 4.0, 4.6 and 4.2 with k in md, respectively. Four producers and one water injection well are completed in the reservoir. The well locations are indicated by the white squares in Fig. 3.7. All producers start producing at time zero at a constant total flow rate of 200 STB/Day and produce for 300 days. The production constraint is the minimum bottom-hole pressure which is set to 50 psi and the economic limit is the maximum WOR which

is set to 49 STB/STB. When the bottom-hole pressure of a well decreases below 50 psi, then the well will be produced at a constant bottom-hole pressure equal to 50 psi. If the WOR exceeds 49, then the well will be shut in. For all examples considered in this section, the production constraint and economic limit are never reached. Bottom-hole pressure data from all five wells, GOR and WOR from all four producers are used as the conditioning data to estimate the gridblock permeabilities, i.e., the porosity is fixed at its true values. A total of 364 data (28 for each type of data at each producing well and 28 pressure data at the water injection well) are history matched. We assume pressure measurement errors to be independent, identically-distributed, normal random variables with mean equal to zero and variance equal to 1 psi^2. GOR measurement errors were modeled similarly except the variance was set equal to 25 (SCF/STB)^2. Following Wu (1999), the variance of WOR measurement errors was specified by

$$\text{Var}[e_{\text{WOR}}] = \text{WOR}_{\text{obs}}^2 \epsilon_o^2 + \frac{1}{q_{o,\text{obs}}^2} \sigma_{q_w,\text{obs}}^2, \tag{3.138}$$

where

$$\epsilon_m^2 = \frac{\text{Var}[q_{m,\text{obs}} - q_m]}{q_{m,\text{obs}}^2}, \tag{3.139}$$

for $m = o, w$ and

$$\sigma_{q_w,\text{obs}} = \max[\epsilon_w q_{w,\text{obs}}, \sigma_{q_w,\text{obs}}^{\min}]. \tag{3.140}$$

Here, $q_{m,\text{obs}}$, $m = o, w$ denotes the observed rate of phase m. In this example, we choose $\sigma_{q_w,\text{obs}}^{\min}$=2.0 STB/Day, ϵ_o=0.001 and ϵ_w=0.02. The variances for different data are used to form the data covariance matrix C_D. The isotropic spherical variogram with the range equal to 240 ft in all three directions and the variance for $\ln(k)$ equal to 1 was used to construct the model covariance matrix, C_M. The objective function given in Eq. 3.10 was minimized; i.e., we generated the maximum a posteriori (MAP) estimate by history matching the production data.

3.6.1 Comparison of the Optimization Algorithms

For the examples considered in this subsection, we did not add any noise to the true data generated by running the simulator with the true model as input. Our focus is on the investigation of the computational efficiency of different optimization algorithms.

The iterative solver (see Zhang (2002)) was applied to solve the adjoint equations involved in the computation of the sensitivity coefficient matrix and the computation of the gradient of the objective function. For the comparison purpose, the Levenberg-Marquardt
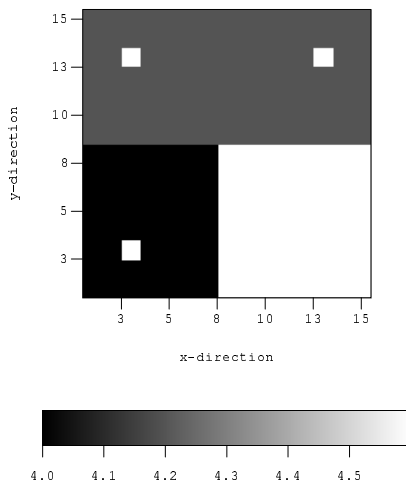
Fig. 3.7: Permeability field for the true model.

(LM), Broyden-Fletcher-Goldfarb-Shanno (BFGS), limited memory BFGS (LBFGS), conjugate gradient with C_M as the preconditioner (CM-PCG) and conjugate gradient with preconditioner generated from limited memory BFGS (LBFGS-PCG) were used to minimize the objective function involved in the history matching procedure. In some cases, GN fails to converge to a legitimate model; see Li (2001). Thus, GN is not compared with the other algorithms. For these algorithms, a uniform value of 4 for $\ln(k)$ was used as the initial guess.

In Levenberg-Marquardt, the initial damping factor was chosen as 10^5. When the objective function increases, the damping factor was simply multiplied by a factor of 10; whereas, when the objective function decreases, the damping factor was simply divided by a factor of 10. Levenberg-Marquardt (LM) converged to 13.343 in 9 iterations. The curve through the circles in Fig. 3.8 shows the behavior of the objective function during the LM iterations. Fig. 3.9(b) shows the final model (i.e., the permeability field) obtained by Levenberg-Marquardt, which is very similar to the true model which is reproduced in Fig. 3.9(a).

The BFGS method we used is the standard Broyden-Fletcher-Goldfarb-Shanno method with initial scaling. In other words, $\gamma_k = 1$ for all $k > 0$ was used when applying Eq. 3.98 to update the inverse Hessian approximation. The initial scaling factor γ_0 was obtained by

$$\gamma_0 = \sigma_0 = \frac{s_0^T y_0}{y_0^T \tilde{H}_0^{-1} y_0^T}, \tag{3.141}$$

where $\tilde{H}_0^{-1}$ is equal to C_M. BFGS converged to 13.448 in 97 iterations. The curve through the diamonds in Fig. 3.8 shows the behavior of the objective function. Fig. 3.9(c) shows
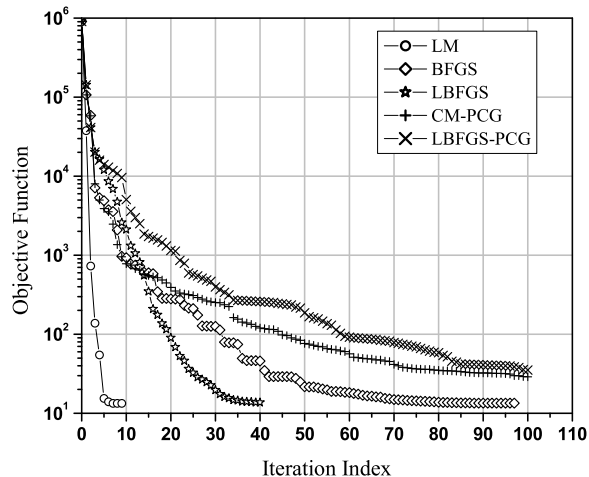
Fig. 3.8: Behavior of the objective function.

the final model obtained by the BFGS. We can see that the final model obtained by BFGS is very similar to the final model obtained by the Levenberg-Marquardt method (compare Figs. 3.9(c) and 3.9(b)).

In limited memory BFGS, we scaled the inverse Hessian approximation at each iteration. Following Oren (1974a), and the results of our comparative studies, the scaling factors were chosen by the following scheme

$$\gamma_k = \tilde{\tau}_{1k} \qquad \text{if} \quad \tilde{\tau}_{1k} < 1.0, \tag{3.142}$$

$$\gamma_k = \tilde{\sigma}_k \qquad \text{otherwise.} \tag{3.143}$$

The diagonal of C_M was chosen as the initial inverse Hessian approximation. The 30 most recent vectors (i.e., s_k's and y_k's) were used to construct the inverse Hessian approximation at each iteration, i.e, $L = 30$. Limited memory BFGS converged to 13.685 in 40 iterations. The curve through the stars in Fig. 3.8 shows the behavior of the objective function. Fig. 3.9(d) shows the final model obtained by the limited memory BFGS, which captures the main characteristics of the true model, but is somewhat rougher than the true model and the MAP estimate obtained with LM.

The two preconditioned conjugate gradient methods discussed earlier were implemented and applied to this history matching problem. In one method, the full C_M^{-1} was used as the preconditioner (we call this method CM-PCG); whereas in the other method, we used an estimated quasi-Newton preconditioner (we call this method LBFGS-PCG). Both algorithms were terminated at 100 iterations, because this was the maximum number of iterations

53

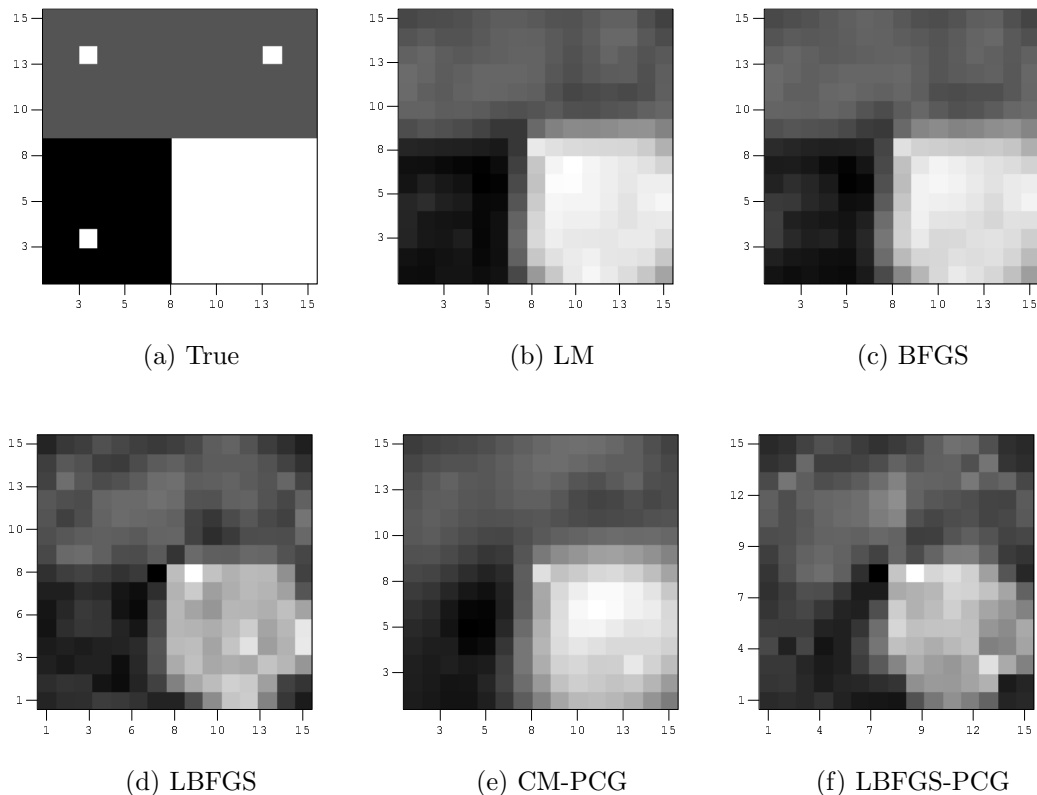|		
(a) True	(b) LM	(c) BFGS
(d) LBFGS	(e) CM-PCG	(f) LBFGS-PCG

Fig. 3.9: Final model obtained by different optimization algorithms.

allowed. However, the convergence criteria of Eqs. 3.112 and 3.113 were not satisfied at the 100th iteration of either method. CM-PCG "converged" to a model corresponding to an objective function value equal to 28.851 in 100 iterations. The curve through the plus signs in Fig. 3.8 shows the behavior of the objective function. Fig. 3.9(e) shows the final model obtained by the CM-PCG. LBFGS-PCG "converged" to a model corresponding to an objective function value equal to 35.187 in 100 iterations. The curve through the crosses in Fig. 3.8 shows the behavior of the objective function. Fig. 3.9(f) shows the final model obtained by the LBFGS-PCG. Note the MAP estimates obtained with the preconditioned CG method are inferior to those obtained by LM and BFGS.

In the quasi-Newton methods, the Hessian inverse approximation is constructed based on the quasi-Newton search direction. The theory guarantees that for a positive definite quadratic function the Hessian inverse approximation becomes the true Hessian inverse at the nth iteration where n is the number of model parameters for the quadratic problem with exact line search. However, in LBFGS-PCG the Hessian inverse approximation is constructed based on the preconditioned conjugate gradient search direction, which is no longer

the quasi-Newton Hessian inverse approximation. When we use the conjugate gradient search direction to find a new model, and based on this new model, construct the Hessian inverse approximation as a preconditioner, it is not clear how good this preconditioner is. For the single-phase gas example presented in the first annual report on this project, this preconditioner results in faster convergence than is obtained by simply using C_M^{-1} as a preconditioner; in the current example, however, convergence is slower.

The final objective function value at convergence and the number of iterations required to converge for different algorithms are summarized in Table 3.3. In the "Scaling Scheme" column, "Initial Scaling" means that only the initial Hessian inverse approximation is scaled by a factor γ_0. In other words, we choose a sequence γ_k such that $\gamma_k = 1$ for all $k > 0$. "All Scaling" means that the inverse Hessian approximation $\tilde{H}_k^{-1}$ was scaled by γ_k at each iteration for the case when BFGS was applied and the initial inverse Hessian approximation $\tilde{H}_0^{-1}$ was scaled by γ_k at each iteration for the case where limited memory BFGS was applied. "N/A" means not available. FCM-LBFGS stands for the LBFGS algorithm in which the initial Hessian inverse approximation is chosen as the full C_M matrix instead of just the diagonal of C_M; see the discussion presented later.

Algorithms	Scaling Scheme	Objective Function	No. of Iterations
LM	N/A	13.343	9
BFGS	Initial Scaling	13.448	99
LBFGS	All Scaling	13.685	40
CM-PCG	N/A	28.851	100
LBFGS-PCG	All Scaling	35.187	100
FCM-LBFGS	All Scaling	13.992	33

Table 3.3: Comparison of different minimization algorithms.

Based on Eq. 3.137 and the number of iterations required (see Table 3.3), LBFGS and FCM-LBFGS, respectively, are 13.7 and 16.6 times faster than Levenberg-Marquardt overall. Table 3.4 shows the CPU time in seconds used by different algorithms. Based on the real CPU time, LBFGS and FCM-LBFGS, respectively, are 10.5 and 11.1 times faster than Levenberg-Marquardt overall. The column labeled "Scaling Scheme" has the same meaning as in Table 3.3.

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

Table 3.4: Comparison of the CPU time used by different minimization algorithms.

3.6.2 Effect of Preconditioning Matrix on Conjugate Gradient Methods

Fig. 3.10 shows the behavior of the objective function obtained by the conjugate gradient without preconditioning (triangles) and the preconditioned conjugate gradient with C_M^{-1} as the preconditioner (circles). We can see clearly that the preconditioned conjugate gradient is slightly better than conjugate gradient without preconditioning. Fig. 3.11 shows the final model (a) obtained by the conjugate gradient method without preconditioning compared to the true model (c) and the model (b) obtained using C_M^{-1} as a preconditioner. We can see that the MAP estimate obtained from CG without preconditioning is far rougher than the model obtained by the CM-PCG. We can conclude that the prior covariance matrix C_M not only acts as a preconditioning matrix but also provides smoothness for the model.

3.6.3 BFGS Scaling Scheme

The BFGS result of the preceding section was obtained by only scaling the initial Hessian inverse approximation $\tilde{H}_0^{-1}$. If we consider the fact that the Hessian matrix changes iteration by iteration for the nonlinear problem, it seems that we should scale the matrix at every iteration based on the new information; see Oren and Luenberger (1974) or Oren (1974b). According to Oren and Spedicato (1976), the optimal scaling factor which minimizes the upper bound of the condition number of $\tilde{H}_{k+1}^{-1}$ at the kth iteration should be

$$\gamma_k = \sigma_k \equiv \frac{s_k^T y_k}{y_k^T \tilde{H}_k^{-1} y_k}. \tag{3.144}$$

The new Hessian inverse approximation $\tilde{H}_{k+1}^{-1}$ was calculated by using Eq. 3.98 with γ_k given by Eq. 3.144. With this all-scaling scheme, BFGS converged to 16.1989 in 66 iterations;
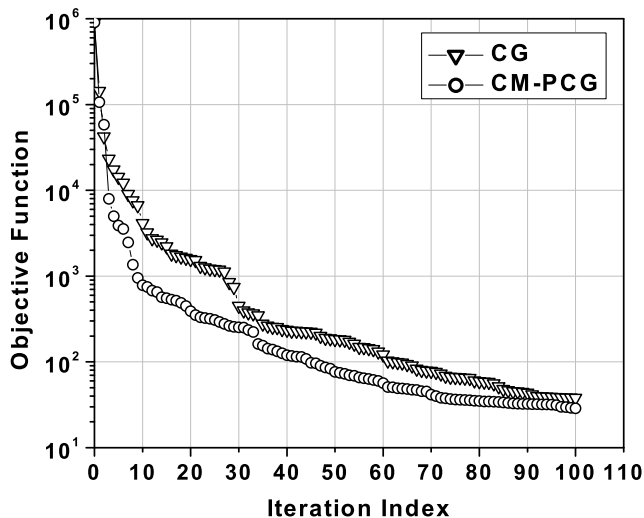
Fig. 3.10: Behavior of the objective function obtained by CG and CM-PCG.

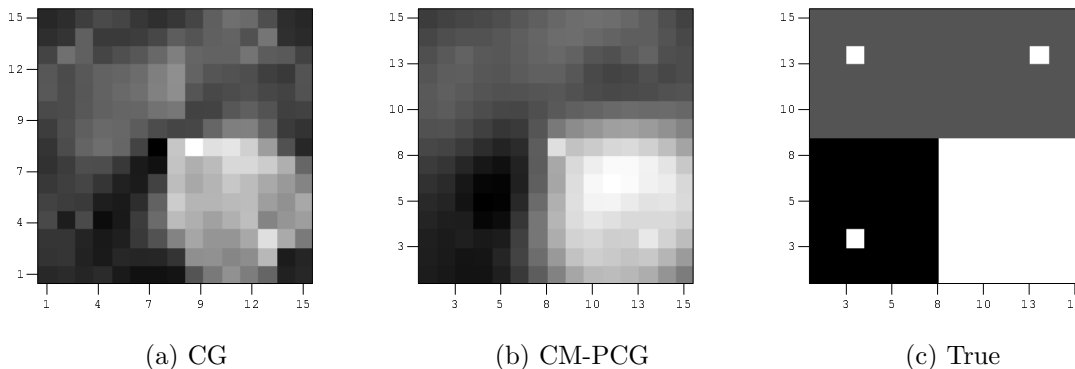(a) CG   (b) CM-PCG   (c) True

Fig. 3.11: Final model obtained by conjugate gradient without preconditioning.

see the curve through circles in Fig. 3.12. The curve through the plus signs in Fig. 3.12 shows the behavior of objective function iteration by iteration for BFGS with initial scaling scheme. This case was presented previously. Although it appears that the all-scaling scheme is better than just initial scaling in terms of the number of iterations required to converge, at the 66th iteration, the objective function value for the two schemes are very close. Yet, the convergence criteria of Eqs. 3.112 and 3.113 were satisfied for all-scaling BFGS but not for the initial scaling BFGS at the 66th iteration. The diamonds shown in Fig. 3.12 represent the objective function values obtained by BFGS without any scaling at each iteration. Clearly, BFGS without scaling is much worse than initial scaling and all-scaling. As presented pre-

viously, scaling does help to improve the convergence rate of BFGS for this three-phase problem. The MAP estimate obtained with the all scaling BFGS is shown in Fig. 3.13 and is very similar to the MAP estimate obtained by applying BFGS with initial scaling (see Fig. 3.9(c)).
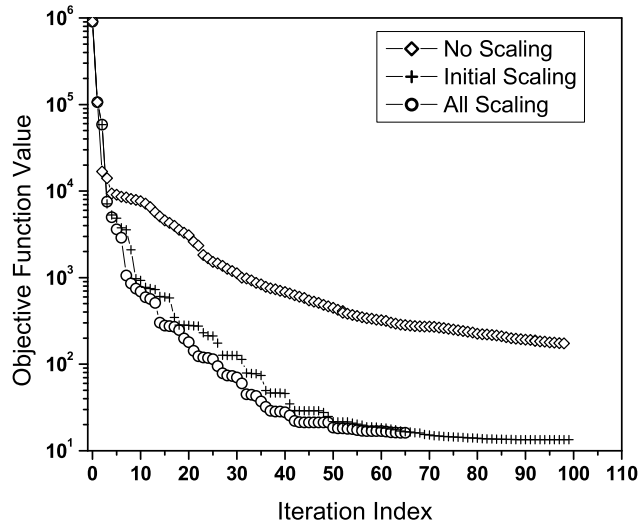

Fig. 3.12: Behavior of the objective function for BFGS with different scaling schemes.
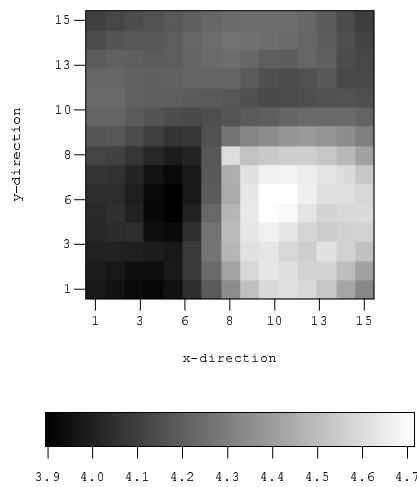

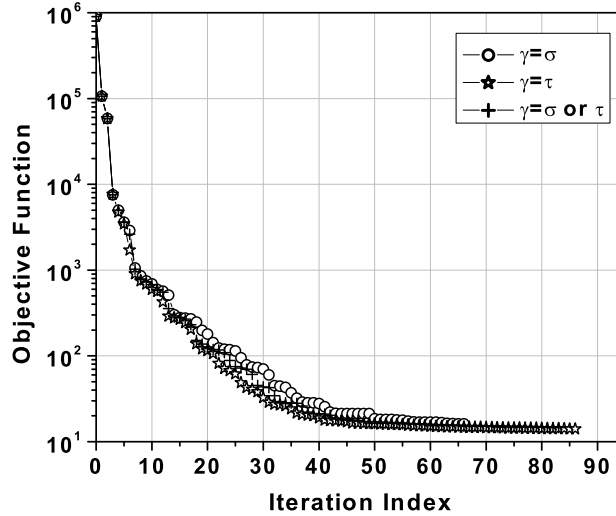Fig. 3.13: Final model obtained by BFGS with scaling at all iterations; $\gamma_k = \sigma_k$.

Fig. 3.14: Behavior of the objective function for BFGS with different schemes for scaling at all iterations.

Based on Oren and Spedicato (1976), σ_k is the optimal scaling factor for γ_k if the BFGS algorithm ($\theta_k = 1$) or one of its variants is applied; see Eq. 3.106. Our experiments discussed later, however, indicate that $\gamma_k = \tau_k$ (Eq. 3.99) is superior to $\gamma_k = \sigma_k$ (Eq. 3.100) for LBFGS; also see Zhang et al. (2001b). Here, we investigate the convergence behavior of BFGS with the scaling factor equal to τ_k. The behavior of the objective function obtained by BFGS with $\gamma_k = \tau_k$ is shown by the curve through stars in Fig. 3.14. The curve through circles in the same figure is obtained by the BFGS with $\gamma_k = \sigma_k$. Even though $\gamma_k = \sigma_k$ is the optimal scaling factor based on Oren's theory, it turns out $\gamma_k = \tau_k$ is not worse than $\gamma_k = \sigma_k$. The curve through plus signs in this same figure is obtained by the BFGS with the modified Oren scaling scheme, i.e., $\gamma_k = \tau_k$ if $\tau_k < 1.0$; otherwise $\gamma_k = \sigma_k$. There is not much difference between these three scaling schemes in terms of the objective function at convergence.

3.6.4 LBFGS Scaling Scheme

As shown in Oren (1974b), $\tilde{\tau}_k$ calculated by Eq. 3.109 or Eq. 3.110 is an approximation to the correct τ_k which is given by Eq. 3.99. In our implementation of LBFGS, we only form the product of $\tilde{H}_k^{-1} g_k$ which is used to generate the search direction. We never form $\tilde{H}_k^{-1}$ or $\tilde{H}_k$ explicitly. Therefore, we cannot obtain τ_k using the exact form of Eqs. 3.102 and 3.104 and cannot obtain σ_k using the exact form of Eq. 3.100. Thus, we have to use Eq. 3.109 or Eq. 3.110 to approximate τ_k and use Eq. 3.108 to approximate σ_k. The scaling factor was

chosen as follows:

$$\gamma_k = \tilde{\tau}_k \qquad \text{if} \quad \tilde{\tau}_k < 1, \tag{3.145}$$

$$\gamma_k = \tilde{\sigma}_k \qquad \text{otherwise}, \tag{3.146}$$

where $\tilde{\tau}_k$ can be obtained by either Eq. 3.109, Eq. 3.110 or Eq. 3.111 and $\tilde{\sigma}_k$ is obtained by Eq. 3.108.

Fig. 3.15 shows the behavior of the objective functions during the LBFGS iterations when different formulas are applied to calculate $\tilde{\tau}_k$. The diamonds represent the objective function obtained by using Eq. 3.110 to calculate $\tilde{\tau}_k$. The circles represent the objective function obtained by using Eq. 3.111 to calculate $\tilde{\tau}_k$. The curve through stars represents the objective function obtained by using Eq. 3.109 to calculate $\tilde{\tau}_k$. The triangles represent the objective function obtained by setting $\gamma_k = \tilde{\sigma}_k$ at every iteration where $\tilde{\sigma}_k$ is calculated from Eq. 3.108. This scaling option was used by Liu and Nocedal (1989). Shanno and Phua (1978) suggested using $\tilde{\sigma}_k$ and only scaling the initial matrix. From these curves, we can see that the curve through the stars in which the $\tilde{\tau}_k$ is calculated by the Eq. 3.109 is the best. The three other scaling factors give roughly the same convergence results as each other. Note that the curve through the circles in which $\tilde{\tau}_k = \tau_k$ is calculated by Eq. 3.111 requires the most iterations to converge. Note that Eq. 3.111 gives the correct value for τ_k. This is somewhat surprising in that it indicates that using an approximation for τ_k, which is defined by Eq. 3.99, gives better convergence results than using the correct value of τ_k. It turns out that when we use Eq. 3.109 and Eq. 3.110 to calculate $\tilde{\tau}_k$, the value of $\tilde{\tau}_k$ is always less than 1 which implies that the scaling factor always takes the value of $\tilde{\tau}_k$. When we use Eq. 3.111 to calculate $\tilde{\tau}_k$, which is the correct τ_k, it is less than 1 at some iterations and is bigger than 1 at other iterations. So whenever $\tilde{\tau}_k > 1$, γ_k takes the value of $\tilde{\sigma}_k$ which is also an approximation. So these four options are all approximations. Based on the example just presented and the gas example presented in the first annual report on this project, Eq. 3.109 is the best approximation. Based on the paper published by Oren and Spedicato (1976), the optimal scaling factor γ_k should always take the value of σ_k for the BFGS method. It is not clear how to form σ_k accurately at every iteration without forming $\tilde{H}_k^{-1}$ explicitly. Moreover, our result presented previously (see Fig. 3.14) indicates that σ_k might not be the optimal scaling factor. For almost all examples that we have considered using either BFGS or LBFGS, $\gamma_k = \tilde{\tau}_k$ gives as good or better convergence results than are obtained using $\gamma_k = \tilde{\sigma}_k$.

Fig. 3.16 shows the behavior of the objective function when LBFGS was applied without scaling, with initial scaling and with all-scaling based on Eqs. 3.145 and 3.146 where $\tilde{\tau}_k$ is
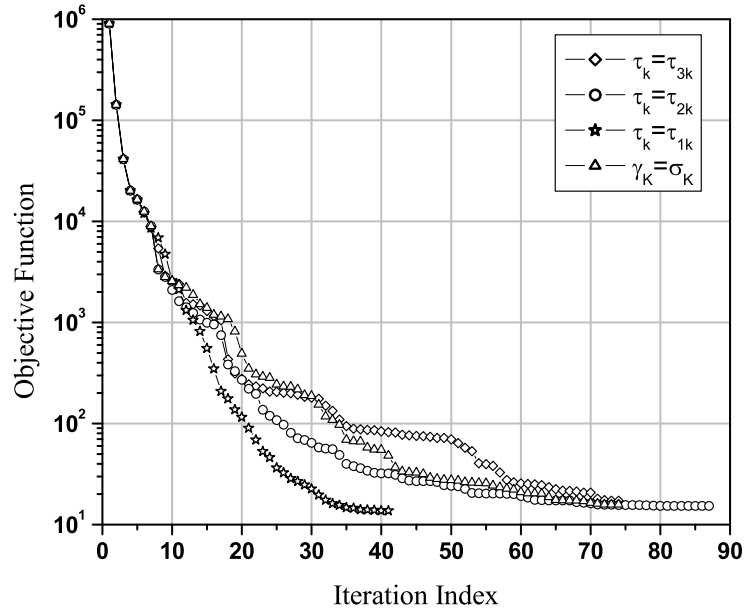
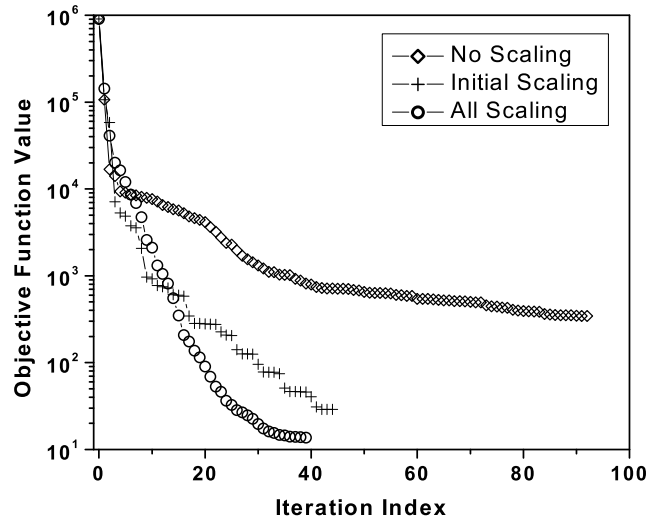Fig. 3.15: Behavior of the objective function for LBFGS with different scaling factors.

Fig. 3.16: Comparison of the behavior of the objective function for different scaling schemes.

calculated by Eq. 3.109. In this figure, the circles represent the objective function values obtained by LBFGS with all-scaling, the plus signs represent the objective function value obtained by LBFGS with initial scaling and diamonds represent the objective function ob-

61

tained by LBFGS without any scaling. As in BFGS, we can see clearly that LBFGS with all-scaling is better than just initial scaling and initial scaling is better than no scaling.

3.6.5 Sensitivity to the Number of Previous Vectors

Here we investigate how the convergence rate is affected by the number of previous vectors used in constructing the Hessian update in the LBFGS algorithm. Fig. 3.17 shows the behavior of the objective function obtained by using a different number of previous vectors to construct the new approximate inverse Hessian. In all cases, the all-scaling scheme with scaling factor given by Eq. 3.109 was applied. Table 3.5 lists the number of iterations required for convergence and the value of the objective function at convergence when using a different number of previous vectors to construct the Hessian inverse updates. L denotes the number of previous vectors. We can see that when we use too few (10 in this example) previous vectors to construct Hessian inverse updates, more iterations are required for convergence and the value of the objective function is higher than when $L = 20, 30$ or 50. When L equals $20, 30$ or 50, there is not too much difference in terms of the number of iterations required to obtain convergence, but $L = 20$ gives a higher value of the objective function at convergence.
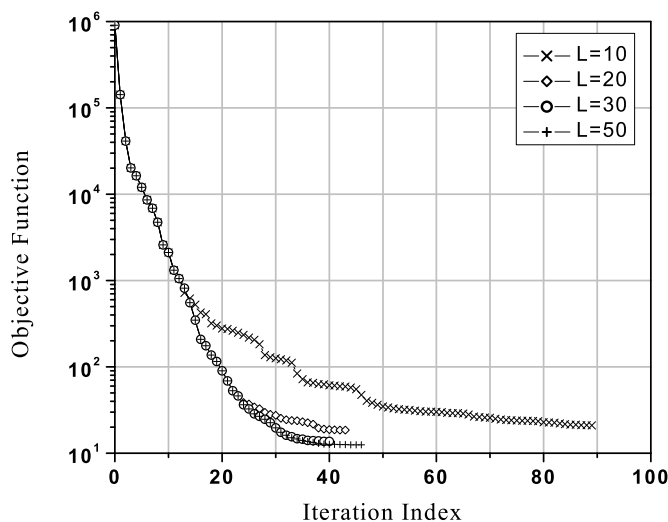

Fig. 3.17: The effect of the number of previous vectors used to construct the new Hessian inverse approximation on the performance of LBFGS.

62

L	Objective Function	No. of Iterations
10	20.922	90
20	18.555	43
30	13.685	40
50	12.512	46

Table 3.5: Sensitivity to the number of previous vectors.

3.6.6   Improvement of the Smoothness of the Model

From the results of Figs. 3.9(c) and 3.9(d), we see that the final model obtained by LBFGS is somewhat rougher than the true model and the MAP estimate obtained with BFGS. It turns out this is caused by the fact that only the diagonal of C_M was used as the initial Hessian inverse approximation. When we form the search direction, i.e., the product of the $\tilde{H}_k^{-1}$ and the gradient of the objective function g_k, we need to perform the operation of multiplying a vector by the initial Hessian inverse approximation. If we use the full C_M instead of just the diagonal elements of C_M as $\tilde{H}_0^{-1}$, we obtain a smoother result. Fig. 3.18(a) shows the MAP estimate obtained by LBFGS using the full C_M as the initial Hessian inverse approximation compared with using only the diagonal of C_M, DCM-LBFGS. Using $\tilde{H}_0^{-1} = C_M$ gives improved results, i.e., gives a MAP estimate much closer to the truth than is obtained with $\tilde{H}_0^{-1} = \text{diag}[C_M]$. With $\tilde{H}_0^{-1} = C_M$, the LBFGS converged in 33 iterations and the value of the objective function at convergence was equal to 13.992. For this case, Fig. 3.19 indicates again that convergence is fastest when Eq. 3.109 is used to calculate the scaling factor. This method is labeled as FCM-LBFGS in Table 3.3.

3.6.7   Effect of Data Noise

In this subsection we consider the case where the observed data were obtained by adding noise to the true data. With this data set, we repeated the example in the previous subsection using LBFGS with the full C_M as the initial Hessian inverse approximation. The objective function evaluated based on Eq. 3.10 is reduced from 900,728 to 309 in 35 iterations. For the case where data are noisy, the approximate results of Tarantola suggest that the expected value of the objective function is $N_d/2 = 182$ with standard deviation equal to $\sqrt{N_d/2} = 13.5$. Thus, the expectation plus five standard deviations equals 249 which is somewhat smaller than the 309. Although the value of the objective function does not satisfy the criterion given by Eq. 3.15, one should recall that the criterion assumed data is a linear function
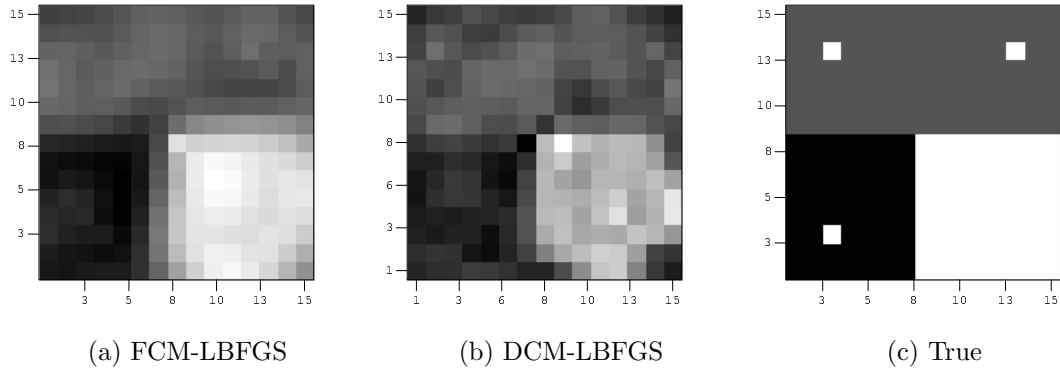
(a) FCM-LBFGS   (b) DCM-LBFGS   (c) True

Fig. 3.18: Final model obtained by LBFGS with full C_M (a) and diagonal C_M (b) as the initial Hessian inverse approximation, compared with the true model (c).
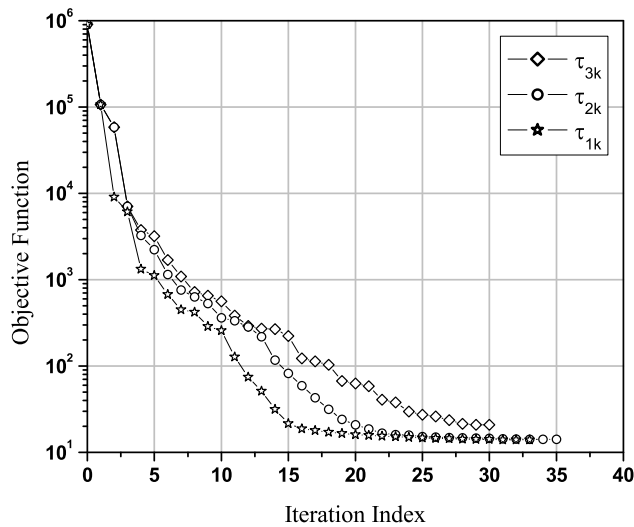

Fig. 3.19: Behavior of the objective function; full C_M as the initial Hessian inverse approximation; different scaling factor options.

of the model, which is not the case. Recall that when using the true data without noise the objective function converged to 13.992 in 33 iterations. The behavior of the objective function for both cases (solid circles for the true data case and the circles for the case where the data with noise) are shown in Fig. 3.20. The model obtained by history matching data with noise (observed data) is shown in Fig. 3.21(a). We can see that even though the model obtained by history matching data with noise captures the main structure of the true model, it is worse than the model obtained by history matching the true data.
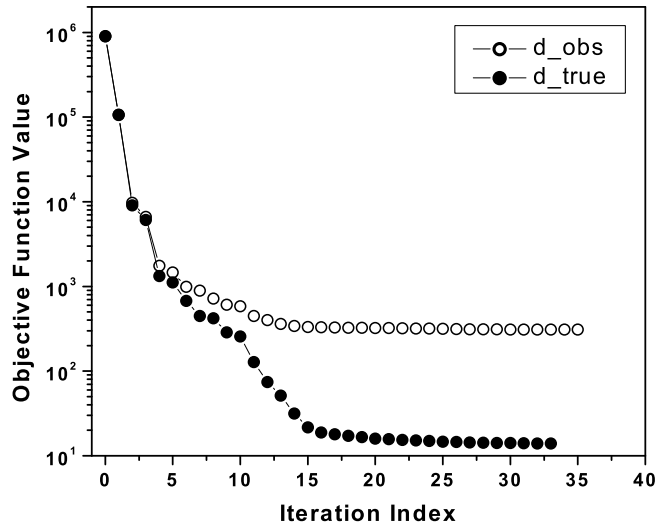
64

Fig. 3.20: Behavior of the objective function; full C_M as the initial Hessian inverse approximation; data with noise and without noise.

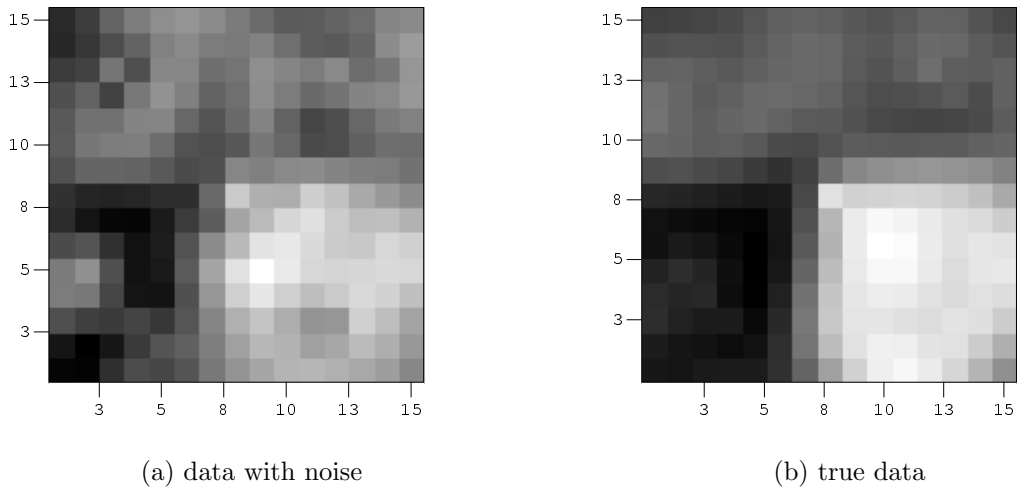(a) data with noise

(b) true data

Fig. 3.21: Final model obtained by LBFGS with full C_M as the initial Hessian inverse approximation by history matching data with noise (a) and true data (b).

