AUTOMATIC HISTORY MATCHING IN BAYESIAN FRAMEWORK

FOR FIELD-SCALE APPLICATIONS

A Dissertation

by

AHMED MOHAMED IBRAHIM DAOUD

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

December 2004

Major Subject: Petroleum Engineering

AUTOMATIC HISTORY MATCHING IN BAYESIAN FRAMEWORK

FOR FIELD-SCALE APPLICATIONS

A Dissertation

by

AHMED MOHAMED IBRAHIM DAOUD

Submitted to Texas A&M University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Approved as to style and content by:

Akhil Datta-Gupta	Daulat D. Mamora
(Chair of Committee)	(Member)
Duane A. McVay	Yalchin R. Efendiev
(Member)	(Member)
Steve A. Holditch	
(Head of Department)	

December 2004

Major Subject: Petroleum Engineering

ABSTRACT

Automatic History Matching in Bayesian Framework for Field-Scale Applications.

(December 2004)

Ahmed Mohamed Ibrahim Daoud, B.S., Cairo University, Egypt;

M.S., Cairo University, Egypt

Chair of Advisory Committee: Dr. Akhil Datta-Gupta

Conditioning geologic models to production data and assessment of uncertainty is generally done in a Bayesian framework. The current Bayesian approach suffers from three major limitations that make it impractical for field-scale applications. These are: first, the CPU time scaling behavior of the Bayesian inverse problem using the modified Gauss-Newton algorithm with full covariance as regularization behaves quadratically with increasing model size; second, the sensitivity calculation using finite difference as the forward model depends upon the number of model parameters or the number of data points; and third, the high CPU time and memory required for covariance matrix calculation. Different attempts were used to alleviate the third limitation by using analytically-derived stencil, but these are limited to the exponential models only.

We propose a fast and robust adaptation of the Bayesian formulation for inverse modeling that overcomes many of the current limitations. First, we use a commercial finite difference simulator, ECLIPSE, as a forward model, which is general and can account for complex physical behavior that dominates most field applications. Second, the production data misfit is represented by a single generalized travel time misfit per well, thus effectively reducing the number of data points into one per well and ensuring the matching of the entire production history. Third, we use both the adjoint method and streamline-based sensitivity method for sensitivity calculations. The adjoint method depends on the number of wells integrated, and generally is of an order of magnitude less than the number of data points or the model parameters. The streamline method is more efficient and faster as it requires only one simulation run per iteration regardless of

the number of model parameters or the data points. Fourth, for solving the inverse problem, we utilize an iterative sparse matrix solver, LSQR, along with an approximation of the square root of the inverse of the covariance calculated using a numerically-derived stencil, which is broadly applicable to a wide class of covariance models.

Our proposed approach is computationally efficient and, more importantly, the CPU time scales linearly with respect to model size. This makes automatic history matching and uncertainty assessment using a Bayesian framework more feasible for large-scale applications. We demonstrate the power and utility of our approach using synthetic cases and a field example. The field example is from Goldsmith San Andres Unit in West Texas, where we matched 20 years of production history and generated multiple realizations using the Randomized Maximum Likelihood method for uncertainty assessment. Both the adjoint method and the streamline-based sensitivity method are used to illustrate the broad applicability of our approach.

DEDICATION

To my dear wife, to the soul of my mother, to my beloved father, to my brother, and to everyone who taught me a letter in my life.

ACKNOWLEDGMENTS

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

FIGURE   Page

CHAPTER I
INTRODUCTION

1.1   Introduction

Automatic history matching is to find the best model parameters that minimize the error between the observed data and that calculated from the model without significant manual intervention. However, to do that there will be three major questions that we should answer.[1] The first question is how accurate is the observed data that we need to match, the second one is how accurate is the forward model response, in other words, does the forward model include all the physics of the problem or not. The last one is that we can find an infinite number of model parameters that can fit the observed data, so which one to select, in other words, what is the prior information about the model parameters that are independent of the data observed. Here comes the important of the statistics to quantify the uncertainties in both the model and the data and the importance of Bayes' theory (1763) to get the model that honor both the prior information about the model parameter and at the same time honor the data.

In petroleum reservoir engineering the main objective is to build a reservoir model that honors both the geological information and the production history of the reservoir. The conventional method to do so is the manual history matching process which involves changing some parameters manually to get a good match with the production history. The selection of this parameter is subjective and always depends upon the engineering sense. This process is very tedious and time consuming; the alternative is using the automatic history-matching concept. The concept of the automatic history matching under Bayes theory is based on selecting a Gaussian probability distribution for the prior model parameter and a Gaussian distribution of the error in the data required to being matched.

The dissertation follows the style and format of *SPE Journal*.

Then using Bayes theory, we get the posterior distribution that honors both the uncertainty of the prior and the data error.

The model parameter can be permeability or porosity or skin factor or relative permeability or all of them and the data to be matched might be gas oil ratio, water cut, bottom-hole pressure or rate or all of them. The model parameter that maximizes the posterior distribution is the model parameter that honors both the data and the prior information. The framework of this study is to get the best permeability realization that honors both the prior information about the permeability distribution in the reservoir and the production history using Bayesian formulation.

1.2   Literature Review

Conditioning geological models to production data typically requires the solution of an inverse problem. Such inverse problems are usually ill-posed and their solutions suffer from difficulties in existence, uniqueness, and stability. To remedy these problems, a regularization term, in the form of data-independent prior information is generally added to the objective function in the inverse problem. Two different approaches to incorporate the regularization term have been used extensively in reservoir characterization literatures. One of these approaches is the Bayesian[2-8], and the other is the deterministic.[9-12] Both approaches have been successfully applied for conditioning geological models to production history and comparison between the two approaches can be found in the literature.[13,14] Unlike the deterministic approach, the Bayesian approach associates probability distribution to the prior models and is thus considered well-suited for post-data inference and uncertainty assessment by defining a posterior distribution of models and sampling multiple realizations from this distribution. That is why Bayesian approach is commonly used for uncertainty assessment during production forecasting.

Typically any inverse problem requires an optimization algorithm. These optimization algorithms can be classified into gradient-based and gradient-free algorithm. The gradient-free algorithms like simulated annealing or genetic algorithms

are not competitive with the gradient-based algorithm and computationally prohibitive for large-scale field applications. For the gradient-based algorithms, it is classified according to its search direction[15] into steepest descent, Newton, quasi-Newton, and conjugate gradient. The fastest among those are the Newton-type search as it has a quadratic rate of convergence in the vicinity of the solution compared to the quasi-Newton method which has a super linear rate of convergence, conjugate gradient and steepest descent which have linear rate of convergence and can be very slow in difficult problems. As Gauss-Newton and Levenberg-Marquard[15] are considered one of the Newton-type of search algorithms, they are commonly used to get the maximum a posteriori estimate (MAP) from the posterior distribution by knowing the **sensitivity coefficients** which measure the change in the production response due to the change in the model parameters. As calculation of sensitivity is considered the key step in conditioning geologic model to production data, most of the literature survey work is focusing on the different approaches used for sensitivity computation.

Jacquard and Jain[16] presented an analytical formula for the sensitivity of pressure to a small perturbation in a uniform permeability field which depends on the number of wells and required about $1+N_w$ (N_w is the number of wells) simulation runs. Their method based on the assumptions of reciprocity theory and the linear relationship between the model parameters and the production response. Jacquard and Jain[16] constructed estimates of two-dimensional permeability fields by history–matching single-phase flow pressure data by minimizing an objective function equal to the sum of squared pressure data mismatch terms. They used less than twenty parameters and they used an optimization procedure similar in spirit to the Levenberg-Marquardt algorithm to avoid the numerical instabilities that can arise when solving an ill-conditioned inverse problem. Jahns[17] adapted the basic ideas of Jacquard and Jain[16] to estimate both transmissibility and storage fields by matching single-phase flow pressure data. Jahns[17] used the perturbation method which requires $M+1$ (M is the number of the model parameters) simulation runs to construct the sensitivities required by Gauss-Newton optimization algorithm and he shows that his method is computationally efficient

compared to Jacquard and Jain's when the number of parameters is less than twice the number of wells. Jahns[17] during inversion applies a sequence of minimization starting from small number of parameters to a bigger one (coarse to fine scale inversion).

After about ten years of the fundamental work discussed in the previous paragraph, Carter et al.[18] published a mathematical derivation of an efficient method to calculate sensitivity coefficients based on Jacquard and Jain's[16] method that require $1+N_w$ simulation runs for sensitivity computation. Carter et al.[18] considered 2D single-phase flow problems with the same assumptions of Jacquard and Jain[16]. Their procedure gives the sensitivity of reservoir simulator grid block pressures to grid block transmissibility and storage. He et al.[19] extended Carter et al.[18] method for 3D single phase flow by estimating the sensitivity of the wellbore pressure with respect to the model parameter by using Peacman[20] equation and a straightforward extension of Carter et al. sensitivities of grid block pressure with respect to the model parameters. The extension of Carter el al.[18] method for 3D single phase reservoir models require a number of simulation runs equivalent to the number of the grid blocks penetrated by the well. However He et al.[19] developed an approximate procedure to estimate the sensitivity of the wellbore pressure with respect to the model parameters that still depends only on the number of wells regardless the number of the penetrated grid blocks by the wells. He at al. method assumed small vertical pressure gradient and is limited to single phase flow problems. The methods for the computation of sensitivity discussed to this point are applicable only for single phase flow problems.

For multi-phase flow, the easiest and the least efficient is the perturbation and sometimes called finite difference method, where the sensitivities are computed by perturbing the value of the model parameter and estimating the change in the production response due to this perturbation. This method has been used by Jans[17] and its difficulties rest on the fact that its accuracy is sensitive towards the selection of the amount of perturbation and it requires M+1 simulation runs, where M is the number of model parameters, which makes it impractical for field-scale applications with thousands of model parameters. A more efficient method for computation of sensitivity coefficients

is the method discussed in Yeh's[21] review of parameter identification methods in the hydrology literature under the name of "sensitivity equation method" and then later introduced to the petroleum literature by Anterion et al.[22] under the name of "gradient simulator method" or "direct method". In this method the sensitivity equation is derived by differentiating the flow equations with respect to a single model parameter to obtain a linear system of equation per each parameter. The solution of this system of equation will give the gradient of the primary variables (pressure and saturation) with respect to the specified model parameter. The right hand side of this system of equation can be extracted directly from the Jacobian matrix at the last Newton iteration during solving for the primary variables using fully implicit method. The left hand side of these linear equations consists of two parts; the first part can also be extracted directly from the first Newton iteration while the second part can be obtained numerically or analytically. In this method the left hand side of the equations is constant, only the right hand side of the equations should be changed with the change of the model parameter. After solving this system of equations M times (M is the number of model parameters) with multiple right hand sides, the sensitivity of the primary variables with respect to the model parameters at each grid block is obtained. This is considered as redundant information especially as one requires only the sensitivity of the primary variables at the grid block penetrated by the wells; hence this method is considered unattractive for large number of model parameters. Tang[23] and Tang et al.[24] introduced the GPST (Generalized Pulse Spectrum Technique) for computing the sensitivity of the primary variables (pressure and saturation) with respect to the model parameters. This method requires only two simulation runs and uses only the information of sensitivities at the well locations; hence it is computationally efficient. Tang[23] and Tang et al.[24] used Tikhonov regularization during parameters updating to alleviate the ill-posed problem and they tested their method for 2D single and two phase flow using permeability and porosity as the model parameters and the results reflect its good accuracy and high rate of convergence for inverting for permeability; however, it does not show good accuracy for porosity inversion. Also, the computation time for the GPST gives quadratic scaling with the

model size; hence it is impractical for field-scale applications. Chu et al.[25] used the basic idea of Tang et al.[24] and they develop the MGPST (Modified Generalized Pulse Spectrum Technique) to get the sensitivity of wellbore pressure to reservoir simulator grid block permeability and porosity.

Chen et al.[26] and Chavent et al.[27] introduced the optimal control theory to estimate the gradient of the objective function with respect to the control variables (rock properties like porosity and permeability). This method is called optimal control or adjoint method and it only requires one simulation run in addition to the solution of linear system of adjoint equations, hence the sensitivity computation require time equivalent to two simulation runs. Chen et al.[26] used both steepest descent and conjugate gradient while Chavent et al.[27] used steepest descent as optimization algorithm for updating the control variables. However, those optimization methods suffer from slow convergence compared to the sensitivity-based optimization algorithm like Gauss-Newton or Levenberg-Marquardt which have a quadratic convergence behavior.[15] Both Chen et al.[26] and Chavent et al.[27] show the derivation of the adjoint system of equations and the gradient of the objective function using the flow equation in continuous form. Wasserman et al.[28] apply the optimal control theory using the flow equation in semi-continuous form by discretizing the right hand side of the flow equation leaving the left hand side in a continuous form. This is equivalent to converting the partial differential equation of flow in space and time into a set of ordinary differential equations in time which is more practical for field applications and they applied their method for three phase 2D field cases. Watson et al.[29] show the application of optimal control theory using the flow equation in a complete discretized form and apply this method for 2D multi phase flow problems. All the applications of the optimal control theory up to this point suffer from one drawback which is the slow convergence as the method gives only the gradient of the objective function which force to the application of slow convergence optimization algorithms like steepest descent or conjugate gradient. However, the method is computationally efficient for the calculation of the gradient of the objective function as it require only two simulation runs and it is independent of the number of the

model parameters or the number of data or the number of wells. Wu et al.[5] used the optimal control theory (adjoint method) to calculate the sensitivity of wellbore pressure and the water–oil ratio with respect to the grid block permeabilities and porosities for 2D two phase flow. This method requires one simulation run and a solution of a linear system of adjoint equations of N_d times (N_d is the number of data points). Li et al.[6] used the optimal control theory to estimate the sensitivity coefficient of wellbore pressure, water-oil ratio, and gas-oil ratio to grid block permeability, porosity, well skin factor, and the parameters used to define the power law relative permeability curves for three dimension, three phase reservoir models. Zhang et al.[7] used the same adjoint method to estimate the gradient of the objective function instead of the sensitivities especially for large scale problems where both the number of model parameters and the number of data are large. The difference between Zhang et al.[7] work and the other works[26-29] is that they used quasi-Newton optimization algorithm called LBFGS (Limited Memory Broyden-Fletcher-Goldfarb-Shano) to minimize the objective function instead of the conjugate gradient or steepest descent that were used before. However, their method still lacks the quadratic convergence of Gauss-Newton and Marquardt-Levenberg optimization algorithms.[15] Wu and Datta-Gupta[8] used the approach of generalized travel time introduced by Luo and Schuster[30] in the context of waveform inversion in seismology which reduces the data misfit into one point per well. This offer a significant advantage in reducing the computation burden of sensitivity using the adjoint method where it requires only one simulation run and solving the adjoint system of equations N_w times (N_w is the number of wells), hence, make the approach well-suited for large-scale field applications. However their method has been applied only for 2D two phase flow.

Up to this point, all the previous discussions for sensitivity calculations are for finite difference models. Recently, streamline models show significant advantage in automatic history matching due to two main reasons; its high speed performance compared to finite difference models[31,32] and the most important is its analytically-derived sensitivities using one single simulation run.[9,10] Vasco et al.[10] presented semi-analytical approach for sensitivity calculation of water cut and tracer response with respect to permeability and

porosity with a total number of 100,000 parameters. They used a two-step inversion approach by matching first the breakthrough time and then the amplitude (water cut or tracer response). Their method based on the assumption that the streamline does not shift; however, for drastic change of flow geometry such as changing well conditions, infill drilling, pattern conversion the stationary streamline assumption is not valid. He et al.[11] modified the sensitivity calculation for changing field conditions and used the approach of generalized travel time inversion introduced by Luo and Schuster.[30] Due to the limitation of streamline models to account for compressible flow and complex physical mechanism, an approach that combine the advantage of streamline models in sensitivity calculations and at the same time using finite difference models has been introduced[33,12] where the velocity field is obtained from the finite difference models which used to trace the streamline and calculate the time of flight required for sensitivity calculation.

1.3 Objectives

The main objective of this study is to use Bayes theory as a tool for integrating static and dynamic data and make it well-suited for field-scale applications and at the same time use the advantage of Bayes theory as a tool for uncertainty assessment. Accordingly, the specific objectives are as follows:

- Generalize the application of the "generalized travel time" inversion to 3D two phase finite difference models by developing the sensitivity of the generalized travel time with respect to the model parameters using the adjoint method.
- Introduce a new Bayesian formulation that is well-suited for field-scale applications with a numerically-derived stencil to compute the square root of the inverse of the covariance matrix required by the new formulation.
- Apply the new and the conventional Bayesian formulation for integrating static and dynamic data from finite difference models using adjoint method-based sensitivity and streamline-based sensitivity.

- Study the computational efficiency with increasing the number of parameters for both the conventional and the new Bayesian formulation using finite difference models with sensitivity calculated from adjoint method and streamline.

- Quantify the uncertainty of the estimate by sampling the posterior distribution resulted from the integration of static and dynamic data using Bayes theory.

- Applications: different synthetic examples used to study the quality of the inversion and the scaling behavior for the new Bayesian compared to the conventional approach using sensitivity from both adjoint method and streamline method. Field example from Goldsmith San Andreas Unit is used to demonstrate the utility of the new Bayesian formulation using finite difference models with sensitivity calculated from streamline and adjoint method and also to assess the uncertainty during the production history matching.

1.4   Dissertation Outline

Chapter II gives a brief introduction about Bayes theory and its application as a tool for integrating static and dynamic data. It is well known that history matching of production (dynamic data) always yield non unique solution especially if the model parameters like permeability and porosity are greater than the data points. Moreover, the history matching problem of this kind is an ill-posed problem and some form of regularization is necessary to avoid instability. Thus incorporating the prior data in the history matching problem using Bayes theory stabilizes the solution and reduce the variability in the reservoir model parameters that provide an acceptable match to the production data. So, the first part of this chapter discusses the three important components of automatic history matching, which are the data misfit, the prior or the regularization, and the optimization algorithms. Those three components are very important and any improvement in one of those components leads to a significant contribution in the automatic history matching process. The second part of this chapter introduces one of the major parts of this dissertation, which is the reformulation of the Bayesian approach for field-scale applications with a special reference to a novel approach used to

approximate the square root of the inverse of the prior covariance matrix required by the new formulation. In addition we examine the scaling of the computational cost required by the conventional and new formulation with increasing the number of parameters. The third part of this chapter is devoted to the uncertainty assessment by briefly discussing the different methods to sample realizations from the posterior distribution for using in the uncertainty analysis.

Chapter III represents the central part of this dissertation which is the sensitivity computation. The first part of this chapter gives a detailed formulation of the sensitivity of the generalized travel time with respect to the model parameter using adjoint method for 3D, two phase reservoir models. The second part shows briefly the sensitivity computation based on streamline for comparison purpose. The forward model used for the both parts is the finite difference models.

Chapter IV presents applications of the procedures discussed in the previous three chapters using 2D and 3D synthetic examples to show the quality of the inversion and the computational time behavior versus the number of parameters for the new Bayesian approach compared to the conventional method using sensitivity from both adjoint method and streamline method. Also, a field example from the Goldsmith San Andreas Unit is used to demonstrate the utility of the new Bayesian formulation using finite difference models with sensitivity calculated from both streamline and adjoint method. We also generate different realizations from the posterior distribution for the uncertainty assessment in production history before and after incorporating production data.

Chapter V summarizes the conclusions and recommendations from this study.

There are three appendices in this dissertation. Appendix A shows the derivation of the Modified Gauss-Newton used in this work and the mathematical equivalence between the new Bayesian formulation and the conventional Gauss-Newton formulation. Appendix B describes the computation of the derivatives in the adjoint system of equations for the adjoint method-based sensitivity. Appendix C represents the computation of the derivatives in the sensitivity coefficient calculation using the adjoint method-based sensitivity.

CHAPTER II
BAYES THEORY AS A TOOL FOR DATA INTEGRATION AND UNCERTAINTY ASSESSMENT

This chapter gives brief introduction about Bayes theory as a tool for integrating static and dynamic data. Due to the ill-posedness of the history matching problems, a regularization term is required to remedy this problem and Bayes theory is considered an appropriate statistical tool that can incorporate the static and dynamic data where the static data serves as a regularization term during the history matching. This chapter consists of three parts; the first part of this chapter discusses the three important components of automatic history matching in Bayesian framework, which are the data misfit, the prior or the regularization, and the optimization. The second part of this chapter introduces the reformulation of the Bayesian approach which is well-suited for field-scale application. A novel approach is used to approximate the square root of the inverse of the prior covariance matrix required by the new formulation. Finally, we examine the computational work required in terms of number of multiplications by the conventional and the new formulation with increasing the number of parameters. The third part of this chapter is devoted to briefly discussing the methods used to sample different realizations from the posterior distribution for using them in the uncertainty analysis.