3.7 Three-Dimensional Three-Phase Example

Here, we consider a three-dimensional three-phase history matching problem. We use a $40 \times 40 \times 6$ grid with $\Delta x = \Delta y = 100$ ft and $\Delta z = 30$ ft. The porosity for the true model is homogeneous and equal to 0.22. The true permeability field is an unconditional realization

generated by Gaussian co-simulation. An isotropic spherical variogram with the range in all directions equal to 600 ft was used to generate unconditional realizations. The variance for $\ln(k)$ is 1 and the mean for $\ln(k)$ is 4.5. One layer of the true permeability field is shown in Fig. 3.22. Fig. 3.22 (b) is the interpolation plot of (a). The interpolation plots make it easier to compare the permeability structure. The initial pressure is 4500 psi and the bubble point pressure is 4417 psi. The formation volume factor (FVF) and the viscosity for oil, water and gas at the bubble point pressure are given in Table 3.6. The capillary pressure is assumed to be negligible. The water-oil and oil-gas two-phase relative permeability are given in Fig. 3.23 (a) and (b), respectively. Stone's model II is used to calculate three-phase oil relative permeability; see Aziz and Settari (1979). Six producers and four water injection wells are completed in the reservoir. The producers and the injectors, respectively, are indicated by black squares and white squares, respectively, in Fig. 3.22 (a). All producers start producing at time 0. We will history match synthetic data generated from running the CLASS simulator up to 500 days using the truth case as input. The injectors start to inject water at 30 days and stop at 500 days. The well operating conditions are summarized in Table 3.7. The wells are operating with the target first. Whenever the constraint is violated at a particular well, then the constraint will be switched to be the target for the corresponding well. When the economic limits are violated at a particular well, then the corresponding well will shut in. In this table, the keyword MAXVOL means maximum total rate in STB/Day; MAXWATINJ means the maximum water injection rate in STB/Day; MINBHP means the minimum bottom-hole pressure in psi and MAXWOR means the maximum water-oil ratio in STB/STB.

B_o (RB/STB)	1.748
μ_o (cp)	0.486
B_g (RB/MSCF)	0.75
μ_g (cp)	0.0284
B_w (RB/STB)	1.006
μ_w (cp)	0.012

Table 3.6: Fluid properties at bubble point pressure.

As in the example presented in the previous section, the production target is a constant total flow rate for the producers and constant water injection rate for injection wells. The production constraints for the producers are the minimum bottom-hole pressure which is set to 50 psi and the maximum WOR which is set to 49 STB/STB. If the bottom-hole pressure
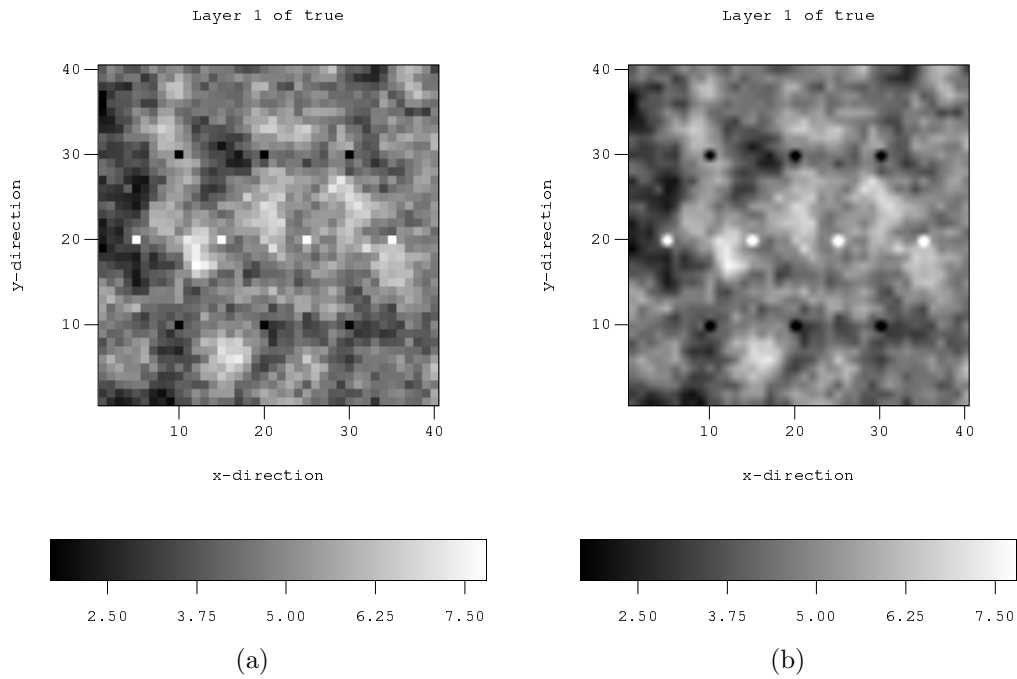
(a)

(b)

Fig. 3.22: The layer 1 log-permeability field of the true model.
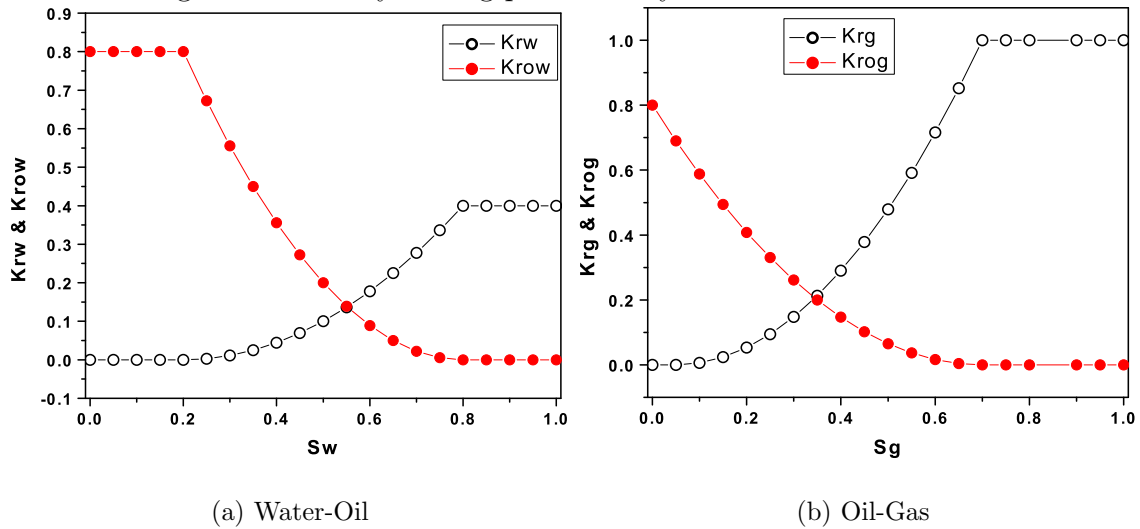

(a) Water-Oil

(b) Oil-Gas

Fig. 3.23: The relative permeability curve used in this example.

of a well decreases below 50 psi, then thereafter, the well will be produced at a constant bottom-hole pressure equal to 50 psi. When the WOR is bigger than 49 STB/STB, then the corresponding well will be shut in. For this example, the production constraint and economic limit are never reached, so all wells are produced at their target rate.

Well No.	Targets		Constraints	Economic Limits
	MAXVOL	MAXWATINJ	MINBHP	MAXWOR
	(STB/Day)	(STB/Day)	(psi)	(STB/STB)
1	4000	-	50	49
2	10000	-	50	49
3	4000	-	50	49
4	15000	-	50	49
5	8000	-	50	49
6	8000	-	50	49
7	-	6000	-	-
8	-	10000	-	-
9	-	8000	-	-
10	-	8000	-	-

Table 3.7: Well operating targets, constraints and economic limits.

3.7.1 Conditioning to True Data

In this subsection, we consider the case where the true data are history matched. Bottom-hole pressure from all ten wells, GOR and WOR from all six producers are used as the conditioning data to estimate the gridblock permeabilities only, i.e., the porosity is fixed at its true values. There are a total of 880 conditioning production data which are history matched. (40 for each type at each producing well and 40 pressure data at the water injection wells.) The variance used for all the measurement errors are the same as used in the 2D three-phase flow example. The variances for different data are used to form the diagonal data covariance matrix C_D. The objective function given in Eq. 3.14 was minimized, except we used d_{obs} instead of d_{uc} and d_{obs} is equal to the true data with no noise added. Although we did not add noise to the data, we refer to the resulting realization as a conditional realization.

The unconditional realization was generated by Gaussian co-simulation. Two layers of the unconditional realization of the log-permeability fields are shown in Fig. 3.24 (a) and Fig. 3.25 (a), respectively. Optimization was done with the LBFGS algorithm using scaling at all iterations. Eq. 3.109 was used to generate the kth scaling factor. The objective function was reduced from 312,164,623 to 649 in 70 iterations. The objective function value was calculated based on the Eq. 3.14 with d_{uc} replaced by the true data without noise. The behavior of the objective function is shown in Fig. 3.26.

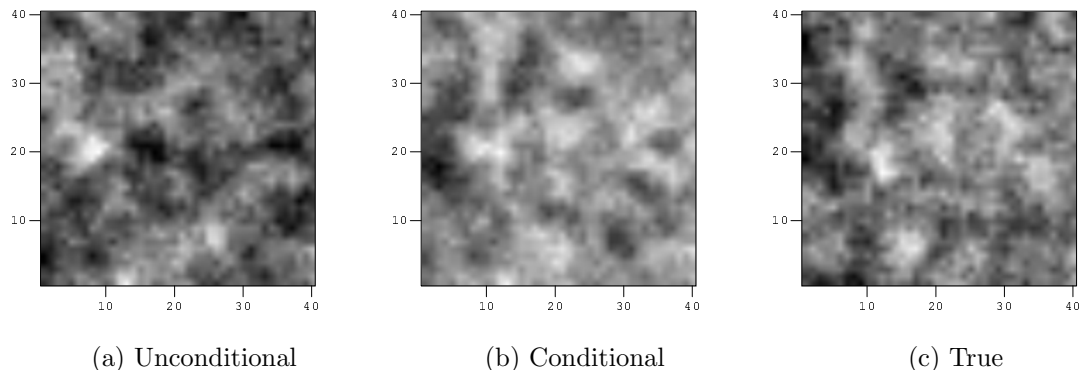(a) Unconditional   (b) Conditional   (c) True

Fig. 3.24: The log-permeability field for layer 1 generated by Gaussian co-simulation (a), by history matching production data (b) and the true model(c).
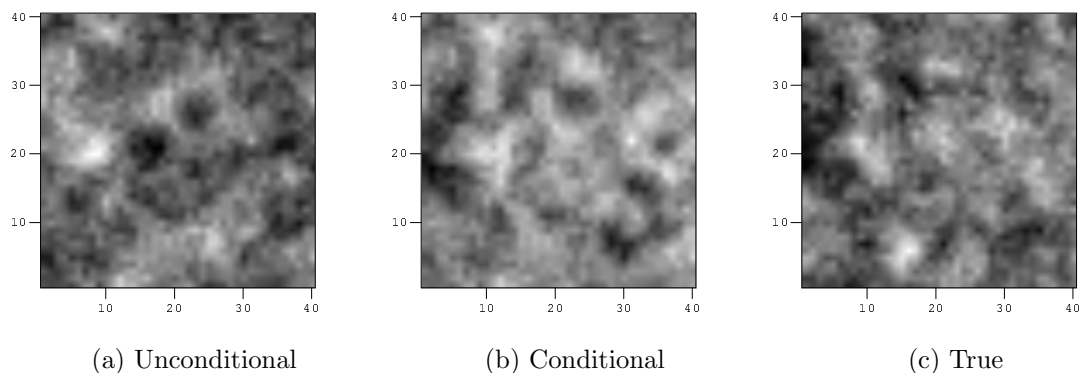

(a) Unconditional   (b) Conditional   (c) True

Fig. 3.25: The log-permeability field for layer 4 generated by Gaussian co-simulation (a), by history matching production data (b) and the true model(c).

Fig. 3.27 shows the pressure match for two injectors (well 7 and well 8). In this figure and in similar figures, the line through circles represents the observed data; the line through the plus signs represents the calculated data based on the conditional realization obtained by history matching the production data and the line through the diamonds represents the calculated data based on the initial model, m_{uc}, before history matching. In Fig. 3.27 (a), the pressure data generated from the initial model falls below the observed data during the injection period, whereas in Fig. 3.27 (b), the pressure data generated using the initial model is greater than the observed pressure data. For both wells, the pressure data are matched very well. Fig. 3.28 (a) and (b) show the pressure data match for well 4 and the WOR match at the same well which is the only well at which water has broken through. We can see that at this well, both the pressure and the WOR are matched very well. Fig. 3.29 shows the
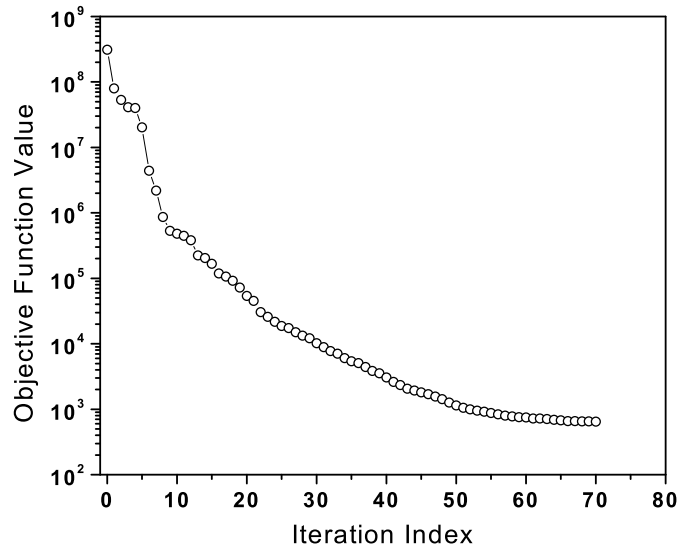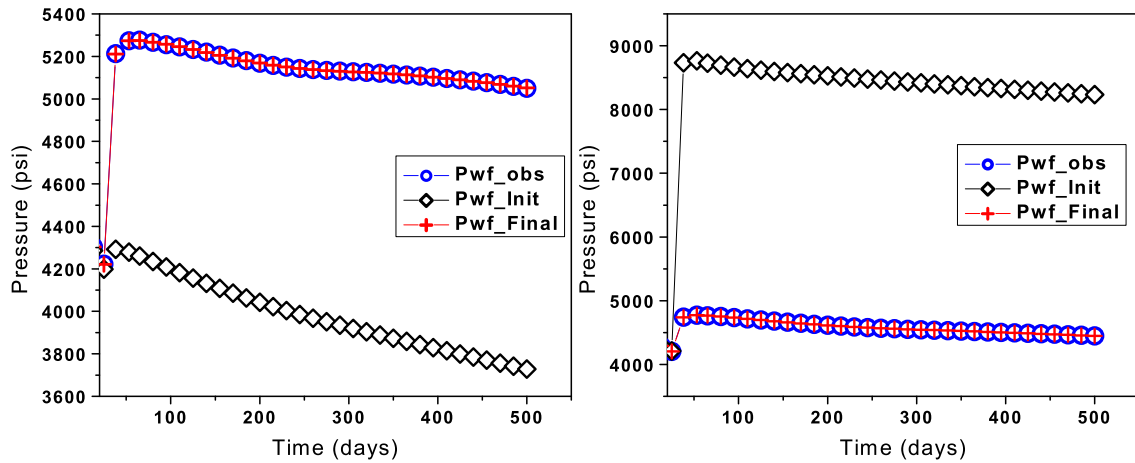
69

Fig. 3.26: Behavior of the objective function for the big model.

gas-oil ratio data match for two producers (well 3 and well 4). We can see that we obtained a very good GOR match for both wells. We obtained matches of comparable quality to these shown in Figs. 3.27 through 3.29 at all wells.
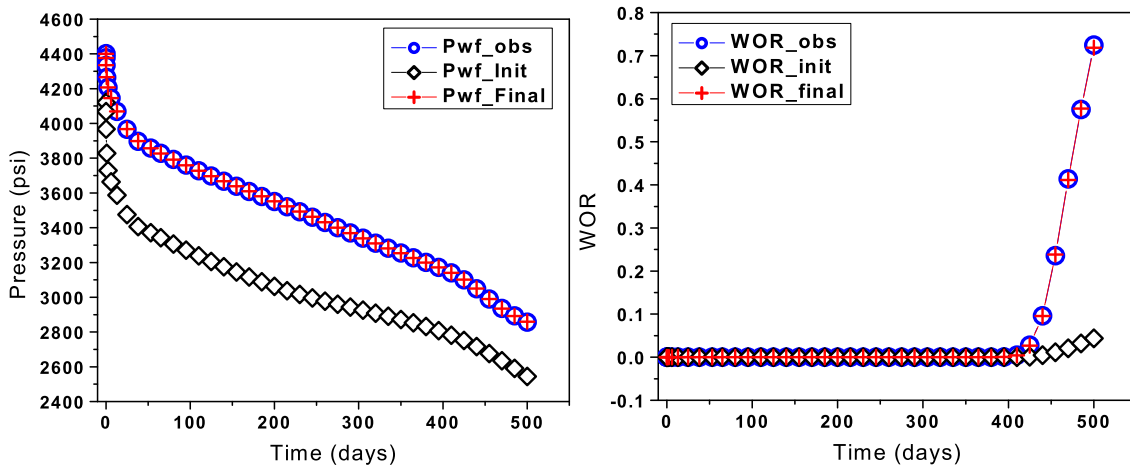

(a) well 7, injector

(b) well 8, injector

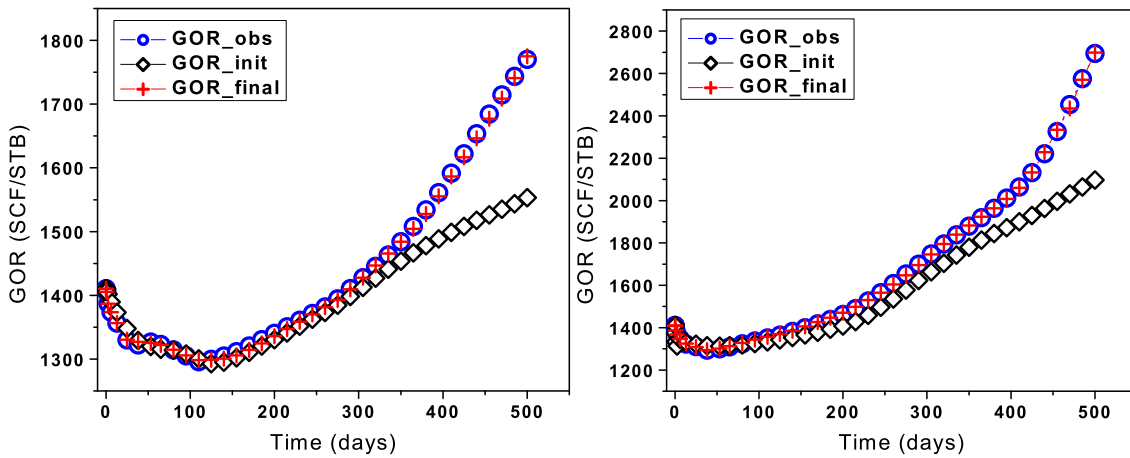Fig. 3.27: Pressure match at two water injection wells.

Two layers of the final permeability fields are shown in Fig. 3.24 (b) and Fig. 3.25 (b) and

(a) Pressure

(b) WOR

Fig. 3.28: Pressure and WOR match at well 4.

(a) well 3

(b) well 4

Fig. 3.29: GOR match from two wells.

the corresponding two layers from the true model are shown in Fig. 3.24 (c) and Fig. 3.25 (c), respectively. The corresponding unconditional permeability distribution is shown in (a) in these two figures. The conditional realization is close to the truth, in fact much closer than would normally be expected. This occurs because of the long correlation length in the vertical direction.

71

3.7.2 Conditioning to Observed Data with Noise Added

In this section, the objective function given by Eq. 3.14 is minimized. So the unconditional data were generated using Eq. 3.13. The objective function value is reduced from 313,023,514 to 5471 in 45 iterations. The squares in Fig. 3.30 show the behavior of the objective function when Eq. 3.14 is minimized. The circles in this figure show the behavior of the objective function when the true data without noise were used in Eq. 3.14. We can see that the objective function value at convergence is much bigger when unconditional data were used than when true data were used. Fig. 3.31 (a) shows the first layer of the model obtained by minimizing the objective function given by Eq. 3.14. Figs. 3.32 and 3.33 show results comparable to those shown in Figs. 3.28 and 3.29. Note although the match of data is not as close as in the case without noise in the data, we still obtain a reasonable match.
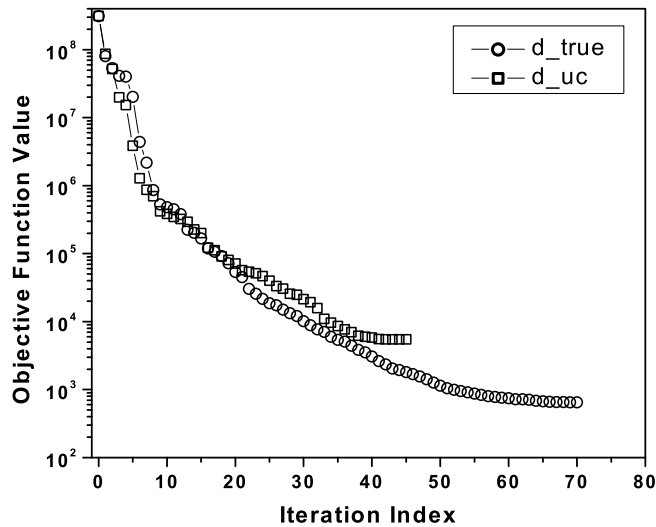

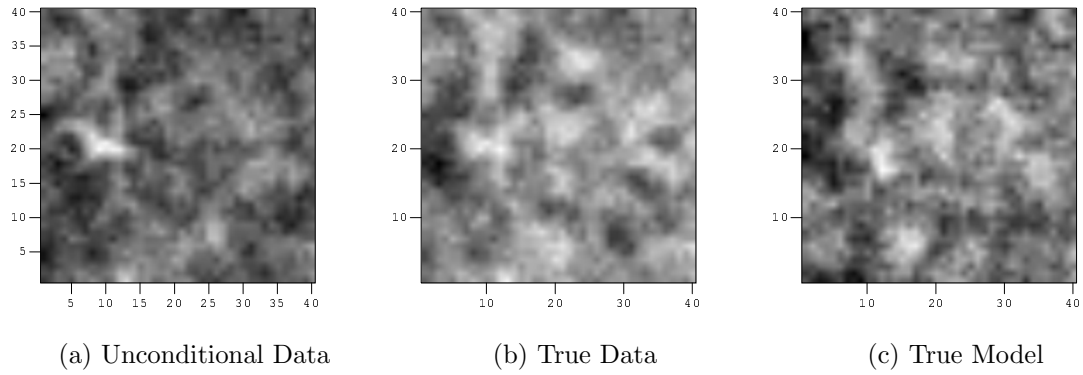Fig. 3.30: Behavior of the objective function when unconditional data were used.

(a) Unconditional Data   (b) True Data   (c) True Model

Fig. 3.31: The log-permeability field for layers 1 generated by history matching d_{uc} (a), by history matching true data (b) and the true model(c).
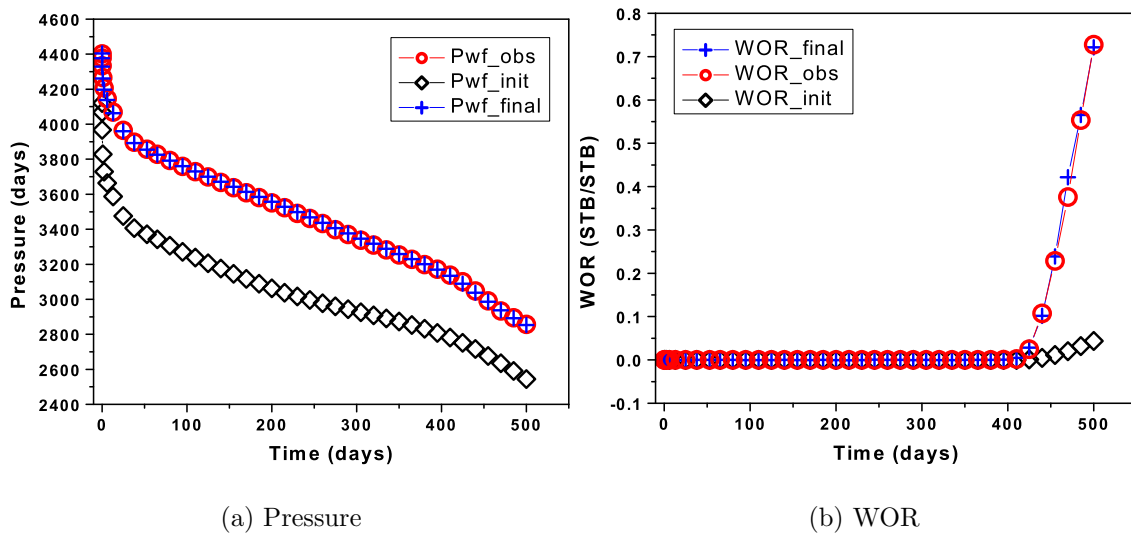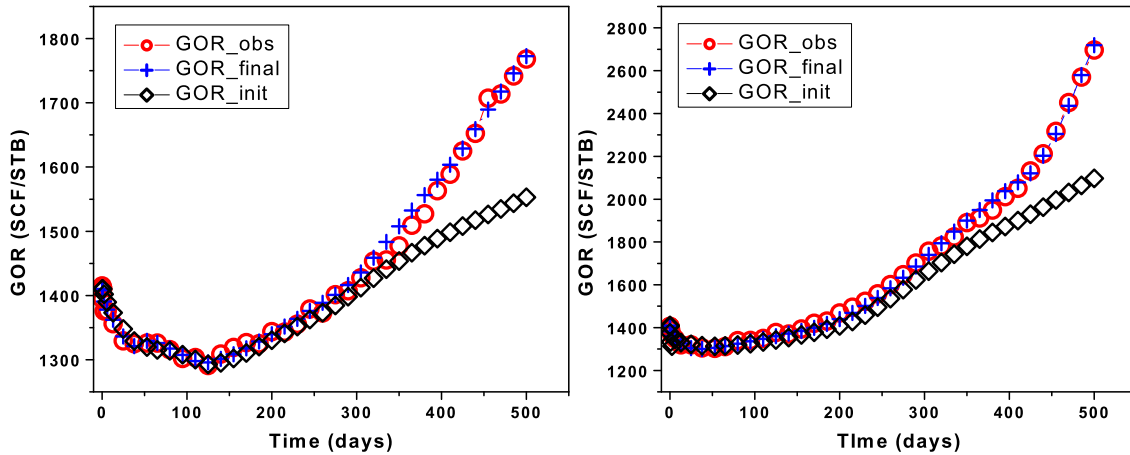

(a) Pressure   (b) WOR

Fig. 3.32: Pressure and WOR match at well 4, d_{uc}.

(a) well 3

(b) well 4

Fig. 3.33: GOR match from two wells, d_{uc}.

Chapter 4

HISTORY MATCHING, RESULTS AND DISCUSSION

4.0.3 Introduction

In this subsection, we consider matching production data for two field cases. The first case is a true field case, where we match static pressures from pressure buildup surveys for the Tengiz reservoir, which is an undersaturated oil reservoir. The second example case is actually a pseudo-field case. Based on known geologic information for a specific North Sea reservoir, we constructed a true reservoir model. Using this model, we generated synthetic production data (pressures and gas-oil ratios) from a forward simulation run. In generating the synthetic data, we used well locations and wells constraints that were similar to the actual wells in the field.

For both examples, production data were history matched using the limited memory BFGS algorithm described in the previous chapter to minimize the appropriate objective function.

4.1 Tengiz

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 4.1 shows that the central or "platform" portion of the reservoir is relativly flat with localized structural highs on the south and eastern edges. The platform is bounded by faults or lithologic breaks and surrounded by gently sloping "flanks" of carbonate debris; see Chambers (1997) and He and Chambers (1999) for additional details
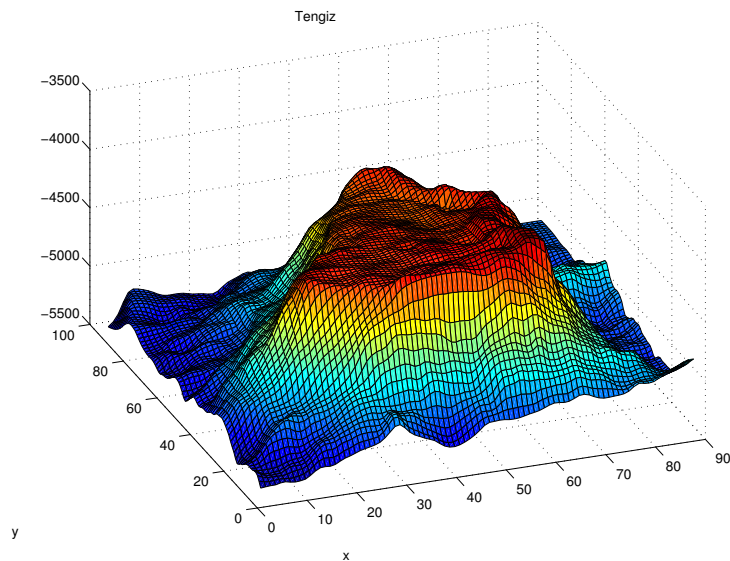
on the geology.

Fig. 4.1: A 3D plot of the Tengiz field.

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 4.2 shows a contour map of the top of the reservoir with well locations.

The initial permeability and porosity field are from a geostatistical model, generated by Chevron geoscientists. The cross plot of the permeability versus porosity in Figure 4.3 shows an overall least square regression trend of increasing porosity with increasing permeability. The porosity varies from less than 2 percent to over 14, while permeability varies from less than 0.01 md to over 9 md.

The main problem we encounter in history matching is that the measured pressures available correspond to buildup pressures. However, from an examination of the oil rate
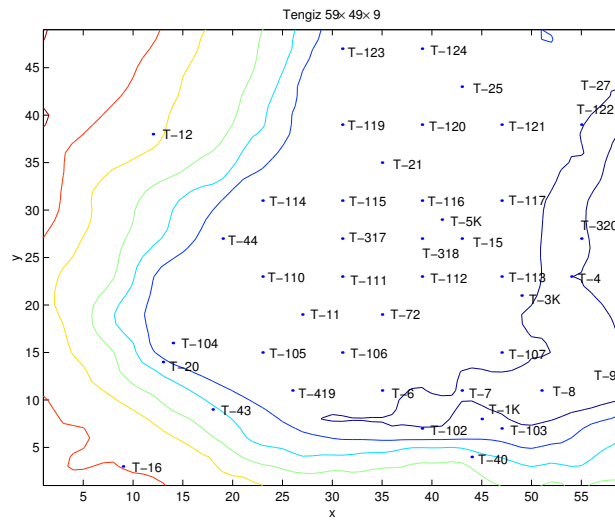
Fig. 4.2: A contour plot of the upscaled model with the well locations.
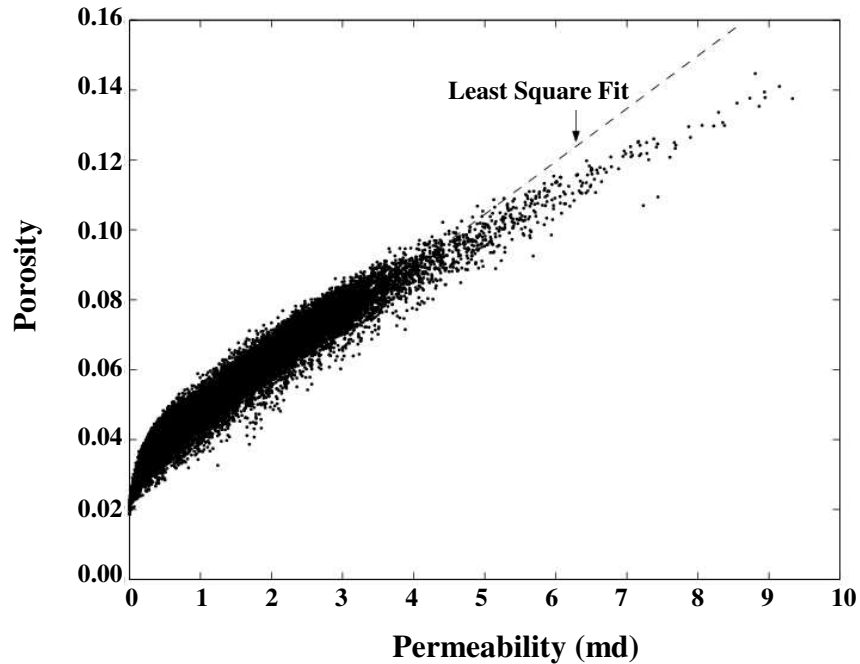

Fig. 4.3: Cross plot of porosity to permeability.

data provided, we found in most cases that the buildup pressure data at a well correspond
to periods when the monthly rate data indicate that the well is still flowing i.e., the actual

shutin time is less than one month, see, for example Figure 4.4. The reason of this behavior is that the input monthly data contain oil-rates for each well, which are averaged over a period of approximately 30 days. A short buildup period during this 30 days period will have a small influence of the total average oil-rate value. In Figure 4.4 and similar figures, the stars denote observed pressure data, i.e., the data we wish to history match; the oil rate versus time is represented by a dot-dashed curve at the bottom of the figure; the solid curve represents the flowing bottom hole pressure predicted from the simulator using the initial reservoir description based on static data; the dashed curve represents the bottomhole pressure predicted from a history matched model using the original geostatistical model, and the dotted curves represent bottomhole pressure predicted from a history matched model with a modified geostatistical model. The modified geostatistical model was obtained by increasing the correlation length in the vertical direction of the log-permeability field. This aspect is discussed in more detail later.
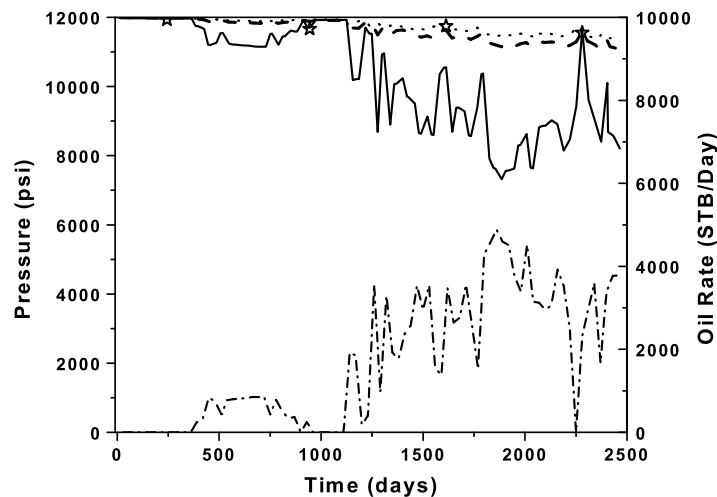

Fig. 4.4: Bottomhole pressure and oil rate of Well T-4. Located in gridblock (54,23).

4.1.1   History Matching

He and Chambers (1999) have previously history matched pressure data from the Tengiz reservoir. Their history matching procedure, however, is quite different from the one used in this work. In their results, they conditioned the permeability field in the neighborhood of a well to well-test pressure data by estimating a single permeability multiplier. The multipliers for each well were then interpolated to the whole field. The final model was obtained by

applying the multiplier field to the initial permeability field generated from the geostatistical model. In our work, we match all measured pressure data simultaneously, and adjust log-permeability in each grid cell individually during the history matching process without using large-scale permeability multipliers.

In the history matching problem, we used the scaled LBFGS algorithm discussed in the previous chapter to estimate the horizontal log-permeability. Porosity values were held fixed in the history matching problem. We fixed vertical permeability equal to ten percent of horizontal permeability. Well skin factors were fixed at each well based on values obtained from well tests. A total of 104 buildup pressures from 40 wells were used in the history matching procedure; buildup pressures were obtained at some time during the period from April 1991 to January 1998. In this history matching procedure, we specified the oil rate for each well based on historical data. The covariance matrix for the prior model for horizontal log-permeability was generated from a spherical variogram with ranges in x, y and z directions, respectively, equal to 6560, 4920 and 165 ft. The variance of horizontal log-permeability was equal to 0.225. After doing a history match based on this geostatistical model, we tried a second history match based on using a modified geostatistical model in which we increased the correlation length (variogram range) to 730 ft in the z direction . We did this experiment to investigate the effect of this range on the convergence properties of the algorithm. With the original variogram model, the vertical dimension of some simulation gridblocks was almost as large as the vertical range (165 ft) which is the same order as the correlation length in the z direction. In this situation, we expect that the horizontal log-permeability model will exhibit little correlation in the z-direction. In the history matching process, we set the variance of pressure measurement errors equal to 1 psi^2.

Figures 4.4 through 4.13 shows the field oil rate (dot-dashed curve) at the bottom of each figure, the observed buildup pressures (stars), the bottomhole pressures predicted from the initial reservoir model (solid curve), the bottomhole pressure predicted from the conditional realization obtained from history matching with short correlation length in z direction (dashed curve) and the bottomhole pressure predicted from the conditional realization with long correlation length in z direction (dotted curve), for wells T-4, T-6, T-15, T-16, T-44, T-102, T-104, T-106, T-120 and T-121. Note the conditional realizations obtained by history matching predict pressures in reasonable agreement with the observed pressures.

Note that, in most cases, the initial model of Tengiz predicts wellbore pressures lower than the measured pressure data, and in instances, these predicted pressures are hundreds of psi below the corresponding measured pressure data. In such situations, we expect that permeability will need to be increased in the neighborhood of a well to obtain a history
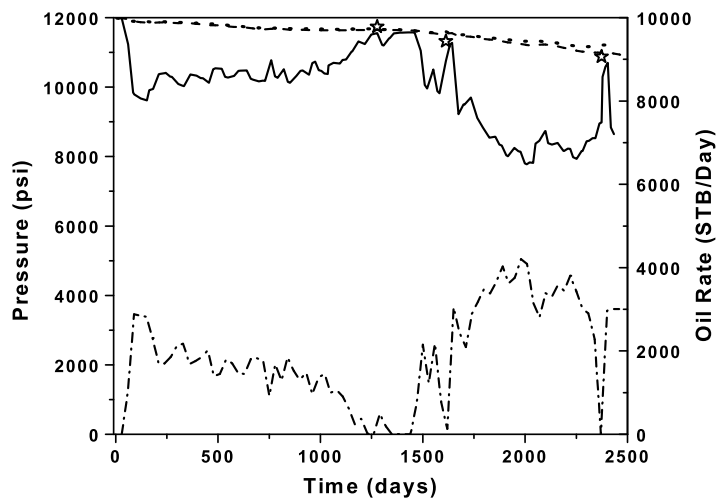
Fig. 4.5: Bottomhole pressure and oil rate of Well T-6. Located in gridblock (35,11).
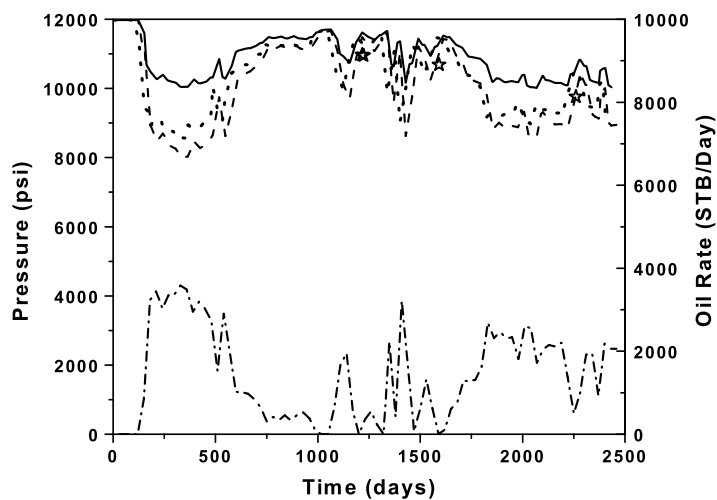

Fig. 4.6: Bottomhole pressure and oil rate of Well T-15. Located in gridblock (43,27).

match. In rare cases, the forward run of the initial model predicts wellbore pressures above the observed bottomhole pressures; see, for example Figs. 4.6 and 4.13. In these cases, one expects that one will need to decrease permeability near the well to obtain a history match of pressure data.