2.1 Bayes Theory Background

Bayes' rule is given as follows:

$$f(\mathbf{m}/\mathbf{d}_{obs}) = f(\mathbf{m}) \cdot \frac{f(\mathbf{d}_{obs}/\mathbf{m})}{f(\mathbf{d}_{obs})} \qquad \text{............(2.1)}$$

Where, $\mathbf{m}$ is a vector of the model parameters, $\mathbf{d}_{obs}$ is a vector of the observed data,

f(**m**) is the prior probability distribution function of the model parameters, f(**d**$_{obs}$/**m**) is the probability distribution of the observed data given the true model parameters is **m,** f(**d**$_{obs}$) is the marginal probability distribution, which is given as $f(\mathbf{d}_{obs}) = \int d\mathbf{m}\; f(\mathbf{d}_{obs}/\mathbf{m}) \cdot f(\mathbf{m})$, f(**m**/**d**$_{obs}$) is the posterior probability distribution of the model parameters given the observed data.

Assuming that the error in the data and the forward model and the uncertainty in the model parameters follow a multi-Gaussian distribution, the posterior probability distribution can be written as[34]:

$$f(\mathbf{m}/\mathbf{d}_{obs}) = const\; \mathbf{exp}\left\{ \frac{-1}{2}\left[\begin{array}{c} \left(g(\mathbf{m}) - \mathbf{d}_{obs}\right)^{T} C_{D}^{-1}\left(g(\mathbf{m}) - \mathbf{d}_{obs}\right) \\ + \left(\mathbf{m} - \mathbf{m}_{prior}\right)^{T} C_{M}^{-1}\left(\mathbf{m} - \mathbf{m}_{prior}\right) \end{array} \right] \right\} \qquad ..……….(2.2)$$

Where, g(**m**) is the forward model that defines the non-linear relationship between the model parameters and the data, C$_D$ is the covariance operator combines both the error in the data and the forward model. If the errors in the data and the forward model are uncorrelated, C$_D$ will be a diagonal matrix. Also, C$_M$ is the covariance operator describing the estimated uncertainties in the prior model, **m**$_{prior}$ is a vector of the prior mean of the model parameter **m**.

Bayes' rule provide a natural framework for automatic history matching of reservoir models by combining the prior geologic model with the production data, where g(**m**) is the reservoir simulation model, which can be finite difference or streamline models, the **d**$_{obs}$ are the production data, and **m** and **m**$_{prior}$ can be the permeability or porosity or the skin factor or the relative permeability control parameters or any other reservoir control parameters assigned by a reservoir engineer.

Eq. 2.2 gives the posterior distribution of the model parameter, where any sample from this distribution can be a plausible model consistent with the production history and the prior geological model. The best estimate of the parameter is the one that maximize the posterior distribution given by **Eq. 2.2,** or minimizing the following objective function:

$$o(\mathbf{m}) = \frac{1}{2}\left[(g(\mathbf{m}) - \mathbf{d}_{obs})^T C_D^{-1}(g(\mathbf{m}) - \mathbf{d}_{obs}) + (\mathbf{m} - \mathbf{m}_{prior})^T C_M^{-1}(\mathbf{m} - \mathbf{m}_{prior})\right] \qquad \text{.…………(2.3)}$$

According to **Eq. 2.3**, the three important components for the automatic history matching process are the first term of **Eq. 2.3**, which is called "**data misfit**", the second component is the second term of **Eq. 2.3**, which is called "**regularization**" or the "**prior model**" and the third component is the **optimization algorithm** used to minimize the objective function given by **Eq. 2.3**. The next section is devoted to discussing those three components.

2.2   Data Misfit

This section discusses three different types of data misfit and the formulation of one of those types that is used during this work which is the "generalized travel time misfit".

2.2.1   Types of Data Misfit

Three different ways used to represent the production data misfit, namely, the "**amplitude misfit**", "**travel time misfit**", and "**generalized travel time misfit**". The most common one is the "**amplitude misfit**" (**Fig. 2.1a**), where the production data misfit is represented as follows

$$J_p = \sum_{j=1}^{n_w} \sum_{i=1}^{n_{dj}} w_{ij}\left(y_j^{cal}(t_i) - y_j^{obs}(t_i)\right)^2 \qquad \text{………..(2.4)}$$

For $i = 1,...,n_{dj}$, $j = 1,...,n_w$

In **Eq. 2.4**, $y_j(t_i)$ denotes the production data for well j at time t_i, n_w and n_{dj} stand for the number of production wells and the number of observed data at each well, respectively and w_{ij} represents the data weights, which are the reciprocal of the variance at each data point under the assumption that the C_D is diagonal matrix, where the errors between the data points are uncorrelated.

The "**travel time misfit**" is represented as shown in **Fig. 2.1b,** where the misfit is obtained by lining up the observed and the predicted data at a reference time such as the breakthrough or the first arrival time. The disadvantage of travel time misfit is that it is only one time point match and it does not take into consideration all the points as in the amplitude misfit. However, the travel time misfit has major advantages compared to the amplitude during inversion, first it has a quasi-linear properties compared to the high non linearity of the amplitude[35] as a result the travel time inversion is robust and converge rapidly even if the initial model is far away from the solution. Second, it is computationally efficient because the number of travel-time is equal to the number of wells, regardless of the number of data points. This leads to considerable savings in computational time during the minimization.

(a)   (b)
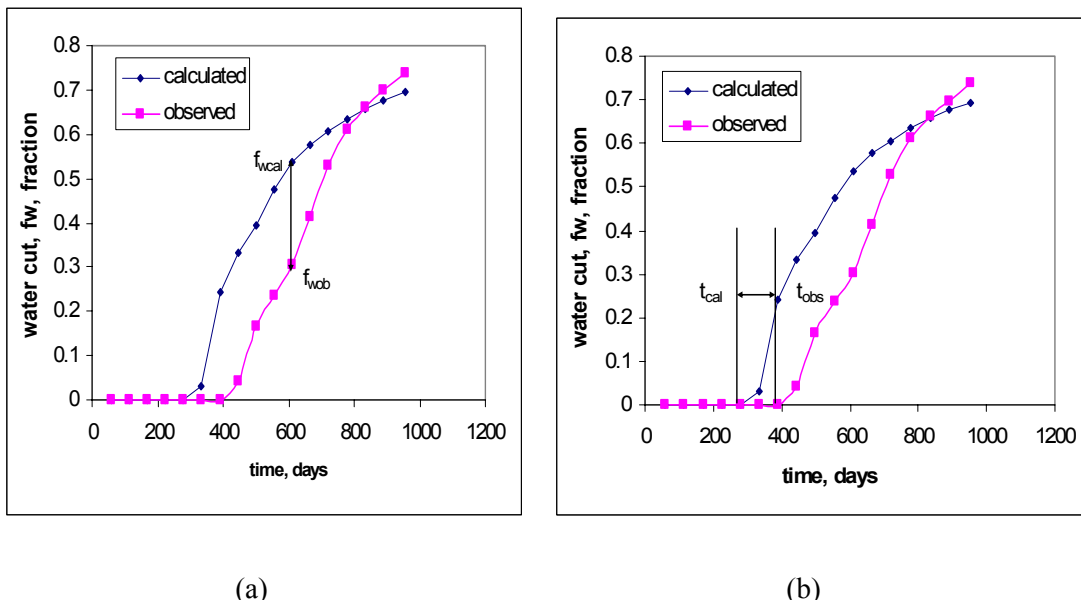
Fig. 2.1−Illustration of different types of data misfit, (a) Amplitude misfit, (b) Travel time misfit, (c) & (d) Generalized travel time misfit
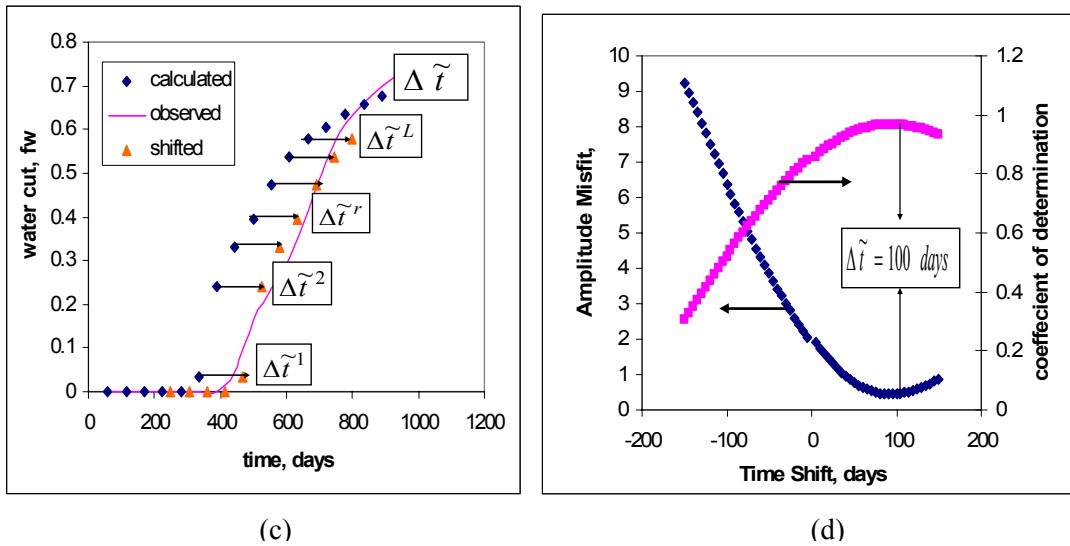
Fig. 2.1−Continued

The "**generalized travel-time misfit**" ensures matching of the entire production history rather than single time point match and at the same time retaining most of the desirable properties of travel-time inversion. In "**generalized travel time match**" we seek an optimal time-shift at each well to minimize the production data misfit at the well. This is illustrated in **Fig. 2.1c** where the calculated water-cut response is systematically shifted in small time increments towards the observed response, and the data misfit is computed for each time increment. Taking well j as an example, the optimal shift will be given by the Δt_j that minimizes the misfit function,

$$J_{pj} = \sum_{i=1}^{n_{dj}} w_{ij} \left[y_j^{obs} \left(t_i + \Delta t_j \right) - y_j^{cal} \left(t_i \right) \right]^2 = f(\Delta t_j) \qquad \text{…..……..(2.5)}$$

Or, alternatively maximizes the coefficient of determination given by:

$$R^2(\Delta t_j) = 1 - \frac{\sum_{i=1}^{n_{dj}} \left[y_j^{obs}(t_i + \Delta t_j) - y_j^{cal}(t_i) \right]^2}{\sum_{i=1}^{n_{dj}} \left[y_j^{obs}(t_i) - \overline{y_j^{obs}} \right]^2} \qquad \text{…..……..(2.6)}$$

Thus, the generalized travel-time at well j is the 'optimal' time-shift $\Delta \tilde{t}_j$ that maximizes the $R^2(\Delta t_j)$ or minimizes J_{pj} as shown in **Fig. 2.1d.** It is important to point out that the computation of the optimal time-shift does not require any additional flow simulations. It is carried out as a post-processing at each well after the calculated production response is derived using a flow simulation. The overall production data misfit can now be expressed in terms of a generalized travel-time misfit at all wells as follows

$$J_{\Delta\tilde{t}} = \sum_{j=1}^{n_w} \frac{\left(\Delta \tilde{t}_j\right)^2}{\sigma_j^2} \qquad\qquad\qquad \text{…..…….(2.7)}$$

Where, σ_j^2 is the error variance of the generalized travel time at well j by assuming C_D a diagonal matrix. It is worth to mention here that using the data misfit in the objective function given by **Eq. 2.3** as the generalized travel time reduces the computational burden during the minimization by reducing the data covariance matrix to be of order $N_w \times N_w$ and the data misfit vector to be of $N_w \times 1$ which are always of order of magnitude lower than the number of data points, N_d. Thus, the concept of generalized travel time shift as the data misfit is well-suited for field-scale application and is used during this study. Accordingly, the objective function given by **Eq. 2.3** using the generalized travel time as the data misfit, will be as follows:

$$O(\mathbf{m}) = \frac{1}{2}\left[\Delta\tilde{\mathbf{t}}^T C_D^{-1}\Delta\tilde{\mathbf{t}} + \left(\mathbf{m} - \mathbf{m}_{prior}\right)^T C_M^{-1}\left(\mathbf{m} - \mathbf{m}_{prior}\right)\right] \qquad \text{……..…….(2.8)}$$

Where $\Delta\tilde{t}$ is the generalized travel time that minimizes the difference between the calculated and the observed data as given by **Eq. 2.3.** A detailed formulation of the generalized travel time shift under different scenarios will be studied in the next section.

It is important to mention that the selection of the standard deviation of the data error is subjective and it depends upon the data itself. However a good guideline for selecting this parameter is given by Wu et al.[5] and Wu[36].

2.2.2 Generalized Travel Time Formulation

In this section, a formulation of a general formula for the generalized travel time with respect to the travel time for two cases is given. The first case is when shifting the calculated response towards the observed and the second is when shifting the observed towards the calculated response.

Case1: Shifting calculated towards the observed

Figs 2.2a, 2.2b show situations when the calculated is to the left of the observed and the calculated to the right of the observed, respectively. The general formula for the generalized travel time as function of the travel time at each point that satisfies the two situations in the next figures is as follows:

$$\Delta \widetilde{t}_i = t_{shift,i} - t_{cal,i} \qquad i = 1,..n_d \qquad\qquad ….……..(2.9)$$

Where, for the first situation as shown in **Fig. 2.2a**, the sign of the generalized travel time is positive while for the other situation as shown in **Fig. 2.2b**, the sign is negative So, irrespective of the relative location of the calculated and the observed, **Eq.2.9** satisfies the both situation for the case of shifting the calculated towards the observed.

As we are shifting all the points with the same amount of shift, so the generalized travel time shift can be written as the average of all the shift for all the points as follows:

$$\Delta \widetilde{t} = \frac{1}{n_d} \sum_{i=1}^{n_d} \left(t_{shift,i} - t_{cal,i} \right) \qquad\qquad ……….(2.10)$$

In the vicinity of the solution or when the shape of the calculated response is close to that of the observed, t_{shift} can be approximately equal to t_{obs} as shown in **Figs. 2.2a,b**. Thus **Eq.2.10** will be as follows:

$$\Delta \widetilde{t} \cong \frac{1}{n_d} \sum_{i=1}^{n_d} \left(t_{obs,i} - t_{cal,i} \right) \qquad\qquad ….……..(2.11)$$

(a)

(b)

(c)

(d)
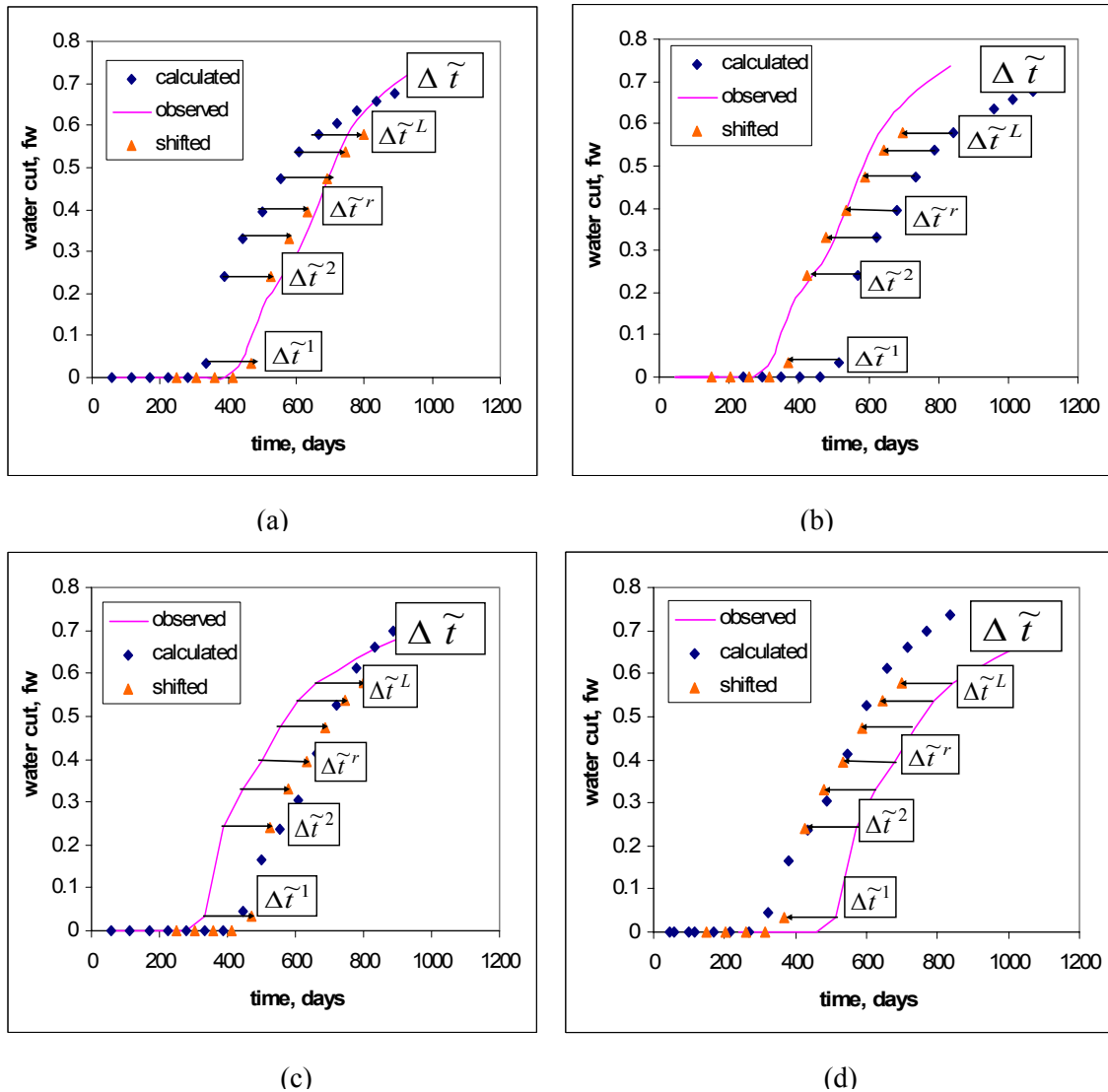
Fig. 2.2–Illustration for the formulation of generalized travel time shift, (a) Shifting the calculated towards the observed: calculated to the right of the observed, (b) Shifting the calculated towards the observed: calculated to the left of the observed, (c) Shifting the observed towards the calculated: calculated to the right of the observed, (d) Shifting the observed towards the calculated: calculated to the left of the observed

Case2: Shifting observed towards the calculated

Figs 2.2c, 2.2d show situations when the calculated is to the left of the observed and the calculated to the right of the observed, respectively. The general formula for the generalized travel time as function of the travel time at each point that satisfies the two situations in the above figures is as follows:

$$\Delta \tilde{t}_i = t_{shift,i} - t_{obs,i} \quad i = 1,..n_d \qquad \qquad \text{……….(2.12)}$$

Where, for the first situation as shown in **Fig. 2.2c**, the sign of the generalized travel time is positive while for the other situation as shown in **Fig. 2.2d**, the sign is negative.

As we are shifting all the points with the same amount of shift, so the generalized travel time shift can be written as the average of all the shift for all the points as follows:

$$\Delta \tilde{t} = \frac{1}{n_d} \sum_{i=1}^{n_d} \left(t_{shift,i} - t_{obs,i} \right) \qquad \qquad \text{……….(2.13)}$$

In the vicinity of the solution or when the shape of the calculated response is close to that of the observed, t_{shift} can be approximately equal to t_{cal} as shown in **Figs. 2.2c, d**. Thus **Eq.2.13** will be as follows:

$$\Delta \tilde{t} \cong \frac{1}{n_d} \sum_{i=1}^{n_d} t_{cal,i} - t_{obs,i} \qquad \qquad \text{……….(2.14)}$$

Notice here the difference in the formulation of the generalized time shift, **Eqs. 2.11** and **2.14** for shifting the calculated towards the observed and the opposite.

It should be mentioned that while using poor initial model, **Eqs. 2.11** and **2.14** might not be good approximation. For example, situations might arise when there is observed water cut response and no calculated response and vise versa. Under such conditions, the generalized travel time shift is given by the difference between the breakthrough time

and the end of the observed response and vise versa. From our experience we have seen that during successive iterations the shape of the production response gets close to the observed and **Eqs. 2.11** and **2.14** can be considered good approximate formulation for the generalized travel time misfit.

2.3 Prior Model

The prior model parameter ($\mathbf{m}$) used in this work is the permeability at each grid block which are modeled as correlated stationary Gaussian random fields with specified means ($\mathbf{m}_{prior}$) and covariance, C_M. The prior covariance is an auto covariance between the permeability at each grid block and it is calculated by knowing the variogram model which consists of three main components; the variogram model, the sill and the range.

For more than one type of model parameter, for example permeability and porosity at each grid block, the covariance matrix will be as follows:

$$C_M = \begin{bmatrix} C_K & C_{K,\phi} \\ C_{\phi,K} & C_\phi \end{bmatrix} \qquad\qquad \text{….....….(2.15)}$$

Where, C_K is the covariance matrix of permeability derived from the permeability variogram modeling, C_ϕ is the covariance matrix of porosity obtained from the porosity variogram modeling, $C_{K,\phi}$ and $C_{\phi,K}$ are the cross covariance matrix between porosity and permeability and is obtained by modeling the cross variogram or by using the screening hypothesis of Xu et al.[37] During this study, the model parameter is the permeability at each grid block which is assumed to have a log normal distribution.

It should be mentioned here that the covariance matrix is a full matrix of order M x M (M is the number of model parameters, i.e. M is equivalent to the number of grid blocks). So for field-scale applications with large number of grid blocks, a certain form of parameterization[38] or approximations using the "stencil" concept[34,39] is required during inversion. The approximation using stencil will be discussed later in this chapter.

2.4 Optimization Algorithms

The minimization of **Eq. 2.8** or **Eq. 2.3** requires an efficient minimization algorithm especially for large field-scale applications where, the number of model parameters is usually high of the order of thousands to million grid block permeabilities or porosities. There are two different methods of minimization algorithms for unconstrained objective function like that given in **Eq. 2.8** or **Eq. 2.3;** the gradient-based algorithms[15] such as the steepest descent, Newton, Gauss-Newton, Levenberg-Marquardt, conjugate gradient and Variable metric (sometimes called quasi-Newton) and the non-gradient based algorithm like simulated annealing, genetic algorithm, Monte Carlo methods, and neural networks. The non gradient-based algorithms are not practical compared to the gradient algorithms for large number of parameters and thus, the gradient-based algorithms are the one that are commonly used in reservoir inverse problems.

   The rates of convergence of each type of the gradient-based algorithms are different. The Newton type of search algorithms like Newton, Gauss-Newton, and Levenberg-Marquardt have quadratic rate of convergence in the vicinity of the solution compared to the super-linear rate of convergence of the variable-metric algorithm and the linear rate of convergence of steepest descent and conjugate gradient.[15] However, the advantage of steepest descent, conjugate gradient, and variable metric is that the computation of sensitivity matrix is not required. Instead, the only requirement is the gradient of the objective function which can be obtained using adjoint method and need only one forward run and a solution of the adjoint system of linear equation only once independent of the number of data or the number of wells.[7, 26-29] Due to the rapid convergence of the Newton type of search algorithms, the next sections will cover briefly the equations used during the minimization for Newton, Gauss-Newton, and Levenberg-Marquardt algorithms.

2.4.1 Newton Algorithm

The Taylor series of the objective function O(**m**), given by **Eq. 2.8**, is as follows:

$$O(\mathbf{m}) = O(\mathbf{m_0}) + \left[\nabla_{\mathbf{m}} O(\mathbf{m_0})\right]^T (\mathbf{m} - \mathbf{m_0}) + \frac{1}{2}(\mathbf{m} - \mathbf{m_0})^T H_0(\mathbf{m} - \mathbf{m_0}) + \ldots \qquad \ldots(2.16)$$

Where, $\nabla_m O(\mathbf{m}_0)$ is the gradient of the objective function with respect to the model parameter, $\mathbf{m}$ at $\mathbf{m} = \mathbf{m}_0$ and H_0 is the Hessian of the objective function at $\mathbf{m} = \mathbf{m}_0$. Taking the gradient of **Eq. 2.16**,

$$\nabla_m O(\mathbf{m}) = \nabla_m O(\mathbf{m}_0) + H_0(\mathbf{m} - \mathbf{m}_0) + ... \qquad\qquad ....(2.17)$$

Locating the point $\mathbf{m}$ at the optimum value of the $O(\mathbf{m})$ is equivalent to locating the point where the gradient of $O(\mathbf{m})$ vanishes. By setting $\nabla_m O(\mathbf{m}) = 0$ in **Eq. 2.17**, **Eq. 2.16** becomes:

$$\mathbf{m} = \mathbf{m}_0 - H_o^{-1} \nabla_m O(\mathbf{m}_0) \qquad\qquad ….....(2.18)$$

Eq. 2.18 is the Newton algorithm and is written in general form as:

$$\mathbf{m}^{l+1} = \mathbf{m}^l - H_l^{-1} \nabla_m O(\mathbf{m}^l) \qquad\qquad …………(2.19)$$

Where, (l) denotes the iteration level.

Newton algorithm, **Eq. 2.19,** requires getting the Hessian and its inverse. For large scale problems, where number of model parameters is extremely high, the inverse of the Hessian matrix which is of order M x M is computationally difficult. In Variable metric method, the inverse of the Hessian in **Eq. 2.19** is updated at each iteration. Zhang et al.[7] used the variable metric method with the gradient of the objective function calculated using adjoint method and they used LBFGS[15] to update the inverse of the Hessian starting with the covariance matrix as the initial guess. However, their method can be computationally efficient if the updated Hessian remains positive definite at each iteration which is not the case in general.

2.4.2 Gauss-Newton Algorithm

By taking the gradient of the objective function given by **Eq. 2.8**,

$$\nabla_{\mathbf{m}} O(\mathbf{m}^{l}) = G_{l}^{T} C_{D}^{-1} \Delta \tilde{\mathbf{t}}_{\mathbf{l}} + C_{M}^{-1}(\mathbf{m}^{l} - \mathbf{m}_{\mathbf{p}}) \qquad \ldots\ldots\ldots(2.20)$$

Where, G_{l} is the sensitivity matrix of the generalized travel time with respect to the model parameter and it is given as:

$$G_{l} = (\nabla_{\mathbf{m}}(\Delta \tilde{\mathbf{t}}_{\mathbf{l}})^{\mathbf{T}})^{T} = \begin{bmatrix} \dfrac{\partial \Delta \tilde{t}_{1,l}}{\partial m_{1}^{l}} & \dfrac{\partial \Delta \tilde{t}_{1,l}}{\partial m_{2}^{l}} & \dfrac{\partial \Delta \tilde{t}_{1,l}}{\partial m_{M}^{l}} \\[2mm] \dfrac{\partial \Delta \tilde{t}_{2,l}}{\partial m_{1}^{l}} & \dfrac{\partial \Delta \tilde{t}_{2,l}}{\partial m_{2}^{l}} & \dfrac{\partial \Delta \tilde{t}_{2,l}}{\partial m_{M}^{l}} \\[3mm] \dfrac{\partial \Delta \tilde{t}_{n_{w},l}}{\partial m_{1}^{l}} & \dfrac{\partial \Delta \tilde{t}_{n_{w},l}}{\partial m_{2}^{l}} & \dfrac{\partial \Delta \tilde{t}_{n_{w},l}}{\partial m_{M}^{l}} \end{bmatrix}_{n_{w} x M} \qquad \ldots\ldots\ldots(2.21)$$

By taking the gradient of **Eq. 2.20**,

$$\nabla_{\mathbf{m}}^{2} O(\mathbf{m}^{l}) = H = (\nabla_{\mathbf{m}} G_{l}^{T}) \cdot \left(C_{D}^{-1} \Delta \tilde{\mathbf{t}}_{\mathbf{l}}\right) + G_{l}^{T} C_{D}^{-1} G_{l} + C_{M}^{-1} \qquad \ldots\ldots\ldots(2.22)$$

For small residual, $\Delta \tilde{\mathbf{t}}_{\mathbf{l}}$, or for quasi-linear problems, the first term in **Eq. 2.22** can be neglected, thus **Eq. 2.22** becomes:

$$H \cong G_{l}^{T} C_{D}^{-1} G_{l} + C_{M}^{-1} \qquad \ldots\ldots\ldots(2.23)$$

Substituting **Eq. 2.20**, and **Eq. 2.23** in the Newton algorithm, **Eq. 2.19**,

$$\mathbf{m}^{l+1} = \mathbf{m}^{l} - \left[G_{l}^{T} C_{D}^{-1} G_{l} + C_{M}^{-1}\right]^{-1} \left[G_{l}^{T} C_{D}^{-1} \Delta \tilde{\mathbf{t}}_{\mathbf{l}} + C_{M}^{-1}(\mathbf{m}^{l} - \mathbf{m}_{\mathbf{p}})\right] \qquad \ldots\ldots\ldots(2.24)$$

Eq. 2.24 is the Gauss-Newton formula used during the minimization. The difficulties of **Eq. 2.24**, is that updating the model parameter at each iteration requires obtaining the inverse of the covariance matrix plus the inverse of the matrix $\left[G_l^T C_D^{-1} G_l + C_M^{-1} \right]$ both of which are of order M x M. Tarantola[34] and Chu et al.[25] used a matrix inverse lemma to convert **Eq. 2.24** in a form computationally efficient when the number of model parameters are greater than the number of data. This form is:

$$\mathbf{m}^{l+1} = \mathbf{m_p} - \mathbf{C_M G_l^T} \left[C_D + G_l C_M G_l^T \right]^{-1} \left[\Delta \tilde{t}_l - G_l (\mathbf{m^l} - \mathbf{m_p}) \right] \qquad \ldots\ldots\ldots\ldots(2.25)$$

The form given in **Eq. 2.25** is called Modified Gauss-Newton, **Appendix A** shows the derivation of the Modified Gauss-Newton formula. **Eq. 2.25** and **Eq. 2.24** are mathematically equivalent, but the computation time for both is completely different. **Eq. 2.25** requires only the inverse of matrix $\left[C_D + G_l C_M G_l^T \right]$ which is of order $N_w \times N_w$ (N_w is the number of wells) in using the generalized travel time as the data misfit.

It is worth to mention here that starting an initial guess with poor model makes the residual too large and the approximation of the Hessian given by **Eq. 2.23** will not be a valid assumption and this lead to a poor convergence of Gauss-Newton. Li[40] shows that using Levenberg-Marquardt algorithm with high value of the damping factor at the initial iteration to damp the model changes can overcome the convergence problem of the high data misfit at the early iterations. Levenberg-Marquardt algorithm is discussed in the next section.

2.4.3 Levenberg-Marquardt Algorithm
Bi[41] modified Levenberg-Marquardt formula for application to the inverse problems to be in the following form:

$$\mathbf{m}^{l+1} = \mathbf{m^l} + \frac{\mathbf{m_p} - \mathbf{m^l}}{\mathbf{1 + \alpha}} - \mathbf{C_M G_l^T} \left[(1 + \alpha) \cdot C_D + G_l C_M G_l^T \right]^{-1}$$
$$\left[\Delta \tilde{t}_l - \frac{1}{1 + \alpha} G_l (\mathbf{m^l} - \mathbf{m_p}) \right] \qquad \ldots\ldots(2.26)$$

α is the damping factor and for large α, the change in the model parameters per iteration is small. Li[40] use high value of α equal to 10^4 or 10^5 at the initial iteration for large residual to ensure reduction in the objective function and whenever there is a reduction in the objective function from one iteration to the other, the value of α decreased by a factor of 10 until it becomes close to zero, where **Eq. 2.26** tends to the original Modified Gauss-Newton, **Eq. 2.25** which is a good assumption at small residual.

The minimization algorithm given by **Eq. 2.25** requires knowledge about the sensitivity matrix, G which is a very critical step during minimization. Chapter III will be devoted to show the calculation of the sensitivity matrix using the finite difference simulator as forward model.

2.5   Bayesian Formulation for Field-Scale Applications

The central point for the second part of this chapter deals with reformulating the objective function, **Eq. 2.8** resulting from the Bayesian approach and use the same approach of Gauss-Newton algorithm to reach to a system of equations for model updating in order to reduce the burden of matrix multiplications during the minimization process using the Modified Gauss-Newton, **Eq. 2.25**. Thus, reducing the computation time and making it well-suited for large-scale field applications.

2.5.1   Bayesian Formulation

The objective function in the Bayesian formulation given by **Eq. 2.8** is re-written in the following from:

$$O(\mathbf{m}) = \frac{1}{2} e^T e \qquad \ldots\ldots\ldots(2.27)$$

Where,

$$e = \begin{bmatrix} C_D^{-\frac{1}{2}} \Delta \widetilde{\mathbf{t}}_{\mathbf{l}} \\ C_M^{-\frac{1}{2}} (\mathbf{m}^{\mathbf{l}} - \mathbf{m}_{\mathbf{p}}) \end{bmatrix} \qquad \ldots\ldots\ldots(2.28)$$

The minimization of the objective function given in **Eq. 2.27** can be obtained by using Newton's optimization algorithm given by **Eq. 2.19** as follows:

$$H\,\delta\mathbf{m} = -\nabla_{\mathbf{m}}O(\mathbf{m})$$
.........(2.29)

Where, $\nabla_{\mathbf{m}}O(\mathbf{m})$ is obtained from **Eq. 2.27** as follows:

$$\nabla_{\mathbf{m}}O(\mathbf{m}) = \left(\nabla_{\mathbf{m}}e^{T}\right)e = \left[G_{l}^{T}C_{D}^{-1/2} \quad C_{M}^{-1/2}\right]e$$
.........(2.30)

Letting the Jacobian, J, be as follows:

$$J = \left(\nabla_{\mathbf{m}}e^{T}\right)^{T} = \begin{bmatrix} C_{D}^{-1/2}\,G_{l} \\ C_{M}^{-1/2} \end{bmatrix}$$
.........(2.31)

Substitute **Eq. 2.31** in **Eq. 2.30**,

$$\nabla_{\mathbf{m}}O(\mathbf{m}) = J^{T}e$$
...........(2.32)

The Hessian is obtained by taking the gradient of **Eq. 2.32** with respect to the model parameter (**m**):

$$H = \nabla_{m}\left(\nabla_{\mathbf{m}}O(\mathbf{m})\right)^{T} = \nabla_{\mathbf{m}}(e^{T}\left(\nabla_{\mathbf{m}}e^{T}\right)^{T}) = \nabla_{\mathbf{m}}e^{T}\,(\nabla_{\mathbf{m}}e^{T})^{T} + e^{T}\nabla_{\mathbf{m}}(\nabla_{\mathbf{m}}e^{T})^{T}$$
$$= J^{T}J + e^{T}\nabla_{\mathbf{m}}J$$
......(2.33)

Similarly, as Gauss-Newton, by neglecting the second term of **Eq. 2.33**, **Eq. 2.33** becomes:

$$H \cong J^{T}J$$
........(2.34)

The approximation for the Hessian, **Eq. 2.34**, is the same as that of the Gauss-Newton algorithm and is strictly valid near the solution (small misfit) or for quasilinear problems. Substituting **Eqs. 2.32** and **2.34** in **Eq. 2.29**;

$$J^T J \, \delta \mathbf{m} = -J^T e \qquad \qquad \ldots\ldots\ldots(2.35)$$

Eq. 2.35 is simply a least-squares solution to the following system of equations

$$J \, \delta \mathbf{m} = -e \qquad \qquad \ldots\ldots\ldots(2.36)$$

Substitute **Eqs. 2.28** and **2.31** in **Eq. 2.36**,

$$\begin{bmatrix} C_D^{-1/2} G_l \\ C_M^{-1/2} \end{bmatrix} \delta \mathbf{m} = \begin{bmatrix} -C_D^{-1/2} \, \Delta \tilde{\mathbf{t}}_l \\ C_M^{-1/2} \left(\mathbf{m_p} - \mathbf{m^l} \right) \end{bmatrix} \qquad \ldots\ldots\ldots(2.37)$$

    Eq. 2.37 is mathematically equivalent to the Gauss-Newton formulation, **Eq. 2.24** and in turn equivalent to the Modified Gauss-Newton formulation, **Eq. 2.25**. **Appendix A** shows the mathematical equivalent between the two formulations, **Eq. 2.37** and **Eq. 2.24**.

    Eq. 2.37 represents a system of linear equations and we use an iterative sparse matrix solver, LSQR[42] for solving this system. LSQR is well suited for highly ill-conditioned systems and is widely used for large-scale tomographic problem in seismology.[43] However, difficulties arise in the computation of the square root of the matrix inverse in **Eq. 2.37**. In practice, the data covariance matrix is assumed to be diagonal and is thus easy to manipulate. However, the covariance matrix for the model parameters can be full and in general, the calculation of $C_M^{-1/2}$ will be computationally prohibitive for large-scale inverse problems. Previous efforts to compute $C_M^{-1/2}$ analytically have been limited to exponential covariance model[14]. Vega[44] proposes an

approach to approximate the square root of the inverse of the covariance using a numerical stencil which is general for any covariance models. The next section will give brief overview for approximating the square root of the inverse of the prior covariance matrix using the numerically derived "stencil".

The scaling of the computation time with respect to the model parameters for the conventional Bayesian formulation, **Eq. 2.25**, and **Eq. 2.37** will be studied in terms of the number of multiplications required by each formulation after discussing the concept of the numerical stencil.

2.5.2 Square Root of the Inverse of the Covariance Using Numerically-Derived Stencil

The exact analytical calculation of the square root of the inverse of the covariance can be done using the concept of matrix diagonalization.[45] Since the covariance matrix is a symmetrical matrix so, its square root of the inverse can be calculated exactly using the following equation:

$$C_M^{-1/2} = U^T \Lambda^{-1/2} U \qquad \qquad \text{.......(2.38)}$$

Where U is the matrix, whose columns are the eigenvectors of C_M, Λ is the diagonal matrix whose diagonal elements are the eigenvalues of the covariance matrix C_M. This computation is very difficult to handle especially for large field-scale cases where the covariance matrix is full and large. Another alternative is to use iterative algorithms like Newton method[46] to get the square root of the inverse of the covariance matrix. However, this method requires the calculation of the inverse of the covariance per iteration, which makes it impractical for large-scale problems. Recent attempt used to approximate the square root of the inverse of the covariance matrix by obtaining analytically its stencil from the covariance kernel[14] based on the previous works for calculating the inverse of the prior covariance matrix.[34,39] However, the analytical approximation suffers from two major limitations; it is applicable only for the

exponential covariance and the ratio of the grid size to the range in the three directions need to be equal.

Due to these limitations, Vega[44] proposed a method that overcomes these limitations which based on two basic principles; First, the covariance matrix and the square root of its inverse can be constructed using their respective kernels, Second, the two kernels remain unchanged regardless of the size of the matrix.

The following are the procedures used to approximate the square root of the inverse of the covariance matrix using a numerically-derived stencil. First, and the most important step, is choosing the size of the stencil, which depends mainly upon the ranges, the grid sizes and the number of gridblocks in the three directions. Selections of the stencil is a tradeoff between speed and accuracy and sensitivity study should be done to best select the stencil required depending upon the behavior of each problem. To make the method understandable, we assume that 5x5x5 stencil provide a good compromise between efficiency and accuracy, so a 5x5x5 stencil can be used to approximate the square root of the inverse of the covariance matrix. Second, the concept of matrix diagonalization, **Eq. 2.38,** is used to get the square root of the inverse of the covariance for 5x5x5 grid block (125x125 covariance matrix) by knowing the kernel of the covariance. This is equivalent to getting the kernel of the square root of the inverse of any covariance function in a discretized or numerical form other than obtaining the kernel analytically as before.[14] Third, the set up of the 5x5x5 stencil is shown in **Fig. 2.3**. This stencil has only 27 distinct elements due to symmetry. Any column or any row of the covariance matrix calculated from the second step can be used to get the magnitude of each stencil presented in **Fig. 2.3**. **Column 63** is selected for convenience, as it is the middle column to construct the magnitude of each stencil. **Table 2.1** shows the location of the stencil in the 125 x 125 matrix constructed in the second step, which in turn gives the magnitude of each stencil. Finally, the approximation of the square root of the inverse of the covariance for the model under study is obtained by using the stencil constructed in **Fig. 2.3** and its magnitude obtained from **Table 2.1.**

Fig. 2.3–5x5x5 stencil used for the numerical approximation of the square root of the inverse of the covariance

Table 2.1–Location of the numerical stencil terms from column 63 of the square root of inverse of covariance of $5 \times 5 \times 5$ grid	
Numerical Stencil Term	**Row Number in Column 63**
G(0)	63
G(1)	62
G(2)	58
G(3)	38
G(4)	33
G(5)	57
G(6)	37
G(7)	32
G(8)	61
G(9)	53
G(10)	13
G(11)	52
G(12)	56
G(13)	8
G14)	28
G(15)	36
G(16)	12
G(17)	31
G(18)	27
G(19)	7
G(20)	51
G(21)	11
G(22)	3
G(23)	2
G(24)	6
G(25)	26
G(26)	1

2.5.3 Computational Scaling Properties: Conventional vs Field-Scale Bayesian Formulation

In comparing the computation effort required by each formulation, we assume that the computational effort is directly proportional to the number of multiplications required by each formulation and our objective is to show how the number of multiplications for each formulation behaves with increasing the number of model parameters. For this

purpose we will assume that the sensitivity matrix required by the both formulation is full and covariance matrix is full in case of the conventional Bayesian, while in case of the new Bayesian formulation, we will use the concept of the stencil discussed before to approximate the square root of the inverse of the covariance. Thus the matrix will be sparse and we will use the maximum number of the non zero values for each row to be equal to the N_s (i.e. N_s is the maximum number of stencil, which is equal to 125 in case of using 5x5x5 stencil).