Both of the reservoir models obtained from the conditional realizations with short and long correlation lengths in the z direction predict pressures (dashed and dotted curves in Figures 4.4-4.13) that are in reasonable agreement with the measured buildup pressures,
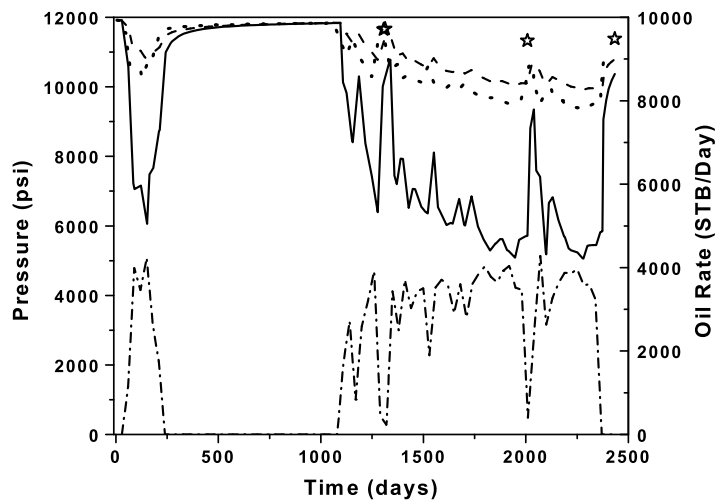
Fig. 4.7: Bottomhole pressure and oil rate of Well T-16. Located in gridblock (9,4).
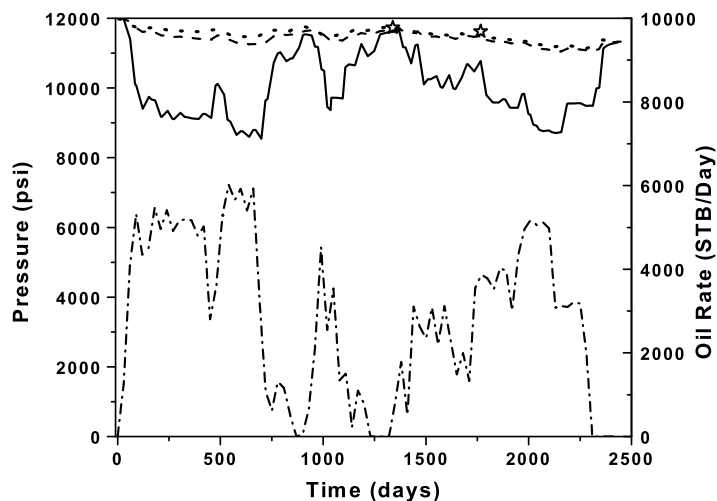

Fig. 4.8: Bottomhole pressure and oil rate of Well T-44. Located in gridblock (19,27).

and the results indicate that the bottomhole pressure mismatches are fairly small. Note, however, that many of these predicted pressures are much higher than those predicted with the initial rock property field. Moreover, in some wells the history matched models predict a relatively small pressure drop throughout the producing history (see Fig. 4.5, 4.8 and 4.10); thus indicates that the history matching process has resulted in significantly higher permeabilities. On the other hand, Figures 4.6 and 4.13 show that the history matched model also predicts pressures that are lower than those predicted from the initial reservoir
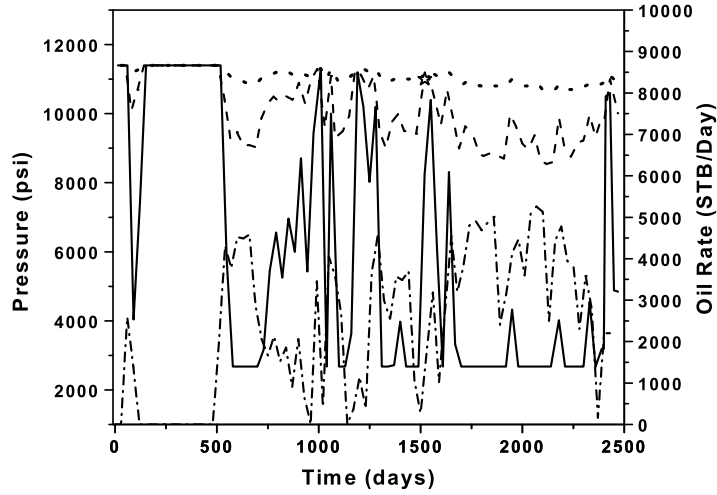
Fig. 4.9: Bottomhole pressure and oil rate of Well T-102. Located in gridblock (39,7).
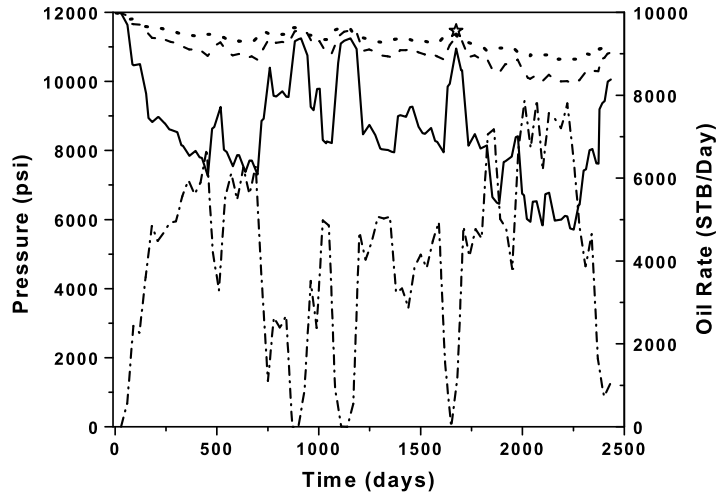

Fig. 4.10: Bottomhole pressure and oil rate of Well T-104. Located in gridblock (14,16).

model, which indicates that the history matching process has decreased the permeability in some regions.

Figures 4.14 through 4.17 show the horizontal log-permeability from the initial model (left plot in each figure) and the conditional realization of horizontal log-permeability based on a short correlation length in the z direction (right plot in each figure); results are shown only for model layers 1 through 4, i.e., only for the first four gridblocks in the vertical direction. Note that the platform part of Tengiz has a higher horizontal log-permeability than that of
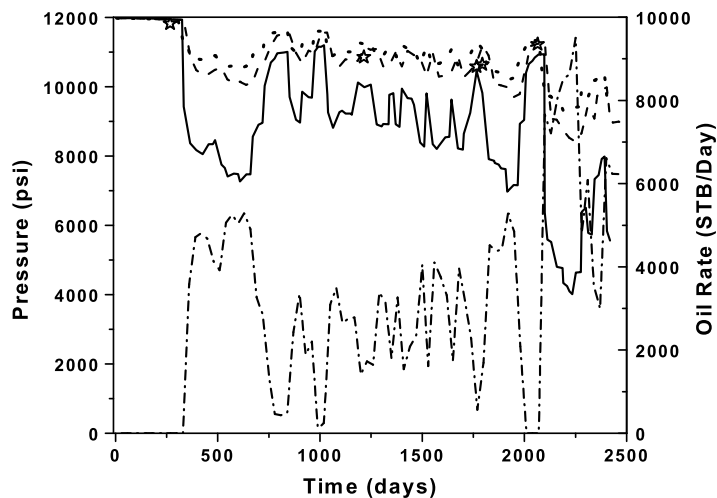
82

Fig. 4.11: Bottomhole pressure and oil rate of Well T-106. Located in gridblock (31,15).
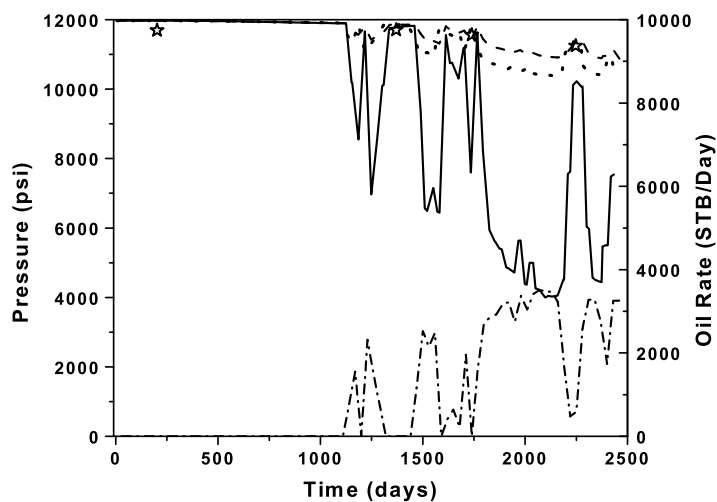

Fig. 4.12: Bottomhole pressure and oil rate of Well T-120. Located in gridblock (39,39).

the flanks, especially in the top layers. Also note that in much of the reservoir, the history matching process has resulted in a large increase in horizontal log-permeability; although it may be difficult to tell from the figures, the largest changes are in gridblocks close to the well locations. Also note that, because of the short vertical range (165 ft) of the variogram, the permeability is not strongly correlated in the vertical direction, e.g., the permeability distributions in layers 1 and 4 are quite different. The distance in the vertical direction between the center of layer 1 and layer 4 is on the order of 506 feet.
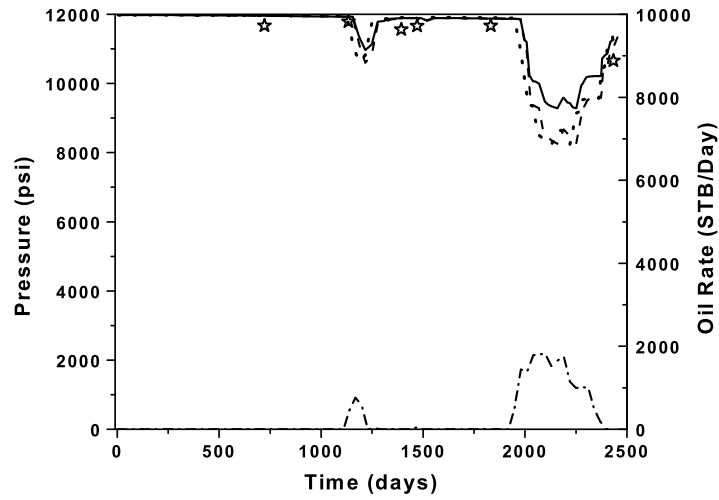
Fig. 4.13: Bottomhole pressure and oil rate of Well T-121. Located in gridblock (47,39).

Fig. 4.14: The conditional realization of horizontal log-permeability conditioned to p_{wf}, based on a short correlation length in z direction, layer 1.

Fig. 4.15: The conditional realization of horizontal log-permeability conditioned to p_{wf}, based on a short correlation length in z direction, layer 2.

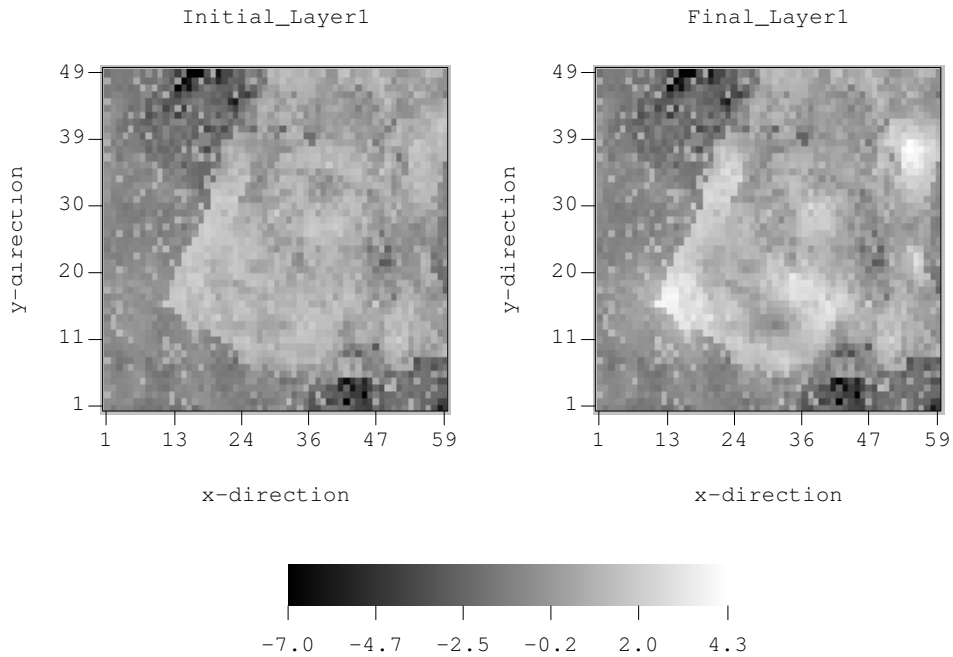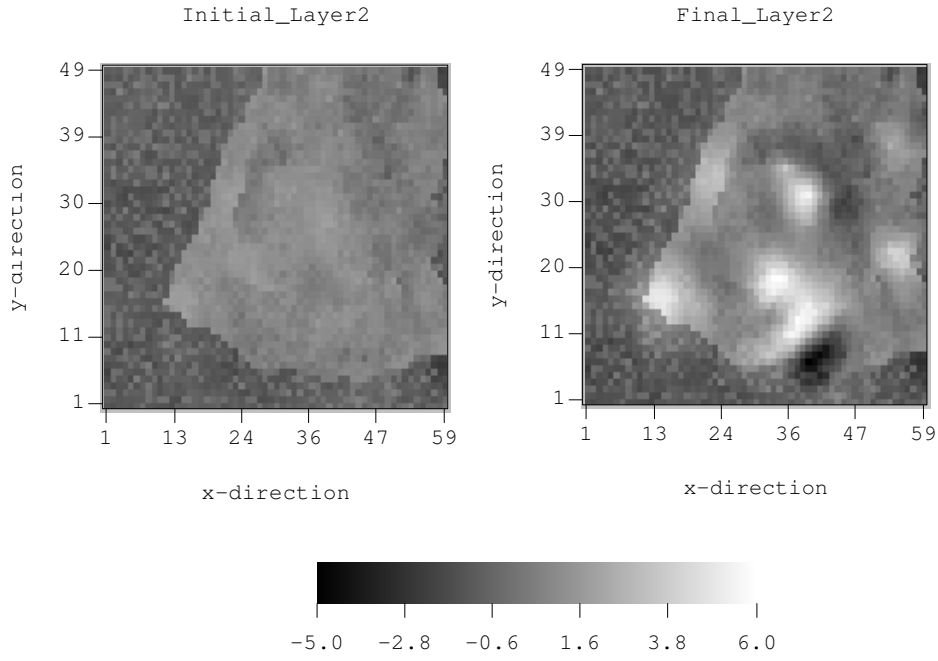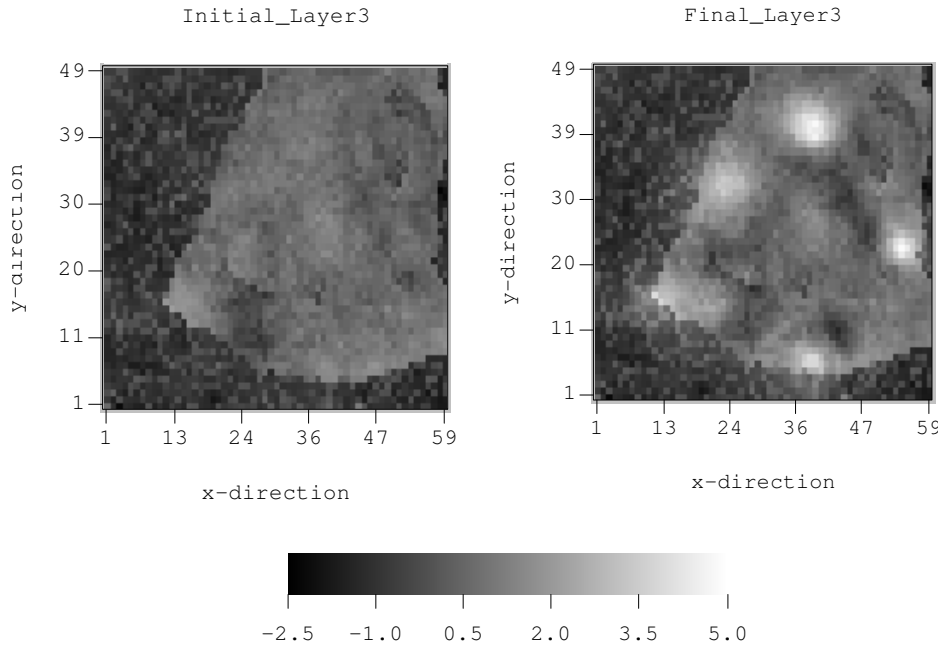Fig. 4.16: The conditional realization of horizontal log-permeability conditioned to p_{wf}, based on a short correlation length in z direction, layer 3.
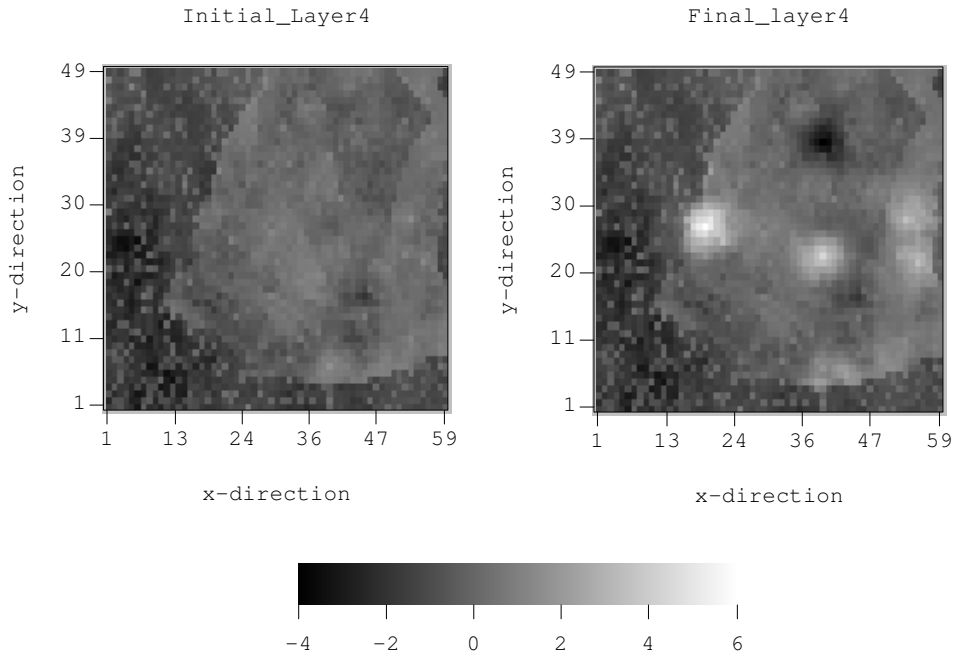
Fig. 4.17: The conditional realization of horizontal log-permeability conditioned to p_{wf}, based on a short correlation length in z direction, layer 4.

Because of the short correlation length, results that may on the surface seem surprising can be explained. For example, Well T-102 is located in areal $(39, 7)$ and completed in model layers 1, 2 and 3. From the results of Fig. 4.9, we see that the history matched model predicts higher wellbore pressures than the initial model, which indicates that permeability in the wellblock and nearby has been increased by history matching. A careful examination of the results depicted in Figs. 4.14 and 4.15 indicates that the permeabilities have actually been decreased in the layer 1 and layer 2 gridblocks that are penetrated by Well T-102. The results of Fig. 4.16, however, indicate that in and near the gridblock of layer 3 that has been penetrated by this well, horizontal log-permeability has been increased. Moreover the thickness averaged horizontal permeability for layers 1 through 3 has been increased at the areal location of this gridblock and this increase is consistent with the increased pressure predicted by the history matched model.

Figure 4.18 shows the difference between the horizontal log-permeability field obtained by history matching and the initial horizontal log-permeability field. Again, the results indicate that large changes have been made to the permeability field. Note that some large decreases have also been made, for example in some regions corresponding to about the the 37th to 45th gridblock in the x-direction. The decrease of permeability in this region appears

to be consistent with what should be expected based on the pressure data. For example, Well T-15 is located in areal gridblock $(47, 27)$ and is completed in layers 2 through 3. The pressure behavior of well T-15 (see Fig. 4.6) indicates that we need decrease permeability in the neighborhood of this well to history match the pressure data because the pressure data predicted with the initial model is higher than the observed pressure data.
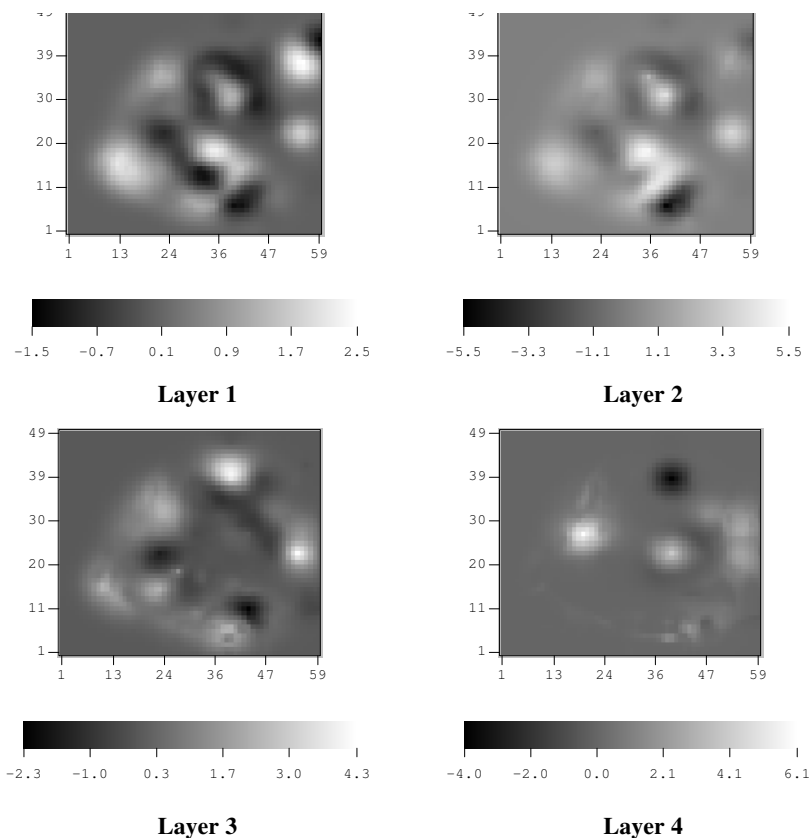

Fig. 4.18: The change in the horizontal log-permeability by conditioning to p_{wf} for the first four layers, based on a short correlation length in z direction.

Figure 4.19 shows the change in the horizontal log-permeability field obtained by history matching when the correlation length was increased from 165 feet to 730 feet. In this case, the changes in horizontal log-permeability are in the top four top layers. Now we can also see clearly that the permeability has been decreased in the vicinity of well T-102 which is consistent with our earlier discussion and the results of Fig. 4.9. Note the results of Figs. 4.4 through 4.13 indicate that we do not get significantly different pressure matches when the long correlation length is used to construct the prior covariance matrix. Figure 4.20 shows the rate of convergence of the LBFGS algorithm for the two different cases. The convergence rates

indicate that the objective function decreases rapidly for both cases in the early iterations, and both cases converge almost to the same value in about 15 iterations. The objective function is reduced from a value of 5.8×10^7 to 3.1×10^6 and the maximum mismatch has been changed from a value of approximately 2300 psi to approximately 500 psi.
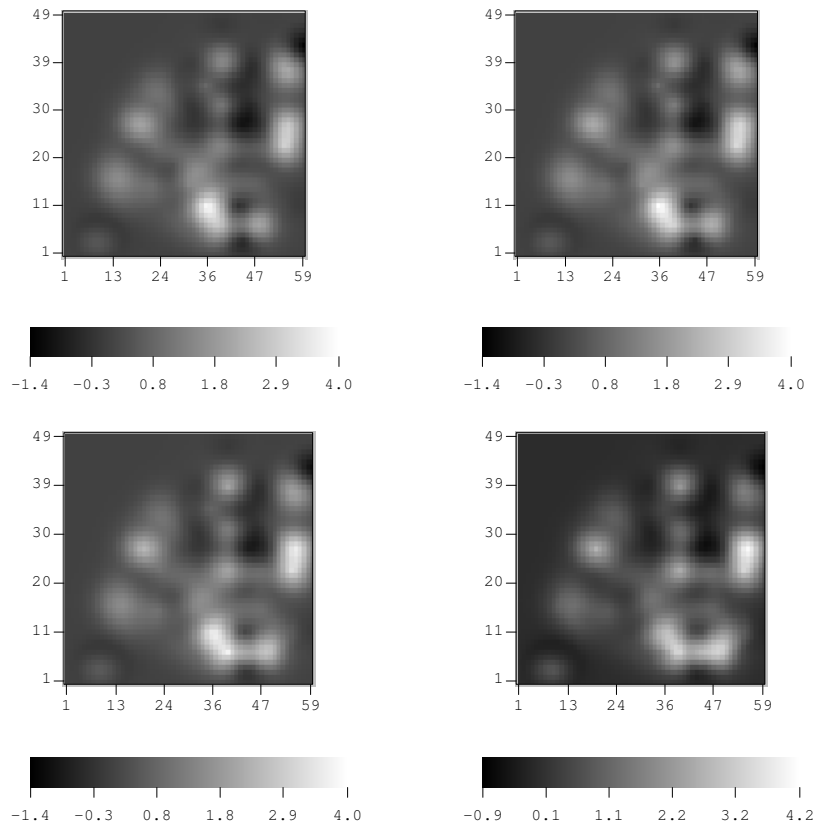
Fig. 4.19: The change in the horizontal log-permeability by conditioning to p_{wf} for the first four layers, based on a long correlation length in z direction.

4.1.2 Comments

Although we greatly improved the pressure match by history matching, it is far from a perfect match and to obtain the match, we made extremely large changes to the log-permeability field during the history matching process. At this point, it is unclear whether we could obtain a match of similar or better quality with smaller changes in the permeability field. It is clear, however that the history matching is complicated by the quality and interpretation of the pressure data. Many "measured" buildup pressure data at a well correspond to periods

Fig. 4.20: Behavior of the objective function.

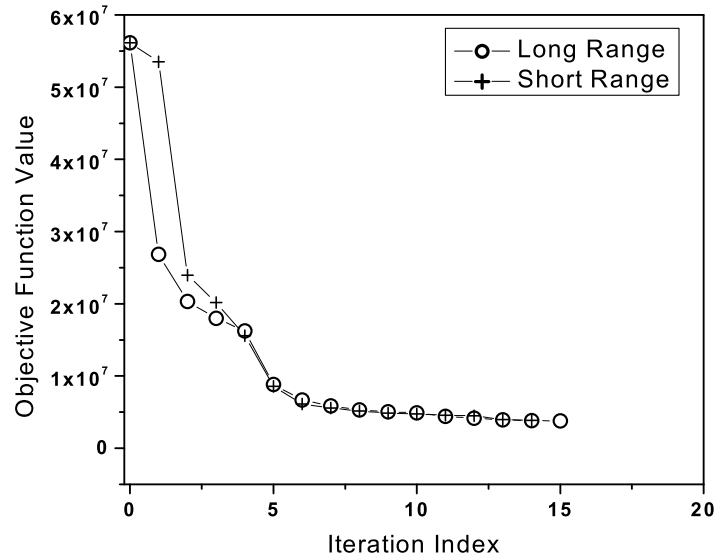when the well is still flowing according to the input monthly rate data, so our automatic history matching procedure treats them as true flowing wellbore pressures. Because the well is flowing at times corresponding to these measured shutin pressures, the optimization algorithm makes large increases in permeability in an attempt to enable the well to flow with little or no pressure drop.

The conditional realizations obtained from the automatic history matching procedure indicated that a short and a long correlation length in the z direction gave quite different results for the permeability field. When we were operating with a short correlation length, the changes of horizontal log-permeability were fairly large layer by layer, but a calculation of the averaged volumetric horizontal log-permeability in each column of gridblocks confirmed that the results were in agreement with expectations based on the bottomhole pressures. The changes layer by layer were relatively small when we were operating with a long correlation length as expected, and the plots of the horizontal log-permeability changes verified that they were in accordance with the predicted bottomhole pressures from the conditional realization. Moreover, we treat the measured pressures as wellbore pressures, whereas, in reality, they represent an approximation to the average pressure in the well's drainage area at the instant of shut-in. This average pressure is higher than the wellbore pressure and would require smaller changes in the permeability in order to match this pressure, i.e., if treated the pressure observations as gridblock pressures in flowing periods, the initial pressure mismatch term

would be smaller in cases when the pressure predicted by the initial model is much lower than the observed pressure. As a consequence, the history matching procedure would make smaller changes in horizontal log-permeability if we matched the observed pressures with gridblock pressures. Our objective, however, was to test our automatic history matching procedure on a representative field problem, and this example demonstrates that the automatic history matching technology we have developed is feasible for field problems.

4.2 Field Example – Oseberg Reservoir

We consider a synthetic problem which is based on the Oseberg reservoir in the Norwegian sector of the North Sea. The reservoir consists of three distinct geological zones, Etive, Rannoch and Oseberg. Etive is the top zone and Oseberg is the bottom zone. These two zones are separated by Rannoch which is a relatively tight layer. There is vertical communication between the three zones. In our simulation study, we simulate only one half of the reservoir using a $39 \times 25 \times 10$ grid. Only one vertical gridblock is used in the Etive and Rannoch layer. So layer 3 through 10 are used to model the Oseberg zone. The gridblock size in the x-direction is equal to 328 ft in the central part of the reservoir, and the gridblock sizes expand gradually towards the ends from 328 ft to 2624 ft. The gridblock sizes in the y-direction are uniform and equal to 656 ft. Gridblock sizes in the z-direction are non-uniform with values equal to 23.0 ft in Etive, 16.5 ft in Rannoch and 11.5 ft in Oseberg. Initial reservoir pressure is 4071 psi at the depth of 8192 ft subsea, and the initial bubble point pressure is 3771 psi at this same datum. The reservoir has a gas cap at the top and an aquifer at the bottom. The initial gas-oil contact is at 8192 ft subsea and the water-oil contact is at 8918 ft subsea. Fig. 4.21 (a) shows the surface plot of the reservoir top and (b) shows the overview of the reservoir top with well locations indicated by black squares for producers and by white squares for injectors. Note that the reservoir has a significant dip. The oil column is separated from the aquifer by a tar mat. The initial permeability and porosity field are based on a geostatistical model, and the true synthetic model is generated by using a Gaussian co-simulation algorithm.

Two gas injection wells, which are indicated by the white squares in Fig. 4.21 (b), are located in the gas cap and five producing wells, which are indicated by the black squares in Fig. 4.21 (b), are located in the oil zone. The producing wells are named PROD1, PROD2, $\cdots$, PROD5 and the two gas injection wells are named INJ1 and INJ2 respectively. All these wells are fully-penetrating, i.e., all layers are open to flow. The areal locations of the five producing wells are in gridblocks (32,3), (32,8), (32,13), (32,18) and (32,23) respectively.
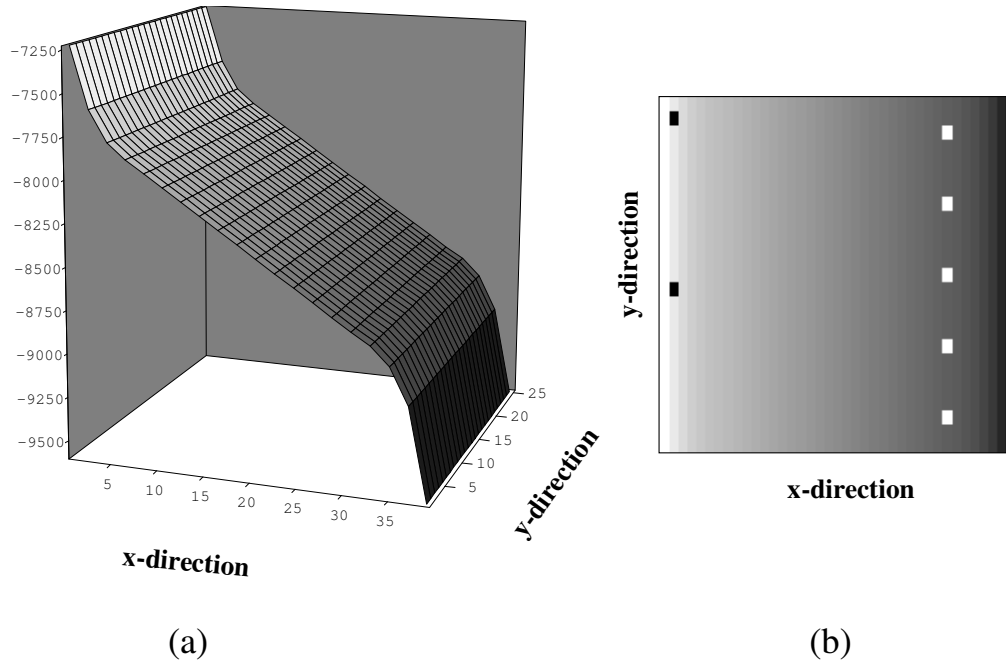
(a)　　　　　　　　　　　　　(b)

Fig. 4.21: Top depth of Oseberg reservoir and well locations.

Fig. 4.22 (a) through (e) show the production rate history for each well respectively. The production rate history for all producing wells are plotted together in Fig. 4.22 (f). We can see that these five wells are open to flow with a high flow rate (15000 STB/Day) in a sequential order (at 30, 90, 180, 270 and 360 days in turn). The oil rates shown in Fig. 4.22 are the rates which were specified as the target rates in the simulation runs. The minimum bottom-hole pressure (MINBHP) which is fixed at 2000 psi is used as the producing constraint whenever production at a well's target rate causes the bottom-hole pressure to fall below 2000 psi, this constraint will be switched to be the producing target, i.e., the well will be produced at a fixed bottom-hole pressure of 2000 psi. The maximum water-oil ratio (MAXWOR) which is specified to be 50 STB/STB and the maximum gas-oil ratio (MAXGOR) which is specified to be 561,000 MSCF/STB are used as the producing economic limits. When these economic limits are violated at a certain well, then the corresponding well will be shut in. INJ1 and INJ2 are gas injection wells which are located in gridblocks (2,12) and (2,24) respectively. The gas injection rate history for the two gas injectors are plotted in Fig. 4.23. INJ1 starts to inject gas at 900 days and INJ2 starts to inject gas at 990 days. No producing constraint is specified for the injection well. The top, bottom and outer reservoir boundaries are all assumed to be no-flow boundaries, and the capillary pressure is assumed to be negligible. In the oil column, initial oil saturation is 0.885 and the initial water saturation is equal to the
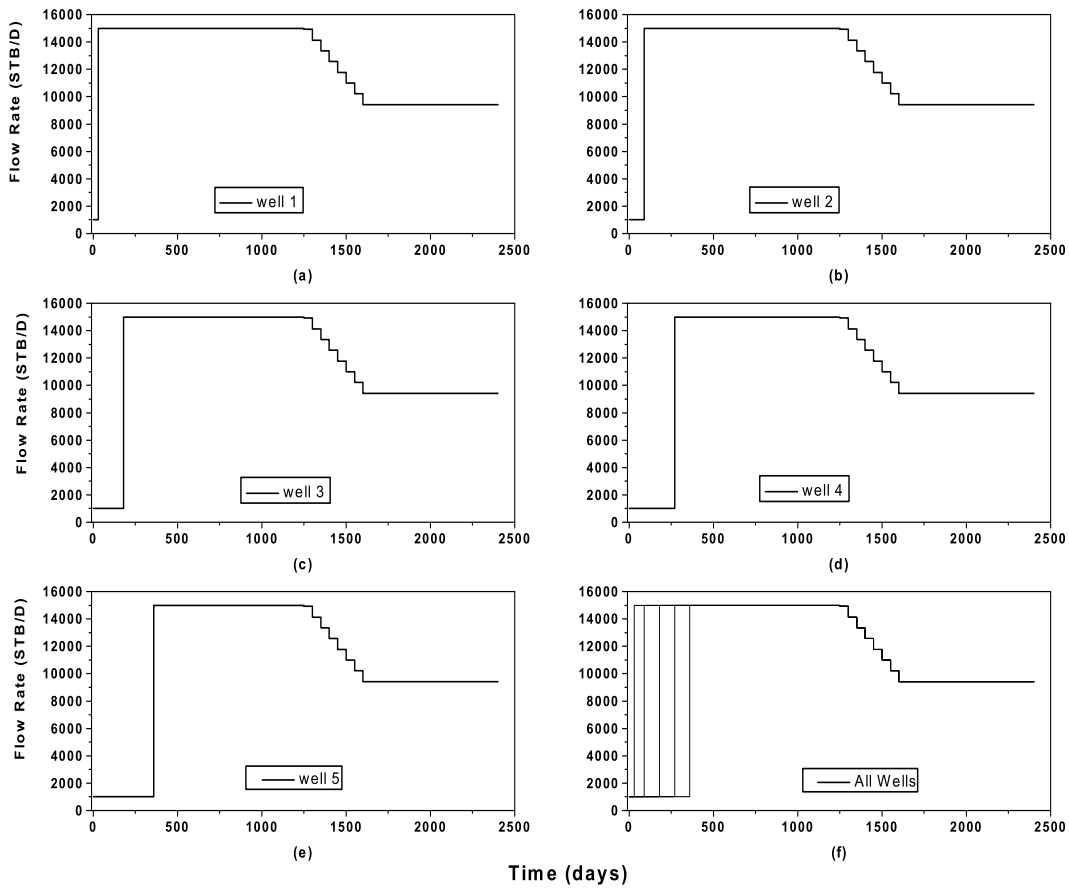
91

Fig. 4.22: Production rate history for the five producing wells.

irreducible water saturation which is equal to 0.115. In the gas cap, the initial gas saturation is 0.885. The water saturation in the gas cap is the irreducible water saturation and there is no oil in the gas cap. The oil formation volume factor (FVF) and the oil viscosity are shown in Fig. 4.24. Gas FVF and viscosity are shown in Fig. 4.25. Water FVF and viscosity are 1.03 RB/STB and 0.34 cp, respectively, at the reference pressure of 4219.5 psi. The water-oil two-phase relative permeability is shown in Fig. 4.26 (a) and oil-gas two-phase relative permeability is shown in Fig. 4.26 (b) respectively. Stone's model II is used to generate three-phase oil relative permeability; see Aziz and Settari (1979).
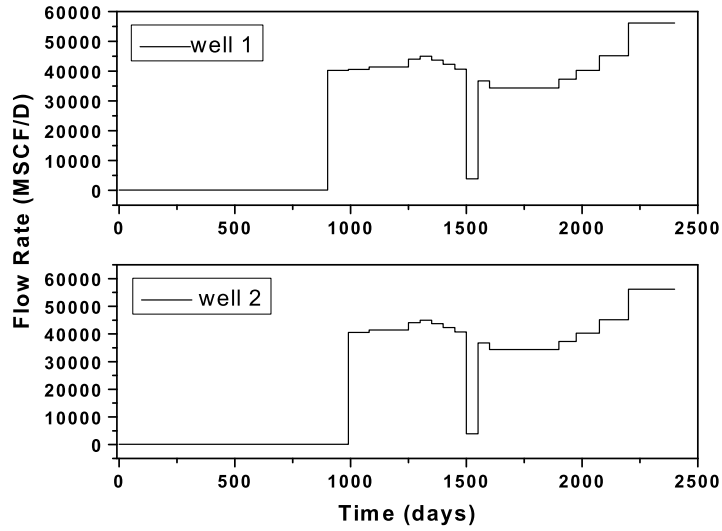
Fig. 4.23: Injection rate history for the gas injection wells.
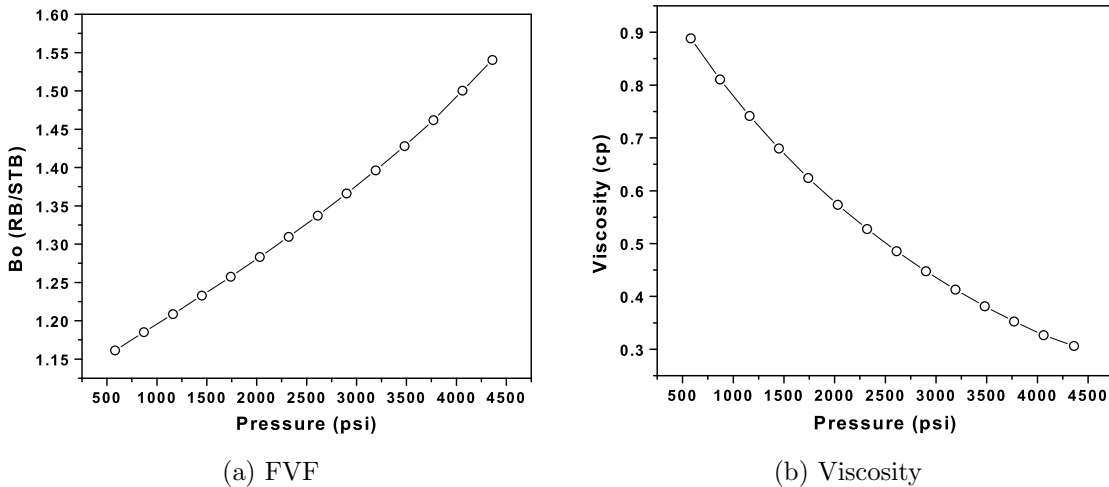
(a) FVF

(b) Viscosity

Fig. 4.24: Oil FVF and viscosity.

4.2.1 Reservoir Model

A non-isotropic exponential covariance structure, with ranges $a_x = 1968$ ft, $a_y = 6555$ ft and $a_z = 20$ ft was used. The statistical descriptions of the prior models for Etive and Rannoch, and Oseberg are given in Tables 4.1 and 4.2 respectively. Note that horizontal and vertical log-permeability in Oseberg decrease linearly from top to bottom, i.e., from layer 3 to layer 10. In Table 4.2, only the means for model layer 3 and 10 are specified. Means for the intermediate model layers are obtained by linear interpolation. Note that Rannoch has a low

(a) FVF

(b) Viscosity

Fig. 4.25: Gas FVF and viscosity.

(a) Water-oil

(b) Oil-gas

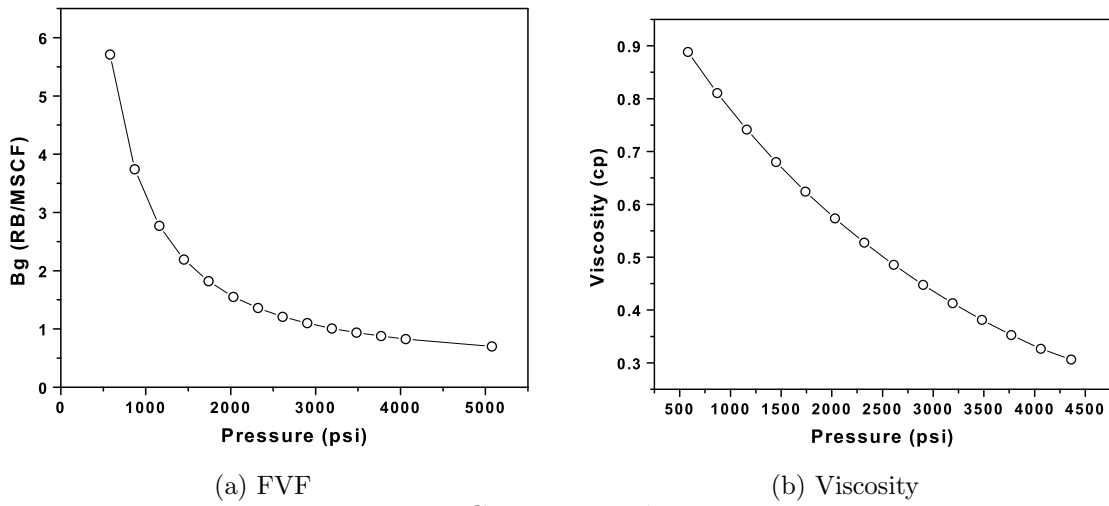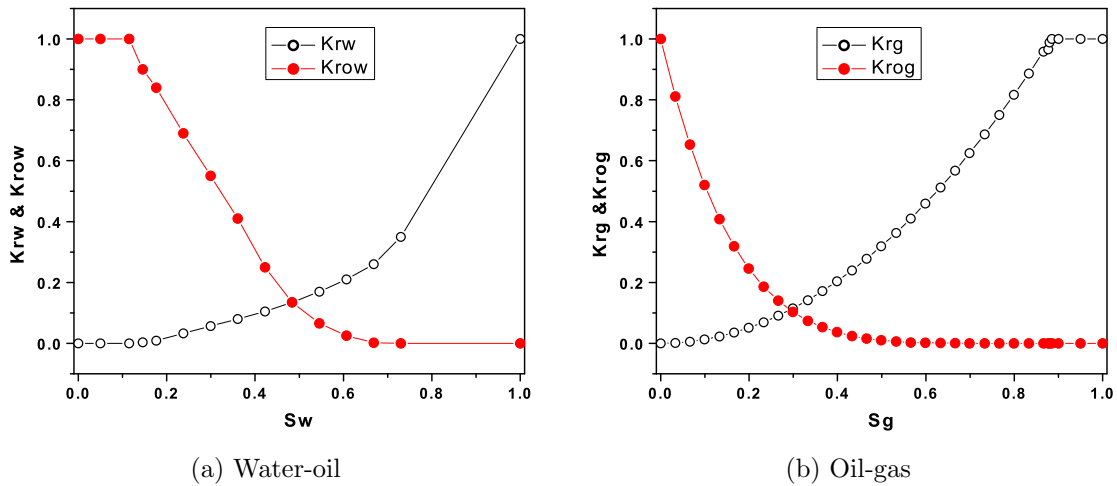Fig. 4.26: Relative permeability.

permeability compared to Etive and Oseberg, and acts as a flow restriction between the Etive and Oseberg formations. The unconditional realization of $\ln(k)$ and $\ln(k_z)$ for Oseberg layers was generated with fixed means. The mean value of 7.48 for $\ln(k)$ and 6.47 for $\ln(k_z)$ were used for generating the unconditional realization. Therefore, the unconditional realization for $\ln(k)$ and $\ln(k_z)$ do not have any trend vertically. The mean used for generating the unconditional realization for $\ln(k)$ and $\ln(k_z)$ for Etive layer are 6.02 and 4.61, respectively. The same means as used in generating the true Rannoch layer were used to generate the unconditional realization for $\ln(k)$ and $\ln(k_z)$ for the Rannoch layer. The correlation between horizontal and vertical log-permeability, ρ_{k,k_z}, was equal to 0.8, and the correlations between

log-permeability and porosity, $\rho_{k,\phi}$, $\rho_{k_z,\phi}$, was equal to 0.3. An unconditional realization generated based on the geostatistical parameters mentioned above with a specific seed was used to generated the true model. In order to provide the decrease trend for the permeability for the Oseberg zone, we added a constant for the ith layer of the Oseberg layer to the unconditional realization by the following way

$$m_{\text{true,i}} = m_{\text{uc,i}} + (7.8 - 7.48) - (i - 3) * (7.8 - 6.3)/8 \tag{4.1}$$

to generate the true horizontal permeability field. The true vertical permeability field was generated similarly. Then we used a different seed to generate initial guesses for horizontal and vertical log-permeability. In the history matching process, the porosity field was fixed and equal to the true porosity. The tar-zone is located in gridblocks centered at (x_i, y_j, z_k), $i = 33, 34$, $1 \leq j \leq 25$, $1 \leq k \leq 10$. In the tar-zone, we set the horizontal and vertical permeability equal to 1 md in the true case. When history matching to generate a realization by the randomized maximum likelihood method, we first generate an unconditional realization m_{uc} from the prior model and then modify m_{uc} by setting the entries corresponding to $\ln(k)$ and $\ln(k_z)$ for the tar-zone to zero. As noted previously, the tar-zone prevents water conning from the aquifer. Fig. 4.27 (a) through (c) show the 3D cube of the horizontal log-permeability, $\ln(k)$, vertical log-permeability, $\ln(k_z)$, and porosity ϕ, respectively, for the true model. Fig. 4.28 (a) through (c) show the middle x-z cross-section corresponding to true $\ln(k)$, $\ln(k_z)$ and ϕ respectively.

	Etive		Rannoch	
	Mean	Variance	Mean	Variance
$\ln(k)$	7.5	1.2	2.1	1.8
$\ln(k_z)$	6.3	1.8	0.15	2.2
ϕ	0.14	0.002	0.10	0.001

Table 4.1: Prior model of Etive and Rannoch.

	Mean top	Mean bottom	Variance
$\ln(k)$	7.8	6.3	0.4
$\ln(k_z)$	6.4	4.4	0.8
ϕ	0.22	0.22	0.001

Table 4.2: Prior model of Oseberg.

(a) Horizontal log-perm. (b) Vertical log-perm. (c) Porosity
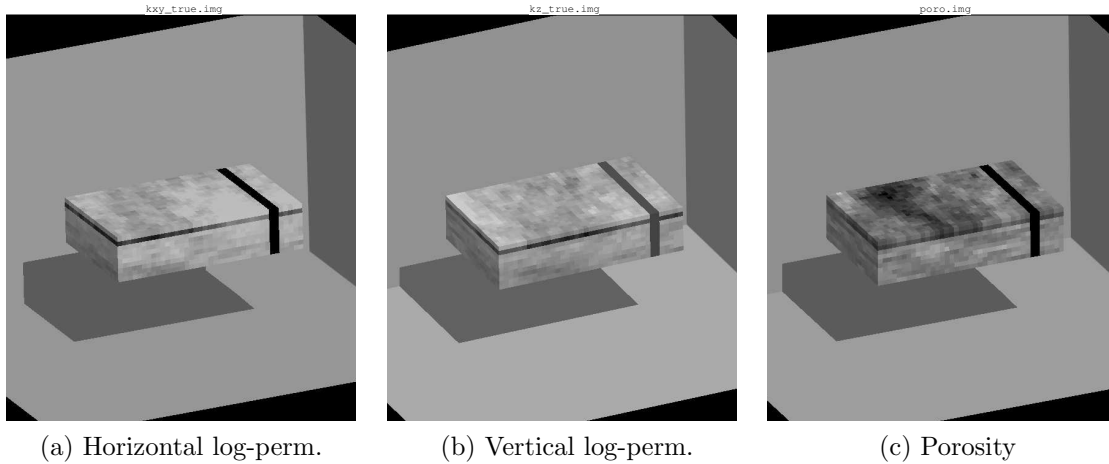
Fig. 4.27: Permeability and porosity for the true model.

(a) Horizontal log-perm. (b) Vertical log-perm. (c) Porosity
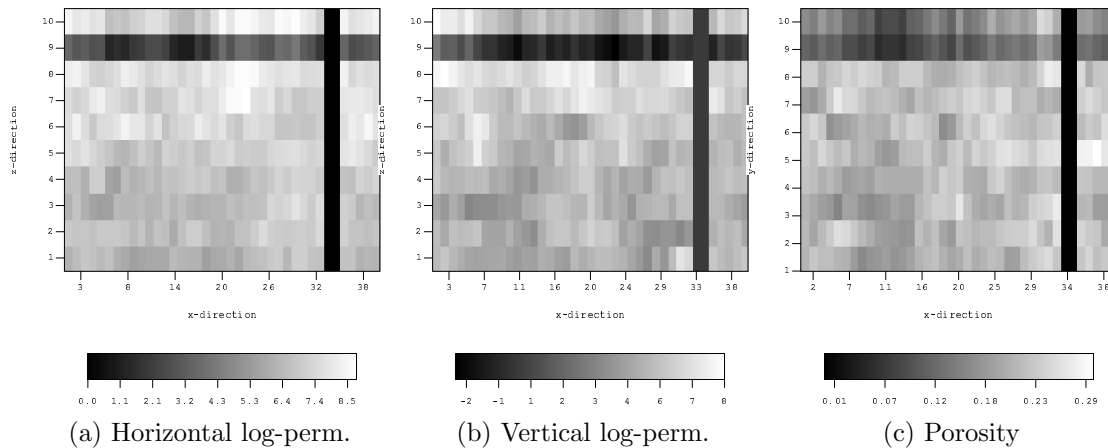
Fig. 4.28: Permeability and porosity for the true model.

Fig. 4.29(a) shows the observed wellbore pressures, which are generated based on the true model, for all seven wells including both producers and injectors. Fig. 4.29(b) shows the GOR data from the five producing wells. From this figure, we can see that the gas breaks through at the five producing wells in a sequential order due to the fact that the producers start to produce in a time-delayed scheme. the exception is that the PROD2 experiences gas breakthrough earlier than PROD1. This occurs because PROD2 is much closer to the injection well than PROD1. Fig. 4.30(a) shows wellbore pressures obtained from the unconditional realization which is used as initial model (initial guess) in the history matching procedure. Comparing Fig. 4.30(a) with Fig. 4.29(a), we can see that the behavior

of the wellbore pressure for the initial model is different from the behavior of the wellbore pressure for the true model. Fig. 4.30(b) shows the GOR data for the initial model. We can see that, for the initial model, only PROD1 and PROD2 had gas breakthrough and the breakthrough happened very late compared with the true model. Breakthrough did not happen for PROD3 to PROD5.
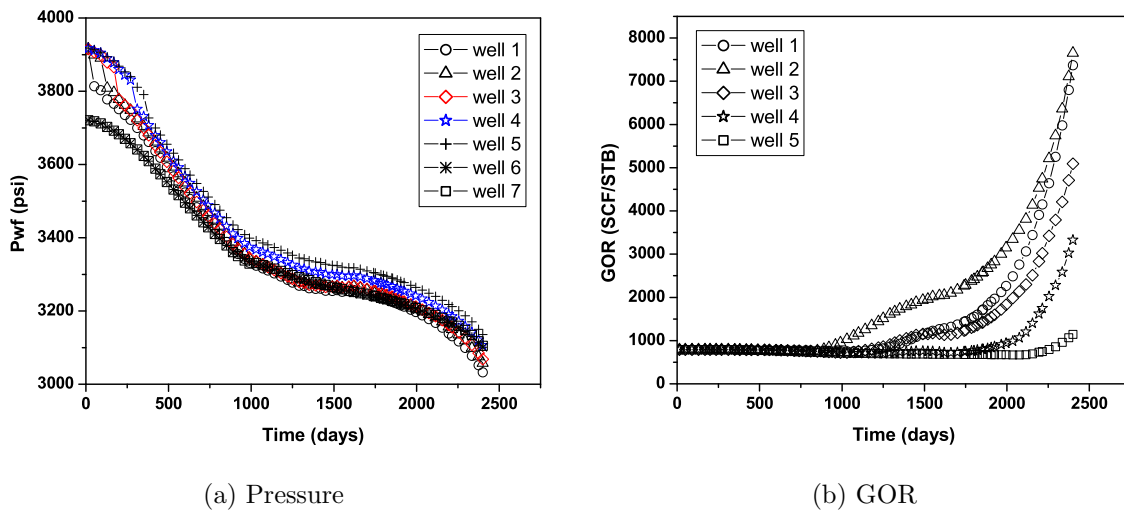

(a) Pressure

(b) GOR

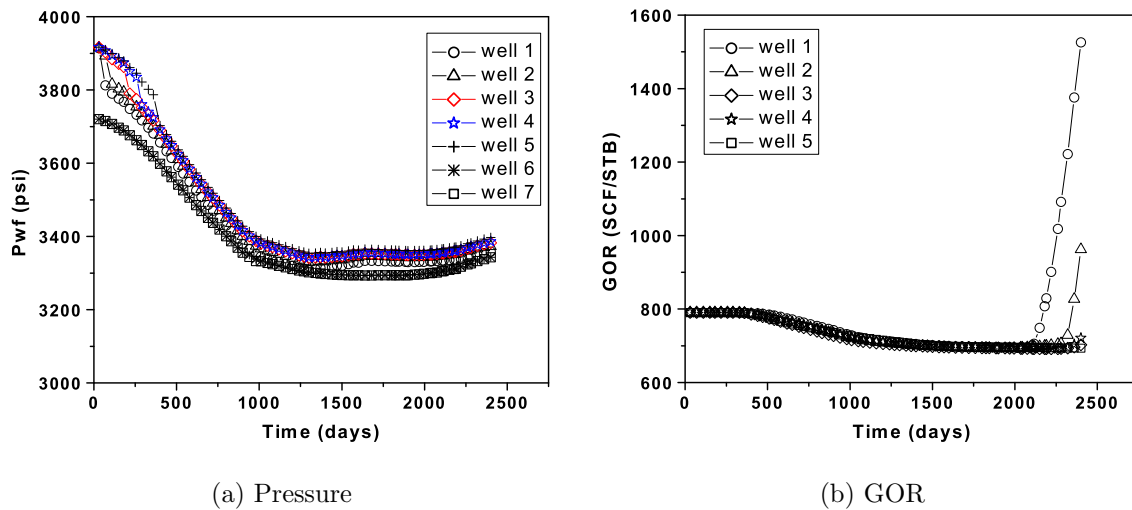Fig. 4.29: Observed production data.

(a) Pressure

(b) GOR

Fig. 4.30: Calculated data for the initial model.

4.2.2 History Matching

The observed data used for history matching were constructed by running the simulator with the true model for a total time of 2400 days. Several data sets are used for history matching. Data set 1 contains the wellbore pressures from both the producing wells and injection wells and the GOR data from the producers. There are 71 measurements for each type of data at each well. So the total number of data history matched is 852 for the first data set. Data set 2 contains only the wellbore pressure from the five producing wells and the two gas injection wells. No GOR data are used inthe history matching process. The total number of data history matched for the second data set is 497. Data set 3 contains only the GOR data from the five producing wells. In this case, the total number of data used for history matching is 375. No noise was added to the data, i.e., we history matched the objective function given by Eq. 3.14 with d_{uc} replaced by d_{true}. However, in generating the covariance matrix for measurement errors, we assumed wellbore pressure measurement errors to be independent identically distributed Gaussian random variable with mean zero and variance equal to 1 psi^2 and gas-oil ratio measurement errors to be independent identically distributed Gaussian random variables with mean zero and variance equal to 25.0. The limited memory BFGS algorithm was used for minimization.
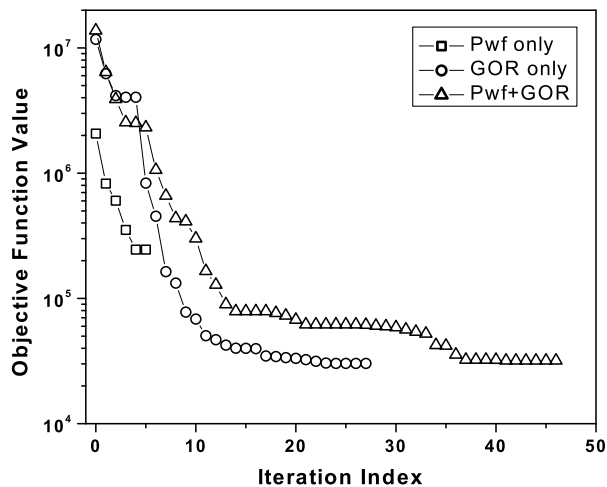

Fig. 4.31: Behavior of the objective function.

With data set 1, even though no noise was added to the true data, the objective function value is only reduced from 1.4×10^7 to 3.2×10^4 in 46 iterations which is much larger than N_d. As shown below, however, we obtained reasonable matches of the data. The behavior

of the objective function is shown by the triangles in Fig. 4.31. Pressure data matches for three producing wells, PROD1, PROD2, PROD4 and one injection well, INJ2, are shown in Fig. 4.32. In these four figures and in similar figures, the circles represent the observed data, the diamonds represent the calculated data based on the initial model and the plus signs represent the calculated data based on the final model after history matching the production data. From these four figures, we can see that we get a reasonably good match for pressure data at both producing wells and the injection well. The largest pressure mismatch at any wells was 19 psi. Matches of comparable quality were obtained at the other three wells. Fig. 4.33 shows the GOR data match for PROD1, PROD2, PROD3 and PROD4. Again a very good match for GOR data at these four wells is obtained.
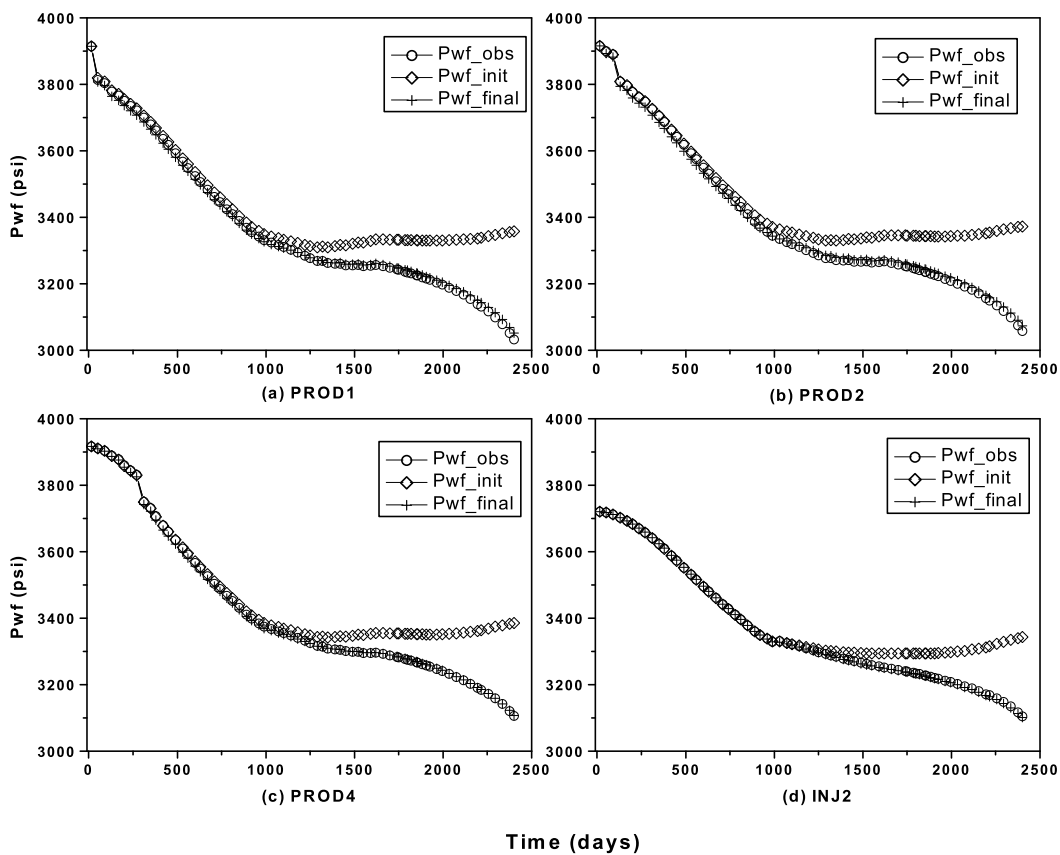

Fig. 4.32: Wellbore pressure data match at four wells, data set 1.

Fig. 4.34 shows the gas saturation at 2400 days for the first layer which corresponds to Etive (top row) and the third layer which is the first layer of the Oseberg (bottom row) corresponding to initial, true and final model which is obtained by history matching both
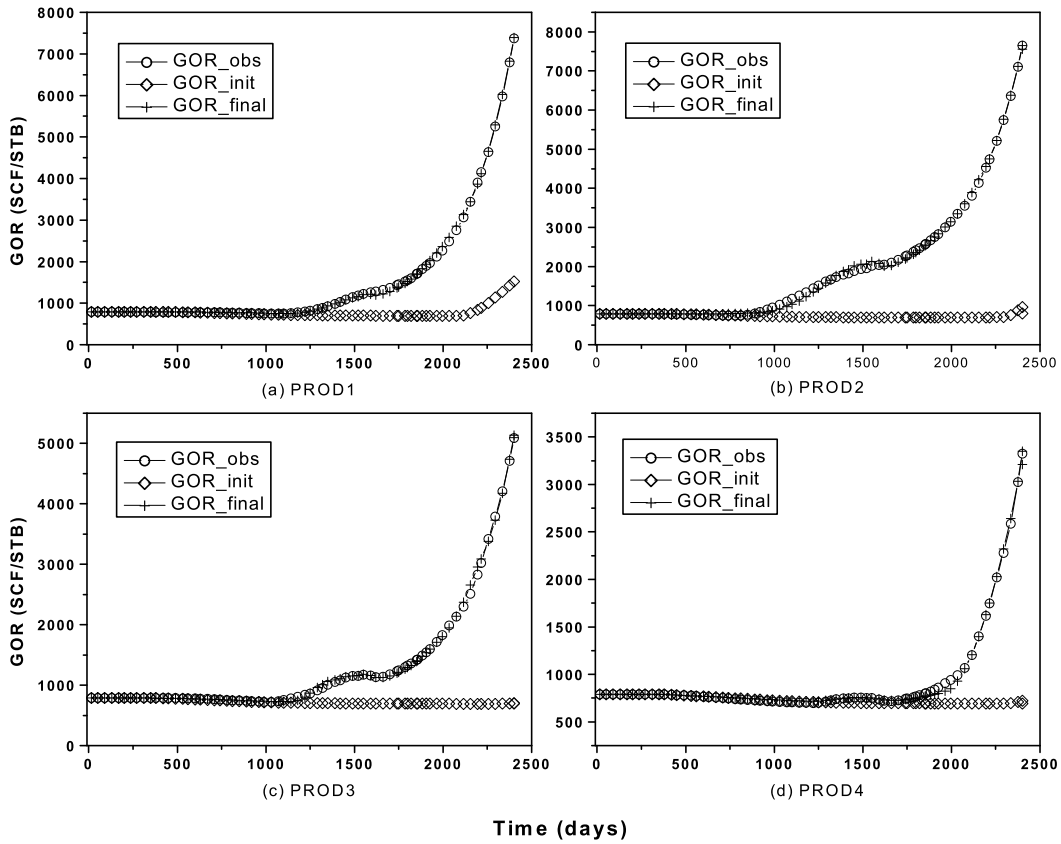
Fig. 4.33: GOR data match at four wells, data set 1.

pressure and GOR data. Fig. 4.35 shows the gas saturation at 2400 days for the 13th x-z cross-section (top row) and the 20th x-z cross-section (bottom row) corresponding to the initial, true and final models. From these figures, we can see that the gas distribution for the final model obtained by history matching the production data is similar to the gas distribution for the true model. For the initial model (see Fig. 4.34 (a) and (d) and Fig. 4.35 (a) and (d)), the gas moves more slowly towards the producing wells. The gas breaks through only at PROD1 and PROD2 and the breakthrough time is much later than for the true model. For the model obtained by history matching the production data, the gas breaks through at all wells at about the same time as for the true model. Fig. 4.36 shows 4 layers of the horizontal log-permeability field corresponding to the initial, true and final model obtained by history matching data set 1. In this figure, the log-permeability fields for the same layer are plotted with the same scale. From these figures, it is not easy to see how the log-permeabilities change by history matching the production data. In order to see
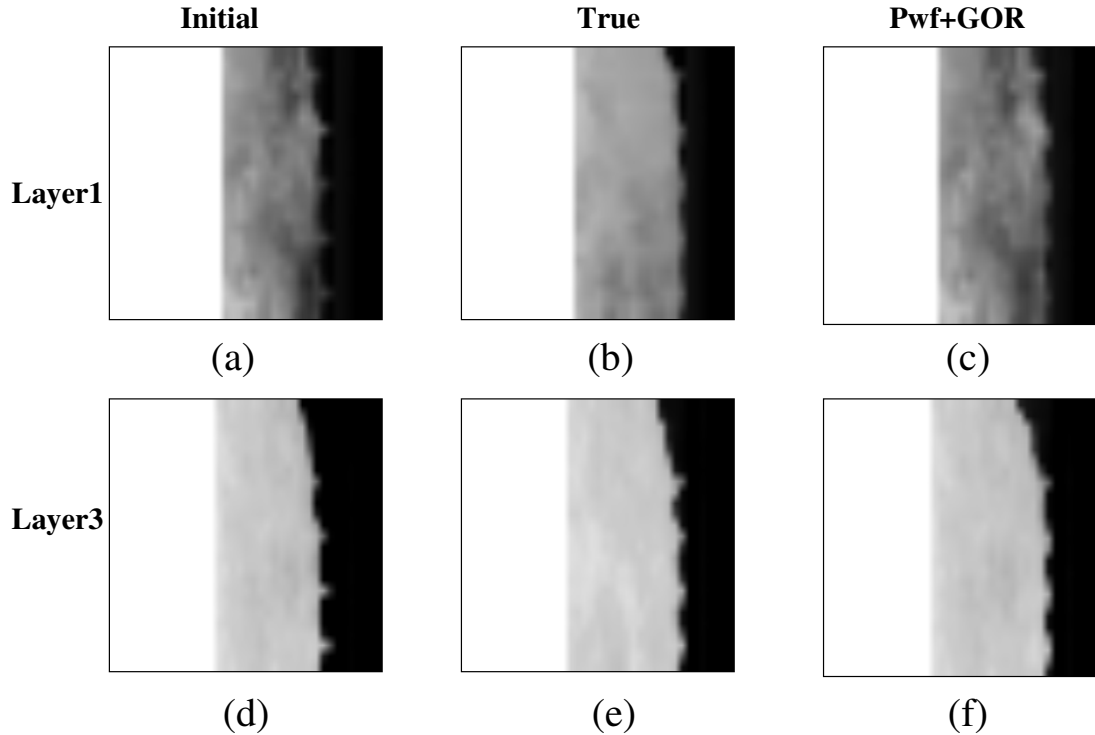
100

Fig. 4.34: Gas saturation at 2400 days in the first layer (top row) and the third layer (bottom row) corresponding to initial, true and final model from history matching wellbore pressure and GOR, data set 1.

how the history matching changes the permeability field from the initial guess, we plotted in Fig. 4.37 the log-permeability change, i.e., the difference between the final model and the initial model, for six layers. From this figure, we can see clearly that for the first three layers, the permeabilities around some wells for the final model are larger than the permeabilities in the same area for the initial model which is an unconditional realization. Recall that the mean of log-permeability for the Etive layer and the top two layers of the Oseberg zone for the initial model is smaller than the true model. Therefore the permeabilities are increased for these layers in order to match the GOR data. We also can see that the permeabilities for the layers 7 through 9 are decreased. Even though we did not show it here, the permeabilities are decreased also for the bottom layer. Recall that the permeability for the true model has the decreasing trend vertically and the permeability for the layer 6 through 10 for the true model are smaller than the initial model. Because flow in the vertical direction is negligible, the vertical log-permeabilities were only changed slightly by history matching (less than 0.001 for the change in log-permeability).
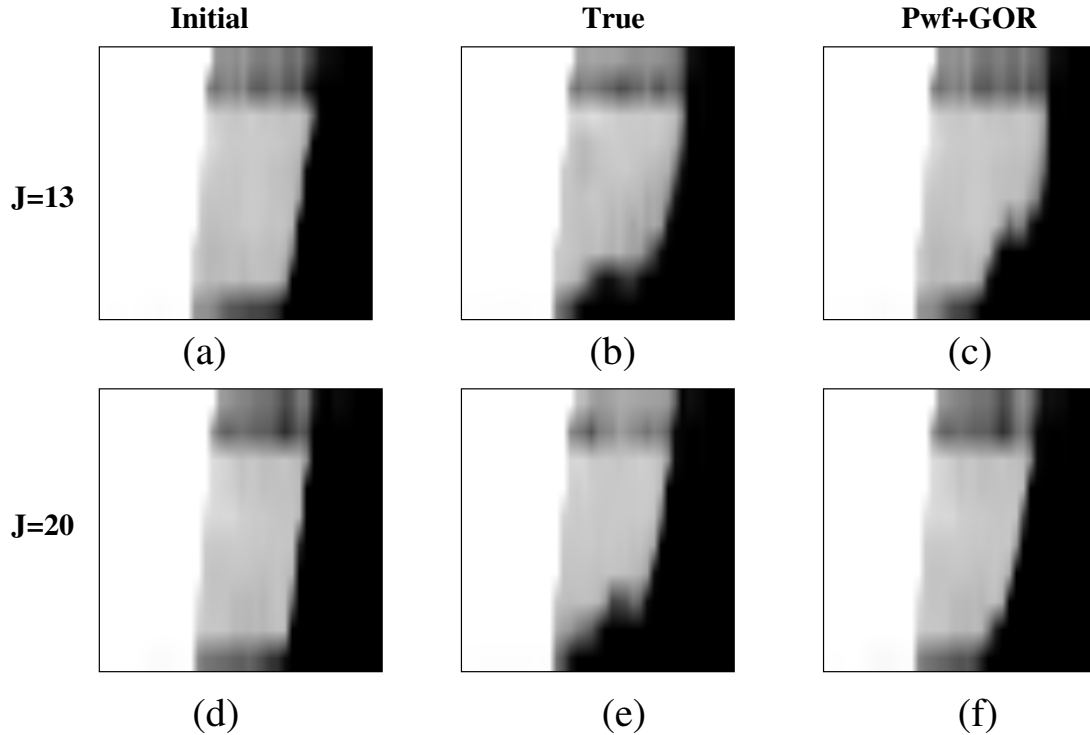
Fig. 4.35: Gas saturation at 2400 days for the 13th cross-section (top row) and the 20th cross-section (bottom row) corresponding to initial, true and final model from history matching wellbore pressure and GOR, data set 1.

When data set 2, which contains only the wellbore pressure data, was history matched, the algorithm converged to a local minimum. I tried several different start point around this minimum, the algorithm always converged to the similar value which may imply a local minimum. The objective function value was reduced from 2.0×10^6 to 2.4×10^5 in 5 iterations. The behavior of the objective function is shown by the squares in Fig. 4.31. Pressure data match for PROD1, PROD2, PROD4 and INJ2 are shown in Fig. 4.38. Comparing this figure with Fig. 3.27, we can see that it is worse than obtained matching both types of data. The reason for that is that the algorithm converges to a local minimum corresponding to a very big objective function value.

The final data set we history matched contains only GOR data from all five producing wells. Each producing well has 71 GOR data. So a total of 355 GOR data are history matched. The behavior of the objective function is shown in Fig. 4.31 by the circles. The value of the objective function was reduced from 1.2×10^7 to 3.0×10^4 in 27 iterations. The algorithm was restarted at the 15th iteration. The GOR data matches for PROD1 through
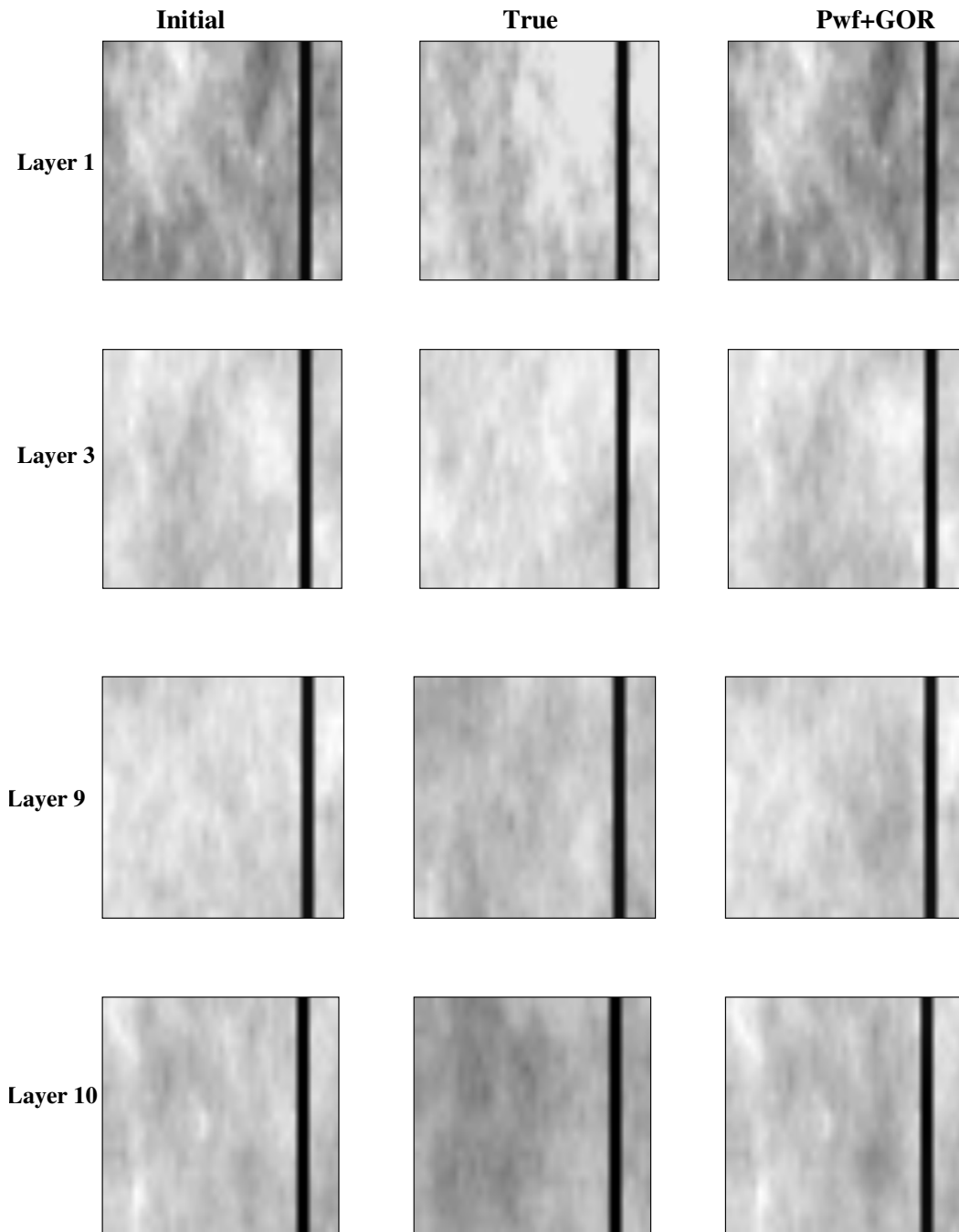
Fig. 4.36: Horizontal permeability field for 4 different layers corresponding to initial, true and final model obtained from history matching wellbore pressure and GOR, data set 1.
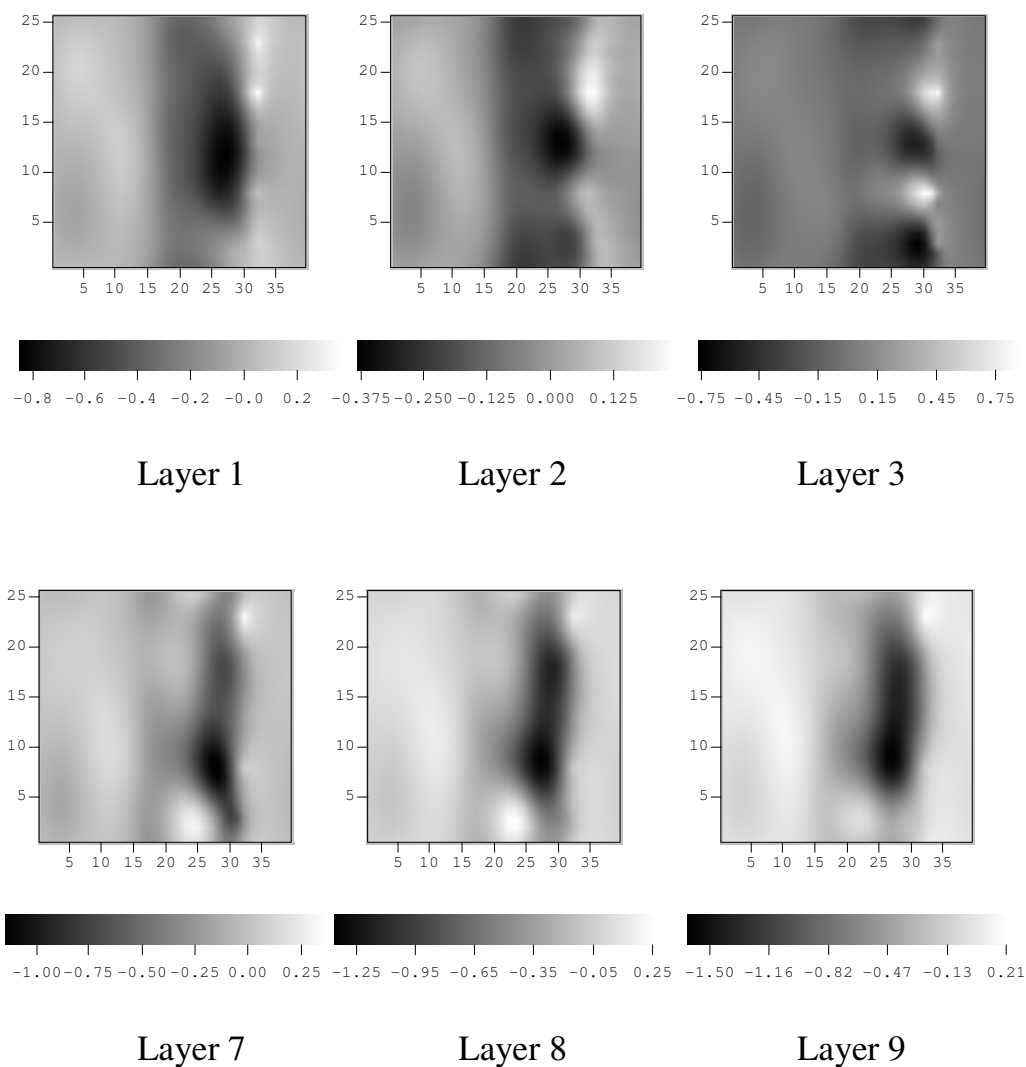
Fig. 4.37: Change in log-permeability for six layers.