Conventional Bayesian formulation

The conventional Bayesian formulation is given by **Eq. 2.25**. The number of multiplication per iteration required is calculated as follows:

- Forming $G_l(\mathbf{m}^l - \mathbf{m}_p)$: $[G_l]_{N_w \times M} \cdot [\mathbf{m}^l - \mathbf{m}_p]_{M \times 1}$ requires $(N_w \cdot M)$ multiplications

- Forming $C_M G_l^T$: $[C_M]_{M \times M} \cdot [\mathbf{G}_\mathbf{l}^\mathbf{T}]_{M \times N_w}$ requires $(N_w \cdot M^2)$ multiplications

- Forming $C_D + G_l C_M G_l^T$: It requires the following multiplication $[G_l]_{N_w \times M} \cdot [\mathbf{C}_\mathbf{M} \mathbf{G}_\mathbf{l}^\mathbf{T}]_{M \times N_w}$, which require $(N_w^2 \cdot M)$ multiplications

- Forming $[C_D + G_l C_M G_l^T]^{-1}[\Delta \tilde{\mathbf{t}}_\mathbf{l} - G_l(\mathbf{m}^l - \mathbf{m}_p)]$: It require solving a system of equations in the form of $[C_D + G_l C_M G_l^T]_{N_w \times N_w} \cdot [g_f]_{N_w \times 1} = [\Delta \tilde{\mathbf{t}}_\mathbf{l} - G_l(\mathbf{m}^l - \mathbf{m}_p)]_{N_w \times 1}$ to get (g_f). The LU Decomposition and LU back substitution is used to get (g_f) and this operation requires[45] (N_w^3) multiplications.

- Forming $C_M G_l^T[C_D + G_l C_M G_l^T]^{-1}[\Delta \tilde{\mathbf{t}}_\mathbf{l} - G_l(\mathbf{m}^l - \mathbf{m}_p)]$: $[\mathbf{C}_\mathbf{M} \mathbf{G}_\mathbf{l}^\mathbf{T}]_{M \times N_w} \cdot [g_f]_{N_w \times 1}$ requires $(N_w \cdot M)$ multiplications.

Adding up all the above operations, results in the total number of multiplications per iteration required by the conventional Bayesian approach.

$$Z_{GN} = N_w M^2 + 2N_w M + N_w^2 M + N_w^3 \qquad \qquad \ldots\ldots\ldots(2.39)$$

Field-scale Bayesian formulation

The field-scale Bayesian formulation is given by **Eq. 2.37**, by assuming that C_D is diagonal matrix, thus **Eq. 2.37** can be written as:

$$\begin{bmatrix} G_I \\ C_M^{-1/2} \end{bmatrix} \delta \mathbf{m} = \begin{bmatrix} -\Delta \tilde{\mathbf{t}}_1 \\ C_M^{-1/2} (\mathbf{m_p} - \mathbf{m^1}) \end{bmatrix} \qquad \ldots\ldots\ldots(2.40)$$

We used LSQR[42] as an iterative sparse matrix solver to solve the above augmented system of equations. I followed the exact algorithm given by Paige and Saunders[42] in counting the number of multiplication required by solving **Eq. 2.40,** which can be written in the following form:

$$[A]_{(N_w+M) \times M} \; [\mathbf{x}]_{M \times 1} = [b]_{(N_w+M) \times 1} \qquad \ldots\ldots\ldots(2.41)$$

The following is the number of multiplications required at each step in the algorithm:

1) **Initializing**:
   - Forming $[u_1]_{(N_w+M) \times 1} = [b]_{(N_w+M) \times 1}$: required the product of $[C_M^{-1/2}]_{M \times M} \cdot [\mathbf{m_p} - \mathbf{m^1}]_{M \times 1}$, which requires $(M \cdot N_s)$ multiplications, where N_s is the maximum number of the stencil used to approximate the square root of the inverse of the covariance (i.e. maximum number of non zero values per each row of $C_M^{-1/2}$).
   - Normalize $[u_1]_{(N_w+M) \times 1}$ and get β_1 : require $(N_w + M)$ multiplications
   - Form $[v_1]_{M \times 1} = [A^T]_{M \times (N_w+M)} [u_1]_{(N_w+M) \times 1}$: It is equivalent to find the product in the following form, $[v_1]_{M \times 1} = [G^T]_{M \times N_w} [u`_1]_{N_w \times 1} + [(C_M^{-1/2})^T]_{M \times M} [u_1``]_{M \times 1}$, this require $(N_w \cdot M + M \cdot N_s)$ multiplications
   - Normalize $[v_1]_{M \times 1}$ and get α_1 : require (M) multiplications

2) Start Iteration inside the LSQR loop for i=1,2,.....N_{iter}, where N_{iter} is the number of iteration

3) Bidiagonalization

- Form $[u_{i+1}]_{(N_w+M)\times1} = [A]_{(N_w+M)\times M}[v_i]_{M\times1} - \alpha_i[u_i]_{(N_w+M)\times1}$: require the following two products;

  a- $\alpha_i[u_i]_{(N_w+M)\times1}$: require (N_w+M) multiplications

  b- $[A]_{(N_w+M)\times M}[v_i]_{M\times1}$: is equivalent to forming the two products;

  $$[G]_{N_w\times M}[v_i]_{M\times1}: \text{require } (N_w \cdot M) \text{ multiplications}$$

  $$[C_M^{-1/2}]_{M\times M}[v_i]_{M\times1}: \text{require } (N_s \cdot M) \text{multiplications}$$

- Normalize $[u_{i+1}]_{(N_w+M)\times1}$ and get β_{i+1}: require (N_w+M) multiplications

- Form $[v_{i+1}]_{M\times1} = [A^T]_{M\times(N_w+M)}[u_{i+1}]_{(N_w+M)\times1} - \beta_{i+1}[v_i]_{M\times1}$: require the following two products;

  a- $\beta_{i+1}[v_i]_{M\times1}$: require (M) multiplications

  b- $[A^T]_{M\times(N_w+M)}[u_{i+1}]_{(N_w+M)\times1}$: It is equivalent to find the product in the following

  form, $[G^T]_{M\times N_w}[u^{`}_{i+1}]_{N_w\times1} + [(C_M^{-1/2})^T]_{M\times M}[u_{i+1}^{``}]_{M\times1}$,

  this require $(N_w \cdot M + M \cdot N_s)$ multiplications

- Normalize $[v_{i+1}]_{M\times1}$ and get α_{i+1}: require (M) multiplications

4) Construct and apply next orthogonal transformation: in this step the number of multiplications is independent upon the number of gridblocks (M) or the number of wells (N_w), so its multiplications count is not considered as they are very trivial.

5) Update $[x]_{M\times1}$, $[w]_{M\times1}$:

- Update $[x]_{M\times1}$: require (M) multiplications

- Update $[w]_{M\times1}$: require (M) multiplications

6) End of the iteration

Adding the number of multiplications mentioned at each step and inside the LSQR iteration loop, the total number of multiplications will be:

$$Z_{Field\ \text{scale Bayesian}} = 2(M \cdot N_s) + (N_w \cdot M) + N_w + 2M +$$
$$N_{iter}\left[2(N_w \cdot M) + 2(M \cdot N_s) + 2N_w + 6M\right] \quad \text{……….(2.42)}$$

Fig. 2.4 shows the behavior of each formulation with respect to the model size using N_w of 15 for the both formulation and using N_s and N_{iter} for the field-scale Bayesian formulation to be 125, and 15 respectively. I assumed that the stencil used is 5x5x5, so the maximum value for N_s is 125.
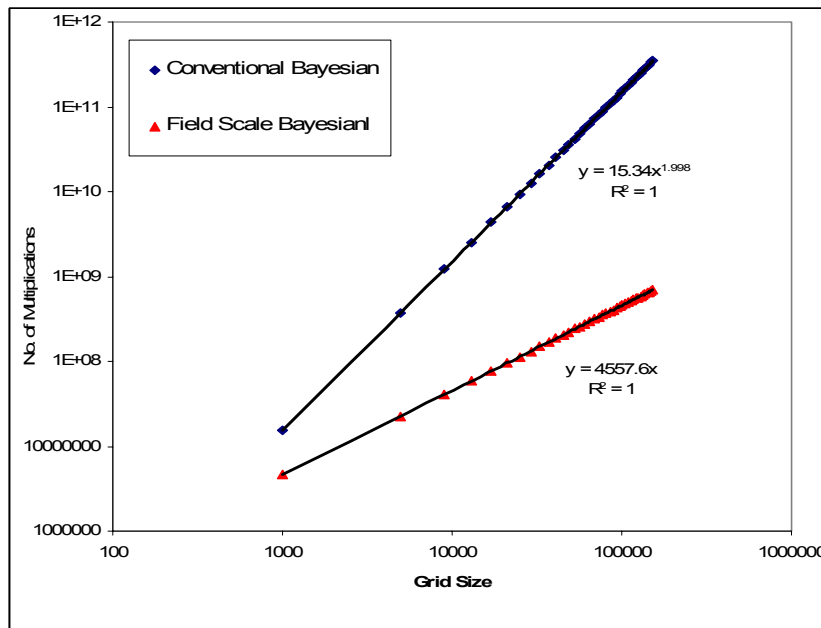
Fig. 2.4–Computational scaling for field-scale Bayesian vs conventional Bayesian

Clearly as shown from **Eqs. 2.39** and **2.42** and from the figure above, the field-scale Bayesian formulation behaves linearly compared to the conventional Bayesian

formulation which behaves quadratically with respect to the model size. This shows that the new Bayesian formulation with the numerically calculated stencil is well suited for field-scale applications where the number of model parameters exceeds hundred thousands to millions. However, it is important to mention that the number of multiplications in the field-scale Bayesian formulation depend upon two additional parameters other than the conventional Bayesian. Those parameters are the inner iteration inside the LSQR loop, N_{iter} and the number of stencil used, N_s. Selecting the stencil is the most important factor in choosing between the two formulations as for high N_s, the Field-scale Bayesian behaves close to the conventional Bayesian and due to the inner iterations inside the LSQR loop it might even behave worse than the conventional Bayesian formulation. That is why selecting the stencil used is the key in choosing which formula to use during the minimization.

2.6 Bayesian Approach as a Tool for Uncertainty Assessment

The Bayesian approach gives very distinct advantage in associating probability distribution known as the posterior distribution with its estimate. Sampling the posterior distribution to generate a suite of realizations provides a good tool to assess the uncertainty in reservoir variables. Moreover, one can predict the uncertainty in the future performance of the reservoir by constructing statistics for the set of outcomes obtained from the realizations generated from the posterior distribution. Accordingly, the correct sampling of the posterior is a vital issue in accurately quantifying the uncertainties and many research works are devoted to find an efficient way for accurately sampling the posterior distribution.

In general, there are two different types of sampling. The first type belongs to the methods that is known to sample rigorously, such as inversion, rejection, sequential realization, Gibbs sampler, Markov Chain Mont Carlo (MCMC), and Genetic algorithms.[34] The second type belongs to those that are known to sample approximately[47,48] such as linearization about the maximum a posteriori estimate (LMAP), randomized maximum likelihood (RML)[49], and pilot point (PP) method.

The most common method used to sample the posterior distribution rigorously is the MCMC method and many of the approximate methods are just an approximation from the MCMC under some specific conditions. In this section, a brief introduction about the MCMC as a rigorous way of sampling and Randomized Maximum Likelihood (RML) as an approximate way of sampling is given for completeness.

2.6.1   Markov Chain Monte Carlo (MCMC)

According to Cunha et al.[50], a Markov chain is a sequence of random variables $X^{(0)}$, $X^{(1)},...X^{(n)}$ where the probability distribution for $X^{(n)}$ is determined by the probability distribution of $X^{(n-1)}$. The set of all possible values for $X^{(i)}$ (i = 0,1,2,...n) is called the state space. The transition probability, P_{ij}^{n} gives the probability of obtaining state j at the nth location in the sequence if the random variable is in state i at the n-1 location in the sequence.

Let us denote a particular realization of a certain reservoir properties by m^i, with the probability associated with it to be π_i, where π is the posterior probability distribution that we want to sample from and π_i is the probability of sampling m^i from the posterior (i.e. $\pi(m^i)$). The objective is to generate a new realization or new state m^j with the probability associated with it to be π_j such that the transition probability, P_{ij} of obtaining the state m^j from state m^i satisfies the following conditions[51],

1- It is possible to get from any one state to another in a finite number of transitions,

2- $\pi_j = \sum_i \pi_i \, P_{ij}$ ………(2.43)

Thus, the Markov chain will be stationary and ergodic (independent on initial conditions) and π will be the stationary distribution, which is the posterior distribution that we want to sample from.

In Metropolis-Hasting algorithm, the transition probability, P_{ij} is partitioned into two parts: $P_{ij} = \alpha_{ij} \, q_{ij}$, where q_{ij} is the probability of proposing transition from state m^i to state m^j and α_{ij} is the probability of accepting the proposed transition m^j as the next state in the Markov chain and it is given by:

$$\alpha_{ij} = \min\left\{1, \frac{\pi_j q_{ji}}{\pi_i q_{ij}}\right\} \qquad \ldots\ldots\ldots(2.44)$$

Two important points should be taken into consideration in sampling the posterior distribution using MCMC. The first is selection of q_{ij} (i.e. the PDF used to sample for the new state, m^j). If q_{ij} is close to the posterior distribution that we want to sample from, most of the transitions will be accepted and the chain converges quickly to the posterior distribution. The second point is the way of sampling, for example, whether the two-point swapping, where two model parameter in two different grid blocks are swapped per each transition to the new state or the local perturbation, where one model parameter is perturbed per transition, or the global perturbation, where all the model parameters are perturbed once per transition. Oliver et al.[51] shows that local perturbation is more efficient in generating independent realizations in the Markov chain for highly non-linear problem, while global perturbation is efficient in linear to slightly non linear problems.

2.6.2 Randomized Maximum Likelihood (RML)

Oliver et al.[49] propose a two step transition to a new state in MCMC that has a high probability of acceptance in the Metropolis-Hasting algorithm. The first step is to propose an unconditional realization from the prior probability distribution using any unconditional simulation technique, like sequential Gaussian simulation.[52] The second step involves the history matching of the unconditional model to the production data that has noise added. The conditional model resulting from the history matching process will be a new state in the Markov chain. Because the acceptance rate was found very high (approximately 95% for a small highly non-linear problem), they suggested to accept all the new state proposed in the chain.

In this study, we applied this method to generate realizations and assessing the uncertainty. The following is the algorithm for generating realizations from RML method:

1. Generate an unconditional realization of the reservoir model parameters, $m_u \leftarrow N[m_p, C_M]$, by assuming that the variogram of the prior model is known, this can be done by using sequential Gaussian simulation or any unconditional simulation technique

2. Generate a realization of the data, $d_u \leftarrow N[d_{obs}, C_D]$, by adding a random noise to the observed data

3. Compute the set of model variables, m, that minimizes the function:

$$O(m) = \frac{1}{2}(m - m_u)^T C_M^{-1}(m - m_u) + \frac{1}{2}[g(m) - d_u]^T C_D^{-1}[g(m) - d_u] \qquad \ldots\ldots\ldots(2.45)$$

2.7 Chapter Summary

Bayes theory provides a good statistical tool to integrate static with dynamic data and assess the uncertainty associated with the estimate. This chapter consists of three major parts. The first part gives a brief introduction about Bayes theory and shows the three important components for automatic history matching in Bayesian framework. Those components are the data misfit, the prior term, and the optimization algorithm. For the data misfit, the general three types of the data misfit are explained, with special reference to the one that is used during this study which is the "Generalized Travel Time Misfit". For the prior term, a brief introduction is given about the covariance matrix calculation in case of single type of model parameter, like permeability and different type of model parameters, for example permeability and porosity. For the optimization algorithm, a brief introduction about gradient-based and gradient-free algorithms with detailed overview about the gradient based algorithms especially; the Newton type algorithms are given.

The second part of this chapter concerns with the new Bayesian formulation that is well-suited for field-scale applications. A numerical approximation of the square root of the inverse of the covariance using the stencil is required by the new formulation and is discussed. Finally the computational scaling in the form of the number of multiplication required by both the conventional and the field-scale Bayesian is investigated. From the

computational scaling it was found that the field-scale Bayesian scales linearly with the increase in the number of model parameters compared to the conventional Bayesian which scales quadratically. This makes the field-scale Bayesian well suited for field-scale applications from the computational point of view.

The last part of this chapter gives one of the most important applications of the Bayes theory which is the uncertainty assessment. In this part a brief introduction about different types of sampling from the posterior distribution for uncertainty assessment are given with a special reference to the rigorous one, which is the MCMC and the approximate one which is used during this study, the Randomized Maximum Likelihood (RML) method.

CHAPTER III
SENSITIVITY COMPUTATION USING FINITE DIFFERENCE MODELS

Sensitivity computation is a very critical step in automatic history matching process when using gradient-based optimization algorithm like Gauss-Newton or Levenberg-Marquardt to minimize the objective function given in the previous chapter. The sensitivity is defined as the change in the production response due to a small change in the model parameter. Chapter I discusses in details the different methods for sensitivity computation using finite difference as the forward model. In this chapter, we present the computation of the generalized travel time sensitivity with respect to the model parameter using the adjoint method (optimal control theory) and the streamline-based sensitivity for finite difference models. A complete detail for developing the generalized travel time sensitivity using adjoint method for 3D, two phase flow is given with a brief introduction for the generalized travel time sensitivity computation from streamline-based sensitivity.

3.1 Adjoint Method-Based Sensitivity

There are three different approaches presented in the literature for sensitivity formulation using the adjoint method. The first approach shows the sensitivity formulation starting from a complete continuous form of the flow equation[26,27], the second approach shows the formulation starting from a semi-continuous form of the flow equation[28,8], where only the left hand side of the flow equation is discretized leaving the right hand side in a continuous form. The third approach shows the formulation in a complete discretized form,[5-7,29] where the flow equation is completely discretized. In this work, we will apply the third approach to develop the formulation of the generalized travel time sensitivity for 3D two phase flow problems.

The sensitivity computation using the adjoint method requires the following three steps; the forward model formulation, adjoint system formulation, and finally the

sensitivity coefficient formulation. The next sections will describe these steps in more details.

3.1.1 Forward Model Formulation

In our work the forward model is IMPES finite difference simulator that is developed to handle 3D, two-phase, oil-water problems. Then later, we used a commercial simulator (viz. ECLIPSE[53]) for modeling fluid flow in porous media and the adjoint method-based sensitivity calculation for practical applications.

The equations governing the two-phase flow equation in the reservoir after neglecting capillary pressure are as follows:

Oil:

$$C_1 \nabla \bullet \left(\frac{K_{ro}}{\mu_o B_o} [K] (\nabla p(x,y,z,t) - \gamma_o \nabla D(x,y,z)) \right) - \hat{q}_o(x,y,z,t) = \frac{\partial}{\partial t} \left(\frac{\phi S_o}{C_2 B_o} \right) \quad \ldots(3.1)$$

Water:

$$C_1 \nabla \bullet \left(\frac{K_{rw}}{\mu_w B_w} [K] (\nabla p(x,y,z,t) - \gamma_w \nabla D(x,y,z)) \right) - \hat{q}_w(x,y,z,t) = \frac{\partial}{\partial t} \left(\frac{\phi S_w}{C_2 B_w} \right) \ldots(3.2)$$

$$S_o + S_w = 1.0 \qquad\qquad\qquad \ldots\ldots\ldots(3.3)$$

Where, C_1 equal 1.127×10^{-3}, C_2 equal 5.615, B_m is the formation volume factor in bbl/STB for m stands for o (oil), w (water), K_{rm} is the relative permeability for phase m, μ_m is the viscosity for phase m in cp, $[K]$ is the permeability tensor,

$\begin{bmatrix} K_x & 0 & 0 \\ 0 & K_y & 0 \\ 0 & 0 & K_z \end{bmatrix}$ K_x, K_y, K_z are the permeability in md, p is the pressure at x,y,z

location and at time t, ϕ is porosity, $\hat{q}(x,y,z,t)$ is the source or the sink term per unit bulk volume, STB/ft^3-day it is zero at all the locations except at the location of the wells. It is positive for production wells and negative for injection wells, S_m is the phase m saturation, γ_m is the specific weight of oil and water, and its unit is in psi/ft, and D is the depth at x,y,z location.