PROD4 are shown in Fig. 4.39. Again, the circles represent the observed data; the plus signs represent the data calculated based on the model obtained after history matching the GOR data and the diamonds represent the calculated data based on the initial model. We can see that we obtained a slightly better match of GOR data compared to the GOR match obtained by history matching both pressure and GOR data; compare Fig. 4.39 and Fig. 4.33.

The stars in Fig. 4.40 shows the bottom-hole pressure data which are calculated based on the model obtained after history matching only GOR data. Again, in these figures, circles
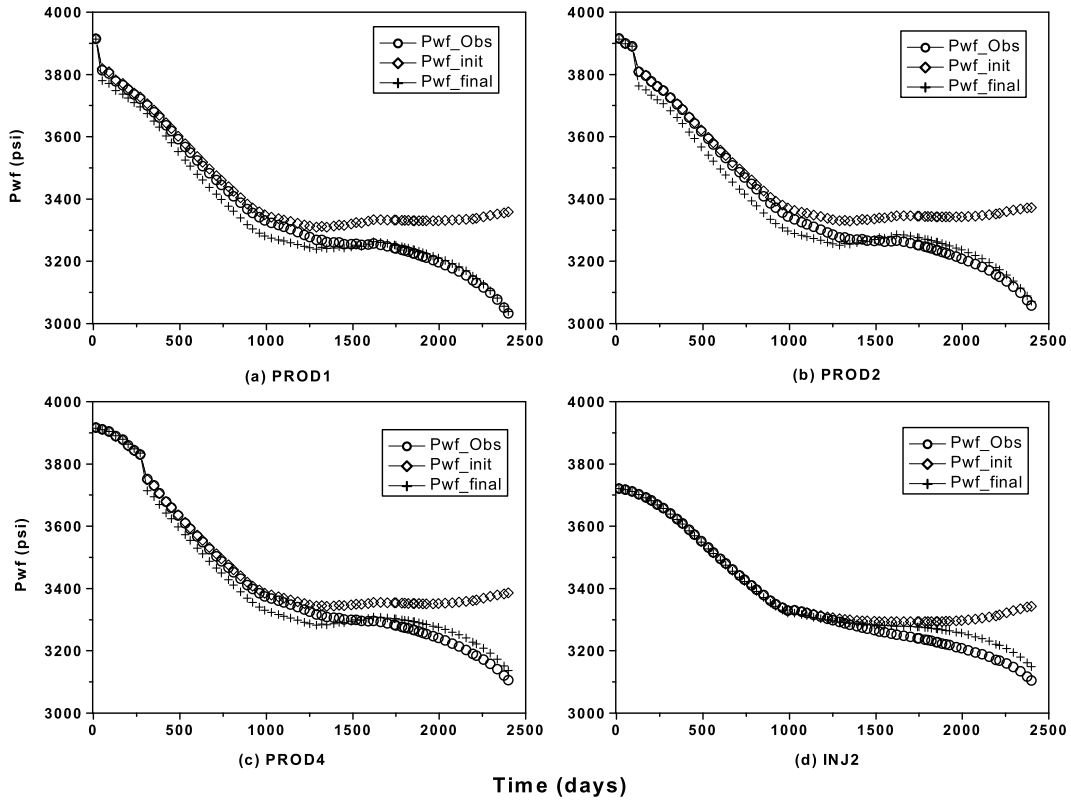
Fig. 4.38: Pwf data obtained from initial model (diamonds), true model (plus signs) and the observed data (circles) at four wells, data set 2.

represent the observed data and diamonds represent the calculated data corresponding to the initial model. We can see that even though pressures are not matched as well as the case where the pressure data were used as the conditioning data, the pressures predicted from the matched model are much closer to the observed data than those predicted from the initial model. Although we did not match any data from the injection wells, the observed pressure data are matched very well at the injection wells which implies that the GOR data at the producing wells can resolve the permeability around the injection wells very well for the case where the flow rates at the injection wells are fixed. Wu (1999) also observed this phenomenon.

Fig. 4.39: GOR data obtained from initial model (diamonds), true model (circles) and final model obtained by history matching only GOR data (plus signs) at four wells, data set 3.
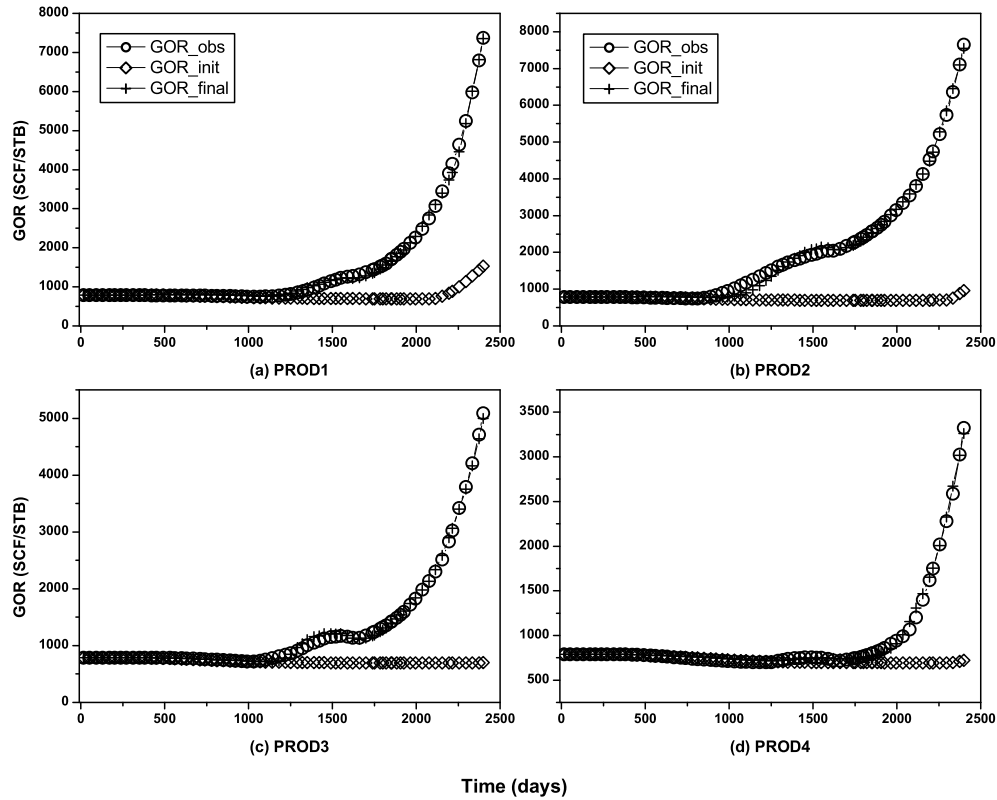
4.2.3 Future Performance Prediction

In making a future performance prediction, after the 2400 days production history, each of the five producing wells is produced at the constant rate of 9000 STB/Day with the minimum bottom-hole pressure of 2000 psi as the constraint for the time period from 2400 days to 2800 days. From 2800 days to 3900 days, the five producing wells are produced at a fixed constant bottom-hole pressure of 2000 psi. After the first 2400 days history, the two injection wells were shut in for 200 days, i.e., from 2400 days to 2600 days; from 2600 days to 2800 days gas was injected at the rate of 18715 MSCF/Day, which is one third of the injection rate at the end of the observed history. From 2800 days to 3900 days, the injection rates were changed to 56145 MSCF/Day, which is equal to the rate at the end of the 2400 days history. Note that the total time span for the future performance prediction is 1500 days, i.e., from 2400 days to 3900 days. The simulation runs were performed for the initial model, the true model and the models obtained by history matching the three different data sets described in the previous section. The total cumulative oil production
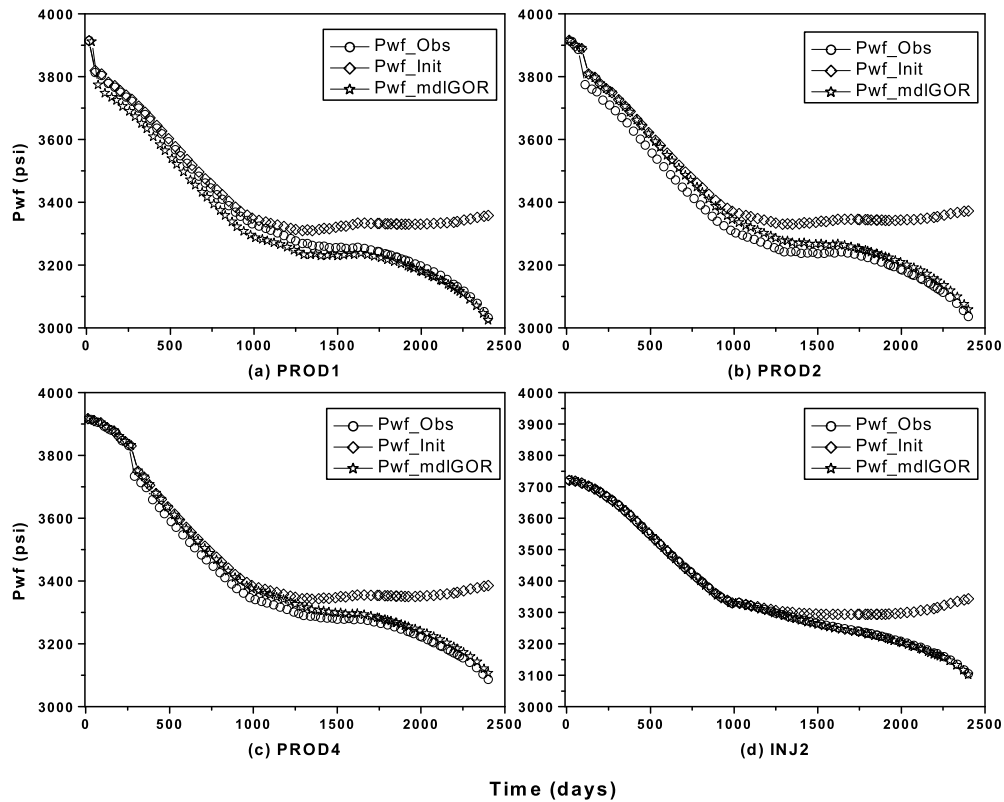
Fig. 4.40: data obtained from initial model (diamonds), true model (circles) and the model obtained by history matching only GOR data (stars) at four wells, data set 3.

obtained for each of the five models is shown in Fig. 4.41. Fig. 4.41 (b) shows the total cumulative oil production only for the period of time corresponding to the future prediction. In this figure, the solid line, dashed line, short dashed line, dot-dashed line and the dotted line, respectively, represent the total cumulative oil production from the true model, the model obtained by history matching both pressure and GOR data, the model obtained by history matching only GOR data, the model obtained by history matching only pressure and the initial model. From this figure, we can see that the cumulative oil productions based on the model obtained by conditioning the initial model to both pressure data and GOR data and the model obtained by conditioning only to the GOR data are closer to the cumulative oil production corresponding to the true model than the model obtained by conditioning only to pressure data. The total cumulative oil production obtained for the initial model is far away from that obtained for the true model. The results of the future performance predictions suggests that the GOR data are more useful than pressure data for reducing the uncertainty in performance prediction. Although this result is consistent with observations of Li et al. (2001),we have not investigated this aspect in sufficient detail to be confident that

107

the conclusion is general. Fig. 4.42 (a) through (e) show the predictions of GOR obtained based on the five different models for PROD1 through PROD5.
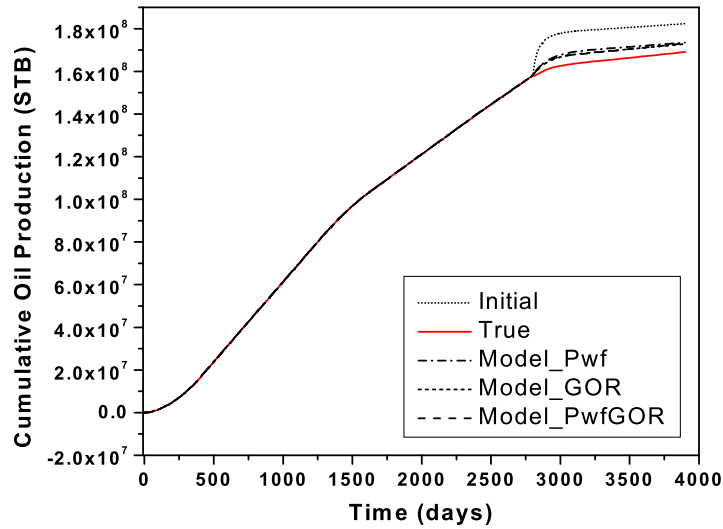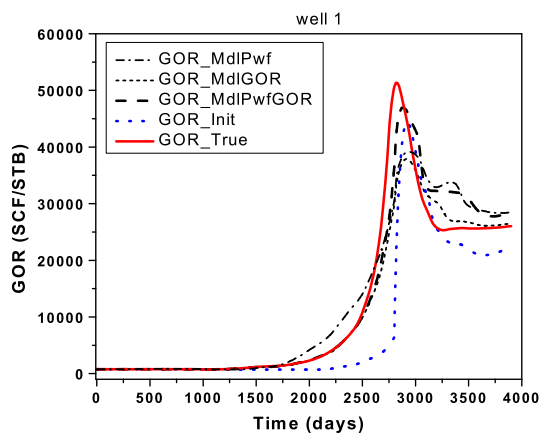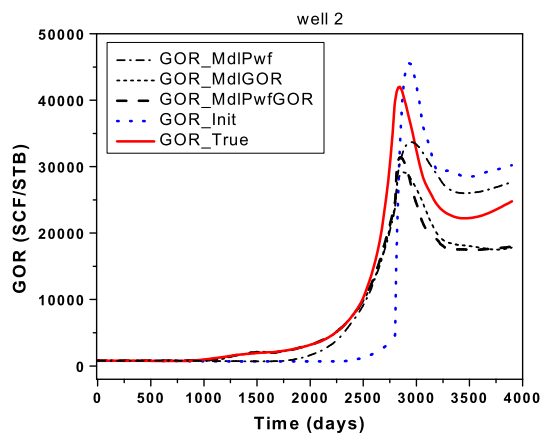

(a) 0-3900 days
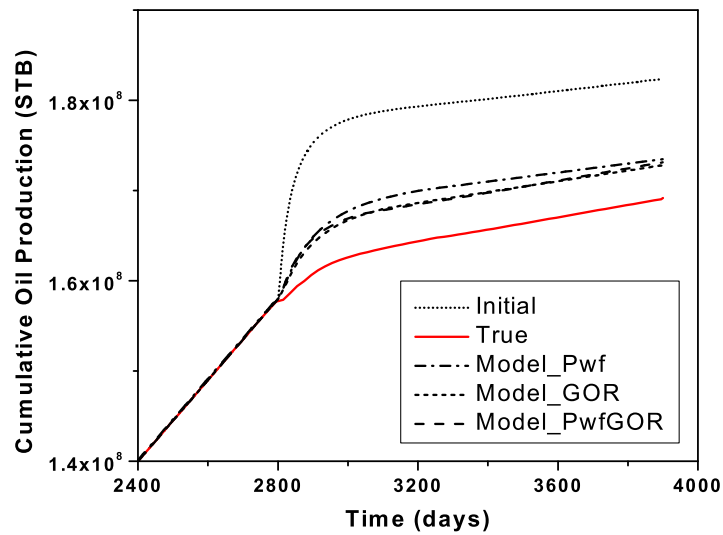
(b) 2400-3900 days

Fig. 4.41: Total field cumulative oil production.

(a) PROD1

(b) PROD2
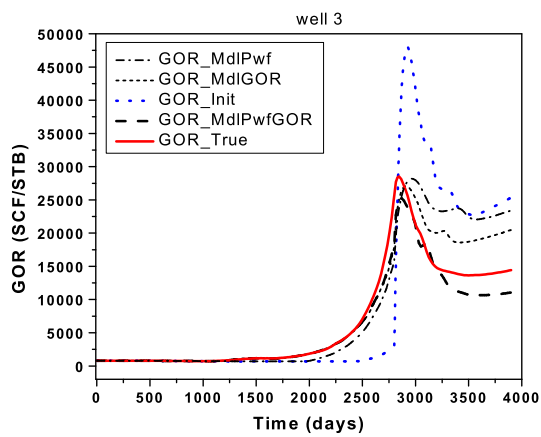
(c) PROD3
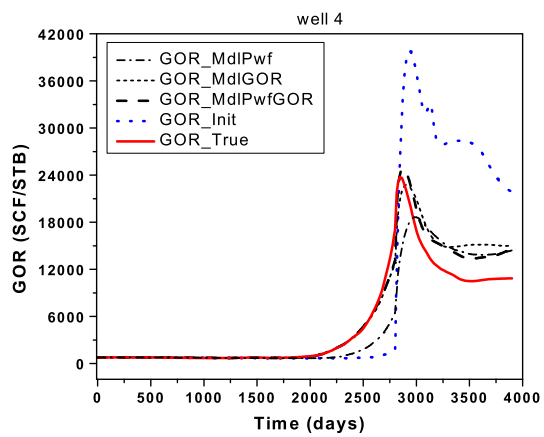
(d) PROD4
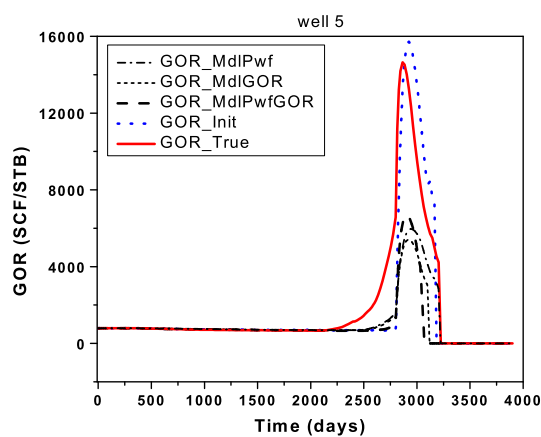
(e) PROD5

Fig. 4.42: GOR data for all producing wells for the production history and the future performance predictions.

109

Chapter 5

SENSITIVITY OF SEISMIC IMPEDANCE CHANGE TO PERMEABILITY AND POROSITY

5.1 Introduction to Time-Lapse Seismic Data

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. Advances in automatic or computer-assisted history matching have allowed researchers to consider the integration of time-lapse seismic data with production data. All quantitative approaches for doing this involve the minimization of an objective function that includes the mismatch between the synthetic changes in seismic data and the observed changes. Using optimization methods, a distribution of parameters that minimize the objective function is sought. The type of seismic data used in the objective function has varied among the researchers. Huang et al. (1997) used amplitude difference or other seismic attributes difference while Arenas et al. (2001) used velocity difference. Landa and Horne (1997) assumed that saturation changes could be obtained directly from time-lapse surveys.

While a number of geophysicists (Tura and D.Lumley, 1999; Landro, 2001; Meadows, 2001) have assumed that changes in saturation and pressure can be estimated directly from time-lapse seismic data (including offset data), it is clearly less restrictive to use all data (including production data) in the estimation of saturation and pressure. Thus we will use the seismic data in the objective function–not saturations and pressures.

Because the number of data and model parameters can be quite large in history matching problems which include time-lapse seismic data, it is common to reduce the number of model parameters by using "pilot points" (Arenas et al., 2001; van Ditzhuijzen et al., 2001) or to divide model into zones with similar properties (Huang et al., 2001).

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. It will also be necessary to use more efficient optimization methods in the history matching than those used previously to get optimum model parameters. A practical method for doing this is outlined in this report.

5.2   Seismic Impedance

Geophysicists often prefer to use amplitude when integrating time-lapse seismic data because amplitude data is always generated and also because it is used to interpret the structure of underground layers. Amplitude, however, depends both on the reflectivity between layers and on the properties of the source. In order to use time-lapse seismic data in history matching for inversion of reservoir parameters, such as permeability and porosity, we must use data which are quantitatively related to the change in value of saturation and pressure. Thus we use seismic P-wave impedance, which is directly related to the values of elastic parameters in reservoir. The definition of impedance is

$$Z = \rho V_p \,, \tag{5.1}$$

where ρ is body density, V_p is P-wave velocity. In time-lapse seismic, the subtraction of such seismic impedance volumes from two different times will show the effect of the phase saturation and pressure change.

Generally, our approach to integration of time-lapse seismic data begins with the creation of a model with initial parameters such as bulk modulus, shear modulus, shaliness, permeability and porosity, etc. We then use the reservoir simulator to get pressure and phase saturations, which can be used to generate synthetic data for GOR, WOR and seismic impedance change, etc. The synthetic data are then compared to field data. The difference between these two groups will tell us whether they match or not under some criteria. If they match, it means that our model is good enough and computation stops. If not, we will solve an adjoint system to compute sensitivities which can be used to compute model corrections. This procedure is repeated until a match is achieved (see Fig. 5.1).

In our approach, we assume that the change in seismic P-wave impedance has been estimated by the geophysicists. In order to compute the data mismatch, we need to compute the theoretical change in impedance due to changes in saturation and pressure in the reservoir, and ultimately, we need to compute the effect of changes in permeability and porosity
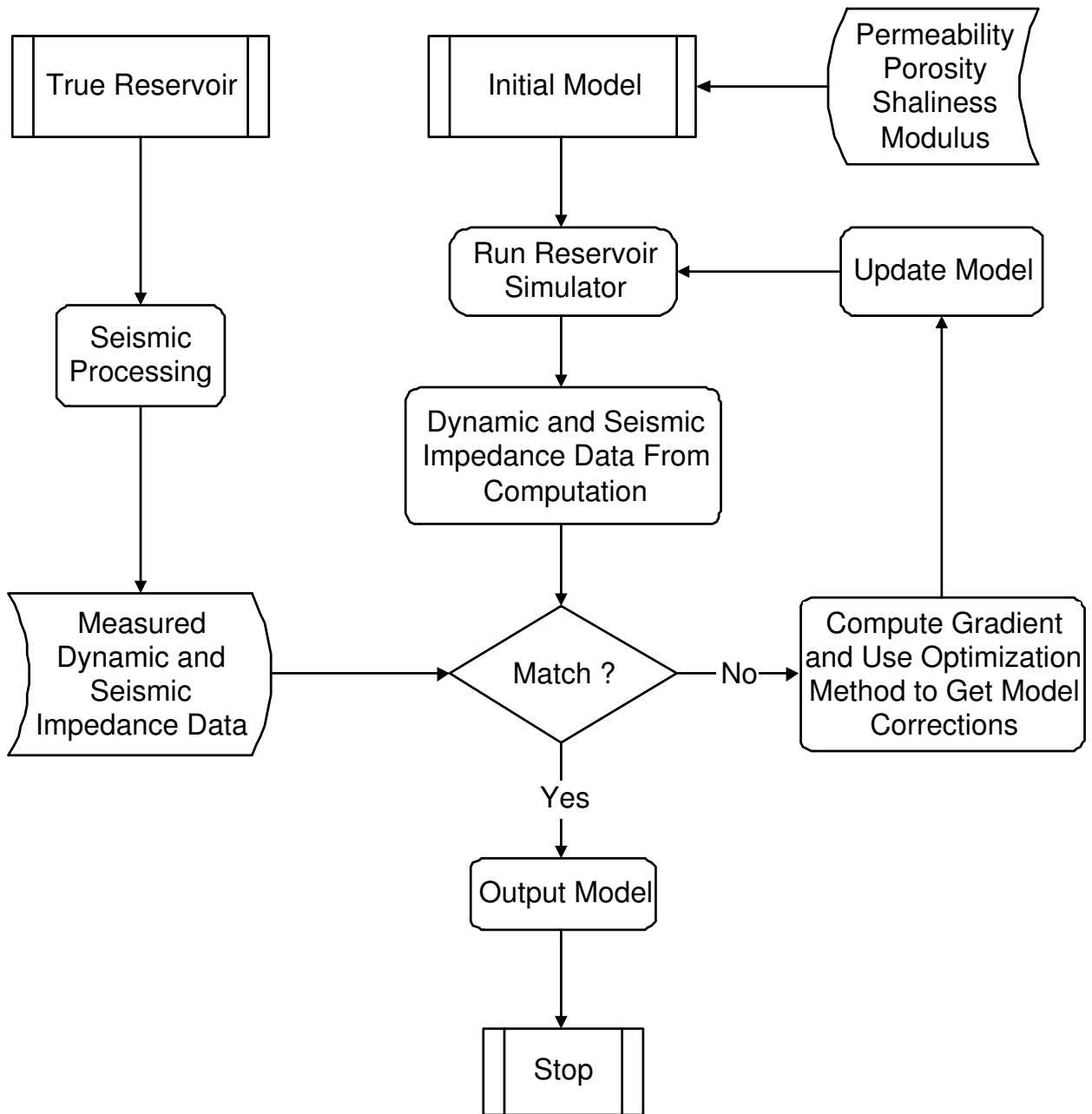
Fig. 5.1: Flow Chart for History Matching with Integration of Time-Lapse Seismic Data

on the impedance. The forward modelling of seismic P-wave impedance is accomplished by adding rock physics models to the reservoir simulator. The following sections describe common approaches to the computation of impedance.

5.2.1 Seismic Velocities

The velocity of the primary wave is

$$V_p = \sqrt{\frac{1}{\rho}\left(K + \frac{4}{3}G\right)},\tag{5.2}$$

where K is the bulk modulus and G is the shear modulus.

The velocity of shear wave is

$$V_s = \sqrt{\frac{G}{\rho}}.\tag{5.3}$$

The seismic impedance is the product of the primary velocity V_p and the bulk density ρ, which can be written in several different ways:

$$Z = \rho V_p = \sqrt{\rho\left(K + \frac{4}{3}G\right)} = \sqrt{\rho K + \frac{4}{3}\rho^2 V_s^2}.\tag{5.4}$$

5.2.2 Poro-Elastic Properties

From Eq. 5.4, we see that to compute seismic impedance, we must know either ρ, K, G or ρ, K, V_s. It is easy to compute ρ using the following equation:

$$\rho = (\rho_o S_o + \rho_w S_w + \rho_g S_g)\phi + (1 - \phi)\rho_{\text{solid}},\tag{5.5}$$

so the main task is to get K, G, V_s. Several formulas have been proposed to compute them. Usually, we can use Gassmann equations to get K and use Han, Kuster plus Toksöz or Ramammorthy equations to get G, V_s (Guerin, 2000).

Gassmann Equations

Gassmann equations (Gassmann, 1951) give us a relation between the bulk modulus of saturated rock and fluid saturation. According to Murphy et al. (1993) and Nolen-Hoeksema (2000), the Gassmann equations provide a low frequency approximation of Biot theory. The

assumption is that the wave must be in low frequency, which is generally true in seismic exploration. It is common to use these relations to explain the dependence of seismic impedance on pressure and saturation. The equations are

$$K = K_{\text{grain}} \frac{K_{\text{frame}} + Q}{K_{\text{grain}} + Q} \,, \tag{5.6}$$

$$Q = \frac{K_{\text{fluid}} \left(K_{\text{grain}} - K_{\text{frame}} \right)}{\phi \left(K_{\text{grain}} - K_{\text{fluid}} \right)} \,, \tag{5.7}$$

$$\log_{10} K_{\text{frame}} = \log_{10} K_{\text{grain}} - 4.25\phi \,, \tag{5.8}$$

$$K_{\text{grain}} = \frac{1}{2} \left[\gamma K_c + (1 - \gamma) K_s + \frac{K_s K_c}{K_s \gamma + K_c (1 - \gamma)} \right] \,, \tag{5.9}$$

$$\frac{1}{K_{\text{fluid}}} = \frac{S_w}{K_w} + \frac{S_g}{K_g} + \frac{S_o}{K_o} \,, \tag{5.10}$$

where K_{grain} is the modulus of grain, K_{frame} is the modulus of the dry frame, K_{fluid} is the modulus of pore fluid, K_s is the sand modulus, K_c is the clay modulus, ϕ is porosity and γ is shaliness.

Equations for Computation of G and V_s

The Han equation (Han et al., 1986) for the velocity of the shear wave is

$$V_s = 3520.0 - 4910.0\phi - 1890.0\gamma \,, \tag{5.11}$$

where ϕ is porosity, γ is shaliness and the units of V_s are m/s.

The KT (Kuster plus Toksöz) equation (Kuster and M.N.Toksöz, 1974) for shear modulus is

$$G = G_m \frac{(6K_m + 12G_m)G_i + (9K_m + 8G_m)((1 - I_c)G_m + I_c G_i)}{(9K_m + 8G_m)G_m + (6K_m + 12G_m)((1 - I_c)G_i + I_c G_m)} \,, \tag{5.12}$$

where G_m denotes the shear modulus of the matrix and G_i is the shear modulus of the inclusion. In the case of a mixture of sand and clay, I_c is replaced by shaliness γ.

The Ramammorthy equations (Ramammorthy et al., 1995) are

$$G = G_{\text{grain}}(1 - 3.48\phi + 2.19\phi^2) \,, \tag{5.13}$$

and

$$G_{\text{grain}} = (0.039 \log_{10} \gamma + 0.072)^{-1} \,, \tag{5.14}$$

115

where G_{grain} is the shear modulus of grain, ϕ is porosity, γ is shaliness. According to Ramammorthy et al. (1995), Eq. 5.14 is valid when γ is between 0.06 and 0.42. If γ is below 0.06, Ramammorthy et al. (1995) recommended using the following equation:

$$G = 42.65(1 - 3.48\phi + 2.19\phi^2)\,, \qquad (5.15)$$

where 42.65 is the shear modulus of pure quartz and the unit of G is GPa. Eq. 5.15 was first derived by Murphy et al. (1993).

For all examples in this report, we use the Gassmann and Han formulas.

5.3 Sensitivity of Seismic Impedance Change to Reservoir Parameters

Because the *static* reservoir parameters, for example geological features, affect the seismic survey, we consider the change of seismic impedance, thereby eliminating such *static* effects and focus on the effects from *dynamic* changes.

5.3.1 The Predominant Effect of Phase Saturation Change in Seismic Impedance Change

The purpose of using seismic impedance change data in history matching is to infer information from phase saturation change due to the production. In practice, it is not clear that this can be done because other poorly known properties of the reservoir such as sand modulus, clay modulus, and porosity will also affect the seismic impedance. Even if these parameters do not change with time, our model will include errors because we usually suppose spatially uniform values for these spatially varying parameters. Before using seismic impedance change data, we must make sure that the change of phase saturation will be the predominant effect. In this investigation, we select some typical values for a sensitivity study to show that the effect from phase saturation change will be the dominant effect on seismic impedance change. To simulate real production, we consider two cases, one of which is an oil reservoir under water flood and the other is a solution gas drive reservoir. The base values of the reservoir parameters are shown in Table 5.1.

Water Flood Case

In addition to these base values, we select reasonable upper and lower bounds for other parameters as shown in Table 5.2. By varying each parameter, we can study the "effect of

Parameter	Value
Porosity	0.2
Shaliness	0.2
Sand Modulus (Pa)	3.8×10^{10}
Clay Modulus (Pa)	21.2×10^{9}
Density of Solid (kg/m^3)	2650
Density of Gas (kg/m^3)	214
Density of Water (kg/m^3)	986
Density of Oil (kg/m^3)	707
Modulus of Gas (Pa)	3.94×10^{7}
Modulus of Water (Pa)	2.39×10^{9}
Modulus of Oil (Pa)	6.71×10^{8}
Initial Water Saturation	0.2
Initial Oil Saturation	0.8
Initial Gas Saturation	0.0

Table 5.1: Base Value in Computation

uncertainty" in these parameters on the change in seismic impedance.

Parameter	Lower Bound	Upper Bound
Water Saturation	0.2	0.6
Shaliness	0.0	0.4
Clay Modulus	1.0×10^{10}	3.0×10^{10}
Sand Modulus	2.8×10^{10}	4.8×10^{10}

Table 5.2: Lower and Upper Bounds for Reservoir Parameters (Water Flood)

We then compute the change in seismic impedance due to the variation in water saturation. We recompute such changes assuming different values of other parameters, changing one parameter at a time. The results are in Table 5.3.

From Table 5.3, we can see that even if the value of the rock mineral parameters were greatly in error, the change of seismic impedance change due to water saturation change would be largely unaffected. That is to say that we can neglect the uncertainty of shaliness, grain modulus and shale modulus in the seismic impedance change data.

Parameter	Change in Seismic Impedance	Change Ratio
Base Value	251978	3.6%
Shaliness in Lower Bound	242795	3.5%
Shaliness in Upper Bound	265121	3.8%
Clay Modulus in Lower Bound	249037	3.6%
Clay Modulus in Upper Bound	252823	3.6%
Sand Modulus in Lower Bound	247481	3.5%
Sand Modulus in Upper Bound	253692	3.6%

Table 5.3: Change in Seismic Impedance from Changing Water Saturation

Solution Gas Drive Case

In the solution gas drive case, the base values are the same as those in Table 5.1. The bounds on parameter values for the sensitivity study are shown in Table 5.4.

Parameter	Lower Bound	Upper Bound
Gas Saturation	0.0	0.5
Shaliness	0.0	0.4
Clay Modulus	1.0×10^{10}	3.0×10^{10}
Sand Modulus	2.8×10^{10}	4.8×10^{10}

Table 5.4: Lower and Upper Bounds for Reservoir Parameters (Solution Gas Drive)

Using the same methods described in the previous section, we obtain the results shown in Table 5.5. It is clear that in solution gas drive, the phase saturation change is still the most important effect. Given these results, it appears that the uncertainty in mineralogy can be dealt with as part of modelling error and that its effect on change in impedance is small compared to those parameters that affect the phase saturation, such as permeability and porosity.

5.3.2 Sensitivity Coefficients Using Adjoint Method

In history matching, we generally use gradient based optimization methods, which require computation of the gradient of the objective functions with respect to model parameters. The most efficient method for obtaining the sensitivity of seismic impedance change to reservoir

Parameter	Change in Seismic Impedance	Change Ratio
Base Value	-526631	-7.5%
Shaliness in Lower Bound	-502306	-7.2%
Shaliness in Upper Bound	-562770	-8.0%
Clay Modulus in Lower Bound	-528152	-7.5%
Clay Modulus in Upper Bound	-525463	-7.5%
Sand Modulus in Lower Bound	-528241	-7.5%
Sand Modulus in Upper Bound	-523081	-7.5%

Table 5.5: Change in Seismic Impedance from Changing Gas Saturation

parameters is using the adjoint method. Although the adjoint method can be used to provide sensitivity to almost any kind of parameter, in this report, we describe only the computation of the sensitivity of impedance to permeability and porosity.

In field case, we will have a lot of seismic impedance data (nearly in every gridblock). Here, we only consider the computation of sensitivity of the seismic impedance in a single gridblock; in future history matching work, we will only need to add more source terms to the adjoint system to calculate all the seismic impedance sensitivities needed.

The Adjoint System

According to Li (2001), the adjoint system and the desired sensitivity coefficients for some functional J, or equivalently β are

$$[\nabla_{y^n}(f^n)^T]\lambda^n = -[\nabla_{y^n}(f^{n+1})^T]\lambda^{n+1} - \nabla_{y^n}\beta \,, \tag{5.16}$$

and

$$\nabla_m J = \nabla_m \beta + \sum_{n=1}^{L}[\nabla_m(f^n)^T](\lambda^n) \,, \tag{5.17}$$

where,

$$y^n = \begin{bmatrix} P_1^n \\ S_{w,1}^n \\ S_{g,1}^n \\ P_2^n \\ \vdots \\ S_{g,N}^n \\ P_{wf,1}^n \\ \vdots \\ P_{wf,N_w}^n \end{bmatrix} \tag{5.18}$$

is the vector of the primary variables that are solved for in a reservoir simulator,

$$m = \begin{bmatrix} m_1 \\ m_2 \\ \vdots \\ m_{N_m} \end{bmatrix} \tag{5.19}$$

is the model parameters vector, and

$$\lambda^{n+1} = \begin{bmatrix} \lambda_1^{n+1}, & \lambda_2^{n+1}, & \dots, & \lambda_{3N+N_w}^{n+1} \end{bmatrix}^T \tag{5.20}$$

is the adjoint variable. Here, β could be any functional for which the sensitivity is required. In this report, β is the seismic impedance of a gridblock.

In our computations, we will compute the sensitivity to horizontal permeabilities (k_h) and porosity (ϕ). In each gridblock, k_h can be divided into k_x and k_y, so the model vectors can be written as

$$m_{k_x} = k_x = \begin{bmatrix} k_{x,1} & k_{x,2} & \cdots & k_{x,M} \end{bmatrix}^T, \tag{5.21}$$

$$m_{k_y} = k_y = \begin{bmatrix} k_{y,1} & k_{y,2} & \cdots & k_{y,M} \end{bmatrix}^T, \tag{5.22}$$

and

$$m_\phi = \phi = \begin{bmatrix} \phi_1 & \phi_2 & \cdots & \phi_M \end{bmatrix}^T. \tag{5.23}$$

Then the equations to calculate sensitivities are

$$\nabla_{k_x} J = \nabla_{k_x} \beta + \sum_{n=1}^{L} [\nabla_{k_x}(f^n)^T](\lambda^n), \tag{5.24}$$

$$\nabla_{k_y} J = \nabla_{k_y} \beta + \sum_{n=1}^{L} [\nabla_{k_y}(f^n)^T](\lambda^n), \tag{5.25}$$

and

$$\nabla_\phi J = \nabla_\phi \beta + \sum_{n=1}^{L} [\nabla_\phi (f^n)^T](\lambda^n) \,. \tag{5.26}$$

In the isotropic case, k_x is equal to k_y, so the sensitivity to horizontal permeability is

$$\nabla_{k_h} J = \nabla_{k_x} J + \nabla_{k_y} J \,, \tag{5.27}$$

where J is seismic impedance at some specified time step L.

From Eq. 5.16 and Eq. 5.17, we know that for the various equations which convert reservoir parameters into seismic impedance, there are different forms for computing the source term $\nabla_{y^n}\beta$ and direct derivative terms $\nabla_{k_h}\beta$ and $\nabla_\phi\beta$, which we summarize in the following sections.

Source Term and Direct Term Computation Using Gassmann and Han Equations

The density of rock with porous fluid is

$$\rho = (\rho_o S_o + \rho_w S_w + \rho_g S_g)\phi + (1 - \phi)\rho_{\text{solid}} \,, \tag{5.28}$$

so the derivative of seismic impedance with respect to P is

$$\frac{\partial Z}{\partial P} = \frac{1}{2}\left(\rho K + \frac{4}{3}\rho^2 V_s^2\right)^{-1/2} \times \left(K\frac{\partial \rho}{\partial P} + \frac{8}{3}V_s^2\rho\frac{\partial \rho}{\partial P}\right) \,, \tag{5.29}$$

where

$$\frac{\partial \rho}{\partial P} = \phi\left(S_o\frac{\partial \rho_o}{\partial P} + S_w\frac{\partial \rho_w}{\partial P} + S_g\frac{\partial \rho_g}{\partial P}\right) \,. \tag{5.30}$$

The derivative of seismic impedance with respect to water saturation S_w is:

$$\frac{\partial Z}{\partial S_w} = \frac{1}{2}\left(\rho K + \frac{4}{3}\rho^2 V_s^2\right)^{-1/2} \times \left(K\frac{\partial \rho}{\partial S_w} + \rho\frac{\partial K}{\partial S_w} + \frac{8}{3}V_s^2\rho\frac{\partial \rho}{\partial S_w}\right) \,, \tag{5.31}$$

where

$$\frac{\partial \rho}{\partial S_w} = \phi\left(\rho_w - \rho_o\right) \,, \tag{5.32}$$

$$
\begin{aligned}
\frac{\partial K}{\partial S_w} =\ & \frac{\left(\left(K_{\text{grain}}^2 - K_{\text{grain}}K_{\text{frame}}\phi - K_{\text{grain}}K_{\text{frame}}\right)\frac{\partial K_{\text{fluid}}}{\partial S_w}\right)}{\left(K_{\text{grain}}^2\phi + K_{\text{fluid}}\left(K_{\text{grain}} - K_{\text{grain}}\phi - K_{\text{frame}}\right)\right)} - \\
& \frac{\left(\left(K_{\text{grain}} - K_{\text{grain}}\phi - K_{\text{frame}}\right)\frac{\partial K_{\text{fluid}}}{\partial S_w}\right)}{\left(K_{\text{grain}}^2\phi + K_{\text{fluid}}\left(K_{\text{grain}} - K_{\text{grain}}\phi - K_{\text{frame}}\right)\right)} \times \\
& \frac{\left(K_{\text{grain}}^2 K_{\text{frame}}\phi + K_{\text{fluid}}\left(K_{\text{grain}}^2 - K_{\text{grain}}K_{\text{frame}}\phi - K_{\text{grain}}K_{\text{frame}}\right)\right)}{\left(K_{\text{grain}}^2\phi + K_{\text{fluid}}\left(K_{\text{grain}} - K_{\text{grain}}\phi - K_{\text{frame}}\right)\right)} \,,
\end{aligned}
\tag{5.33}
$$

and

$$\frac{\partial K_{\text{fluid}}}{\partial S_w} = \frac{\frac{1}{K_o} - \frac{1}{K_w}}{\left(\frac{S_w}{K_w} + \frac{1-S_w-S_g}{K_o} + \frac{S_g}{K_g}\right)^2} \, . \tag{5.34}$$

The derivative of seismic impedance with respect to gas saturation S_g is:

$$\frac{\partial Z}{\partial S_g} = \frac{1}{2}\left(\rho K + \frac{4}{3}\rho^2 V_s^2\right)^{-1/2} \times \left(K\frac{\partial \rho}{\partial S_g} + \rho\frac{\partial K}{\partial S_g} + \frac{8}{3}V_s^2\rho\frac{\partial \rho}{\partial S_g}\right), \tag{5.35}$$

where

$$\frac{\partial \rho}{\partial S_g} = \phi\left(\rho_g - \rho_o\right), \tag{5.36}$$

$$\frac{\partial K}{\partial S_g} = \frac{\left(\left(K_{\text{grain}}^2 - K_{\text{grain}}K_{\text{frame}}\phi - K_{\text{grain}}K_{\text{frame}}\right)\frac{\partial K_{\text{fluid}}}{\partial S_g}\right)}{\left(K_{\text{grain}}^2\phi + K_{\text{fluid}}\left(K_{\text{grain}} - K_{\text{grain}}\phi - K_{\text{frame}}\right)\right)} - \frac{\left(\left(K_{\text{grain}} - K_{\text{grain}}\phi - K_{\text{frame}}\right)\frac{\partial K_{\text{fluid}}}{\partial S_g}\right)}{\left(K_{\text{grain}}^2\phi + K_{\text{fluid}}\left(K_{\text{grain}} - K_{\text{grain}}\phi - K_{\text{frame}}\right)\right)} \times \frac{\left(K_{\text{grain}}^2 K_{\text{frame}}\phi + K_{\text{fluid}}\left(K_{\text{grain}}^2 - K_{\text{grain}}K_{\text{frame}}\phi - K_{\text{grain}}K_{\text{frame}}\right)\right)}{\left(K_{\text{grain}}^2\phi + K_{\text{fluid}}\left(K_{\text{grain}} - K_{\text{grain}}\phi - K_{\text{frame}}\right)\right)}, \tag{5.37}$$

and

$$\frac{\partial K_{\text{fluid}}}{\partial S_g} = \frac{\frac{1}{K_o} - \frac{1}{K_g}}{\left(\frac{S_w}{K_w} + \frac{1-S_w-S_g}{K_o} + \frac{S_g}{K_g}\right)^2} \, . \tag{5.38}$$

There is no direct dependance between seismic impedance and permeability k, so the derivative of seismic impedance with respect to permeability is zero.

The derivative with respect to porosity is:

$$\frac{\partial Z}{\partial \phi} = \frac{1}{2}\left(\rho K + \frac{4}{3}\rho^2 V_s^2\right)^{-1/2} \times \left(\left(K + \frac{8}{3}\rho V_s^2\right)\frac{\partial \rho}{\partial \phi} + \rho\frac{\partial K}{\partial \phi} + \frac{8}{3}\rho^2 V_s\frac{\partial V_s}{\partial \phi}\right), \tag{5.39}$$

where

$$\frac{\partial V_s}{\partial \phi} = -4910.0 \, , \tag{5.40}$$

$$\frac{\partial \rho}{\partial \phi} = \rho_o\left(1 - S_w - S_g\right) + \rho_w S_w + \rho_g S_g - \rho_{\text{solid}} \, , \tag{5.41}$$

$$\frac{\partial K}{\partial \phi} = \frac{\left(\left(K_{\text{grain}}^2 - K_{\text{fluid}}K_{\text{grain}}\right)\left(\phi\frac{\partial K_{\text{frame}}}{\partial \phi} + K_{\text{frame}}\right) - K_{\text{fluid}}K_{\text{grain}}\frac{\partial K_{\text{frame}}}{\partial \phi}\right)}{\phi\left(K_{\text{grain}}^2 - K_{\text{fluid}}K_{\text{grain}}\right) + K_{\text{fluid}}K_{\text{grain}} - K_{\text{fluid}}K_{\text{frame}}} - \frac{\left(K_{\text{frame}}\phi\left(K_{\text{grain}}^2 - K_{\text{fluid}}K_{\text{grain}}\right) + K_{\text{fluid}}K_{\text{grain}}^2 - K_{\text{fluid}}K_{\text{grain}}K_{\text{frame}}\right)}{\left(\phi\left(K_{\text{grain}}^2 - K_{\text{fluid}}K_{\text{grain}}\right) + K_{\text{fluid}}K_{\text{grain}} - K_{\text{fluid}}K_{\text{frame}}\right)} \times \frac{\left(K_{\text{grain}}^2 - K_{\text{fluid}}K_{\text{grain}} - K_{\text{fluid}}\frac{\partial K_{\text{frame}}}{\partial \phi}\right)}{\left(\phi\left(K_{\text{grain}}^2 - K_{\text{fluid}}K_{\text{grain}}\right) + K_{\text{fluid}}K_{\text{grain}} - K_{\text{fluid}}K_{\text{frame}}\right)} \, , \tag{5.42}$$

and

$$\frac{\partial K_{\text{frame}}}{\partial \phi} = -4.25 \ln(10) 10^{\left(\log_{10}(K_{\text{grain}}) - 4.25\phi\right)} . \tag{5.43}$$

Source Term and Direct Term Computation Using Gassmann and KT Equations

Here, we assume that the Gassmann and KT equations are applied when computing the seismic impedance given by

$$Z = \sqrt{\rho \left(K + \frac{4}{3}G \right)} , \tag{5.44}$$

where G is shear modulus.

The density of rock with porous fluid is

$$\rho = (\rho_o S_o + \rho_w S_w + \rho_g S_g)\phi + (1 - \phi)\rho_{\text{solid}} , \tag{5.45}$$

so the derivative of seismic impedance with respect to P is

$$\frac{\partial Z}{\partial P} = \frac{1}{2} \left(\rho K + \frac{4}{3}\rho G \right)^{-1/2} \times \left(K \frac{\partial \rho}{\partial P} + \frac{4}{3}G \frac{\partial \rho}{\partial P} \right) , \tag{5.46}$$

where

$$\frac{\partial \rho}{\partial P} = \phi \left(S_o \frac{\partial \rho_o}{\partial P} + S_w \frac{\partial \rho_w}{\partial P} + S_g \frac{\partial \rho_g}{\partial P} \right) . \tag{5.47}$$

The derivative of seismic impedance with respect to water saturation S_w is

$$\frac{\partial Z}{\partial S_w} = \frac{1}{2} \left(\rho K + \frac{4}{3}\rho G \right)^{-1/2} \times \left(\left(K + \frac{4}{3}G \right) \frac{\partial \rho}{\partial S_w} + \rho \frac{\partial K}{\partial S_w} \right) , \tag{5.48}$$

where $\frac{\partial \rho}{\partial S_w}$, $\frac{\partial K}{\partial S_w}$ and $\frac{\partial K_{\text{fluid}}}{\partial S_w}$ can be computed using Eqs. 5.32 to 5.34.

The derivative of seismic impedance with respect to gas saturation S_g is

$$\frac{\partial Z}{\partial S_g} = \frac{1}{2} \left(\rho K + \frac{4}{3}\rho G \right)^{-1/2} \times \left(\left(K + \frac{4}{3}G \right) \frac{\partial \rho}{\partial S_g} + \rho \frac{\partial K}{\partial S_g} \right) , \tag{5.49}$$

where $\frac{\partial \rho}{\partial S_g}$, $\frac{\partial K}{\partial S_g}$ and $\frac{\partial K_{\text{fluid}}}{\partial S_g}$ can be computed using Eqs. 5.36 to 5.38.

The derivative of impedance with respect to permeability k is also zero in this case.

The derivative with respect to porosity ϕ is

$$\frac{\partial Z}{\partial \phi} = \frac{1}{2} \left(\rho K + \frac{4}{3}\rho G \right)^{-1/2} \times \left(\left(K + \frac{4}{3}G \right) \frac{\partial \rho}{\partial \phi} + \rho \frac{\partial K}{\partial \phi} \right) , \tag{5.50}$$

where $\frac{\partial \rho}{\partial \phi}$, $\frac{\partial K}{\partial \phi}$ and $\frac{\partial K_{\text{frame}}}{\partial \phi}$ can be computed using Eqs. 5.41 to 5.43.

Source Term and Direct Term Computation Using Gassmann and Ramammorthy Equations

Here the Gassmann and Ramammorthy equations are used to compute the seismic impedance, which is given by

$$Z = \sqrt{\rho \left(K + \frac{4}{3}G \right)}, \tag{5.51}$$

where G is shear modulus. In this case, all derivatives are the same as the previous section except

$$G = G_{\text{grain}} \left(1 - 3.48\phi + 2.19\phi^2 \right), \tag{5.52}$$

so

$$\frac{\partial G}{\partial \phi} = G_{\text{grain}} \left(4.38\phi - 3.48 \right). \tag{5.53}$$

In this report, we have supposed that porosity ϕ does not change with pressure. In the future, we will add the derivative of ϕ with respect to pressure P in Eq. 5.29 and Eq. 5.46.

We have assumed that we will use the change in impedance for history matching, not the impedance. We therefore need to compute the sensitivity of the change in impedance to porosity and permeability. There are two ways to do this. The most computationally efficient method would be to set $d_{obs} = Z_{obs}(t_f) - Z_{obs}(t_i)$ where t_f and t_i are the final and initial times for the acquisition of time-lapse data. When computing sensitivities using the adjoint system, this would result in sources at t_f and t_i. What we have done in this study is to compute the sensitivities at times t_f and t_i, then take the difference of them.

5.4   Results and Discussion

In order to test the computation of the sensitivity of impedance to permeability and porosity, two synthetic models were created. Both of them use a $15 \times 15 \times 1$ grid with $\triangle x = \triangle y = 40$ ft and $\triangle z = 30$ ft. The porosity field is homogenous and equal to 0.22. The permeability field is heterogenous and isotropic. The mean of $\ln(k)$ is 3.95 (52 mD) and the variance is 0.52.

In one model, there are five wells. One injector is in gridblock $(8, 8)$ and four producers are located symmetrically in gridblocks $(3, 3)$, $(3, 13)$, $(13, 3)$ and $(13, 13)$ (see white points in Fig. 5.2(a)). We use this model to simulate a water flood for a case where the initial pressure in the oil reservoir is 4500 psi, which is slightly above the initial bubble point pressure, 4417 psi. Each producer produces at total fluid rate 220 RB/D. The minimum bottom hole pressure for each producer is 50 psi and the maximum water-oil ratio for each

producer is 0.49. For the injector, the maximum water injection rate is 550 RB/D. Initial oil saturation is 0.8 and the initial water saturation is equal to irreducible water saturation $S_{wc} = 0.2$.

The other model has one producer in gridblock $(3, 3)$ (see white points in Fig. 5.2(b)). We use it to simulate production by solution gas drive. This producer produces at total fluid rate 700 RB/D. The minimum bottom hole pressure is 50 psi and the maximum water-oil ratio is 0.49. All initial reservoir pressure, initial bubble point pressure, oil saturation and water saturation are the same as those in the water flood case.

In both cases, we compute the sensitivity of seismic impedance change in gridblocks $(4, 5)$ and $(6, 10)$ (see black points in Fig. 5.2) to horizontal permeability and porosity at the 30th day. Also we output the corresponding water and gas saturation distribution maps and pressure maps.

From these maps, we make the following observations,

1. In the water flood case, the seismic impedance change map (Fig. 5.4) has the same shape as the water saturation change map (Fig. 5.3(a)). In the solution gas drive case, there is also great similarity between the seismic impedance change map and the gas saturation change map (Fig. 5.5 and Fig. 5.3(b)). In this case the change in seismic impedance is mainly due to the change in fluid saturation due to production. Using the seismic impedance change, we could qualitatively monitor the reservoir production.

2. From the sensitivity of seismic impedance change to log-permeability in the water flood case (Fig. 5.6(a) and Fig. 5.7(a)) and the solution gas drive case (Fig. 5.6(b) and Fig. 5.7(b)), we know that seismic impedance change has sensitivity dependence on reservoir permeability because the change in reservoir permeability will affect the change in phase saturation. So seismic impedance change is clearly "sensitive" to the change in permeability.

3. In addition to the direct effect of porosity on impedance, the seismic impedance change also has an indirect dependence on porosity, which can be understood from the phase saturation equation:
$$\frac{\partial S}{\partial t} + \left(\frac{U_T}{\phi} \frac{df}{dS} \right) \frac{\partial S}{\partial x} = 0 \,, \tag{5.54}$$
where the term in parenthesis is called the "phase velocity", which describes the rate of advance of the phase saturations. Because this velocity includes porosity ϕ, we see that the seismic impedance change must be sensitive to porosity through the effect on saturation.

4. The shapes of the maps of sensitivity to porosity in the water flood case (Fig. 5.6(c) and Fig. 5.7(c)) and the solution gas drive case (Fig. 5.6(d) and Fig. 5.7(d)) do not look at all like the shapes of the maps of sensitivity to log-permeability. The reason is that the sensitivity can be divided into two terms as following:

$$\frac{\partial J}{\partial \phi} = \frac{\partial Z}{\partial \phi} + \frac{\partial f}{\partial \phi}\lambda \tag{5.55}$$

where $\partial Z/\partial \phi$ is the direct derivative term and $\lambda \partial f/\partial \phi$ is contribution to the derivative from the flow equations. In porosity sensitivity, the direct term is very large compared to the effect from the flow equations. This is different from permeability sensitivity because seismic impedance change has no direct dependence on permeability.

5. A comparison of the sensitivity of seismic impedance change to log-permeability and porosity from the perturbation method and the adjoint method shows that the different methods give almost identical results for the water flood case (Fig. 5.8) and the solution gas drive case (Fig. 5.9). The relative errors between them are very small (Fig. 5.10, Fig. 5.11, Fig. 5.12 and Fig. 5.13), which shows that our adjoint method is giving the correct results.

(a) Water Flood

(b) Solution Gas Drive

Fig. 5.2: ln(k) Field

(a) Water Saturation for Water Flood

(b) Gas Saturation for Solution Gas Drive

(c) Pressure for Water Flood

(d) Pressure for Solution Gas Drive
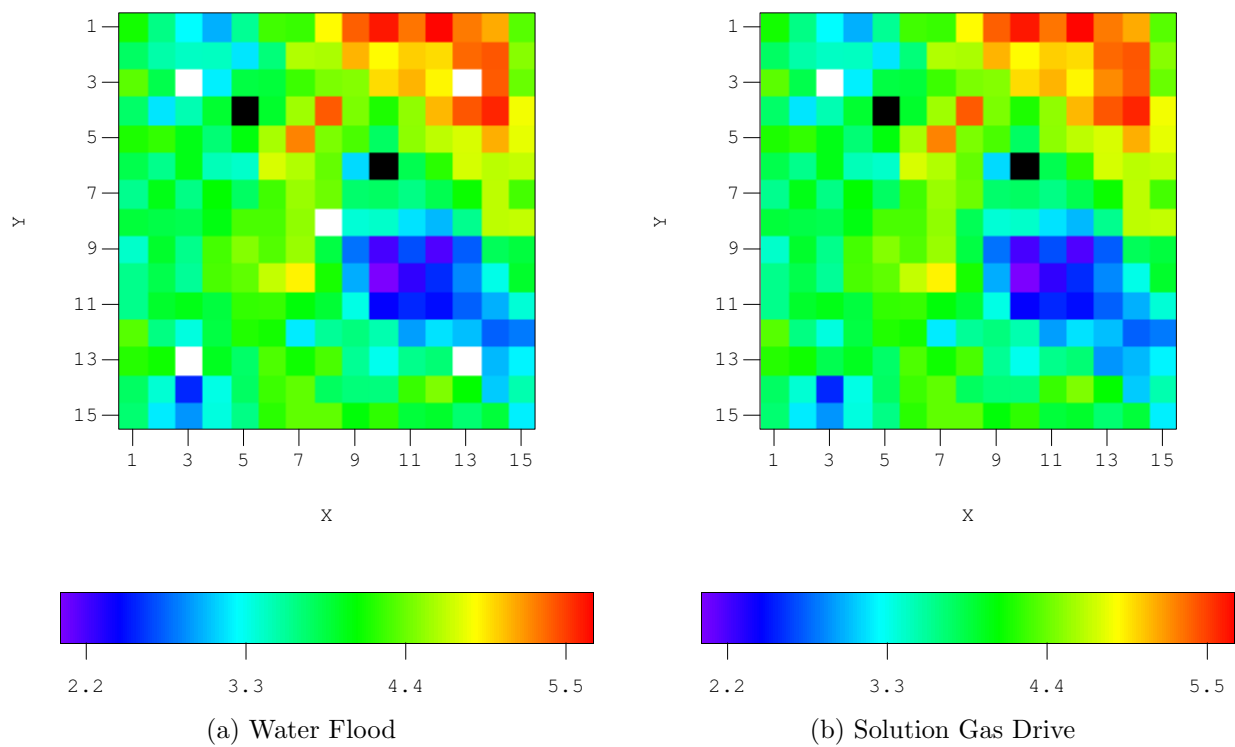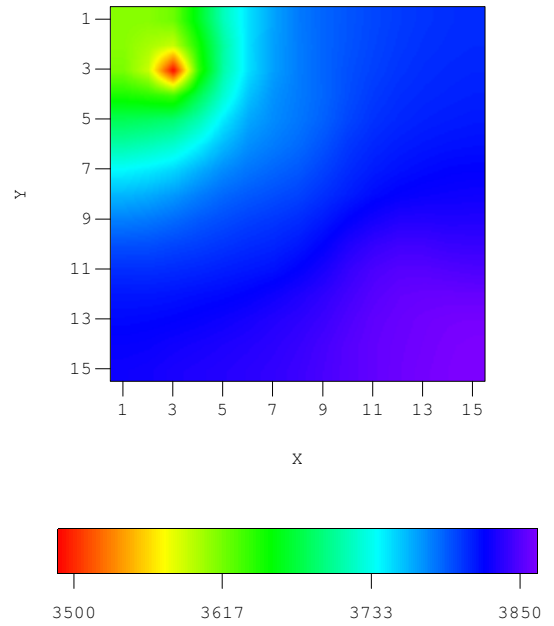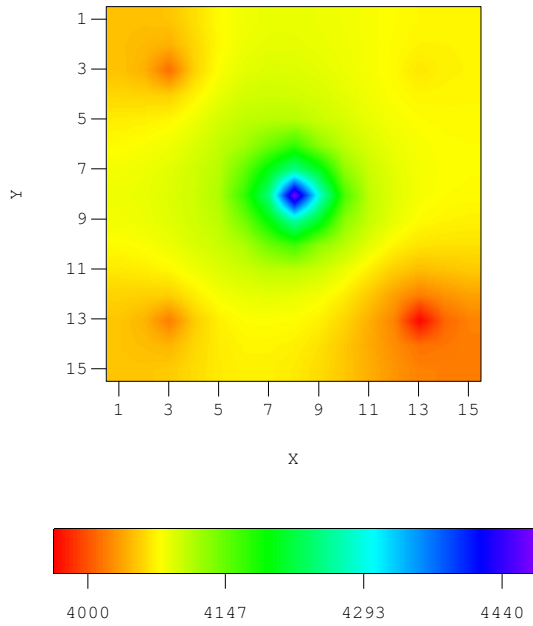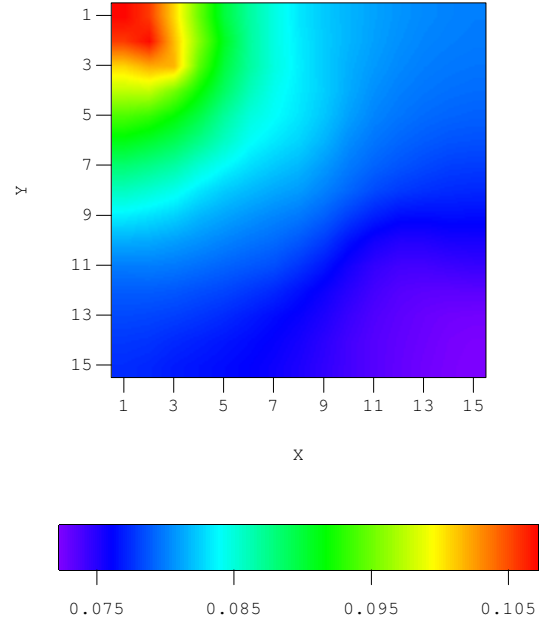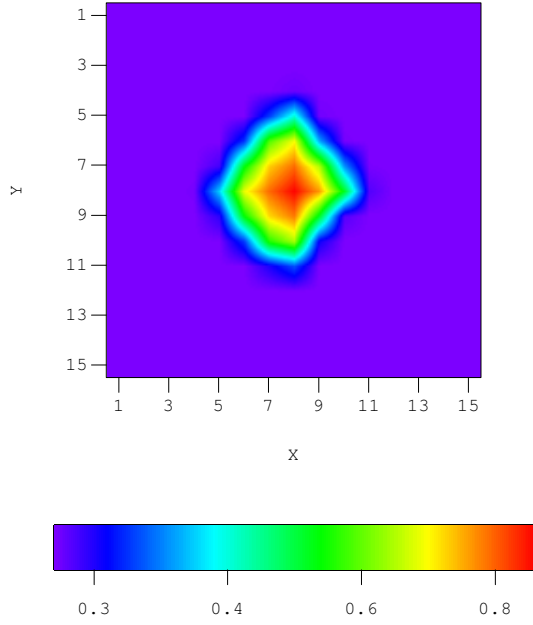
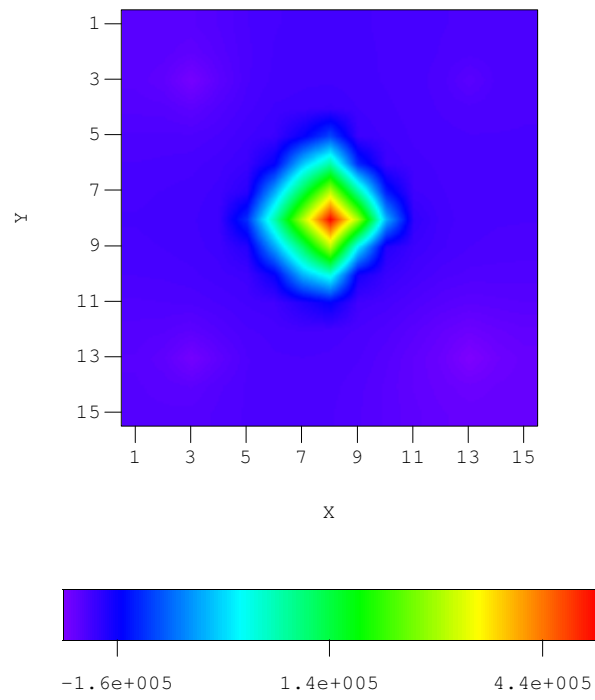Fig. 5.3: Water, Gas Saturation and Pressure Distribution at the 30th Day

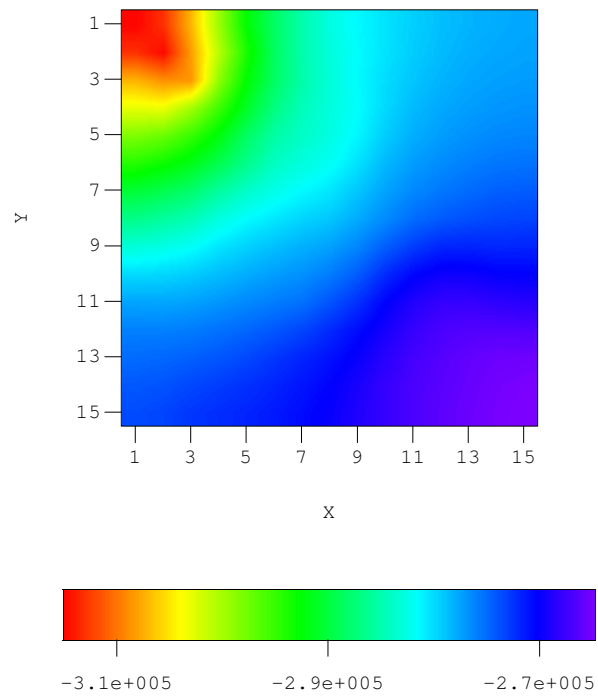Fig. 5.4: Seismic Impedance Change at the 30th Day (Water Flood)

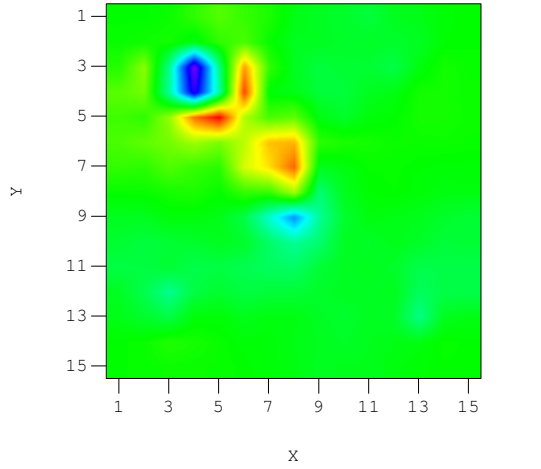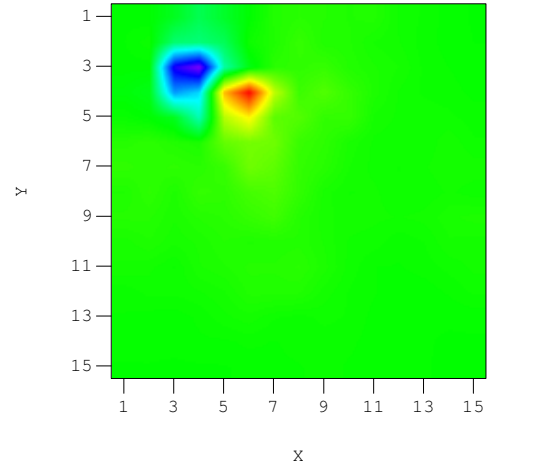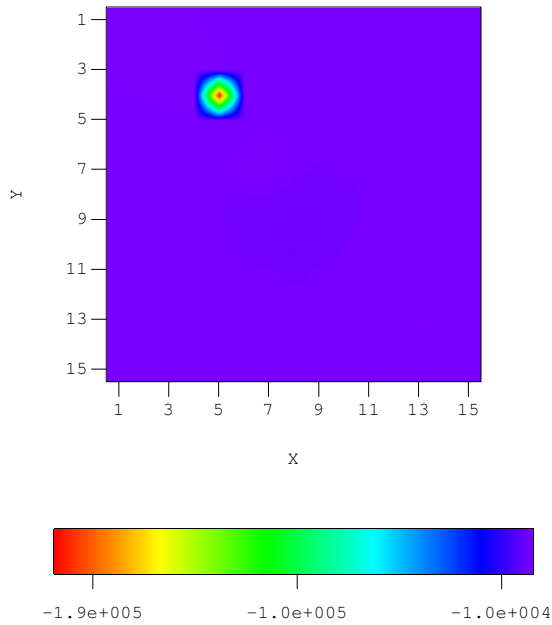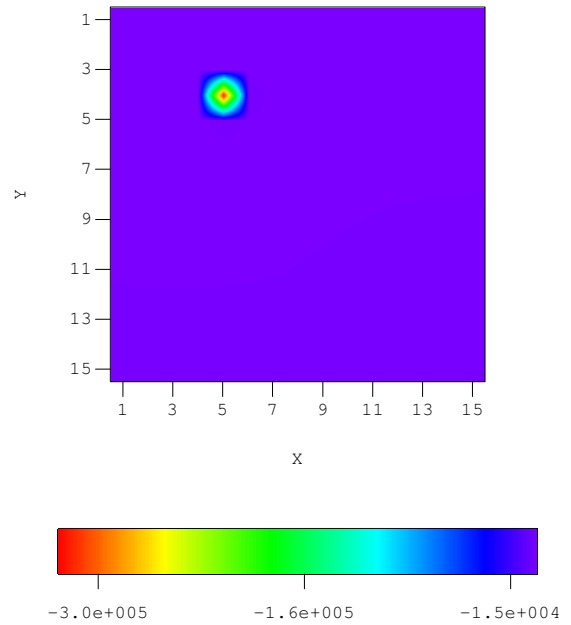Fig. 5.5: Seismic Impedance Change at the 30th Day (Solution Gas Drive)

(a) Sensitivity to ln(k) for Water Flood

(b) Sensitivity to ln(k) for Solution Gas Drive

(c) Sensitivity to ϕ for Water Flood

(d) Sensitivity to ϕ for Solution Gas Drive

Fig. 5.6: Sensitivity of Seismic Impedance Change in Gridblock $(4, 5)$ to $\ln(k)$ and ϕ at the 30th Day
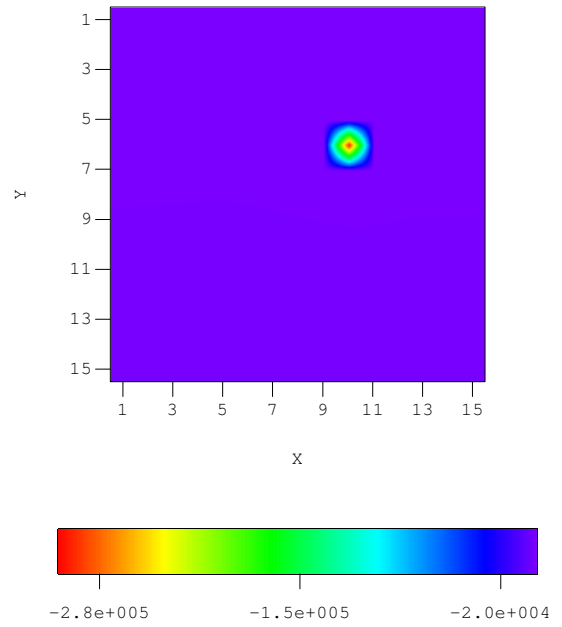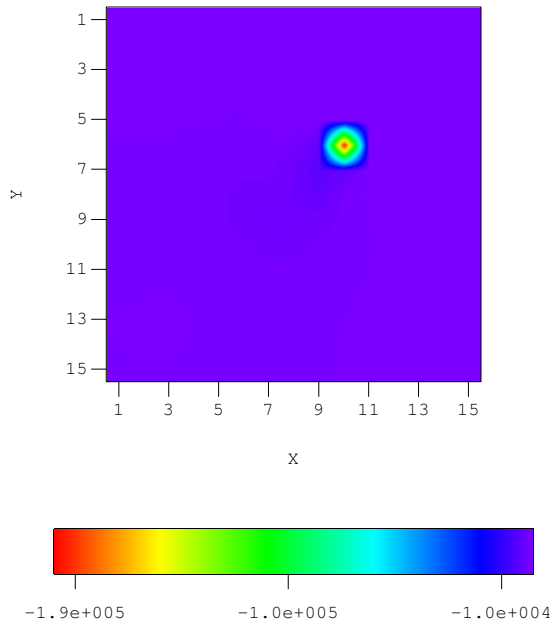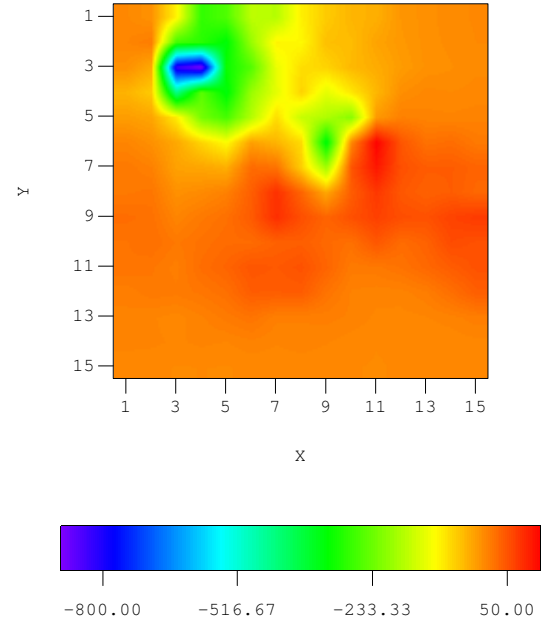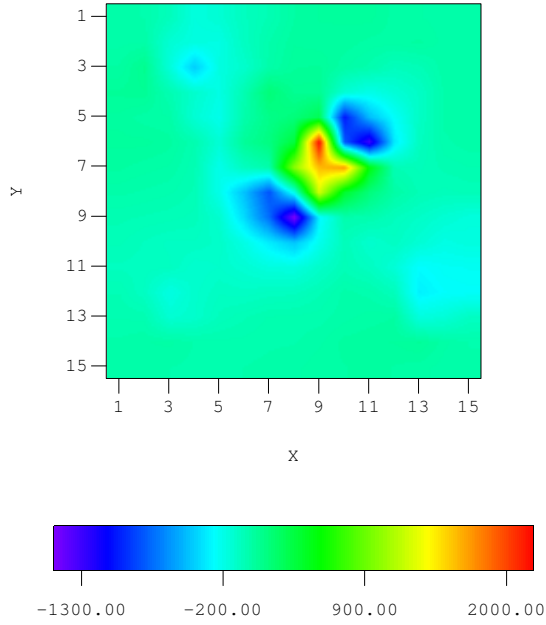
(a) Sensitivity to $\ln(k)$ for Water Flood

(b) Sensitivity to $\ln(k)$ for Solution Gas Drive

(c) Sensitivity to ϕ for Water Flood

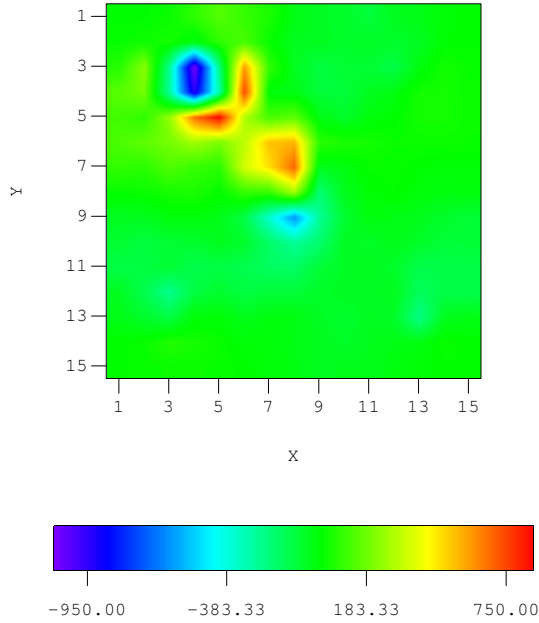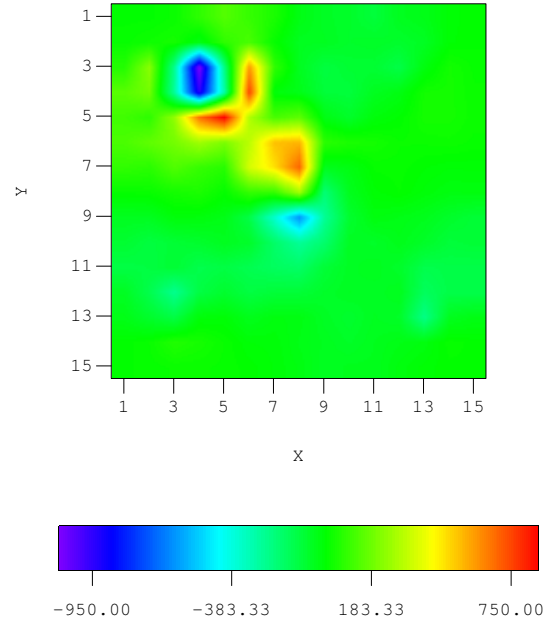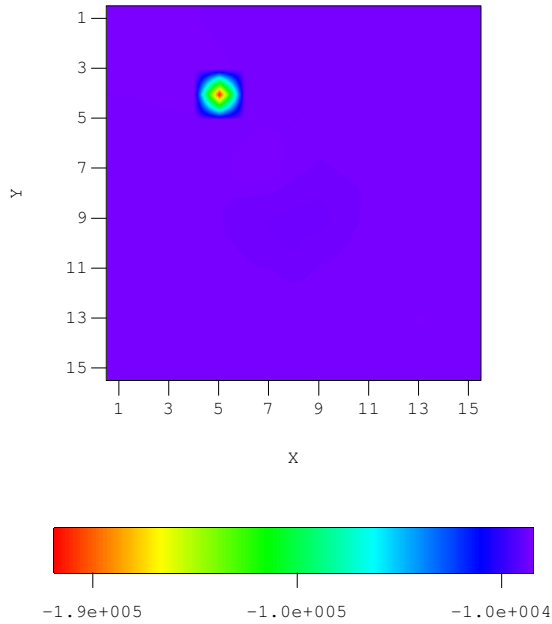(d) Sensitivity to ϕ for Solution Gas Drive

Fig. 5.7: Sensitivity of Seismic Impedance Change in Gridblock $(6, 10)$ to $\ln(k)$ and ϕ at the 30th Day

132

(a) Sensitivity to $\ln(k)$ by Adjoint Method

(b) Sensitivity to $\ln(k)$ by Perturbation Method

(c) Sensitivity to ϕ by Adjoint Method

(d) Sensitivity to ϕ by Perturbation Method

Fig. 5.8: Comparison between Adjoint Method and Perturbation Method in Computing Sensitivity of Seismic Impedance Change in Gridblock $(4,5)$ to $\ln(k)$ and ϕ at the 30th Day (Water Flood)
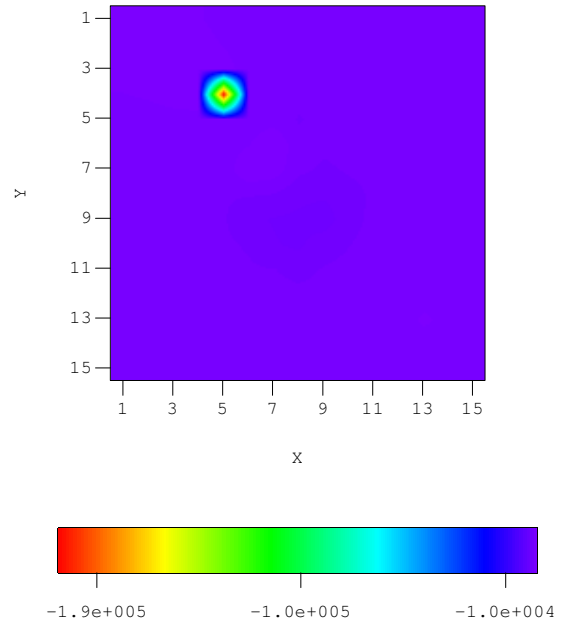
133

(a) Sensitivity to $\ln(k)$ by Adjoint Method

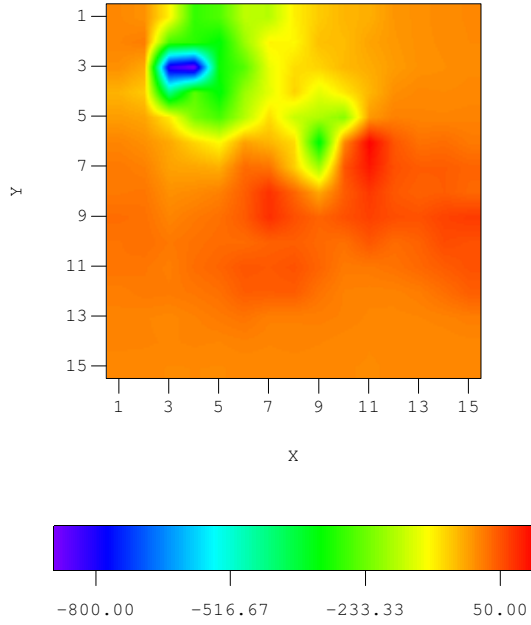(b) Sensitivity to $\ln(k)$ by Perturbation Method