By multiplying **Eqs. 3.1** and **3.2** by bulk volume, V_b, of each grid block and using finite difference to discretize **Eqs. 3.1** and **3.2**, and neglecting the formation compressibility, the following discretized equations are formed:

Oil:

$$T_{ox,i+1/2,j,k}(p_{i+1,j,k} - p_{i,j,k} - \gamma_{oi+1/2,j,k}(D_{i+1,j,k} - D_{i,j,k}))$$
$$-T_{ox,i-1/2,j,k}(p_{i,j,k} - p_{i-1,j,k} - \gamma_{oi-1/2,j,k}(D_{i,j,k} - D_{i-1,j,k}))$$
$$+T_{oy,i,j+1/2,k}(p_{i,j+1,k} - p_{i,j,k} - \gamma_{oi,j+1/2,k}(D_{i,j+1,k} - D_{i,j,k}))$$
$$-T_{oy,i,j-1/2,k}(p_{i,j,k} - p_{i,j-1,k} - \gamma_{oi,j-1/2,k}(D_{i,j,k} - D_{i,j-1,k})) \quad\quad\quad\ldots\ldots\ldots(3.4)$$
$$+T_{oz,i,j,k+1/2}(p_{i,j,k+1} - p_{i,j,k} - \gamma_{oi,j,k+1/2}(D_{i,j,k+1} - D_{i,j,k}))$$
$$-T_{oz,i,j,k-1/2}(p_{i,j,k} - p_{i,j,k-1} - \gamma_{oi,j,k-1/2}(D_{i,j,k} - D_{i,j,k-1})) - q_{o,i,j,k}$$
$$= \frac{V_b\,\phi_{i,j,k}}{C_2\,\Delta t^l}\left[(S_o\,b_o)_{i,j,k}^{l+1} - (S_o\,b_o)_{i,j,k}^{l}\right]$$

Water:

$$T_{wx,i+1/2,j,k}(p_{i+1,j,k} - p_{i,j,k} - \gamma_{wi+1/2,j,k}(D_{i+1,j,k} - D_{i,j,k}))$$
$$-T_{wx,i-1/2,j,k}^{l}(p_{i,j,k} - p_{i-1,j,k} - \gamma_{wi-1/2,j,k}(D_{i,j,k} - D_{i-1,j,k}))$$
$$+T_{wy,i,j+1/2,k}(p_{i,j+1,k} - p_{i,j,k} - \gamma_{wi,j+1/2,k}(D_{i,j+1,k} - D_{i,j,k}))$$
$$-T_{wy,i,j-1/2,k}(p_{i,j,k} - p_{i,j-1,k} - \gamma_{wi,j-1/2,k}(D_{i,j,k} - D_{i,j-1,k})) \quad\quad\quad\ldots\ldots\ldots(3.5)$$
$$+T_{wz,i,j,k+1/2}(p_{i,j,k+1} - p_{i,j,k} - \gamma_{wi,j,k+1/2}(D_{i,j,k+1} - D_{i,j,k}))$$
$$-T_{wz,i,j,k-1/2}(p_{i,j,k} - p_{i,j,k-1} - \gamma_{wi,j,k-1/2}(D_{i,j,k} - D_{i,j,k-1})) - q_{w,i,j,k}$$
$$= \frac{V_b\,\phi_{i,j,k}}{C_2\,\Delta t^l}\left[(S_w\,b_w)_{i,j,k}^{l+1} - (S_w\,b_w)_{i,j,k}^{l}\right]$$

Where, q_m is the phase m rate in STB/day, T_m is the phase m transmissibility in x, y, and z directions, b_m is the reciprocal of the formation volume factor of phase m, Δt is the time step, and i = 1,2,……n_x, j = 1,2,….n_y, k = 1,2,….,n_z.

Eqs. 3.4, 3.5 with **Eq. 3.3** are solved to calculate the pressure and water saturation at each grid block per each time step, in addition to the production history at each well. Since we are interested in matching only the water cut from the wells, so the production history reported is only the water cut at each well. Also, the pressure and water saturation at each grid block per each time step should be stored, as it will be used during sensitivity computation.

3.1.2 Adjoint System Formulation

In formulating the adjoint system of equations, the discretized flow equations, **Eqs. 3.4** and **3.5** are given in the following form:

$$F_m^{l+1}(p^{l+1}, S_w^{l+1}, K_x, K_y, K_z) = V_{\phi,i,j,k}^{l+1}\left[(b_m S_m)_{i,j,k}^{l+1} - (b_m S_m)_{i,j,k}^{l} \right] \qquad \ldots\ldots\ldots\ldots(3.6)$$

It is important to note that **Eq. 3.6** gives the difference equations in fully implicit form, thus it is important that the simulator used to be in fully implicit form for consistency and for the pressure and water saturation calculated from the simulator to satisfy **Eq. 3.6**, which is used during our formulation of the adjoint system of equations as our forward model.

Throughout, l refers to time step index; 0, 1, 2,…… $L-1$, where L is the total time step index used in the forward model.

$$V_{\phi,i,j,k}^{l+1} = \frac{V_b\, \phi_{i,j,k}}{C_2 \Delta t^l} = \frac{\Delta x_i \Delta y_j \Delta z_k \phi_{i,j,k}}{C_2 \Delta t^l} \qquad \ldots\ldots\ldots\ldots(3.7)$$

From this point on, we will assume that there are M simulator grid blocks that is ordered from 1,2,3,…,M and then

$V_{\phi,i,j,k}^{l+1}$ will be written as, $V_{\phi}^{l+1} = \begin{bmatrix} V_{\phi,1}^{l+1} & 0 & 0 & & 0 \\ 0 & V_{\phi,2}^{l+1} & & & \\ 0 & 0 & V_{\phi,3}^{l+1} & & \\ & & & & \\ 0 & 0 & 0 & & V_{\phi,M}^{l+1} \end{bmatrix}_{M \times M}$ …………(3.8)

Pressure will be written in the vector form as: $p^l = \begin{bmatrix} p_1^l & p_2^l & & p_M^l \end{bmatrix}^T$ ………..(3.9)

Saturation will be also written as: $S_m^l = \begin{bmatrix} S_{m1}^l & S_{m2}^l & & S_{mM}^l \end{bmatrix}^T$ ………..(3.10)

The reciprocal of the formation factor is also written in the vector form as:

$b_m^l = \begin{bmatrix} b_{m1}^l & b_{m2}^l & & b_{mM}^l \end{bmatrix}^T$ …………(3.11)

The rate also can be written as: $q_m^l = \begin{bmatrix} q_{m1}^l & q_{m2}^l & & q_{mM}^l \end{bmatrix}^T$ …………(3.12)

However, q_m vector is sparse as small numbers of grids, which are the well grid blocks are only have non zero q_m

Similarly porosity, ϕ, permeability, K_x, K_y, K_z will be written in the vector as shown in **Eqs. 3.9 – 3.12**.

Accordingly, **Eq. 3.6** can be written in the following form:

$F_m^{l+1}(P^{l+1}, S_w^{l+1}, K_x, K_y, K_z) = A_m^{l+1} - A_m^l$ …………(3.13)

Where,

$A_m^{l+1}(P^{l+1}, S_w^{l+1}, \phi) = V_\phi^{l+1}(b_m S_m)^{l+1}$ …………(3.14)

$A_m^l(P^l, S_w^l, \phi) = V_\phi^{l+1}(b_m S_m)^l$ …………(3.15)

F_m is independent of porosity and depends on permeability, while A_m^{l+1} and A_m^l are porosity-dependent and permeability-independent. Accordingly the following equations hold:

$$\nabla_\phi \left[(F_m^{l+1})^T \right] = O \qquad \qquad \text{..........(3.16)}$$

$$\nabla_{K_x} \left[(A_m^{l+1})^T \right] = \nabla_{K_y} \left[(A_m^{l+1})^T \right] = \nabla_{K_z} \left[(A_m^{l+1})^T \right] = O \qquad \qquad \text{..........(3.17)}$$

$$\nabla_{K_x} \left[(A_m^{l})^T \right] = \nabla_{K_y} \left[(A_m^{l})^T \right] = \nabla_{K_z} \left[(A_m^{l})^T \right] = O \qquad \qquad \text{..........(3.18)}$$

O in **Eqs. 3.16 -3.18** is a null M x M matrix.

Our objective is to get the sensitivity of the generalized travel time shift, $\Delta \tilde{t}$, at each well, let us denoted now by $g(p^1,....., p^l, S_w^1,........, S_w^l, K_x, K_y, K_z, \phi)$ subject to the set of constraint equations in the form of finite difference equations given by **Eq. 3.13** at each grid block for each time step. In other words, we want to see how the perturbation in K_x, K_y, K_z, and ϕ at each grid block will affect the generalized travel time at each well. So, a two M–dimensional vectors of Lagrange multipliers are used to adjoin g with the finite difference equations to form the augmented objective function J, where M is the number of grid blocks, as follows:

$$\lambda_o^l = \left[\lambda_{o1}^l \ \lambda_{o2}^l \qquad \qquad \lambda_{oM}^l \right]^T \qquad \qquad \text{..........(3.19)}$$

$$\lambda_w^l = \left[\lambda_{w1}^l \ \lambda_{w2}^l \qquad \qquad \lambda_{wM}^l \right]^T \qquad \qquad \text{..........(3.20)}$$

Thus the equation after adjoining the g function with the constraint equations will be:

$$J = g + \sum_{m=o,w} \sum_{l=0}^{L-1} (\lambda_m^{l+1})^T \left[F_m^{l+1} - A_m^{l+1} + A_m^l \right] \qquad \ldots\ldots\ldots(3.21)$$

From **Eq. 3.21** and **Eq. 3.13**, it is clear that J and g are equivalent. Thus, the partial derivative of J with respect to the model parameter is equivalent to the partial derivative of g. The g function in our case is the generalized travel time shift for each well at the well location, but to keep the formulation general the adjoint equation will be formulated for any arbitrary function g, which is at certain location in reservoir.

By taking the total differentiation of **Eq. 3.21** with respect to the state variables, pressure and water saturation and with respect to the control variables, K_x, K_y, K_z, and ϕ.

$$dJ = dg + \sum_{m=o,w} \sum_{l=1}^{L-1} (\lambda_m^l)^T \left[\left(\nabla_{p^l} \left[F_m^l - A_m^l \right]^T \right)^T dp^l + \left(\nabla_{S_w^l} \left[F_m^l - A_m^l \right]^T \right)^T dS_w^l \right]$$
$$+ \sum_{m=o,w} \sum_{l=0}^{L-1} \left((\lambda_m^{l+1})^T \left[\begin{array}{c} \left(\nabla_{k_x} \left[F_m^{l+1} \right]^T \right)^T dK_x + \left(\nabla_{k_y} \left[F_m^{l+1} \right]^T \right)^T dK_y + \left(\nabla_{k_z} \left[F_m^{l+1} \right]^T \right)^T dK_z - \\ \left(\nabla_{\phi} \left[A_m^{l+1} \right]^T \right)^T d\phi + \left(\nabla_{\phi} \left[A_m^l \right]^T \right)^T d\phi + \left(\nabla_{p^l} \left[A_m^l \right]^T \right)^T dp^l + \left(\nabla_{S_w^l} \left[A_m^l \right]^T \right)^T dS_w^l \end{array} \right] \right)$$
$$+ BT^L$$

$$\ldots\ldots\ldots(3.22)$$

Where, BT^L is calculated from the following equation:

$$BT^L = \sum_{m=o,w} (\lambda_m^L)^T \left[\left(\nabla_{p^L} \left[F_m^L - A_m^L \right]^T \right)^T dp^L + \left(\nabla_{S_w^L} \left[F_m^L - A_m^L \right]^T \right)^T dS_w^L \right] \qquad \ldots\ldots\ldots(3.23)$$

Since initially P, S_w are always known and the variation in the control variables **m** (K_x, K_y, K_z, and ϕ) will not have any effect on the initial pressure and water saturation at each grid block, thus,

$$dp^0 = dS_w^0 = 0 \qquad\qquad\qquad\qquad \text{……..…(3.24)}$$

Thus, any term multiplied by dp^0 and dS_w^0 had to be neglected from **Eq. 3.22**. Similarly, by taking the total differentiation of g as it is also depend on state and control variables, thus the total differentiation of g will be:

$$dg = \left\{ \sum_{l=1}^{L-1} \left(\left[\nabla_{p^l} g \right]^T dp^l + \left[\nabla_{S_w^l} g \right]^T dS_w^l \right) \right\} + \left[\nabla_{K_x} g \right]^T dK_x + \left[\nabla_{K_y} g \right]^T dK_y + \left[\nabla_{K_z} g \right]^T dK_z + \left[\nabla_\phi g \right]^T d\phi$$

$$\text{...………..(3.25)}$$

Substituting **Eq. 3.25** in **Eq. 3.22**,

$$
dJ = \sum_{l=1}^{L-1} \left\{
\begin{aligned}
&\left(\left[\sum_{m=o,w} \left(\lambda_m^l\right)^T \left(\nabla_{p^l} \left[F_m^l - A_m^l \right]^T \right)^T \right] + \left(\nabla_{p^l} g\right)^T + \left[\sum_{m=o,w} \left(\lambda_m^{l+1}\right)^T \left(\nabla_{p^l} \left[A_m^l \right]^T \right)^T \right] \right) dp^l + \\
&\left(\left[\sum_{m=o,w} \left(\lambda_m^l\right)^T \left(\nabla_{S_w^l} \left[F_m^l - A_m^l \right]^T \right)^T \right] + \left(\nabla_{S_w^l} g\right)^T + \left[\sum_{m=o,w} \left(\lambda_m^{l+1}\right)^T \left(\nabla_{S_w^l} \left[A_m^l \right]^T \right)^T \right] \right) dS_w^l
\end{aligned}
\right\} +
$$

$$
\left\{ \left(\sum_{m=o,w} \sum_{l=0}^{L-1} \left(\lambda_m^{l+1}\right)^T \left(\nabla_{K_x} \left[F_m^{l+1} \right]^T \right)^T \right) + \left(\nabla_{K_x} g\right)^T \right\} dK_x + \left\{ \left(\sum_{m=o,w} \sum_{l=0}^{L-1} \left(\lambda_m^{l+1}\right)^T \left(\nabla_{K_y} \left[F_m^{l+1} \right]^T \right)^T \right) + \left(\nabla_{K_y} g\right)^T \right\} dK_y +
$$

$$
\left\{ \left(\sum_{m=o,w} \sum_{l=0}^{L-1} \left(\lambda_m^{l+1}\right)^T \left(\nabla_{K_z} \left[F_m^{l+1} \right]^T \right)^T \right) + \left(\nabla_{K_z} g\right)^T \right\} dK_z +
$$

$$
\left\{ \left(\sum_{m=o,w} \sum_{l=0}^{L-1} \left(\lambda_m^{l+1}\right)^T \left[\left(\nabla_\phi \left[A_m^l \right]^T \right)^T - \left(\nabla_\phi \left[A_m^{l+1} \right]^T \right)^T \right] \right) + \left(\nabla_\phi g\right)^T \right\} d\phi + BT^L
$$

$$\text{……….(3.26)}$$

Eq. 3.26 is very important to understand it physically. It shows how the change in the control variables **m** (K_x, K_y, K_z, and ϕ) at each grid block affect the pressure and water saturation at each time step in each grid block. This in turn affect the augmented objective function (J) which is equivalent to the change in g (generalized travel time at each well) as seen from **Eq. 3.21**.

Since, the objective is to get the sensitivity of (g) with respect to the control variables, which are the permeability and porosity, thus to remove the dependency of J from pressure and water saturation, we choose the adjoint variables to insure that the coefficients of dp^l and dS_w^l in **Eq. 3.26** vanish. This will get the following adjoint system of equations.

$$\sum_{m=o,w} \left[\left(\nabla_{p^l} \left[F_m^l - A_m^l \right]^T \right) \lambda_m^l + \nabla_{p^l} \left[A_m^l \right]^T \lambda_m^{l+1} \right] = -\nabla_{p^l} g \qquad \ldots\ldots\ldots(3.27)$$

$$\sum_{m=o,w} \left[\left(\nabla_{S_w^l} \left[F_m^l - A_m^l \right]^T \right) \lambda_m^l + \nabla_{S_w^l} \left[A_m^l \right]^T \lambda_m^{l+1} \right] = -\nabla_{S_w^l} g \qquad \ldots\ldots\ldots(3.28)$$

Also, to remove the dependency of the change of pressure and water saturation at the end of simulation, dp^L and dS_w^L due to the change of the control variables, we set

$$\lambda_w^L = \lambda_o^L = 0 \qquad \ldots\ldots\ldots(3.29)$$

This will be the initial condition to solve the adjoint system of equation, **Eqs. 3.27 and 3.28** backward in time, to get the Lagrange multipliers, λ_m at each grid block per each time step. The right hand side of **Eq. 3.29** is M-dimensional column vector of zero values.

Thus, from **Eq. 3.23** and **Eq. 3.29**,

$$BT^L = 0 \qquad \ldots\ldots\ldots(3.30)$$

From **Eqs. 3.27, 3.28,** and **3.30**, the change in the augmented function J in **Eq. 3.26**, will be:

$$dJ = \left\{ \left(\sum_{m=o,w} \sum_{l=0}^{L-1} \left(\lambda_m^{l+1}\right)^T \left(\nabla_{K_x}\left[F_m^{l+1}\right]^T\right)^T \right) + \left(\nabla_{K_x} g\right)^T \right\} dK_x +$$

$$\left\{ \left(\sum_{m=o,w} \sum_{l=0}^{L-1} \left(\lambda_m^{l+1}\right)^T \left(\nabla_{K_y}\left[F_m^{l+1}\right]^T\right)^T \right) + \left(\nabla_{K_y} g\right)^T \right\} dK_y +$$

$$\left\{ \left(\sum_{m=o,w} \sum_{l=0}^{L-1} \left(\lambda_m^{l+1}\right)^T \left(\nabla_{K_z}\left[F_m^{l+1}\right]^T\right)^T \right) + \left(\nabla_{K_z} g\right)^T \right\} dK_z +$$

$$\left\{ \left(\sum_{m=o,w} \sum_{l=0}^{L-1} \left(\lambda_m^{l+1}\right)^T \left[\left(\nabla_\phi\left[A_m^l\right]^T\right)^T - \left(\nabla_\phi\left[A_m^{l+1}\right]^T\right)^T\right] \right) + \left(\nabla_\phi g\right)^T \right\} d\phi \qquad \ldots\ldots\ldots(3.31)$$

In our work, the (g) function is the generalized travel time shift at the well j, and according to the formulation given in the previous chapter assuming we are shifting the calculated towards the observed; the generalized travel time shift at well j is given by:

$$g = \Delta\tilde{t}_j \cong \frac{1}{n_{dj}} \sum_{i=1}^{n_{dj}} \left(t_{obs,j}^i - t_{cal,j}^i\right) \qquad \ldots\ldots\ldots(3.32)$$

Where, n_{dj} are the number of data points for well j, (i) is the index for the data point at time t^i. The gradient of the scalar function g in the adjoint system of equations, **Eqs. 3.27** and **3.28** are given by taking the gradient of **Eq. 3.32** as follows:

$$\nabla_{p^l} g = \nabla_{p^l} \Delta\tilde{t}_j = \left[\frac{\partial\Delta\tilde{t}_j}{\partial P_1^l} \quad \frac{\partial\Delta\tilde{t}_j}{\partial P_2^l} \quad \frac{\partial\Delta\tilde{t}_j}{\partial P_M^l}\right]^T \qquad \ldots\ldots\ldots(3.33)$$

$$\nabla_{S_w^l} g = \nabla_{S_w^l} \Delta\tilde{t}_j = \left[\frac{\partial\Delta\tilde{t}_j}{\partial S_{w1}^l} \quad \frac{\partial\Delta\tilde{t}_j}{\partial S_{w2}^l} \quad \frac{\partial\Delta\tilde{t}_j}{\partial S_{wM}^l}\right]^T \qquad \ldots\ldots\ldots(3.34)$$

It should be noted that **Eqs. 3.33** and **3.34** are vectors of non zero elements at grid blocks containing producing wells only. At grid blocks of producing wells, the partial

derivatives of the generalized travel time with respect to pressure and water saturation are given as follows:

$$\frac{\partial \Delta \widetilde{t}_j}{\partial P^l} \cong \frac{1}{n_{dj}} \cdot \frac{\partial}{\partial P^l} \left[\sum_{i=1}^{n_{dj}} \left(t_{obs,j}^i - t_{cal,j}^i \right) \right] \cong -\frac{1}{n_{dj}} \frac{\partial t_{cal,j}^i}{\partial p^l} \qquad \text{.........(3.35)}$$

$$\frac{\partial \Delta \widetilde{t}_j}{\partial S_w^l} \cong \frac{1}{n_{dj}} \cdot \frac{\partial}{\partial S_w^l} \left[\sum_{i=1}^{n_{dj}} \left(t_{obs,j}^i - t_{cal,j}^i \right) \right] \cong -\frac{1}{n_{dj}} \frac{\partial t_{cal,j}^i}{\partial S_w^l} \qquad \text{...........(3.36)}$$

Where at time step index, l, that is corresponding to the same observed time, t^i, the derivatives of **Eqs. 3.35 and 3.36** exist, otherwise, the derivative will vanish. So, strictly speaking in solving the adjoint system of equations backward in time the vectors of the source term or the right hand side of the adjoint system of equations will be null vectors except at the time corresponding to observed data point time, where the non zero elements in those vectors will be corresponding to the grid blocks containing producing wells only.