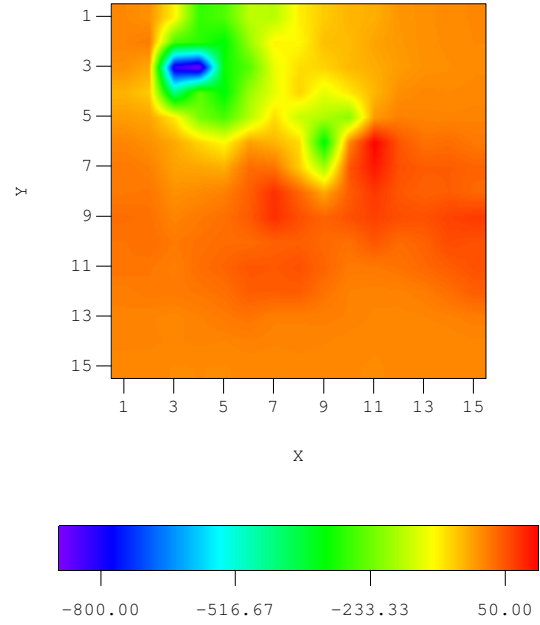(c) Sensitivity to ϕ by Adjoint Method

(d) Sensitivity to ϕ by Perturbation Method

Fig. 5.9: Comparison between Adjoint Method and Perturbation Method in Computing Sensitivity of Seismic Impedance Change in Gridblock $(6, 10)$ to $\ln(k)$ and ϕ at the 30th Day (Solution Gas Drive)

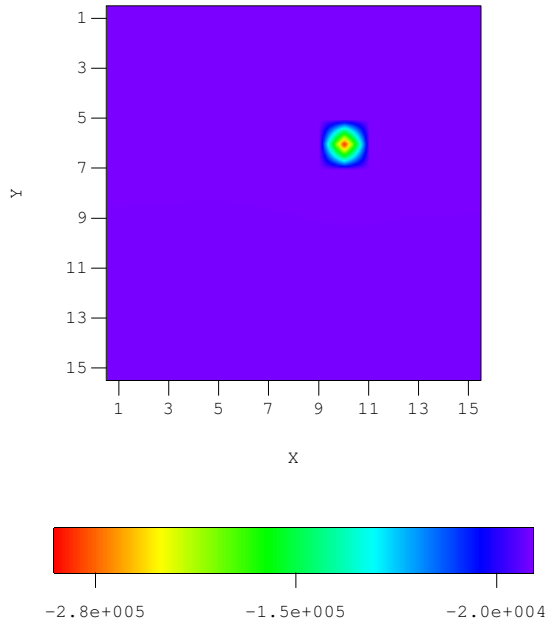Fig. 5.10: Relative Error between Adjoint Method and Perturbation Method in Computing Sensitivity of Seismic Impedance Change in Gridblock $(4, 5)$ to $\ln(k)$ at the 30th Day (Water Flood)
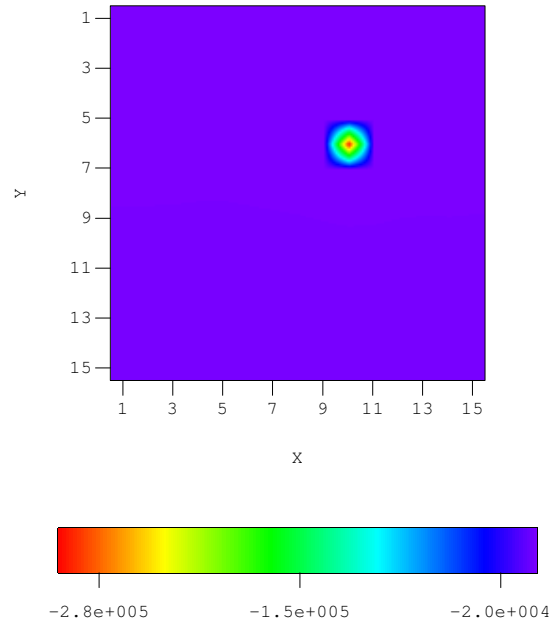
Fig. 5.11: Relative Error between Adjoint Method and Perturbation Method in Computing Sensitivity of Seismic Impedance Change in Gridblock $(4, 5)$ to ϕ at the 30th Day (Water Flood)

Fig. 5.12: Relative Error between Adjoint Method and Perturbation Method in Computing Sensitivity of Seismic Impedance Change in Gridblock $(6, 10)$ to $\ln(k)$ at the 30th Day (Solution Gas Drive)
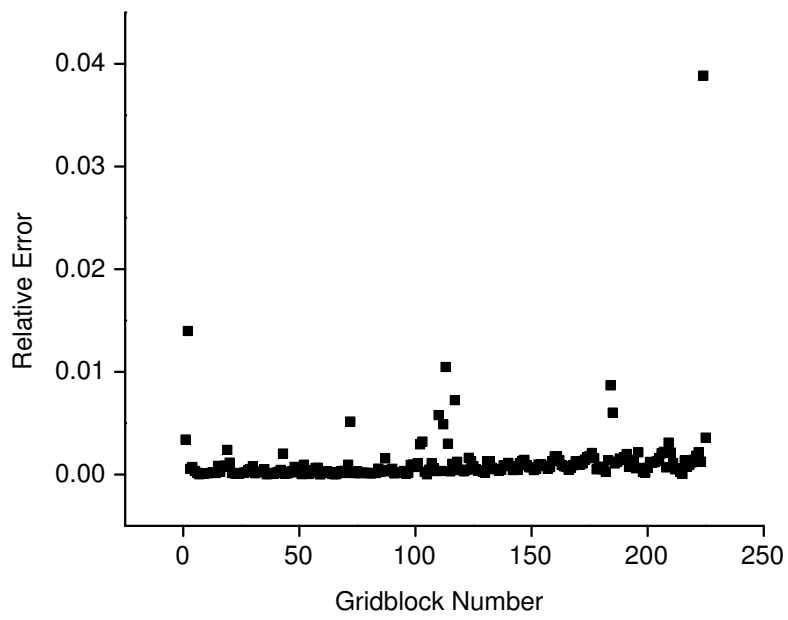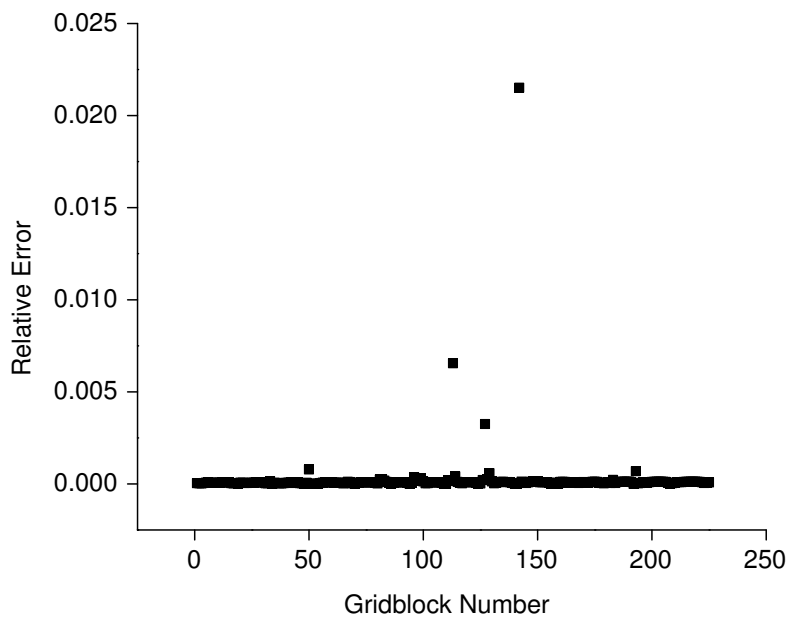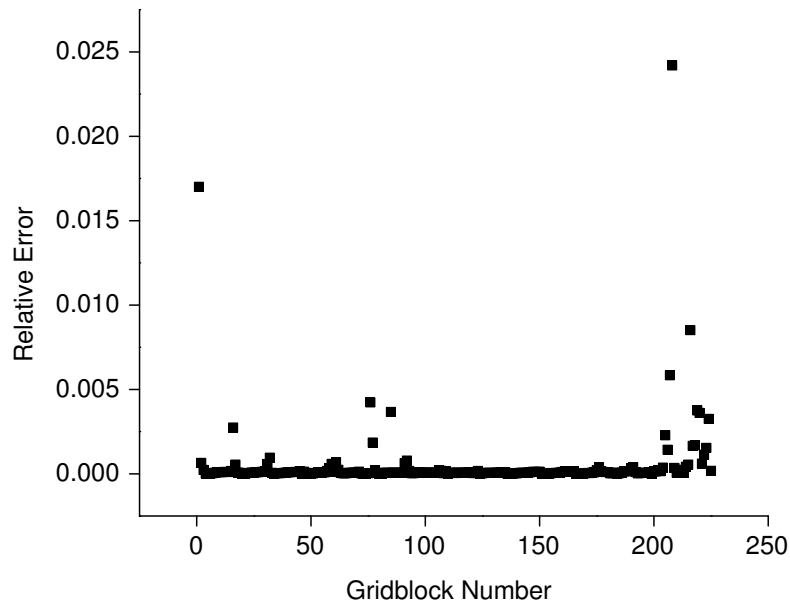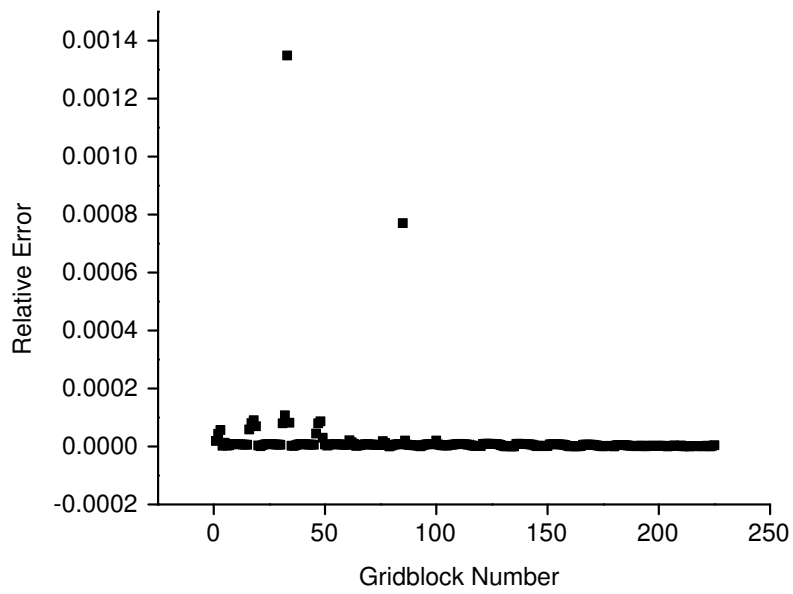
Fig. 5.13: Relative Error between Adjoint Method and Perturbation Method in Computing Sensitivity of Seismic Impedance Change in Gridblock $(6, 10)$ to ϕ at the 30th Day (Solution Gas Drive)

Chapter 6

HISTORY MATCHING OF FACIES DISTRIBUTIONS

6.1 Overview of the Truncated Gaussian Model

The truncated plurigaussian random field model is fairly appealing as a method for generating facies distributions. Galli and his coworkers at the center for Geostatistics in the School of Mines in Fontainebleau have established that it can be used to generate a remarkably wide variety of facies textures and shapes. Besides the qualitative visual appeal of the images, the method uses the covariance in a consistent way — unlike most indicator models. The truncated plurigaussian model is not, however, widely applied outside of France. The reason seems to be that it is fairly difficult to select model parameters that are consistent with a particular image such as might be provided by a geologist. We use the term parameters to refer to quantities such as the range, the variance, the covariance type (Gaussian, Exponential, Spherical, etc.) and the thresholds for discrimination of facies. While most of these parameters are obvious, parameterization of the thresholds can be done in many different ways. In the third section, we discuss our investigation of the model parameterization in some detail.

The variables of the model are the elements of the two Gaussian random fields that are truncated to determine facies. For a gridded model, there are two model variables for each grid block. In general, neither the parameters or the variables are known. The distributions of both are described by probability density functions. The prior pdfs for these parameters and variables will typically be quite simple and largely uninformative. To establish a more informative prior, we will assume that we will be provided with an example of the distribution for facies — one realization of the geology.

Our ultimate goal is to sample from the pdf of reservoir facies realizations conditional to the prior geology and to well observations and production data. We will separate this problem into two phases. In the first phase, we will develop a method for estimating the prior pdf of the model parameters conditional to the observation of a sample reservoir geology. In the second phase, we will develop the methods for applying RML to generate realizations of facies conditional to production data and well observations.

Three major improvements in the application of the truncated Gaussian method for lithofacies simulations based on indicators have been reported in the literature. 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 complex 3-D problem with vertical variation in the proportions of lithofacies. They also showed that this method preserved the consistency of the indicator variograms and cross variograms. The major achievement of this paper is the introduction of the truncated plurigaussian method. In this paper, facies were defined by truncating linear combinations of two or more Gaussian random functions which can be correlated or not. By doing so, the truncated Gaussian method becomes very flexible for general geological simulation problems. Although we have mapped out possible approaches for both phases of the project, this report includes only the initial sensitivity study establishing the features of the problem.

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 uncorrelated, the resulting facies sets obtained by truncating are not independent. 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. The choice of variogram models also will affect the resulting facies distributions. 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.

The paper by Le Loc'h and Galli (1997) presents 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.

6.2   Generating Thresholds

The choice of the method of truncation of the Gaussian variables is important in applying truncated Gaussian simulation in automatic history matching to generate reservoir models satisfying geological requirements. Le Loc'h et al. (1994) have applied rectangular thresholds in truncated Gaussian simulation. In their approach, lithofacies F_i is modeled by

$$F_i = \{x \in R^3; S^i_{j-1} \leq Y_j(x) < S^i_j, j = 1, \ldots, p\}, \tag{6.1}$$

where $Y_j(x)$, for $j = 1, \ldots, p$, are Gaussian random functions which can be dependent or independent. In practice, two Gaussian random functions were used to determine lithotypes, i.e. $p = 2$. The thresholds S^i_{j-1} and S^i_j were decided by the proportion of each lithotypes. Field facies maps with a variety of textures and patterns were generated by truncating groups of Gaussian functions with different combinations of variogram types. Fig. 6.1 shows some of their results. Although the rectangular thresholds approach is useful for geological simulation problems when conditioning to lithotype proportion data, it does not seem easy to be applicable to the problem of optimization of lithotype grouping in automatic history matching problems.

Alternatively Voronoi tessellation could be used to create the thresholds. Given a set of points in a plane, a Voronoi diagram can be created by dividing the plane into regions, so that every location in the region around one point is closer to that point than to any of the other points. Shirriff (1993) described an algorithm for creating Voronoi diagrams on a set of points to generate fractal patterns which resemble the randomness in nature. By attributing each region with a lithotype, a threshold map could be designed from a Voronoi diagram with great flexibility in optimization. Fig. 6.2 shows Voronoi diagrams generated by Shirriff (1993). However, the difficulty in calculating sensitivities of lithotype at any location in the threshold map to the locations of Voronoi points is the main obstacle in coupling the Voronoi type thresholds with efficient optimization methods which require computation of gradients.

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
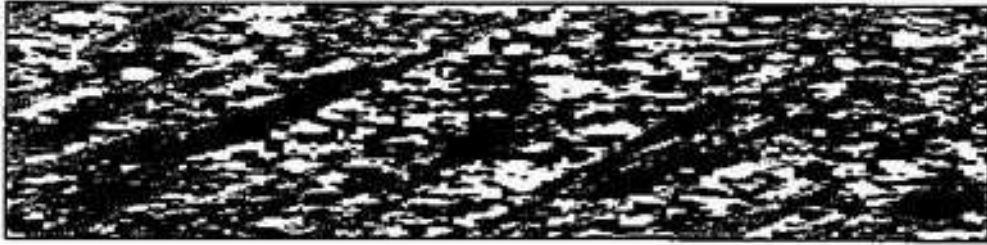
a: Exponential and spherical variograms.

b: Exponential and Gaussian variograms.

c: Exponential and Gaussian variograms.

Fig. 6.1: Simulation using independent Gaussian functions (Le Loc'h et al., 1994).

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:

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

(6.2)

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.

The coordinates of the threshold map are y_1 and y_2 respectively. Calculation of (y_1, y_2) for each gridblock in the field will be introduced in the next section.
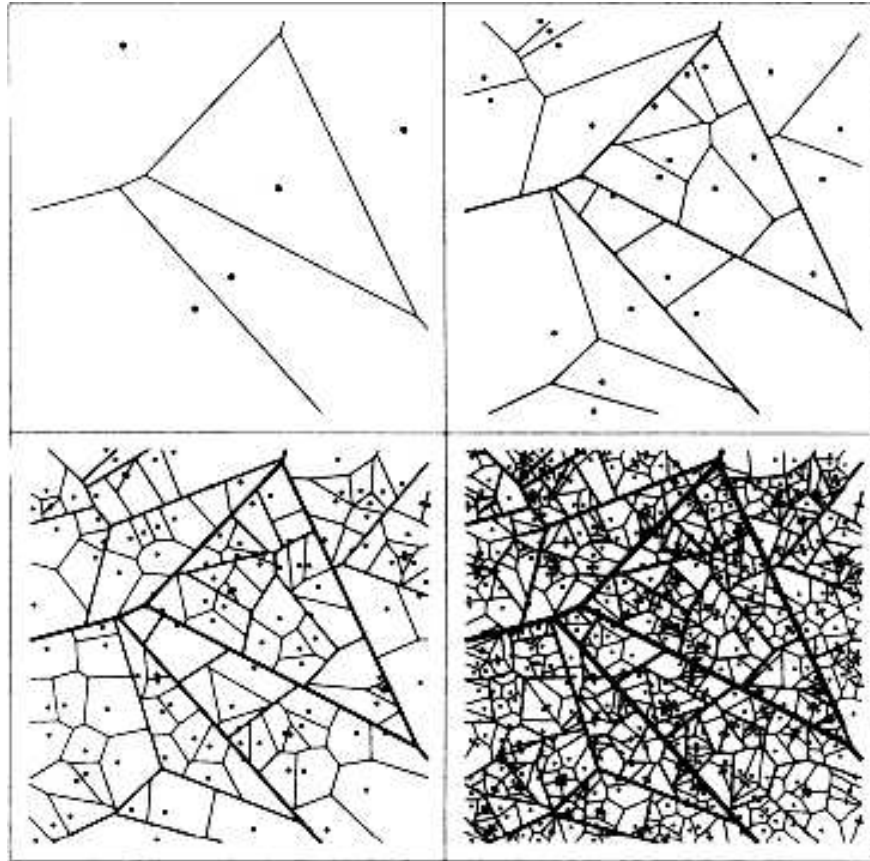
Fig. 6.2: Examples of Voronoi diagrams by Shirriff (1993).

6.3 Bayesian Model

The "model" will consist of model parameters and model variables. Model parameters, m_P, will include the discrete parameters that quantify the number of threshold lines, the type of covariance, and the number of facies regions occupied by each facies. Discrete parameters will be denoted as m_{PD}. Continuous model parameters, m_{PC}, include the range of the covariance and the locations of threshold lines. The prior (noninformative) distribution for covariance type will probably be assumed to be uniform. It seems likely that in the initial implementation, the number of threshold lines will probably be set to three unless we determine that is insufficient in which case we may allow more generality. The prior distribution for discrete parameters will be denoted as $P(m_{\mathrm{PD}})$. The continuous model parameters will describe the ranges of the covariance in the two principal directions for y_1 and y_2, the angle of the covariance axes and the locations of the threshold lines.

Our objective in phase I will be to develop a method for sampling from the pdf of the

model parameters conditional to an observation of a sample cross-section, d_{XS}.

$$P(m_{\text{PD}}, m_{\text{PC}}, d_{\text{XS}}) = P(m_{\text{PD}}, m_{\text{PC}}|d_{\text{XS}})P(d_{\text{XS}}) \tag{6.3}$$

and

$$P(m_{\text{PD}}, m_{\text{PC}}, d_{\text{XS}}) = P(m_{\text{PC}}|m_{\text{PD}}, d_{\text{XS}})P(m_{\text{PD}}|d_{\text{XS}})P(d_{\text{XS}}), \tag{6.4}$$

so

$$P(m_{\text{PC}}, m_{\text{PD}}|d_{\text{XS}}) = P(m_{\text{PC}}|m_{\text{PD}}, d_{\text{XS}})P(m_{\text{PD}}|d_{\text{XS}}). \tag{6.5}$$

We believe that we will be able to sample quite well from the distribution $P(m_{\text{PC}}|m_{\text{PD}}, d_{\text{XS}})$ using the method of RML. We will develop a method for sampling from $P(m_{\text{PD}}|d_{\text{XS}})$.

6.4 The Initial Facies Model

For this example, the reservoir model is designed as two dimensional and heterogeneous, with the size 300 ft × 300 ft. It is discretized into 128×128 gridblocks. The facies fields were generated using the square-root method for Gaussian random fields. y_1 is a Gaussian random variable with the Gaussian covariance $C_1(r) = (\sigma_r)^2 \exp(-3r^2/a^2)$. y_2 is a Gaussian random variable with the Exponential covariance $C_2(r) = (\sigma_r)^2 \exp(-3|r|/a)$, where a is the range of the covariance and r is the distance between two locations. The parameters used for the truncated Gaussian simulation examples are summarized in Table 6.1.

	y_1	y_2
Variance	1.0	1.0
Range (ft)	90.	120.
Prior mean	0	0
Covariance	Gaussian	Exponential

Table 6.1: Geostatistical parameters for simulation of the facies fields.

Random fields y_1 and y_2 are computed by

$$Y = \mu + LZ, \tag{6.6}$$

where the mean $\mu = 0$ for both fields, and Z is a vector of independent identically distributed Gaussian variables with mean 0 and variance 1. L is a square root of the covariance matrix as shown in Eq. 6.7:

$$C = LL^T. \tag{6.7}$$

144

Both vectors y_1 and y_2 have the same dimension as the number of total gridblocks. The facies in the nth gridblock is determined by the region of the threshold diagram in which the point $(y_1(n), y_2(n))$ is located. Fig. 6.3 shows gray-scale plots of y_1 and y_2 in the field. To have a better comparison of the fields with the Gaussian and the Exponential covariance, another field was generated using the Gaussian kernel but multiplied by the same vector used to generate the Exponential covariance field (see Fig. 6.4).
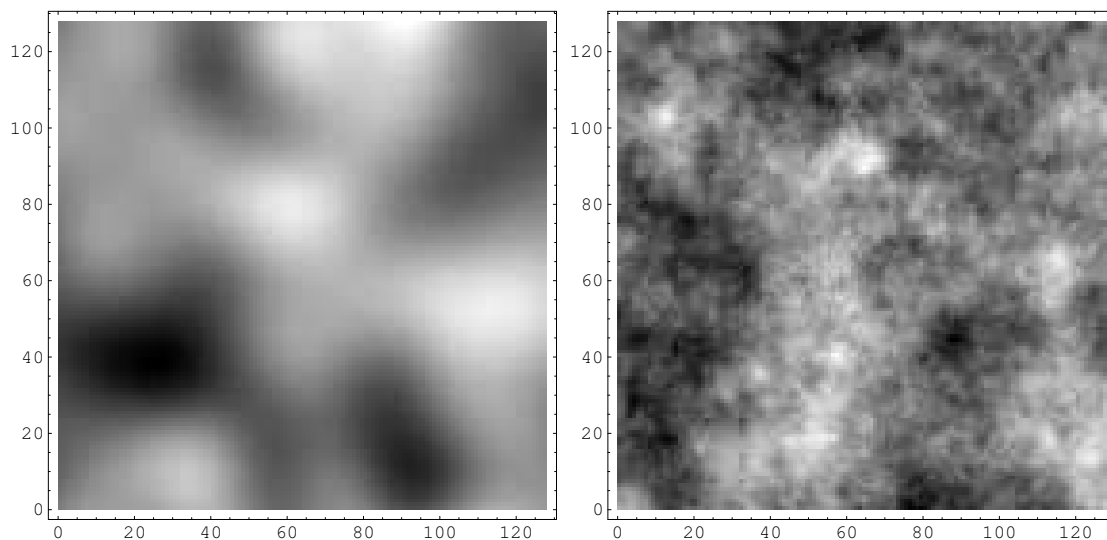

Fig. 6.3: The gray-scale plots of y_1 and y_2 in the field. The kernel of y_1 is from Gaussian covariance and y_2 is from Exponential covariance.

6.5 Relationship between r, θ and the Facies Map

To acquire insight into the relationship of the model parameters and the facies map of the field, the following investigations were made.

A base threshold map was generated using the initial r and θ sets as:

$$R = [-0.140178, -0.680754, -1.10564],$$

$$\Theta = [2.50385, 1.77335, 0.379948].$$

The plot of each of the threshold lines numbered from 1 to 3 and the resulting map of facies regions are shown in Fig. 6.5. Starting from the top-most region, counterclockwise, the regions segregated by the three threshold lines are given numbers 1 to 6, and the region in the middle is the facies region 7. We consider the situation in which there are three types
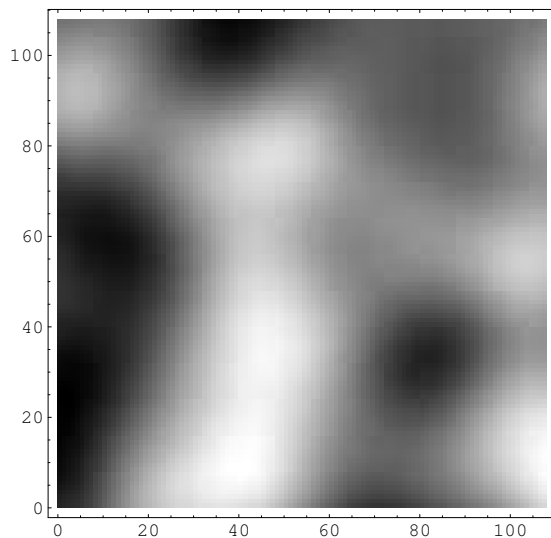
145

Fig. 6.4: The gray-scale plot of the convolution product of the Gaussian kernel with the random multivariate normal array used in computing y_2.

of facies, namely, A, B and C, in the field, and they are attributed to each region as in Table 6.2. The plot on the left of Fig. 6.6 shows the distribution of facies on the threshold map. The plot on the right is the resulting facies map of the field from this set of thresholds. Black indicates facies A. Grey and white indicate facies B and C respectively.

Region $No.$	1	2	3	4	5	6	7
Facies Type	A	B	A	C	C	B	B

Table 6.2: Facies type for each region in the threshold map.

In the first phase of the project, we will develop the methodology for estimating the pdf of the parameters describing the threshold lines given a sample realization of the reservoir, as provided from a geologist or from an outcrop of the formation. Although rock facies are discrete and hence not differentiable, we believe that the most efficient methods for generating conditional realizations may be using the method of Randomized Maximum Likelihood. In this case, we will need to approximate the discrete function by one that is differentiable and then compute the gradient analytically. For this report, the effect of relatively small perturbations of the model parameters will be investigated by perturbing a base model, one parameter at a time.
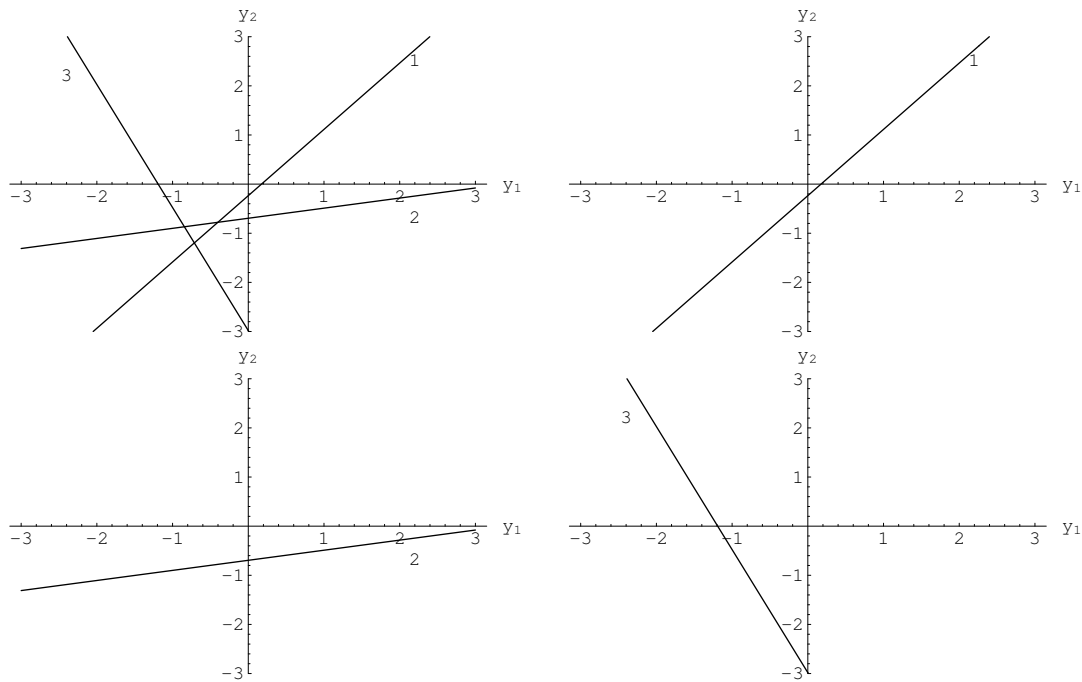
146

Fig. 6.5: The base thresholds map and each threshold line numbered from 1 to 3.

6.5.1 Perturbing r

In the first step, the distance of a randomly picked threshold line to the origin was perturbed by parallel translation of the line. Fig. 6.7, 6.8 and 6.9 show the threshold maps and the resulting facies maps by increasing r values of the second threshold line by 0.5 at a time. The fraction of facies A, i.e. the black area, was reduced and scattered gradually. And the facies C, the white area, was increased.

6.5.2 Perturbing θ

Now starting from the same base threshold map as in the previous test, the second threshold line was rotated by only increasing its slope angle step by step. The distance of the second threshold line to the origin point was not changed and the remaining threshold lines were also kept as in the base case. The threshold maps and the resulting facies maps while rotating the second threshold line are recorded in Figs. 6.10 to 6.14. A variety of textures and shapes are observed from these facies maps.
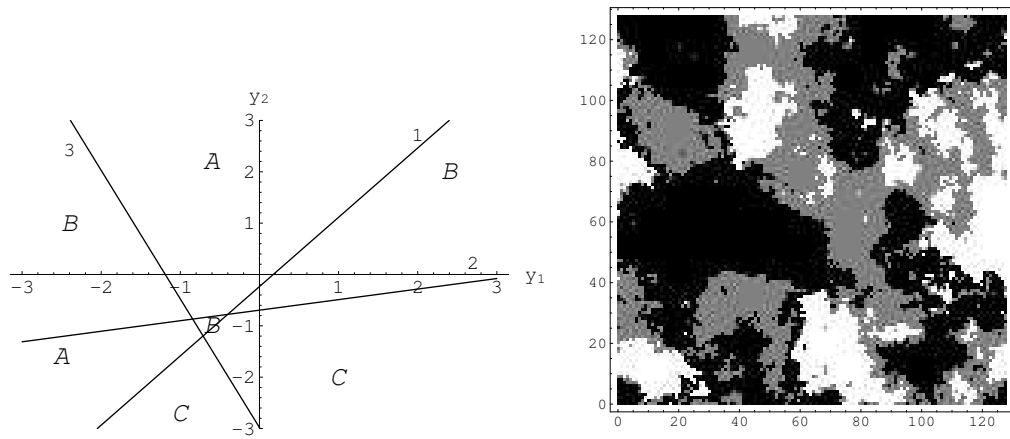
147

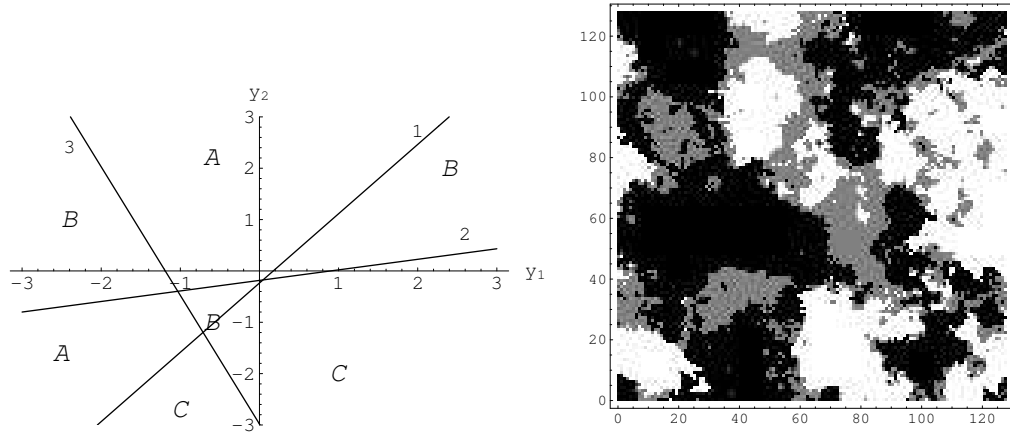Fig. 6.6: The threshold map and the facies map of the 2-D field for the base case.

Fig. 6.7: The thresholds map and the facies map after increasing r_2 by 0.5.

6.5.3 Sensitive Regions

In the threshold map, the most sensitive region of changing facies types to the moving of the thresholds should correspond to the region adjacent to the threshold lines. Similarly, on the facies map, the boundaries between different facies types should be the most sensitive region to changes in threshold parameters. To show this, the parameters, r_1, r_2, r_3, θ_1, θ_2 and θ_3, were perturbed one by one with $\Delta r = 0.25$ and $\Delta \theta = 10°$. For the plots of sensitivity, the facies were arbitrarily assigned integer values, 1, 2 and 3. The difference of the values is displayed in gray scale maps. Figs. 6.15 to 6.20 illustrate the sensitive region to each of the parameters.
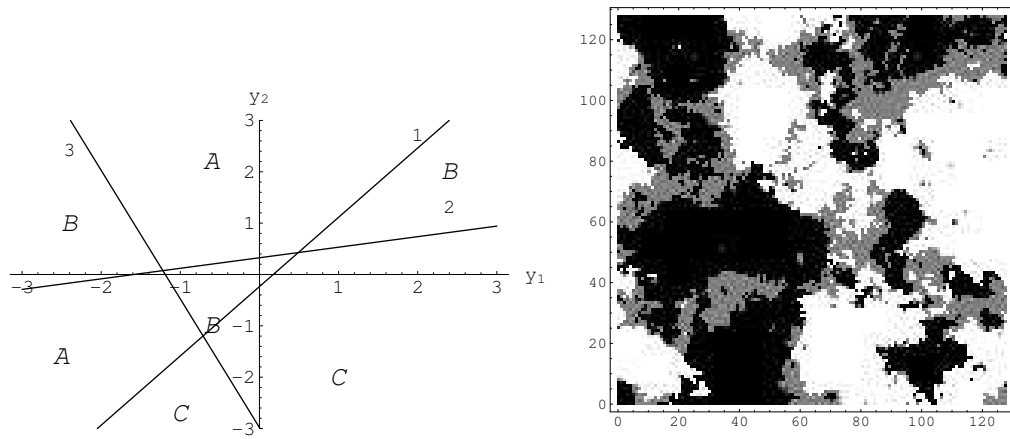
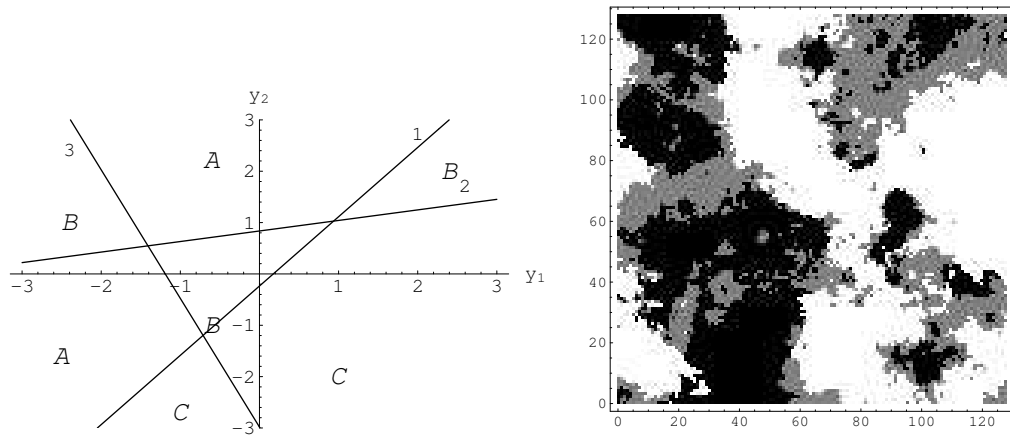Fig. 6.8: The thresholds map and the facies map after increasing r_2 by 1.0.

Fig. 6.9: The thresholds map and the facies map after increasing r_2 by 1.5.

6.6 Facies Arrangements

By correctly arranging the facies types in the threshold map, the resulting distribution of facies in the field can be controlled to match geological requirements of rock type distributions. To show this, four different facies arrangement schemes, besides the base case, were made and are shown together with corresponding facies maps in Figs. 6.21, 6.22, 6.23 and 6.24. The case shown in Fig. 6.21 allows facies B be adjacent to both facies A and facies C, while restricting direct contact between facies A and facies C. Reducing the portion of facies A by replacing A with facies C in the region 1, a field map for a very different geological condition was generated as shown in Fig. 6.22. In Fig. 6.23, facies B was wrapped in facies A. The facies map is generally divided into two regions, the black and gray region and the white region. This kind of facies map might be typical for alluvial plains. In Fig. 6.24, any
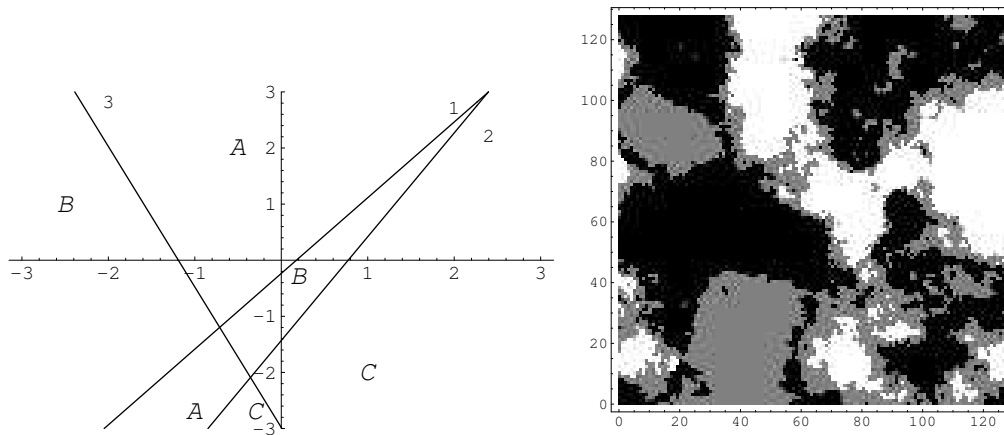
149

Fig. 6.10: The thresholds map and the facies map after increasing θ_2 by 50 degrees.
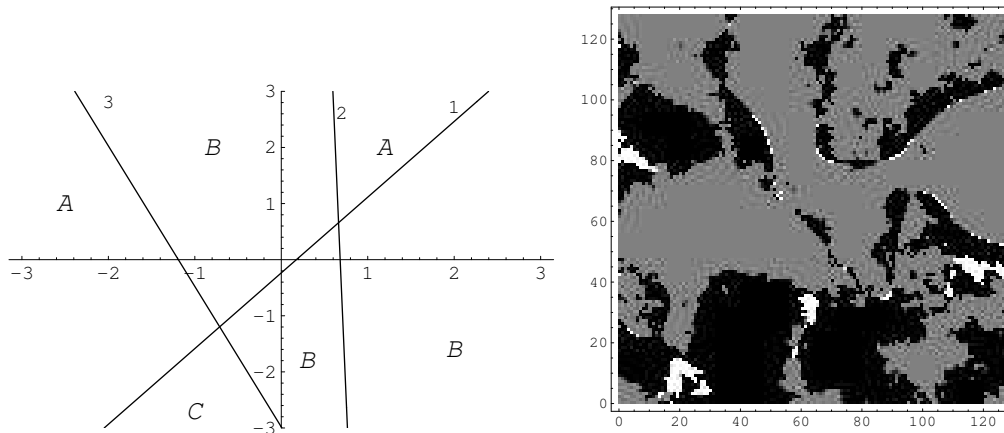

Fig. 6.11: The thresholds map and the facies map after increasing θ_2 by 80 degrees.

facies among the three has similar possibilities for contacting each of the other two facies.

6.7 Anisotropic Covariance

Gaussian fields with different ranges in the main directions can also be simulated using the truncated Gaussian method. In this case, the Gaussian covariance of the Gaussian random variable y_1 is calculated as $C_1(x, y) = \sigma^2 \exp(-3(\frac{x^2}{a_x^2} + \frac{y^2}{a_y^2}))$, and the Exponential covariance of the Gaussian random variable y_2 is $C_2(x, y) = \sigma^2 \exp(-3\sqrt{\frac{x^2}{a_x^2} + \frac{y^2}{a_y^2}})$. Here, a_x and a_y are the range of the covariance in the x and y directions. The parameters used for the truncated Gaussian simulation examples are summarized in Table 6.3. Gaussian random fields y_1 and y_2 were calculated the same way as described before. The same two sets of Z vectors were
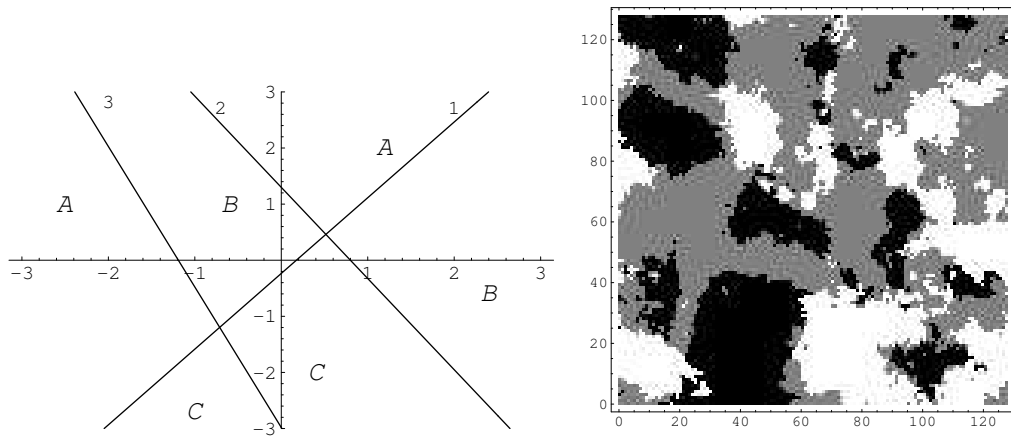
Fig. 6.12: The thresholds map and the facies map after increasing θ_2 by 110 degrees.
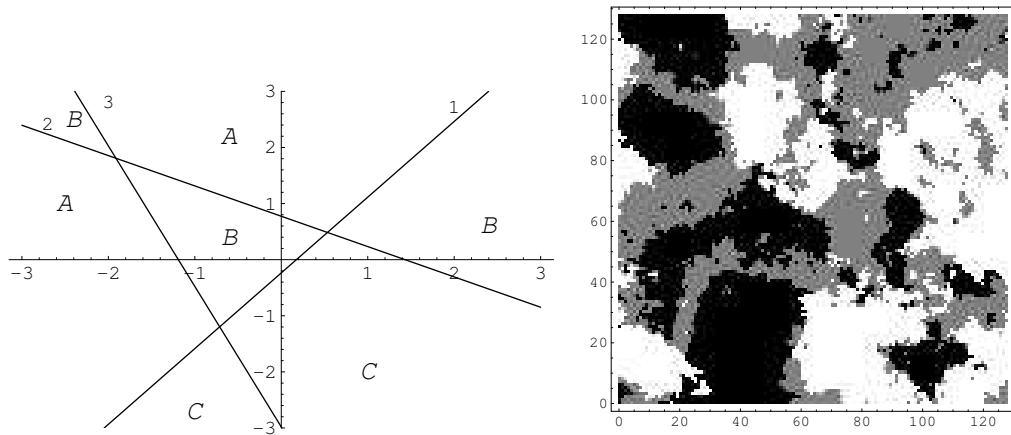
Fig. 6.13: The thresholds map and the facies map after increasing θ_2 by 140 degrees.

used in the calculation. Fig. 6.25 shows gray-scale plots of y_1 and y_2, which have the same dimensions as the field map. Fig. 6.26 is the facies field generated using the preceding y_1 and y_2. Facies types were decided by using the base facies threshold map. So, even though other conditions are exactly the same with base isotropic covariance case, the directional difference in ranges has made a distinct appearance in the facies map.

6.8 Comments

The preceding facies distribution plots obtained by perturbing each of the threshold parameters, such as r, θ and facies attribution in the threshold map, have shown that the truncated Gaussian simulation method is able to generate facies maps with a wide variety of textures
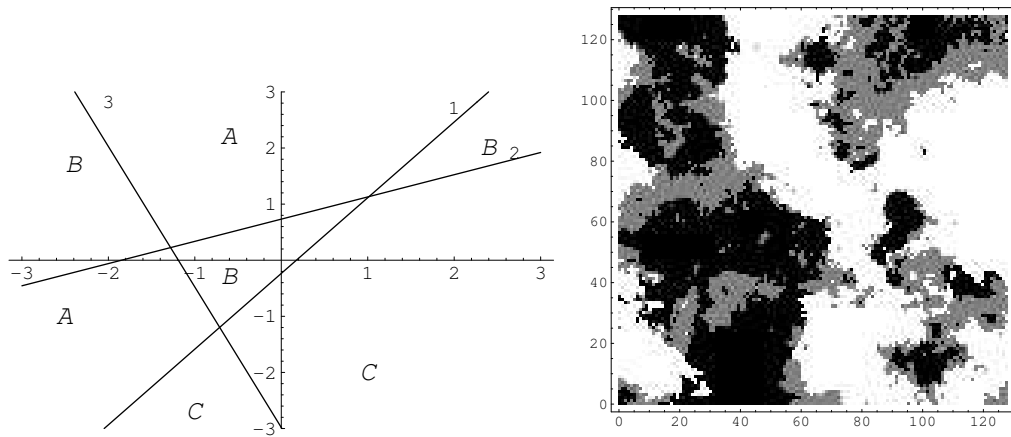
Fig. 6.14: The thresholds map and the facies map after increasing θ_2 by 190 degrees.
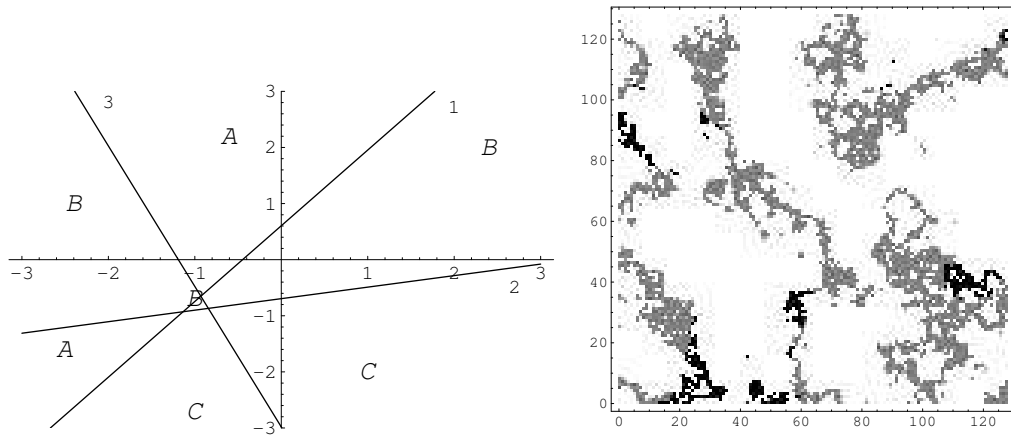
Fig. 6.15: Sensitive regions by increasing r_1 by 0.5.

and shapes. Using intersecting lines as thresholds of truncation, facies map can be easily adjusted. More importantly, for any location in the field, the sensitivities of the facies type to the threshold parameters can be conveniently calculated by using intersecting lines as thresholds. As expected, the plots of the sensitive regions have shown that on the field facies map, the most sensitive regions to changes of threshold parameters are the boundaries between different facies. Introduction of Gaussian functions with anisotropic covariances into truncated Gaussian simulation have brought more flexibility to this method in matching facies models.
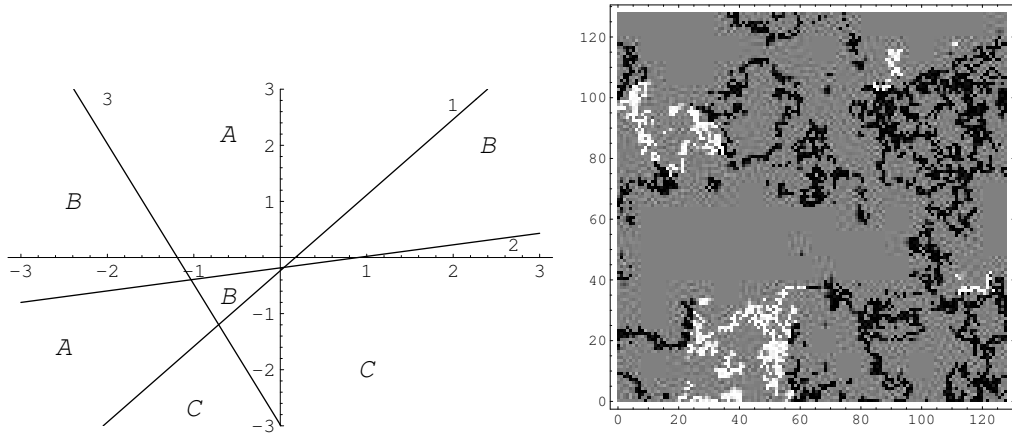
Fig. 6.16: Sensitive regions by increasing r_2 by 0.5.
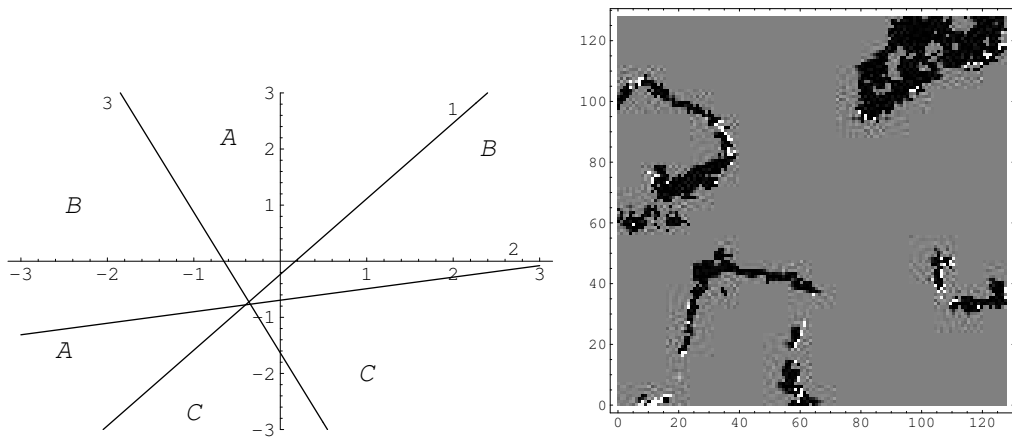
Fig. 6.17: Sensitive regions by increasing r_3 by 0.5.
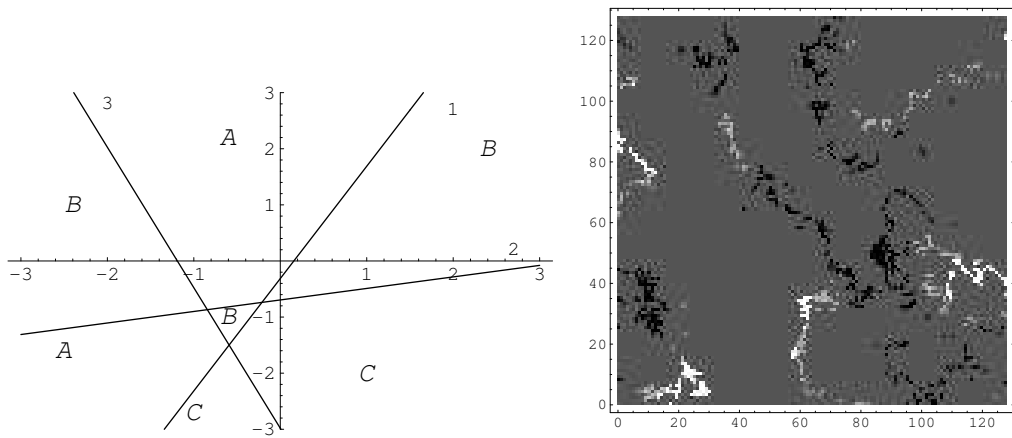
Fig. 6.18: Sensitive regions by increasing $\theta 1$ by 10 degrees.

153
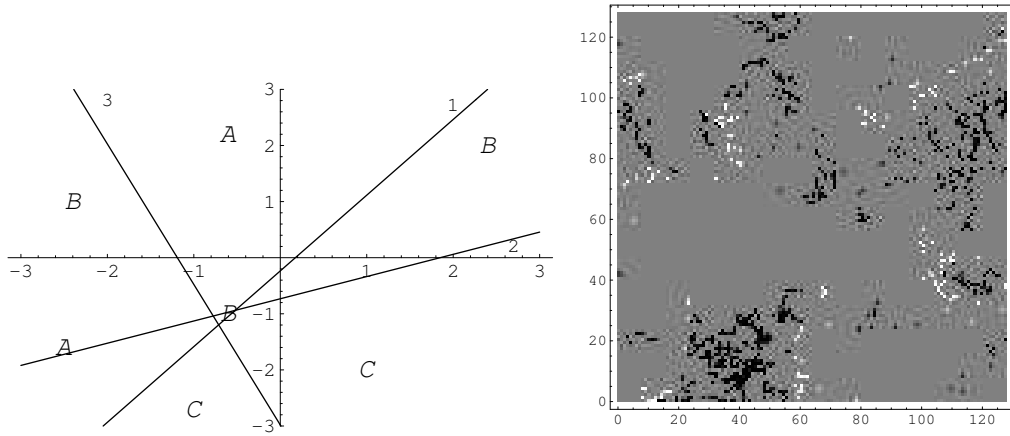
Fig. 6.19: Sensitive regions by increasing $\theta2$ by 10 degrees.
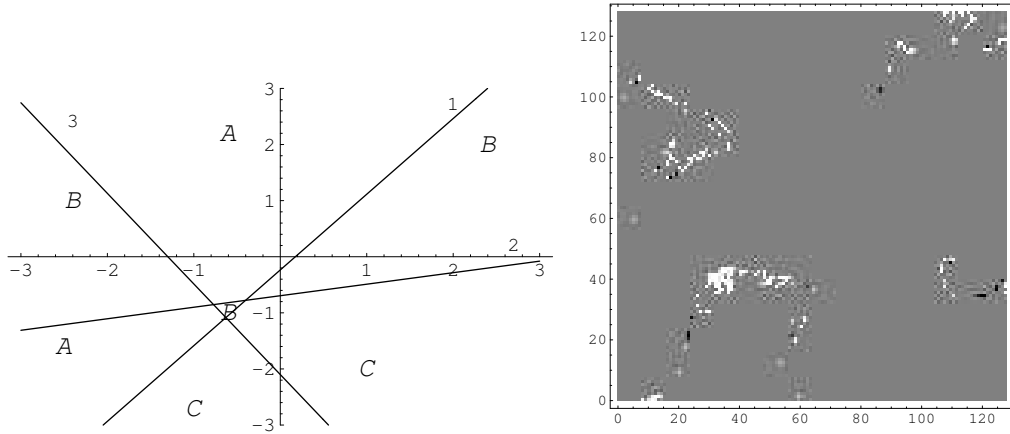
Fig. 6.20: Sensitive regions by increasing $\theta3$ by 10 degrees.
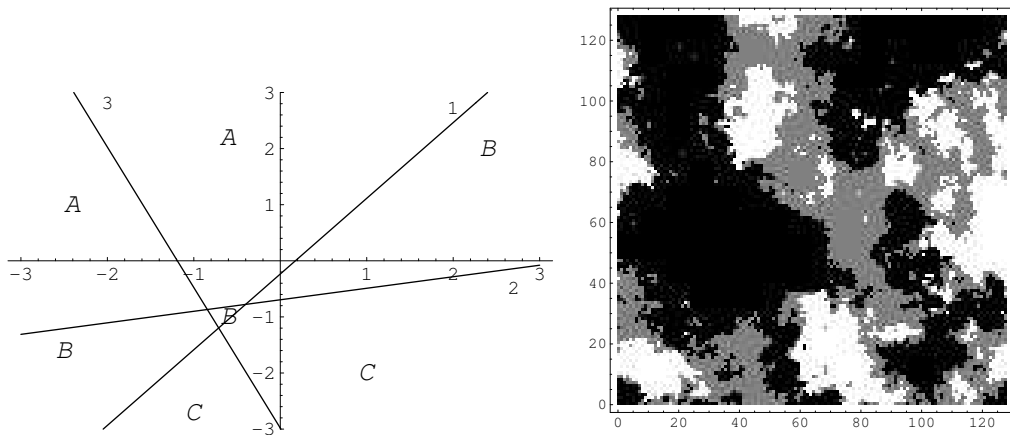
Fig. 6.21: The thresholds map and the facies map. Facies A (black) and facies C (white) were restricted from having direct contact.

154
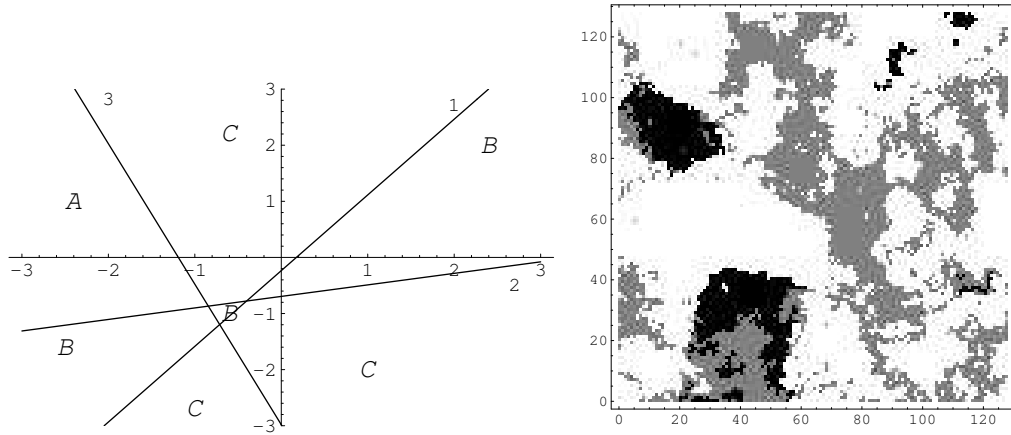
Fig. 6.22: The thresholds map and the facies map. A big portion of facies A (black) in last case was replaced by facies C (white).
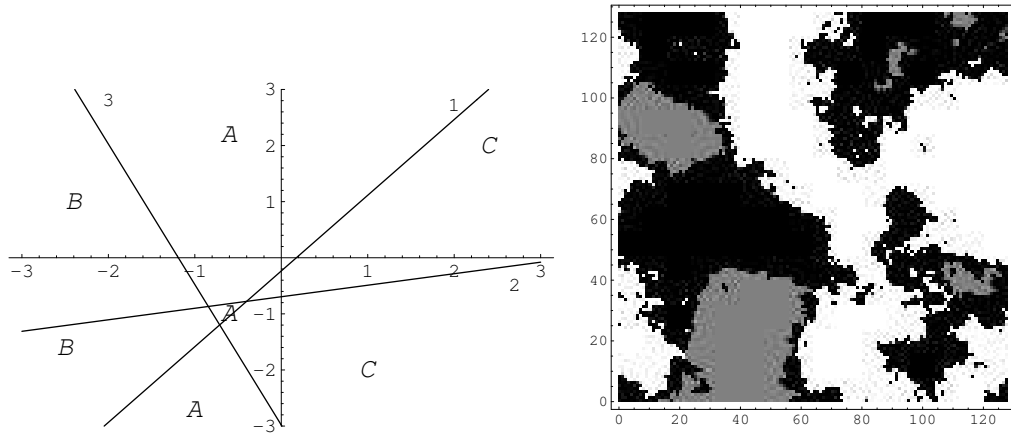
Fig. 6.23: The thresholds map and the facies map. Facies B (gray) is wrapped in facies A (black).

	y_1	y_2
variance	1.0	1.0
range x (ft)	60.	120.
range y (ft)	120.	60.
prior mean	0	0
Distribution	Gaussian	Exponential

Table 6.3: Geostatistical parameters for simulation of the facies fields with anisotropic co-variances.

155
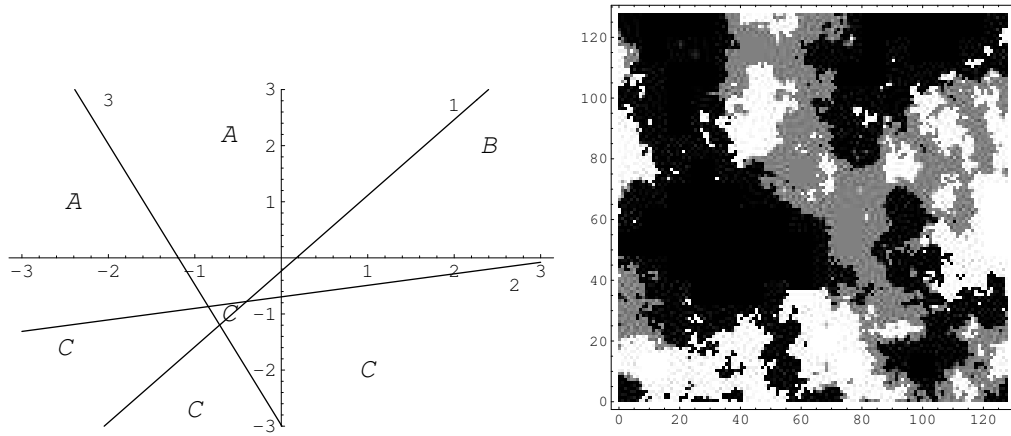
Fig. 6.24: The thresholds map and the facies map. The three facies have similar possibility of contacting each other.
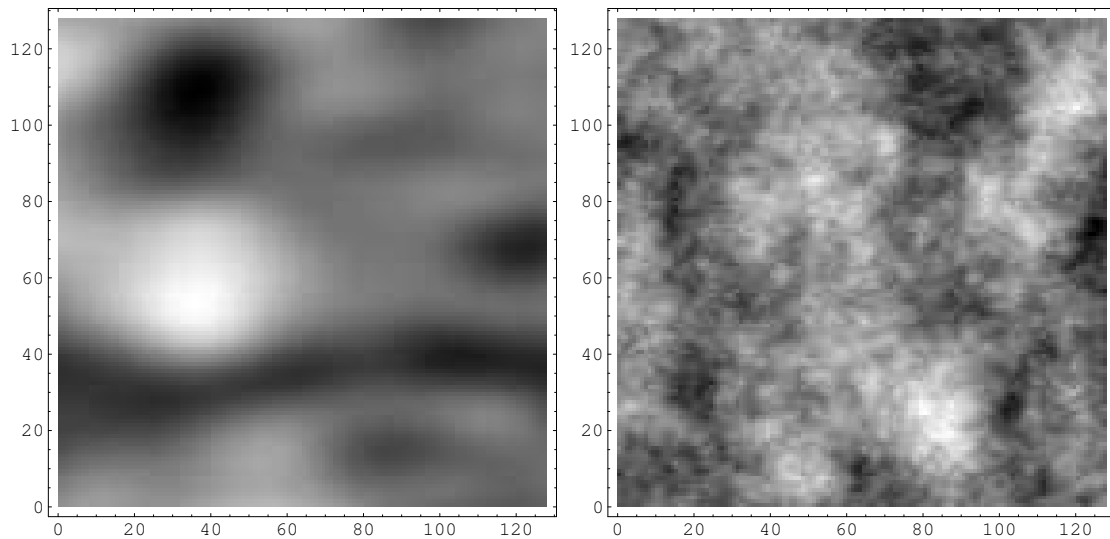
Fig. 6.25: The gray-scale plots of y_1 and y_2 in the field. The kernel of y_1 is from Gaussian covariance and y_2 is from Exponential covariance.
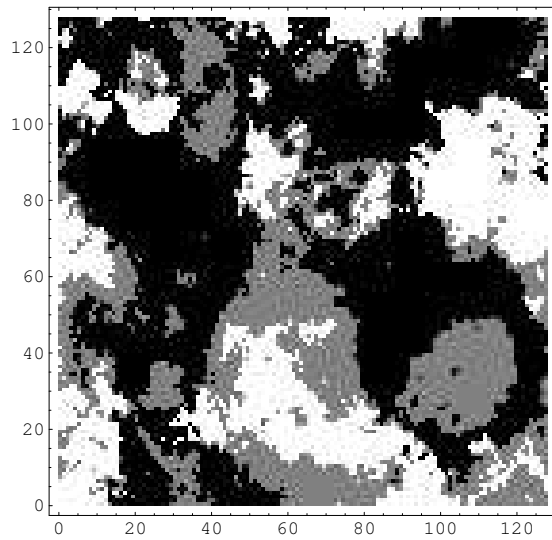
Fig. 6.26: The facies map of the field with anisotropic covariances.

Chapter 7

CONCLUSIONS

The Bayesian approach to automatic history matching of production data requires the application of an optimization algorithm to minimize an objective function which includes the sum of squared production data misfit terms plus a regularization term arising from a prior geostatistical model constructed from static data. For field scale problems where the number of production data history matched ranges from several dozen to a few thousand, the number of reservoir variables (model parameters) estimated by history matching ranges from many hundred to tens of thousands, standard gradient based optimization algorithms such as the Gauss-Newton and Levenberg-Marquardt algorithms which require calculation of the sensitivity of each production data to each model parameter require too much time and memory to be useful. Thus we have investigated the application of two algorithms which require only the gradient of the objective function. Specifically, we have implemented and tested a nonlinear conjugate gradient method and a scaled limited memory Broyden-Fletcher-Goldfarb-Shanno (LBFGS) algorithm for the automatic history matching problems. We have found an appropriate procedure for scaling in the LBFGS algorithm, but have not found a preconditioner for the conjugate gradient which makes this method competitive with the scaled LBFGS algorithm. The LBFGS algorithm and preconditioned conjugate gradient algorithms require calculation of the gradient of the objective function. We have implemented and tested an adjoint procedure for calculating this gradient. The LBFGS has been successfully applied to two field type history matching examples. Based on our investigation of application of optimization algorithms to the problem of history matching production data, we make the following specific conclusions:

- The adjoint method yields sufficiently accurate estimates of the gradient of the objective for history matching purposes.

- The scaled LBFGS algorithm implemented is sufficiently robust and computationally efficient for application to field scale history matching problems.

- For history matching of production data, the scaled LBFGS is a more robust and computationally efficient than the Gauss-Newton, Levenberg-Marquardt and preconditioned conjugate gradient algorithms tested.

- The limited memory BFGS algorithm requires significantly less time and less memory than the modified Levenberg-Marquardt and BFGS algorithms, but yields results of comparable quality based on the value of the objective function obtained at convergence, and the model obtained at convergence.

- Scaling has a significant effect on the performance of the LBFGS and BFGS algorithms. The scaling factors used here result in significant improvement in the convergence properties of the algorithm as compared to the no scaling case.

As our goal is to develop and implement technology to condition a reservoir description generated from a prior geostatistical model to both production data and time-lapse seismic data, we have developed procedures to integrate seismic data. In our approach, it is assumed that the change in seismic impedance will be provided as data by the geophysicists. Given a specific reservoir description and pressure and saturations distributions, the corresponding seismic impedance can be predicted by a simple rock physics model based on the Gassmann equations. Similar to production data, the square of the mismatch between observed time-lapse seismic data and corresponding calculated time-lapse seismic data for the given reservoir model and pressure/saturation distribution is included in the objective function. Thus, we have implemented the adjoint method to calculate the sensitivity of seismic data to rock property fields. We have also shown that the effect of reasonable changes in shaliness and the sand modulus and clay modulus on time-lapse seismic data is relatively small compared to the effect of saturation and pressure changes. This leads to the first of the following three specific conclusions on history matching time lapse impedance data.

- Uncertainties in mineralogy will not have a significant error on time-lapse impedance data; thus, we should be able to integrate time-lapse impedance data to reduce the uncertainty in rock property fields even when the mineralogy is not accurately known.

- The sensitivity of seismic impedance data to rock property fields can be accurately computed with the adjoint method.

- Even though the number of seismic data are large, using a simple forward model based on the Gassmann equations and the adjoint method for computing the gradient of the part of the objective function involving the mismatch term for time-lapse impedance data, it should be feasible to condition reservoir models to time-lapse seismic data.

The set of reservoir variables that can be estimated by the automatic history matching of production and time-lapse seismic data include gridblock log-permeabilities and porosities. As the variation of these variables between facies is usually much larger than the variation within a facies, it may be critical 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 generate a rich variety of textures and shapes, but the new approach should make it easier to generate approximations of the sensitivity coefficients needed to condition reservoir models to facies distributions.

Chapter 8

FUTURE WORK

The history matching process formulated here involves the minimization of an objective function O which always includes a regularization term O_m arising from a prior model and a production data misfit term O_p which in the simplest form is equal to sum of the squared difference between predicted and observed production data with each squared misfit term divided by the variance of the associated measurement error. The objective function may include a similar misfit term O_s for seismic data based on differences between observed time-lapse impedance data and time-lapse impedance data calculated for a particular set of values of reservoir variables. At this point in the project, we have developed and implemented the adjoint method for computing the sensitivity of each production datum to each reservoir variable where the reservoir variables may include gridblock horizontal permeabilities, vertical permeabilities and porosities. With these sensitivities, we have been able to implement Gauss-Newton and Levenberg-Marquardt algorithms to history match production data when the objective function minimized is given by

$$O = O_m + O_d. \tag{8.1}$$

We have also implemented an adjoint method to compute the gradient of O_d. The calculation of this gradient is one of many steps necessary to implement nonlinear preconditioned and conjugate gradient and quasi-Newton methods to minimize the objective function of 8.1. These four optimization algorithms have been implemented and compared. Based on computational efficiency, memory requirements and robustness, we have selected a scaled limited memory Broyden-Fletcher-Goldfard-Shanno (LBFGS) algorithm as the most useful for large scale history matching problems. Based on realistic history matching problems we have done, however, we expect that it may be possible to improve the efficiency of this algorithm. We have also implemented an adjoint method to compute the sensitivity of time-

lapse impedance data to gridblock permeabilities and porosities, but as yet have not done any history matching of seismic data either individually or together with production data. Moreover, the options that have actually been implemented for matching production data are restricted to estimating only two rock property fields, for example horizontal and vertical log-permeabilities with porosity fixed, or an isotropic permeability field together with porosity. Our investigation into the possibility of estimating boundaries between facies is in its early stages.

In the future, we intend to consider the following aspects.

- Implement code so that the scaled LBFGS algorithm can be used to estimate any combination of $\ln(k_x)$, $\ln(k_y)$, $\ln(k_z)$ and porosity.

- Implement and investigate a partially doubly stochastic model that the mean log-permeabilities and mean porosities may different in different geologic regions or layers. In this case, we may wish to estimate these means together with individual gridblock values of rock properties by history matching production data.

- Develop procedures to estimate the shape of relative permeability curves together with rock property fields during the history matching process.

- Attempt to improve the computational efficiency of the LBFGS algorithm by considering damping or model scaling procedures.

- Quantify the measurement and modelling error of time-lapse impedance data. This is important because an esimate of these errors is needed to time the objective function O_s which includes the seismic data misfit terms.

- Implement the LBFGS method for estimating rock property fields by history matching seismic data by themselves or seismic data and production data simultaneously.

- Develop a method to estimate the probability density function (pdf) for the model parameters which define the plurigaussian model for facies so that this pdf is consistent with cross section maps provided by geologists.

- Develop a method to determine effective gridblock permeabilities and porosities for gridblocks containing more than one facies.

- Develop a procedure to calculate the sensitivity of production data to the location of boundaries between facies so that facies boundaries can be adjusted as part of the history matching process.

162

Technical References

Y. Abacioglu, D. S. Oliver, and A. C. Reynolds. Efficient history-matching using subspace vectors. In *TUPREP Research Report 17 (May 22, 2000)*, pages 69–90, 2000.

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.

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.

O. Axelsson. *Iterative Solution Methods*. Cambridge University Press, New York, 1994.

Khalid Aziz and A. Settari. *Petroleum Reservoir Simulation*. Elsevier Applied Science Publishers, London, 1979.

Zhuoxin Bi. *Conditioning 3D Stochastic Channels to Well-Test Pressure Data*. Ph.D. thesis, University of Tulsa, Tulsa, Oklahoma, 1999.

Robert Bissell. Calculating optimal parameters for history matching. In *4th European Conference on the Mathematics of Oil Recovery*, 1994.

Robert Bissell, O. Dubrule, P. Lamy, P. Swaby, and O. Lepine. Combining goestatistical modelling with gradient information for history matching: The pilot point method, SPE 38730. In *Proceedings of the 1997 SPE Annual SPE Technical Conference and Exhibition*, pages 139–154, 1997.

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

C. G. Broyden. Quasi-Newton methods and their application of function minimization. *Maths. Comp.*, 21:368–381, 1967.

C. G. Broyden. The convergence of a class of double rank minimization algorithm parts i and ii. *J. of Institute of Mathematics and its Applications*, 6:76–90 and 222–231, 1970.

A.G. Buckley. A combined conjugate-gradient quasi-Newton minimization algorithm. *Mathematical Programming*, 15:200–210, 1978.

A.G. Buckley and A. Lenir. QN-like variable storage conjugate gradients. *Mathematical Programming*, 27:155–175, 1983.

R. D. Carter, L. F. Kemp, and A. C. Pierce. Discussion of comparison of sensitivity coefficient calculation methods in automatic history matching. *Soc. Petrol. Eng. J.*, pages 205–208, 1982.

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

Kevin T. Chambers. 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.

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.

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.

G. de Marsily, G. Lavedan, M. Boucher, and G. Fasanino. Interpretation of interference tests in a well field using geostatistical techniques to fit the permeability distribution in a reservoir model. In *Geostatistics for Natural Resources Characterization, Part 2*, pages 831–849. D. Reidel, 1984.

R. Fletcher. A new approach to variable metric algorithms. *Computer Journal*, 13:317–322, 1970.

R. Fletcher and M. J. D. Powell. A rapidly convergent descent method for minimization. *Computer Journal*, 6:163–168, 1963.

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.

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.

F. Gassmann. Elastic waves through a packing of spheres. *Geophysics*, 16:673–685, 1951.

G. R. Gavalas, P. C. Shah, and John H. Seinfeld. Reservoir history matching by Bayesian estimation. *Soc. Petrol. Eng. J.*, 16(6):337–350, 1976.

D. Goldfarb. A family of variable-metric methods derived by variational means. *Maths. Comp.*, 24:23–26, 1970.

A. A. Goldstein. On steepest descent. *SIAM J. control*, 3(1):147–151, 1965.

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.

Gilles Guerin. *Simulation of Production and Elastic Properties of Reservoirs to Validate Time-Lapse Seismic*. PhD thesis, Columbia University, 2000.

D. Han, A.Nur, and D.Morgan. Effects of porosity and clay content on wave velocities in sandstones. *Geophysics*, 51(11):2093–2107, 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.

Nanqun He, Albert C. Reynolds, and Dean S. Oliver. Three-dimensional reservoir description from multiwell pressure data and prior information (SPE-36509). In *1996 SPE Annual Technical Conference and Exhibition*, 1996.

M. R. Hestenes and E. S. Stiefel. Methods of conjugate gradients for solving linear systems. *J. Res. Nat. Bur. Stand*, 46:409–536, 1952.

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. Théorie de l'interprétation des mesures de pression. *Revue de L'Institut Français du Pétrole*, 19(3):297–334, 1964.

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.

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.

J. E. Killough, Yogeshwar Sharma, Alain Dupuy, Robert Bissell, and John Wallis. A multiple right hand side iterative solver for history matching SPE 29119. In *Proceedings of the 13th SPE Symposium on Reservoir Simulation*, pages 249–255, 1995.

Peter K. Kitanidis. Quasi-linear geostatistical theory for inversing. *Water Resour. Res.*, 31 (10):2411–2419, 1995.

Tamara K. Kolda, Dianne P. O'Leary, and Larry Nazareth. BFGS with update skipping and varying memory. *SIAM J. Optim*, 8(4):1060–1083, 1998.

G.T. Kuster and M.N.Toksöz. Velocity and attenuation of seismic waves in two-phase media: Part 1. theoretical formulations. *Geophysics*, 39:587–606, 1974.

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.

M. Landro. Discrimination between pressure and fluid saturation changes from time-lapse seismic data. *Geophysics*, 66(3):836–844, 2001.

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.

T. Y. Lee and John H. Seinfeld. Estimation of two-phase petroleum reservoir properties by regularization. *J. Computational Physics*, 69:397–419, 1987a.

Tai-Yong Lee and John H. Seinfeld. Estimation of absolute and relative permeabilities in petroleum reservoirs. *Inverse Problems*, 3(4):711–728, 1987b.

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*, 2001.

D. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. *Math. Programming*, 45:503–528, 1989.

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.

M.A. Meadows. Enhancements to landro's method for separating time-lapse pressure and saturation changes. *San Antonio 2001 Expanded Abstracts*, 2001.

W. Murphy, A.Reischer, and K.Hsu. Modulus decomposition of compressional and shear velocities in sand bodies. *Geophysics*, 58(2):227–239, 1993.

Stephen G. Nash and Jorge Nocedal. A numerical study of the limited memory BFGS method and the truncated-Newton method for large scale optimization. *SIAM. J. OPTM.*, 1(3): 358–372, 1991.

L. Nazareth. A relationship between the BFGS and conjugate gradient algorithms and its implications for new algorithms. *SIAM. J. NUMER. ANAL.*, 16(5):794–800, 1979.

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

R.C. Nolen-Hoeksema. Modulus-porosity relations, gassmann's equations, and the low-frequency elastic-wave response to fluids. *Geophysics*, 65(5):1355–1363, 2000.

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.

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 algorithm without line-search for unconstrained minimization. *Mathematics of Computation*, 27:873–885, 1973.

S. S. Oren. On the selection of parameters in self-scaling variable metric algorithms. *Mathematical Programming*, 7:351–367, 1974a.

S. S. Oren. Self-scaling variable metric (SSVM) algorithms II: Implementation and experiments. *Management Science*, 20:863–874, 1974b.

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.

Robert L. Parker. *Geophysical Inverse Theory*. Princeton University Press, Princeton, New Jersey, 1994.

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.

E. Polak. *Computational methods in optimization: a unified approach*. Academic Press, London, 1971.

M.J.D. Powell. Restart procedures for the conjugate gradient method. *Mathematical Programming*, 12:241–254, 1977.

R. Ramammorthy, W.F.Murphy, and C.Coll. Total porosity estimation in shaly sands from shear modulus. *SPWLA 36th Annual Logging Symposium Transactions*, 1995.

Banda S. RamaRao, A. Marsh LaVenue, Ghislain de Marsily, and Melvin G. Marietta. Pilot point methodology for automated calibration of an ensemble of conditionally simulated transmissivity fields, 1. Theory and computational experiments. *Water Resour. Res.*, 31 (3):475–493, 1995.

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.

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. Conjugate gradient methods with inexact searches. *Mathematics of Operation Research*, 3:244–256, 1978a.

D. F. Shanno. On the convergence of a new conjugate gradient algorithm. *SIAM Journal on Numerical Analysis*, 15:1247–1257, 1978b.

D. F. Shanno and Kang-Hoh Phua. Matrix conditioning and nonlinear optimization. *Mathematical Programming*, 14:149–160, 1978.

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

Jan Arild Skjervheim. *Automatic history matching.* M.S. thesis, University of Tulsa and Norweigian University of Science and Technology, Tulsa, Oklahoma, 2002.

Thomas B. Tan. A computationally efficient gauss-newton method for automatic history matching, SPE-29100. *Proceedings of the 13th SPE Symposium on Reservoir Simulation*, pages 61–70, 1995.

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

A. N. Tikhonov. Regularization of incorrectly posed problems. *Soviet Math. Dokl.*, 4:1624–1627, 1963.

A. Tura and D.Lumley. Estimating pressure and saturation changes from time-lapse avo data. *Extended Abstracts, SEG Annual Meeting*, 1999.

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.

M. L. Wasserman, A. S. Emanuel, and J. H. Seinfeld. Practical applications of optimal-control theory to history-matching multiphase simulator models. *Soc. Petrol. Eng. J.*, 15 (4):347–355, 1975.

P. Wolfe. Convergence conditions for ascent methods. *SIAM Rev.*, 11(2):226–235, 1969.

Zhan Wu. *Conditioning Geostatistical Models to Two-Phase Flow Production Data*. PhD thesis, University of Tulsa, 1999.

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.

William W-G Yeh. Review of parameter identification in groundwater hydrology: The inverse problem. *Water Resour. Res.*, 22(2):95–108, 1986.

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.

Fengjun Zhang. *Automatic History Matching of Production Data for Large Scale Problems*. Ph.D. thesis, University of Tulsa, Tulsa, Oklahoma, 2002.