The derivatives in the right hand side of **Eqs. 3.35** and **3.36** are given as follows:

$$-\frac{1}{n_{dj}} \frac{\partial t_{cal,j}^i}{\partial p^l} = -\frac{1}{n_{dj}} \cdot \frac{\partial t_{cal,j}^i}{\partial K} \cdot \frac{\partial K}{\partial f_{wcal,j}^i} \cdot \frac{\partial f_{wcal,j}^i}{\partial p^l} \qquad \text{.........(3.37)}$$

The change in permeability has opposite effect with respect to travel time and water cut; the increase in permeability leads to a decrease in travel time and increase in water cut, thus:

$$\frac{\partial t_{cal,j}^i}{\partial K} \cdot \frac{\partial K}{\partial f_{wcal,j}^i} = -\frac{1}{\partial f_{wcal,j}^i / \partial t_{cal,j}^i} = -\frac{1}{\left(f_{wcal,j}^l - f_{wcal,j}^{l-1} \right) / \left(t_{cal,j}^l - t_{cal,j}^{l-1} \right)} \qquad \text{.........(3.37a)}$$

Substituting **Eq. 3.37a** in **Eq. 3.37**

$$-\frac{1}{n_{dj}}\frac{\partial t^i_{cal,j}}{\partial p^l} = \frac{1}{n_{dj}}\cdot\frac{1}{(f^l_{wcal,j}-f^{l-1}_{wcal,j})\big/(t^l_{cal,j}-t^{l-1}_{cal,j})}\cdot\frac{\partial f^i_{wcal,j}}{\partial p^l} \qquad\ldots\ldots\ldots(3.37b)$$

Similarly, the derivative with respect to water saturation is as follows:

$$-\frac{1}{n_{dj}}\frac{\partial t^i_{cal,j}}{\partial S^l_w} = \frac{1}{n_{dj}}\cdot\frac{1}{(f^l_{wcal,j}-f^{l-1}_{wcal,j})\big/(t^l_{cal,j}-t^{l-1}_{cal,j})}\cdot\frac{\partial f^i_{wcal,j}}{\partial S^l_w} \qquad\ldots\ldots\ldots(3.38)$$

Where, l in **Eqs. 3.37b** and **3.38** is corresponding to time, t^i at the observed point (i) as mentioned before, while $l-1$ is corresponding to time t^{l-1} which is ($t^i - \Delta t$), where Δt is the time step used in the simulation. This is a backward finite difference approximation to get numerically the derivative of water cut with respect to time at time t^i assuming small Δt used during simulation. It is important to mention that in general $t^i - \Delta t$ is not equal to t^{i-1}, which is the time for the next observed data point.

The partial derivative of water cut with respect to pressure and water saturation is obtained as follows:

$$\frac{\partial f^i_{wcal,j}}{\partial p^l} = \frac{\partial}{\partial p^l}\left[\frac{\sum_{k=1}^{K}\left[\frac{K^i_{rw}}{\mu_w B^l_w}\right]_{k,j}\cdot WI_{k,j}\cdot\left(P^i_{j,k}-P^i_{wf,j}\right)}{\sum_{k=1}^{K}\left[\frac{K^i_{rw}}{\mu^i_w B^i_w}\right]_{k,j}\cdot WI_{k,j}\cdot\left(P^i_{j,k}-P^i_{wf,j}\right)+\sum_{k=1}^{K}\left[\frac{K^i_{ro}}{\mu^i_o B^i_o}\right]_{k,j}\cdot WI_{k,j}\cdot\left(P^i_{j,k}-P^i_{wf,j}\right)}\right]$$

$$\ldots\ldots\ldots(3.39)$$

$$\frac{\partial f_{wcal,j}^{i}}{\partial S_{w}^{l}} = \frac{\partial}{\partial S_{w}^{l}}\left[\frac{\sum_{k=1}^{K}\left[\frac{K_{rw}^{i}}{\mu_{w}B_{w}^{l}}\right]_{k,j}\cdot WI_{k,j}\cdot\left(P_{j,k}^{l}-P_{wf,j}^{l}\right)}{\sum_{k=1}^{K}\left[\frac{K_{rw}^{i}}{\mu_{w}^{i}B_{w}^{i}}\right]_{k,j}\cdot WI_{k,j}\cdot\left(P_{j,k}^{i}-P_{wf,j}^{i}\right)+\sum_{k=1}^{K}\left[\frac{K_{ro}^{i}}{\mu_{o}^{i}B_{o}^{i}}\right]_{k,j}\cdot WI_{k,j}\cdot\left(P_{j,k}^{i}-P_{wf,j}^{i}\right)}\right]$$

.........(3.40)

Where, K is the total number of layers opened for production for well (j), $WI_{k,j}$ is the well index for layer k at well j, which is independent of pressure and water saturation, K_{rw}^{i}, K_{ro}^{i} are the relative permeability to water and oil at the grid block of well (j) and at the observed point (i) corresponding to time t^{i}. B_{w}^{i}, B_{o}^{i} are the water and oil formation volume factors at well j grid block and at the observed point (i) corresponding to time t^{i}. μ_{w}^{i}, μ_{o}^{i} are the water and oil viscosity at well j grid block and at the observed time t^{i}. $P_{j,k}^{i}$ is the pressure at the well j grid block and at time t^{i}. $P_{wf,j}^{i}$ is the bottom hole pressure at well j grid block and at time t^{i} and it is assumed constant through the whole perforation intervals by neglecting the friction loss of the tubing across the perforation intervals. **Appendix B** shows a detailed derivative of **Eqs. 3.39** and **3.40** with respect to pressure and water saturation at each individual grid block opened for production.

It is important to mention that at each time step the adjoint system of equations, **Eqs. 3.27 and 3.28** are solved only n_w times with the same matrix but with different right hand side. While in case of the conventional amplitude inversion the adjoint systems have to be solved n_d (number of data points) times. This reduces the computational difficulties especially in large field-scale problems.

3.1.2.1 Adjoint System Formulation in i,j,k Notation
Till this point the adjoint system of equations are obtained in the form of M simulator grid blocks arranged from 1,2,3,.. M. The objective of this section to obtain the adjoint system of equations given by **Eqs. 3.27** and **3.28** in a conventional form of i,j,k notation.

We will show the procedures for changing **Eq. 3.27** into the i,j,k notation and similarly the same procedures will be for **Eq. 3.28**. **Eq. 3.27** can be written in the following form:

$$\nabla_{p^l}\left[f_o^l\right]^T \lambda_o^l - \nabla_{p^l}\left[q_o^l\right]^T \lambda_o^l - \nabla_{p^l}\left[A_o^l\right]^T \lambda_o^l + \nabla_{p^l}\left[f_w^l\right]^T \lambda_w^l - \nabla_{p^l}\left[q_w^l\right]^T \lambda_w^l$$
$$-\nabla_{p^l}\left[A_w^l\right]^T \lambda_w^l + \nabla_{p^l}\left[A_o^l\right]^T \lambda_o^{l+1} + \nabla_{p^l}\left[A_w^l\right]^T \lambda_w^{l+1} = -\nabla_{p^l} g$$

$$\text{.........(3.41)}$$

Where, from **Eqs. 3.4**, **3.5**, **3.6**, and **3.7**, f_o^l, f_w^l are given as follows:

$$
\begin{aligned}
f_{o,i,j,k}^l &= T_{ox,i+1/2,j,k}^l (p_{i+1,j,k}^l - p_{i,j,k}^l - \gamma_{oi+1/2,j,k}^l (D_{i+1,j,k} - D_{i,j,k})) \\
&- T_{ox,i-1/2,j,k}^l (p_{i,j,k}^l - p_{i-1,j,k}^l - \gamma_{oi-1/2,j,k}^l (D_{i,j,k} - D_{i-1,j,k})) \\
&+ T_{oy,i,j+1/2,k}^l (p_{i,j+1,k}^l - p_{i,j,k}^l - \gamma_{oi,j+1/2,k}^l (D_{i,j+1,k} - D_{i,j,k})) \\
&- T_{oy,i,j-1/2,k}^l (p_{i,j,k}^l - p_{i,j-1,k}^l - \gamma_{oi,j-1/2,k}^l (D_{i,j,k} - D_{i,j-1,k})) \\
&+ T_{oz,i,j,k+1/2}^l (p_{i,j,k+1}^l - p_{i,j,k}^l - \gamma_{oi,j,k+1/2}^l (D_{i,j,k+1} - D_{i,j,k})) \\
&- T_{oz,i,j,k-1/2}^l (p_{i,j,k}^l - p_{i,j,k-1}^l - \gamma_{oi,j,k-1/2}^l (D_{i,j,k} - D_{i,j,k-1}))
\end{aligned}
$$

$$\text{...........(3.42)}$$

$$
\begin{aligned}
f_{w,i,j,k}^l &= T_{wx,i+1/2,j,k}^l (p_{i+1,j,k}^l - p_{i,j,k}^l - \gamma_{wi+1/2,j,k}^l (D_{i+1,j,k} - D_{i,j,k})) \\
&- T_{wx,i-1/2,j,k}^l (p_{i,j,k}^l - p_{i-1,j,k}^l - \gamma_{wi-1/2,j,k}^l (D_{i,j,k} - D_{i-1,j,k})) \\
&+ T_{wy,i,j+1/2,k}^l (p_{i,j+1,k}^l - p_{i,j,k}^l - \gamma_{wi,j+1/2,k}^l (D_{i,j+1,k} - D_{i,j,k})) \\
&- T_{wy,i,j-1/2,k}^l (p_{i,j,k}^l - p_{i,j-1,k}^l - \gamma_{wi,j-1/2,k}^l (D_{i,j,k} - D_{i,j-1,k})) \\
&+ T_{wz,i,j,k+1/2}^l (p_{i,j,k+1}^l - p_{i,j,k}^l - \gamma_{wi,j,k+1/2}^l (D_{i,j,k+1} - D_{i,j,k})) \\
&- T_{wz,i,j,k-1/2}^l (p_{i,j,k}^l - p_{i,j,k-1}^l - \gamma_{wi,j,k-1/2}^l (D_{i,j,k} - D_{i,j,k-1}))
\end{aligned}
$$

$$\text{..........(3.43)}$$

The first term in **Eq. 3.41** is given in matrix form as follows:

$$
\nabla_{p^l}\left[f_o^l\right]^T =
\begin{bmatrix}
\dfrac{\partial f_{o,1}^l}{\partial p_1^l} & \dfrac{\partial f_{o,2}^l}{\partial p_1^l} & & \dfrac{\partial f_{o,M}^l}{\partial p_1^l} \\[2ex]
\dfrac{\partial f_{o,1}^l}{\partial p_2^l} & \dfrac{\partial f_{o,2}^l}{\partial p_2^l} & & \dfrac{\partial f_{o,M}^l}{\partial p_2^l} \\[4ex]
\dfrac{\partial f_{o,1}^l}{\partial p_M^l} & \dfrac{\partial f_{o,2}^l}{\partial p_M^l} & & \dfrac{\partial f_{o,M}^l}{\partial p_M^l}
\end{bmatrix}_{M\times M}
\qquad \ldots\ldots(3.44)
$$

Similarly, we can write $\nabla_{p^l}\left[f_w^l\right]^T$, and $\nabla_{p^l}\left[q_m^l\right]^T$, $\nabla_{p^l}\left[A_m^l\right]^T$ for m = o, w in the same form as given by **Eq. 3.44.** The source term of **Eq. 3.41**, $\nabla_{p^l}g$, which is a vector is given before in **Eq. 3.33.** By substituting **Eq. 3.44** and the gradients of the other terms of **Eq. 3. 41** and writing the rth row equation, the resultant equation will be:

$$
\sum_{s=1}^{M}\frac{\partial f_{o,s}^l}{\partial p_r^l}\cdot\lambda_{o,s}^l - \sum_{s=1}^{M}\frac{\partial q_{o,s}^l}{\partial p_r^l}\cdot\lambda_{o,s}^l - \sum_{s=1}^{M}\frac{\partial A_{o,s}^l}{\partial p_r^l}\cdot\lambda_{o,s}^l + \sum_{s=1}^{M}\frac{\partial f_{w,s}^l}{\partial p_r^l}\cdot\lambda_{w,s}^l - \sum_{s=1}^{M}\frac{\partial q_{w,s}^l}{\partial p_r^l}\cdot\lambda_{w,s}^l
$$
$$
-\sum_{s=1}^{M}\frac{\partial A_{w,s}^l}{\partial p_r^l}\cdot\lambda_{w,s}^l + \sum_{s=1}^{M}\frac{\partial A_{o,s}^l}{\partial p_r^l}\cdot\lambda_{o,s}^{l+1} + \sum_{s=1}^{M}\frac{\partial A_{w,s}^l}{\partial p_r^l}\cdot\lambda_{w,s}^{l+1} = -\frac{\partial g_m^i}{\partial p_r^l}
\qquad \ldots\ldots(3.45)
$$

For (r) corresponds to grid block i,j,k , so from **Eq. 3.42** and **3.43** , the only terms of $f_{m,s}^l$ in the sums of **Eq. 3.45** that depend on $p_{i,j,k}^l$ are $f_{m,i-1,j,k}^l$, $f_{m,i,j-1,k}^l$, $f_{m,i,j,k-1}^l$, $f_{m,i,j,k}^l$, $f_{m,i+1,j,k}^l$, $f_{m,i,j+1,k}^l$, $f_{m,i,j,k+1}^l$. Also for $q_{m,s}^l$ the terms in the sums that depend on $p_{i,j,k}^l$ is only $q_{m,i,j,k}^l$ and for $A_{m,s}^l$ the only term that depends on $p_{i,j,k}^l$ is $A_{m,i,j,k}^l$ as seen from **Eqs. 3.14, 3.15, and 3.7**. For the source term of the adjoint equation, as mentioned before the only term that is non zero is the term that at the same time corresponding to the time step index l and at $m = i,j,k$.

Thus **Eq. 3.45** can be written in the i,j,k notation as follows:

$$\frac{\partial f_{w,i,j,k-1}^{l}}{\partial p_{i,j,k}^{l}} \cdot \lambda_{w,i,j,k-1}^{l} + \frac{\partial f_{o,i,j,k-1}^{l}}{\partial p_{i,j,k}^{l}} \cdot \lambda_{o,i,j,k-1}^{l} + \frac{\partial f_{w,i,j-1,k}^{l}}{\partial p_{i,j,k}^{l}} \cdot \lambda_{w,i,j-1,k}^{l} + \frac{\partial f_{o,i,j-1,k}^{l}}{\partial p_{i,j,k}^{l}} \cdot \lambda_{o,i,j-1,k}^{l}$$

$$+ \frac{\partial f_{w,i-1,j,k}^{l}}{\partial p_{i,j,k}^{l}} \cdot \lambda_{w,i-1,j,k}^{l} + \frac{\partial f_{o,i-1,j,k}^{l}}{\partial p_{i,j,k}^{l}} \cdot \lambda_{o,i-1,j,k}^{l} + \left[\frac{\partial f_{w,i,j,k}^{l}}{\partial p_{i,j,k}^{l}} - \frac{\partial q_{w,i,j,k}^{l}}{\partial p_{i,j,k}^{l}} - \frac{\partial A_{w,i,j,k}^{l}}{\partial p_{i,j,k}^{l}} \right] \cdot \lambda_{w,i,j,k}^{l}$$

$$+ \left[\frac{\partial f_{o,i,j,k}^{l}}{\partial p_{i,j,k}^{l}} - \frac{\partial q_{o,i,j,k}^{l}}{\partial p_{i,j,k}^{l}} - \frac{\partial A_{o,i,j,k}^{l}}{\partial p_{i,j,k}^{l}} \right] \cdot \lambda_{o,i,j,k}^{l} + \frac{\partial f_{w,i+1,j,k}^{l}}{\partial p_{i,j,k}^{l}} \cdot \lambda_{w,i+1,j,k}^{l} + \frac{\partial f_{o,i+1,j,k}^{l}}{\partial p_{i,j,k}^{l}} \cdot \lambda_{o,i+1,j,k}^{l} \quad \text{....(3.46)}$$

$$+ \frac{\partial f_{w,i,j+1,k}^{l}}{\partial p_{i,j,k}^{l}} \cdot \lambda_{w,i,j+1,k}^{l} + \frac{\partial f_{o,i,j+1,k}^{l}}{\partial p_{i,j,k}^{l}} \cdot \lambda_{o,i,j+1,k}^{l} + \frac{\partial f_{w,i,j,k+1}^{l}}{\partial p_{i,j,k}^{l}} \cdot \lambda_{w,i,j,k+1}^{l} + \frac{\partial f_{o,i,j,k+1}^{l}}{\partial p_{i,j,k}^{l}} \cdot \lambda_{o,i,j,k+1}^{l}$$

$$= -\frac{\partial A_{w,i,j,k}^{l}}{\partial p_{i,j,k}^{l}} \cdot \lambda_{w,i,j,k}^{l+1} - \frac{\partial A_{o,i,j,k}^{l}}{\partial p_{i,j,k}^{l}} \cdot \lambda_{o,i,j,k}^{l+1} - \frac{\partial g_{i,j,k}^{l}}{\partial p_{i,j,k}^{l}}$$

Similarly, **Eq. 3.28** can be written in the i,j,k notation as follows:

$$\frac{\partial f_{w,i,j,k-1}^{l}}{\partial S_{wi,j,k}^{l}} \cdot \lambda_{w,i,j,k-1}^{l} + \frac{\partial f_{o,i,j,k-1}^{l}}{\partial S_{wi,j,k}^{l}} \cdot \lambda_{o,i,j,k-1}^{l} + \frac{\partial f_{w,i,j-1,k}^{l}}{\partial S_{wi,j,k}^{l}} \cdot \lambda_{w,i,j-1,k}^{l} + \frac{\partial f_{o,i,j-1,k}^{l}}{\partial S_{wi,j,k}^{l}} \cdot \lambda_{o,i,j-1,k}^{l}$$

$$+ \frac{\partial f_{w,i-1,j,k}^{l}}{\partial S_{wi,j,k}^{l}} \cdot \lambda_{w,i-1,j,k}^{l} + \frac{\partial f_{o,i-1,j,k}^{l}}{\partial S_{wi,j,k}^{l}} \cdot \lambda_{o,i-1,j,k}^{l} + \left[\frac{\partial f_{w,i,j,k}^{l}}{\partial S_{wi,j,k}^{l}} - \frac{\partial q_{w,i,j,k}^{l}}{\partial S_{wi,j,k}^{l}} - \frac{\partial A_{w,i,j,k}^{l}}{\partial S_{wi,j,k}^{l}} \right] \cdot \lambda_{w,i,j,k}^{l}$$

$$+ \left[\frac{\partial f_{o,i,j,k}^{l}}{\partial S_{wi,j,k}^{l}} - \frac{\partial q_{o,i,j,k}^{l}}{\partial S_{wi,j,k}^{l}} - \frac{\partial A_{o,i,j,k}^{l}}{\partial S_{wi,j,k}^{l}} \right] \cdot \lambda_{o,i,j,k}^{l} + \frac{\partial f_{w,i+1,j,k}^{l}}{\partial S_{wi,j,k}^{l}} \cdot \lambda_{w,i+1,j,k}^{l} + \frac{\partial f_{o,i+1,j,k}^{l}}{\partial S_{wi,j,k}^{l}} \cdot \lambda_{o,i+1,j,k}^{l} \quad \text{...(3.47)}$$

$$+ \frac{\partial f_{w,i,j+1,k}^{l}}{\partial S_{wi,j,k}^{l}} \cdot \lambda_{w,i,j+1,k}^{l} + \frac{\partial f_{o,i,j+1,k}^{l}}{\partial S_{wi,j,k}^{l}} \cdot \lambda_{o,i,j+1,k}^{l} + \frac{\partial f_{w,i,j,k+1}^{l}}{\partial S_{wi,j,k}^{l}} \cdot \lambda_{w,i,j,k+1}^{l} + \frac{\partial f_{o,i,j,k+1}^{l}}{\partial S_{wi,j,k}^{l}} \cdot \lambda_{o,i,j,k+1}^{l}$$

$$= -\frac{\partial A_{w,i,j,k}^{l}}{\partial S_{wi,j,k}^{l}} \cdot \lambda_{w,i,j,k}^{l+1} - \frac{\partial A_{o,i,j,k}^{l}}{\partial S_{wi,j,k}^{l}} \cdot \lambda_{o,i,j,k}^{l+1} - \frac{\partial g_{i,j,k}^{l}}{\partial S_{wi,j,k}^{l}}$$

The derivative of the flow terms, $f_{m,i-1,j,k}^{l}$, $f_{m,i,j-1,k}^{l}$, $f_{m,i,j,k-1}^{l}$, $f_{m,i,j,k}^{l}$, $f_{m,i+1,j,k}^{l}$, $f_{m,i,j+1,k}^{l}$, $f_{m,i,j,k+1}^{l}$, the accumulation terms, $A_{m,i,j,k}^{l}$, the source/sink terms, $q_{m,i,j,k}^{l}$, and the source term of the adjoint system of equations, $g_{i,j,k}^{l}$, with respect to pressure and water saturation as required by **Eq. 3.46** and **3.47** is shown in **Appendix B.**

3.1.3 Sensitivity Coefficients Formulation

From the previous section, the adjoint variables λ_m, $m = o, w$ has been calculated for each grid block at each time step. This section gives the sensitivity coefficient calculation by knowing the adjoint variables calculated from the previous section.

By considering J in **Eq. 3.21**, as a function of K_x, K_y, K_z, and ϕ, the total differentiation of J will be:

$$dJ = \left[\nabla_{K_x} J\right]^T dK_x + \left[\nabla_{K_y} J\right]^T dK_y + \left[\nabla_{K_z} J\right]^T dK_z + \left[\nabla_\phi J\right]^T d\phi \qquad \text{.........(3.48)}$$

Comparing **Eq. 3.31** and **Eq. 3.48**, leads to the sensitivity coefficients calculation equations, which are:

$$\left(\nabla_{K_x} J\right)^T = \left(\sum_{m=o,w} \sum_{l=0}^{L-1} \left(\lambda_m^{l+1}\right)^T \left(\nabla_{K_x} \left[F_m^{l+1}\right]^T\right)^T\right) + \left(\nabla_{K_x} g\right)^T \qquad \text{..........(3.49)}$$

$$\left(\nabla_{K_y} J\right)^T = \left(\sum_{m=o,w} \sum_{l=0}^{L-1} \left(\lambda_m^{l+1}\right)^T \left(\nabla_{K_y} \left[F_m^{l+1}\right]^T\right)^T\right) + \left(\nabla_{K_y} g\right)^T \qquad \text{....……(3.50)}$$

$$\left(\nabla_{K_z} J\right)^T = \left(\sum_{m=o,w} \sum_{l=0}^{L-1} \left(\lambda_m^{l+1}\right)^T \left(\nabla_{K_z} \left[F_m^{l+1}\right]^T\right)^T\right) + \left(\nabla_{K_z} g\right)^T \qquad \text{..........(3.51)}$$

$$\left(\nabla_\phi J\right)^T = \left(\sum_{m=o,w} \sum_{l=0}^{L-1} \left(\lambda_m^{l+1}\right)^T \left[\left(\nabla_\phi \left[A_m^l\right]^T\right)^T - \left(\nabla_\phi \left[A_m^{l+1}\right]^T\right)^T\right]\right) + \left(\nabla_\phi g\right)^T \qquad \text{..........(3.52)}$$

In our work, the (g) function is the generalized travel time shift, which is given by **Eq. 3.32**. Since, $\Delta\tilde{t}$ is not an explicit function of permeability and porosity so the last term in **Eqs. 3.49 – 3.52** will vanish. As stated before from **Eq. 3.21** and **Eq. 3.13** that the (J) and (g) are equivalent, so $J = \Delta\tilde{t}_j$ ($j = 1,..n_w$).

By taking the transpose of **Eqs. 3.49 – 3.52**, the sensitivity of $\Delta\tilde{t}_j$ with respect to the model parameter will be:

$$\nabla_{K_x} \Delta \tilde{t}_j = \left(\sum_{m=o,w} \sum_{l=0}^{L-1} \nabla_{K_x} \left[F_m^{l+1} \right]^T \lambda_m^{l+1} \right) \qquad \ldots\ldots\ldots(3.53)$$

$$\nabla_{K_y} \Delta \tilde{t}_j = \left(\sum_{m=o,w} \sum_{l=0}^{L-1} \nabla_{K_y} \left[F_m^{l+1} \right]^T \lambda_m^{l+1} \right) \qquad \ldots\ldots\ldots(3.54)$$

$$\nabla_{K_z} \Delta \tilde{t}_j = \left(\sum_{m=o,w} \sum_{l=0}^{L-1} \nabla_{K_z} \left[F_m^{l+1} \right]^T \lambda_m^{l+1} \right) \qquad \ldots\ldots\ldots(3.55)$$

$$\nabla_{\phi} \Delta \tilde{t}_j = \left(\sum_{m=o,w} \sum_{l=0}^{L-1} \left[\nabla_{\phi} \left[A_m^l \right]^T - \nabla_{\phi} \left[A_m^{l+1} \right]^T \right] \lambda_m^{l+1} \right) \qquad \ldots\ldots\ldots(3.56)$$

It is important to mention that the adjoint system of equations given by **Eqs. 3.46** and **3.47** and the calculation of the sensitivity coefficients, given by **Eqs. 3.53 – 3.56** has to be performed n_w times to calculate the sensitivity matrix, G. This is a considerable savings in computation time compared to the conventional method using amplitude misfit, where the adjoint system of equations have to be solved n_d times which can be order of magnitude larger than the number of wells. The sensitivity matrix for the generalized travel time with respect to K_x as an example is given as follows:

$$G = \begin{bmatrix} \dfrac{\partial \Delta \tilde{t}_1}{\partial k_{x1}} & \dfrac{\partial \Delta \tilde{t}_1}{\partial k_{x2}} & & \dfrac{\partial \Delta \tilde{t}_1}{\partial k_{xM}} \\ \dfrac{\partial \Delta \tilde{t}_2}{\partial k_{x1}} & \dfrac{\partial \Delta \tilde{t}_2}{\partial k_{x2}} & & \dfrac{\partial \Delta \tilde{t}_2}{\partial k_{xM}} \\ & & & \\ \dfrac{\partial \Delta \tilde{t}_{n_w}}{\partial k_{x1}} & \dfrac{\partial \Delta \tilde{t}_{n_w}}{\partial k_{x2}} & & \dfrac{\partial \Delta \tilde{t}_{n_w}}{\partial k_{xM}} \end{bmatrix}_{n_w \times M} \qquad \ldots\ldots\ldots(3.57)$$

$\Delta \tilde{t}_1$, $\Delta \tilde{t}_2$, ... $\Delta \tilde{t}_{n_w}$ are the generalized travel time shift at well 1 ,2, ...n_w respectively.

3.1.3.1 Sensitivity Coefficients Formulation in i,j,k Notation

The sensitivity coefficients formulation given by **Eqs. 3.53 − 3.56** are obtained in the form of M simulator grid blocks arranged from 1,2,3,.. M. The objective of this section is to obtain the sensitivity coefficients in a conventional form of i,j,k notation.

We will show the procedures for changing **Eq. 3.53** into the i,j,k notation and similarly the same procedures will be applied for the rest of equations, **Eqs. 3.54 – 3.56**. **Eq. 3.53** is given as follows:

$$\nabla_{K_x} \Delta \tilde{t}_j = \sum_{l=0}^{L-1} \left[\nabla_{K_x} \left[f_w^{l+1} \right]^T \lambda_w^{l+1} - \nabla_{K_x} \left[q_w^{l+1} \right]^T \lambda_w^{l+1} + \nabla_{K_x} \left[f_o^{l+1} \right]^T \lambda_o^{l+1} - \nabla_{K_x} \left[q_o^{l+1} \right]^T \lambda_o^{l+1} \right] \quad \ldots\ldots..(3.58)$$

Where, f_o^{l+1}, f_w^{l+1} are similar to that given before in **Eqs. 3.42**, and **3.43**. The terms in **Eq. 3.58** are given as follows:

$$\nabla_{K_x} \Delta \tilde{t}_j = \left[\frac{\partial \Delta \tilde{t}_j}{\partial K_{x,1}} \quad \frac{\partial \Delta \tilde{t}_j}{\partial K_{x,2}} \qquad \frac{\partial \Delta \tilde{t}_j}{\partial K_{x,M}} \right]^T \quad \ldots\ldots..(3.59)$$

$$\nabla_{K_x} \left[f_m^{l+1} \right]^T = \begin{bmatrix} \dfrac{\partial f_{m,1}^{l+1}}{\partial K_{x,1}} & \dfrac{\partial f_{m,2}^{+1l}}{\partial K_{x,1}} & \dfrac{\partial f_{m,M}^{l+1}}{\partial K_{x,1}} \\[2ex] \dfrac{\partial f_{m,1}^{l+1}}{\partial K_{x,2}} & \dfrac{\partial f_{m,2}^{l+1}}{\partial K_{x,2}} & \dfrac{\partial f_{m,M}^{l+1}}{\partial K_{x,2}} \\[3ex] \dfrac{\partial f_{m,1}^{l+1}}{\partial K_{x,M}} & \dfrac{\partial f_{m,2}^{l+1}}{\partial K_{x,M}} & \dfrac{\partial f_{m,M}^{l+1}}{\partial K_{x,M}} \end{bmatrix}_{M \times M} , \quad m = o,w \qquad \ldots\ldots..(3.60)$$

$$\nabla_{K_x}\left[q_m^{l+1}\right]^T = \begin{bmatrix} \dfrac{\partial q_{m,1}^{l+1}}{\partial K_{x,1}} & \dfrac{\partial q_{m,2}^{+1l}}{\partial K_{x,1}} & \dfrac{\partial q_{m,M}^{l+1}}{\partial K_{x,1}} \\[2em] \dfrac{\partial q_{m,1}^{l+1}}{\partial K_{x,2}} & \dfrac{\partial q_{m,2}^{l+1}}{\partial K_{x,2}} & \dfrac{\partial q_{m,M}^{l+1}}{\partial K_{x,2}} \\[3em] \dfrac{\partial q_{m,1}^{l+1}}{\partial K_{x,M}} & \dfrac{\partial q_{m,2}^{l+1}}{\partial K_{x,M}} & \dfrac{\partial q_{m,M}^{l+1}}{\partial K_{x,M}} \end{bmatrix}_{M\times M} , \quad m = o,w \qquad \ldots\ldots\ldots(3.61)$$

Substituting **Eqs. 3.59 – 3.61** in **Eq. 3.58** and writing the rth row equation, the resulting equation will be:

$$\frac{\partial \Delta \tilde{t}_j}{\partial K_{x,r}} = \sum_{l=0}^{L-1}\left[\sum_{s=1}^{M}\frac{\partial f_{w,s}^{l+1}}{\partial K_{x,r}}\cdot\lambda_{w,s}^{l+1} - \sum_{s=1}^{M}\frac{\partial q_{w,s}^{l+1}}{\partial K_{x,r}}\cdot\lambda_{w,s}^{l+1} + \sum_{s=1}^{M}\frac{\partial f_{o,s}^{l+1}}{\partial K_{x,r}}\cdot\lambda_{o,s}^{l+1} - \sum_{s=1}^{M}\frac{\partial q_{o,s}^{l}}{\partial K_{x,r}}\cdot\lambda_{o,s}^{l}\right] \ldots\ldots(3.62)$$

For (r) corresponds to grid block i,j,k , so from **Eq. 3.42** and **3.43** , the only terms of $f_{m,s}^{l+1}$, $m = o,w$, in the sums of **Eq. 3.62** that depend on $K_{x,i,j,k}$ are $f_{m,i-1,j,k}^{l+1}$, $f_{m,i,j,k}^{l+1}$, $f_{m,i+1,j,k}^{l+1}$, which are in the transmissibility terms $T_{mx,i+1/2,j,k}^{l+1}$ and $T_{mx,i-1/2,j,k}^{l+1}$. Similarly, the terms that depend on $K_{y,i,j,k}$ are $f_{m,i,j-1,k}^{l+1}$, $f_{m,i,j,k}^{l+1}$, $f_{m,i,j+1,k}^{l+1}$, and those that depend on $K_{z,i,j,k}$ are $f_{m,i,j,k-1}^{l+1}$, $f_{m,i,j,k}^{l+1}$, $f_{m,i,j,k+1}^{l+1}$. Also for $q_{m,s}^{l+1}$ the terms in the sums that depend on $K_{x,i,j,k}$ is only $q_{m,i,j,k}^{l+1}$ and similarly for the sensitivity with respect to $K_{y,i,j,k}$ and $K_{z,i,j,k}$. Thus, **Eq. 3.62** will be as follows:

$$\frac{\partial \Delta \tilde{t}_j}{\partial K_{x,i,j,k}} = \sum_{l=0}^{L-1}\begin{bmatrix} \left(\dfrac{\partial f_{w,i,j,k}^{l+1}}{\partial K_{x,i,j,k}} - \dfrac{\partial q_{w,i,j,k}^{l+1}}{\partial K_{x,i,j,k}}\right)\cdot\lambda_{w,i,j,k}^{l+1} + \dfrac{\partial f_{w,i-1,j,k}^{l+1}}{\partial K_{x,i,j,k}}\cdot\lambda_{w,i-1,j,k}^{l+1} \\[2em] + \dfrac{\partial f_{w,i+1,j,k}^{l+1}}{\partial K_{x,i,j,k}}\cdot\lambda_{w,i+1,j,k}^{l+1} + \left(\dfrac{\partial f_{o,i,j,k}^{l+1}}{\partial K_{x,i,j,k}} - \dfrac{\partial q_{o,i,j,k}^{l+1}}{\partial K_{x,i,j,k}}\right)\cdot\lambda_{o,i,j,k}^{l+1} \\[2em] + \dfrac{\partial f_{o,i-1,j,k}^{l+1}}{\partial K_{x,i,j,k}}\cdot\lambda_{o,i-1,j,k}^{l+1} + \dfrac{\partial f_{o,i+1,j,k}^{l+1}}{\partial K_{x,i,j,k}}\cdot\lambda_{o,i+1,j,k}^{l+1} \end{bmatrix} \qquad \ldots\ldots(3.63)$$

As we are interested to get the sensitivity with respect to permeability only, the following two equations give the sensitivity formulation with respect to $K_{y,i,j,k}$ and $K_{z,i,j,k}$ as follows:

$$\frac{\partial \Delta \tilde{t}_j}{\partial K_{y,i,j,k}} = \sum_{l=0}^{L-1} \begin{bmatrix} \left(\dfrac{\partial f_{w,i,j,k}^{l+1}}{\partial K_{y,i,j,k}} - \dfrac{\partial q_{w,i,j,k}^{l+1}}{\partial K_{y,i,j,k}} \right) \cdot \lambda_{w,i,j,k}^{l+1} + \dfrac{\partial f_{w,i,j-1,k}^{l+1}}{\partial K_{y,i,j,k}} \cdot \lambda_{w,i,j-1,k}^{l+1} \\[3mm] + \dfrac{\partial f_{w,i,j+1,k}^{l+1}}{\partial K_{y,i,j,k}} \cdot \lambda_{w,i,j+1,k}^{l+1} + \left(\dfrac{\partial f_{o,i,j,k}^{l+1}}{\partial K_{y,i,j,k}} - \dfrac{\partial q_{o,i,j,k}^{l+1}}{\partial K_{y,i,j,k}} \right) \cdot \lambda_{o,i,j,k}^{l+1} \\[3mm] + \dfrac{\partial f_{o,i,j-1,k}^{l+1}}{\partial K_{y,i,j,k}} \cdot \lambda_{o,i,j-1,k}^{l+1} + \dfrac{\partial f_{o,i,j+1,k}^{l+1}}{\partial K_{y,i,j,k}} \cdot \lambda_{o,i,j+1,k}^{l+1} \end{bmatrix} \qquad \ldots\ldots\ldots(3.64)$$

$$\frac{\partial \Delta \tilde{t}_j}{\partial K_{z,i,j,k}} = \sum_{l=0}^{L-1} \begin{bmatrix} \left(\dfrac{\partial f_{w,i,j,k}^{l+1}}{\partial K_{z,i,j,k}} - \dfrac{\partial q_{w,i,j,k}^{l+1}}{\partial K_{z,i,j,k}} \right) \cdot \lambda_{w,i,j,k}^{l+1} + \dfrac{\partial f_{w,i,j,k-1}^{l+1}}{\partial K_{z,i,j,k}} \cdot \lambda_{w,i,j,k-1}^{l+1} \\[3mm] + \dfrac{\partial f_{w,i,j,k+1}^{l+1}}{\partial K_{z,i,j,k}} \cdot \lambda_{w,i,j,k+1}^{l+1} + \left(\dfrac{\partial f_{o,i,j,k}^{l+1}}{\partial K_{z,i,j,k}} - \dfrac{\partial q_{o,i,j,k}^{l+1}}{\partial K_{z,i,j,k}} \right) \cdot \lambda_{o,i,j,k}^{l+1} \\[3mm] + \dfrac{\partial f_{o,i,j,k-1}^{l+1}}{\partial K_{z,i,j,k}} \cdot \lambda_{o,i,j,k-1}^{l+1} + \dfrac{\partial f_{o,i,j,k+1}^{l+1}}{\partial K_{z,i,j,k}} \cdot \lambda_{o,i,j,k+1}^{l+1} \end{bmatrix} \qquad \ldots\ldots\ldots(3.65)$$

The derivative of the flow terms, $f_{m,i-1,j,k}^{l+1}, f_{m,i,j,k}^{l+1}, f_{m,i+1,j,k}^{l+1}$, with respect to $K_{x,i,j,k}$, the derivative of $f_{m,i,j-1,k}^{l+1}, f_{m,i,j,k}^{l+1}$, and $f_{m,i,j+1,k}^{l+1}$ with respect to $K_{y,i,j,k}$, the derivative of $f_{m,i,j,k-1}^{l+1}, f_{m,i,j,k}^{l+1}, f_{m,i,j,k+1}^{l+1}$ with respect to $K_{z,i,j,k}$, and the derivative of the source/sink terms, $q_{m,i,j,k}^{l+1}$ with respect to $K_{x,i,j,k}, K_{y,i,j,k}, K_{z,i,j,k}$ are given in **Appendix C.**

3.2 Streamline-Based Sensitivity

The streamline-based sensitivity calculation using finite difference as forward model follows the following steps:

i- Obtaining pressure, water saturation, and flux distribution from finite difference models:

The pressure, water saturation and flux distribution are obtained at time steps corresponding only to the pressure update times due to changing filed conditions, like infill drillings, changing well conditions, etc. We use a commercial finite-difference simulator (viz. ECLIPSE[53]) for modeling flow in the reservoir to calculate the pressure, water saturation, and the flux distribution. The two-phase black oil model used here is completely general and includes comprehensive physical mechanisms such as compressibility, gravity effects and other cross-streamline fluxes such as mobility effects, rate changes, infill drilling etc.

ii- Calculating the total velocity field

The total velocity field is calculated from the fluxes distribution obtained from the previous step.

iii- Tracing the streamline

The streamline tracing takes place by knowing the velocity and calculating the time of flight according to the algorithm given by Datta-Gupta and King[54].

iv- Sensitivity calculation

The generalized travel time sensitivity is calculated by knowing the velocity field and the streamlines traced along the grid blocks. A detailed discussion about the sensitivity computation is given in the next sections.

3.2.1 Generalized Travel Time Sensitivity without Pressure Update

The formulation for the generalized travel time shift is given in the previous chapter for two different cases; shifting the calculated towards the observed and the opposite. In this section, we will assume that we are shifting the calculated towards the observed, so the sensitivity of the generalized travel time shift for well j with respect to the model parameter, $\mathbf{m}$, is given as follows[11]:

$$\frac{\partial \Delta \tilde{t}_j}{\partial m} = -\frac{\sum_{i=1}^{n_{dj}} \partial t_{i,j}/\partial m}{n_{dj}} \qquad \ldots\ldots\ldots(3.66)$$

The sensitivity of travel time at producing well j can be obtained in terms of the streamline time of flight as discussed by He et al.[11],

Considering two-phase incompressible flow of oil and water in a non-deformable permeable medium. The transport equation can be written in the streamline time of flight coordinates as follows

$$\frac{\partial S_w}{\partial t} + \frac{\partial F_w}{\partial \tau} = 0 \qquad \ldots\ldots\ldots (3.67)$$

Rearranging **Eq. 3.67**,

$$\frac{\partial S_w}{\partial t} = -\frac{\partial F_w}{\partial S_w}\frac{\partial S_w}{\partial \tau} \qquad \ldots\ldots\ldots (3.68)$$

In the above expression, τ represents the streamline time of flight which is the travel time of a neutral tracer along a streamline[10],

$$\tau = \int_{\Sigma} s(\mathbf{x})\,dx \qquad \ldots\ldots\ldots(3.69)$$

where, the integral is along the streamline trajectory, Σ, and $s(\mathbf{x})$ is the 'slowness' defined as the reciprocal of the total interstitial velocity[10]

$$s(\mathbf{x}) = \frac{\phi(\mathbf{x})}{\lambda_t K(\mathbf{x})|\nabla P(\mathbf{x})|} \qquad \ldots\ldots\ldots(3.70)$$

Where λ_t represents the total phase mobility, if we assume that the streamlines do not shift because of small perturbations in reservoir properties, then the changes in the water saturation at the outlet node of a streamline is given by

$$\delta S_w = \frac{\partial S_w}{\partial t} \delta t + \frac{\partial S_w}{\partial \tau} \left[\frac{\partial \tau}{\partial \mathbf{m}} \right]^T \delta \mathbf{m} \qquad \ldots\ldots\ldots(3.71)$$

The propagation of a fixed saturation can be expressed by simply setting $\delta S_w = 0$ as follows

$$0 = \frac{\partial S_w}{\partial t} \delta t + \frac{\partial S_w}{\partial \tau} \left[\frac{\partial \tau}{\partial \mathbf{m}} \right]^T \delta \mathbf{m} \qquad \ldots\ldots\ldots(3.72)$$

We can now combine **Eq. 3.72** with **Eq. 3.68** in order to obtain the following expression for travel time sensitivity with respect to a model parameter m_i in terms of the streamline time of flight,

$$\frac{\partial t}{\partial m_i} = \frac{\dfrac{\partial \tau}{\partial m_i}}{\dfrac{\partial F_w}{\partial S_w}} \qquad \ldots\ldots\ldots(3.73)$$

In the above expression, the fractional flow derivatives are computed at the saturation of the outlet node of the streamline. The time of flight sensitivities can be obtained analytically in terms of simple integrals along streamline from **Eq. 3.69** by assuming no shift of streamline due to change in reservoir parameters as mentioned before, so the change in time of flight can be expressed in terms of slowness as follows:

$$\delta\tau = \int_{\psi} \delta s(\mathbf{x}) dr \qquad\qquad ………..(3.74)$$

As the slowness is a composite function involving reservoir properties, its first order variation is as follows:

$$\delta s(\mathbf{x}) = \frac{\partial s(\mathbf{x})}{\partial K} \delta K(\mathbf{x}) + \frac{\partial s(\mathbf{x})}{\partial \phi} \delta\phi(\mathbf{x}) \qquad\qquad ………(3.75)$$

where, the partial derivatives are

$$\frac{\partial s(\mathbf{x})}{\partial K} = \frac{-\phi(\mathbf{x})}{\lambda_t K^2(\mathbf{x})|\nabla P|} = -\frac{s(\mathbf{x})}{K(\mathbf{x})} \qquad\qquad …....…..(3.76)$$

$$\frac{\partial s(\mathbf{x})}{\partial \phi} = \frac{1}{\lambda_t K(\mathbf{x})|\nabla P|} = \frac{s(\mathbf{x})}{\phi(\mathbf{x})} \qquad\qquad ………..(3.77)$$

Accordingly, from **Eq. 3.74 – 3.77**, the time of flight sensitivities can be obtained analytically in terms of simple integrals along streamline. For example, the time of flight sensitivity with respect to permeability will be given by

$$\frac{\partial\tau}{\partial K(\mathbf{x})} = \int_{\Sigma} \frac{\partial s(\mathbf{x})}{\partial K(\mathbf{x})} dx = -\int_{\Sigma} \frac{s(\mathbf{x})}{K(\mathbf{x})} dx \qquad\qquad …....…..(3.78)$$

where the integrals are evaluated along the streamline trajectory. It is to be noted that the quantities in the sensitivity expressions are either contained in the initial reservoir model or are produced by a single simulation run regardless of the number of parameters or the number of data points.

3.2.2 Generalized Travel Time Sensitivity with Pressure Update

The travel time sensitivities derived before assume a stationary streamline, for the cases of changing field conditions or infill drillings this assumption is no longer valid and a new sensitivity equation should be derived to account for pressure updating and remapping of water saturation. According to He et al.[11], the sensitivity of the travel time will be as follows:

$$\frac{\partial t}{\partial m_i} = \frac{\dfrac{\partial \tau}{\partial m_i}}{\dfrac{\partial F_w}{\partial S_w}} + \frac{\partial t^n}{\partial m_i} \qquad\qquad\text{.........(3.79)}$$

In the above equation, $\partial t^n / \partial m_i$ is the travel time sensitivity at the beginning of the pressure update. For multiple pressure updates, $\partial t^n / \partial m$ will correspond to that of the last update. Again, all the quantities in **Eq. 3.79** can be obtained analytically from a single forward simulation. Thus the sensitivity computations are extremely efficient and do not require any additional simulations regardless of the number of data points or the number of parameters.

3.3 Chapter Summary

In this chapter, we give a detailed discussion about calculating generalized travel time sensitivity using the adjoint method with a brief overview of the streamline-based sensitivity using finite difference models. For the adjoint method, the sensitivity calculation per minimization iteration require one forward simulation run and solving 2M adjoint system of equations each time step for N_w times to get the Lagrange multipliers. For the streamline-based sensitivity, the sensitivity calculation requires one simulation run and retracing the streamline and updating the sensitivity calculation each pressure update. So, the major computational advantages of streamline-based sensitivity computation compared to the adjoint method are in two points. First, the streamline

requires retracing the streamline and updating the sensitivity calculation each pressure update, which is far less than the time steps used to solve the 2M adjoint system of equations. The second point, which is most important, is that all the information required to calculate the sensitivity can be obtained from only one simulation run. So it is independent of the number of wells, compared to the adjoint method where for sensitivity calculation, the adjoint system of equations are required to be solved N_w times to get all the information for sensitivity calculation. To improve the adjoint method to be comparable to the streamline-based sensitivity is to solve the adjoint system of equation using larger time step[55], for example equivalent to the number of pressure update used in streamline; however, still the second advantage of streamline-based sensitivity makes it superior compared to the adjoint method or any other rigorous sensitivity calculation method.

CHAPTER IV

APPLICATIONS

In this chapter we shows applications for automatic history matching on Bayesian framework using two different approaches of sensitivity calculation required by any gradient based optimization algorithms, those are adjoint method-based sensitivity and streamline-based sensitivity which discussed before in Chapter III. Also, we used two different approaches to include the regularization term in the objective function, one based on the conventional approach of using the full covariance and the other based on the approximation of the square root of the inverse of the covariance using numerical stencil which is well suited for field-scale applications. In addition, we use two different techniques during the minimization process for updating the mode parameter, one is the exact calculation of the model parameter update during the minimization using the conventional Gauss-Newton with full covariance, and the other is using LSQR as a sparse matrix solver for calculating the model parameter update with an approximation of the square root of the inverse of the covariance using numerical stencil which discussed before in Chapter II.

Fig. 4.1 and **Fig. 4.2** show a flow chart for the automatic history matching process in Bayesian framework used during this work for adjoint method-based sensitivity and streamline-based sensitivity, respectively.

The first part of this chapter gives the comparison between the travel time sensitivity obtained from the perturbation with that obtained from adjoint method after we formulate it to 3D, two phase flow problems and using commercial simulator (viz. Eclipse[53]) as a forward model.

The second part shows two different synthetic examples to test the accuracy of the history matching using adjoint method-based sensitivity with two different approaches used during the minimization process, one with Gauss-Newton with full covariance, which we call it "conventional Bayesian Approach" and the other with LSQR as sparse matrix solver with an approximation of the square root of the inverse of the covariance
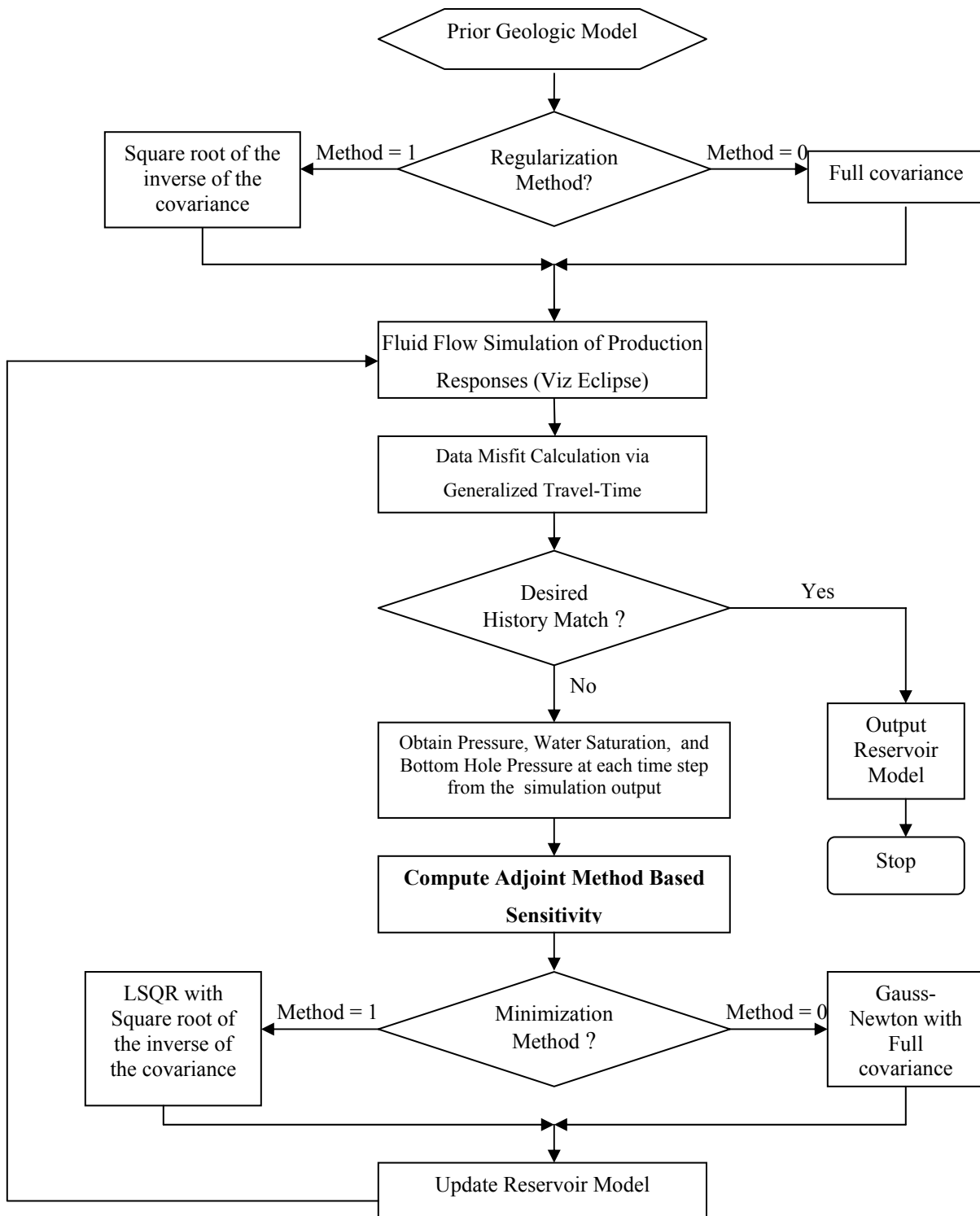
Fig. 4.1−Flowchart for automatic history matching process using adjoint method-based sensitivity
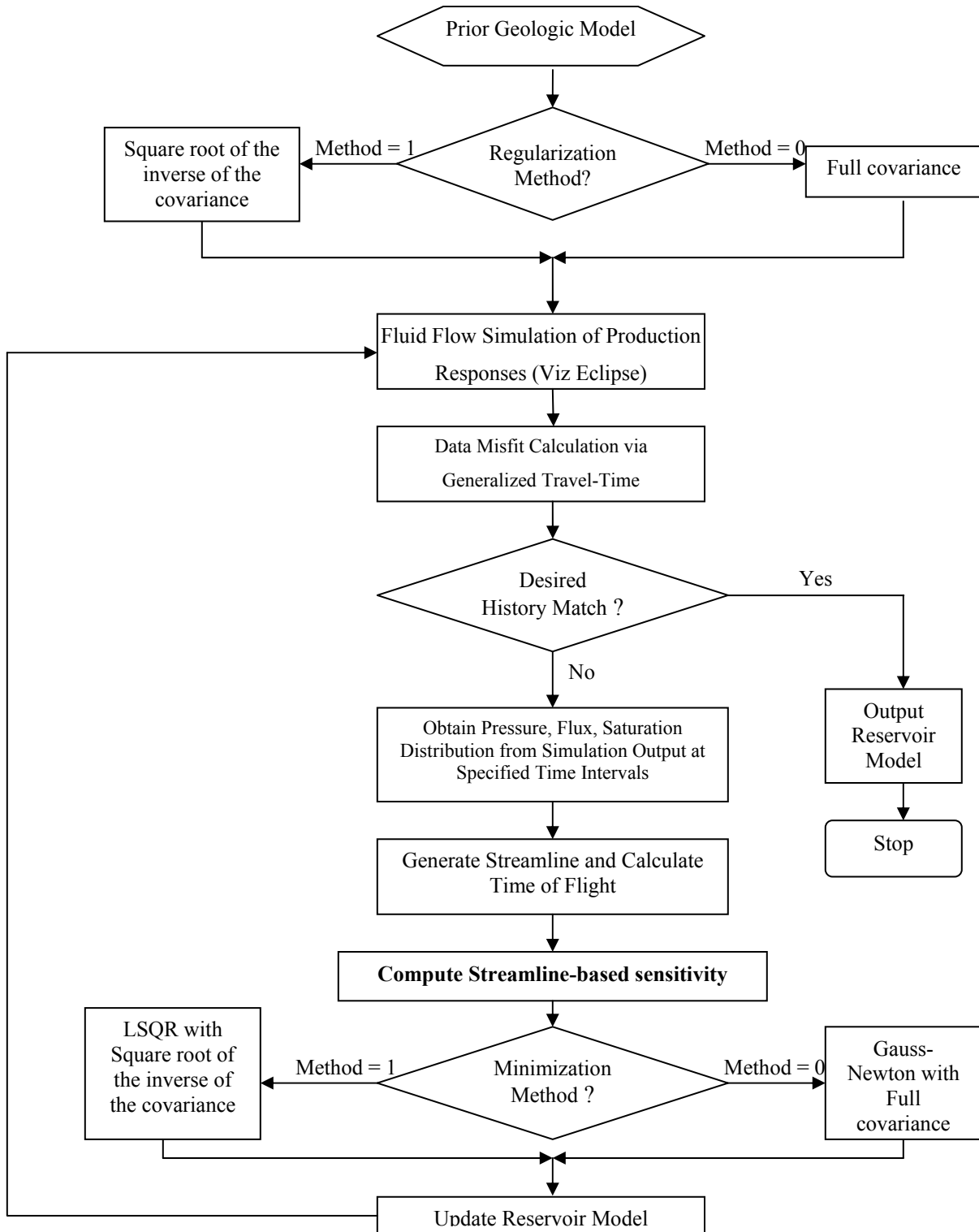
Fig. 4.2–Flowchart for automatic history matching process using streamline-based sensitivity

using numerical stencil, which we call it "Field-scale Bayesian Approach" as it is well suited for field-scale applications.

The third part of this chapter shows the CPU time scaling with increasing the model parameters for both the conventional and field-scale Bayesian approach using four synthetic cases of model size ranges from 8000 to 80,000 grid blocks and also shows a comparison between the CPU time required for the sensitivity calculation using both adjoint and streamline-based sensitivity.

The fourth part of this chapter shows a field case application taken from Goldsmith San Andreas unit in West Texas using both adjoint method-based sensitivity and streamline- based sensitivity with the field-scale Bayesian approach during the minimization process.

Finally, the last part of this chapter uses the Randomized Maximum Likelihood to generate multiple realizations from the posterior distribution which can be further used to assess the uncertainty in the production forecast for Goldsmith field case.

4.1 Sensitivity Comparison

In this section we show the comparison between the travel time sensitivity obtained from perturbation and that obtained using adjoint method for 3D two phase flow as formulated before in Chapter III. The only different between the generalized travel time sensitivity given before in chapter III and the travel time sensitivity is in the way of formulating the source term (g) in the adjoint system of equations given by **Eqs. 3.27** and **3.28**, where the source term is given as:

$$g = t^i_{cal,j}$$
………..(4.1)

Where, $t^i_{cal,j}$ is the travel time at well j and at observed point t^i, which is one single point on the production response. The gradient of $t^i_{cal,j}$ with respect to pressure and water saturation is obtained in the same manner as given by **Eqs. 3.33** and **3.34** by

replacing generalized travel time with travel time. At grid blocks of producing wells and at the observed time t^i corresponds to the simulation time index t^l , the partial derivatives of the travel time with respect to pressure and water saturation are given as follows:

$$\frac{\partial t^i_{cal,j}}{\partial p^l} = \frac{\partial t^i_{cal,j}}{\partial K} \cdot \frac{\partial K}{\partial f^i_{wcal,j}} \cdot \frac{\partial f^i_{wcal,j}}{\partial p^l} \qquad \ldots\ldots\ldots\ldots(4.2)$$

Similarly as mentioned before, the change in permeability has opposite effect with respect to travel time and water cut, thus:

$$\frac{\partial t^i_{cal,j}}{\partial K} \cdot \frac{\partial K}{\partial f^i_{wcal,j}} = \frac{-1}{\partial f^i_{wcal,j}/\partial t^i_{cal,j}} = \frac{-1}{(f^l_{wcal,j} - f^{l-1}_{wcal,j})/(t^l_{cal,j} - t^{l-1}_{cal,j})} \qquad \ldots\ldots\ldots\ldots(4.3)$$

Substituting **Eq. 4.3** in **Eq. 4.2**

$$\frac{\partial t^i_{cal,j}}{\partial p^l} = \frac{-1}{(f^l_{wcal,j} - f^{l-1}_{wcal,j})/(t^l_{cal,j} - t^{l-1}_{cal,j})} \cdot \frac{\partial f^i_{wcal,j}}{\partial p^l} \qquad \ldots\ldots\ldots\ldots(4.4)$$

Similarly, the derivative with respect to water saturation is as follows:

$$\frac{\partial t^i_{cal,j}}{\partial S^l_w} = \frac{-1}{(f^l_{wcal,j} - f^{l-1}_{wcal,j})/(t^l_{cal,j} - t^{l-1}_{cal,j})} \cdot \frac{\partial f^i_{wcal,j}}{\partial S^l_w} \qquad \ldots\ldots\ldots\ldots(4.5)$$

Where, l in **Eqs. 4.4** and **4.5** is corresponding to time, t^i at the single observed point (i) selected to get the travel time sensitivity, while $l-1$ is corresponding to time t^{l-1} which is ($t^i - \Delta t$), where Δt is the time step used in the simulation.

The derivative of water cut with respect to pressure and water saturation is given before in **Eqs. 3.39** and **3.40**.

The sensitivity of travel time with respect to horizontal permeability from perturbation is obtained by perturbing the permeability in x and y direction at each grid and getting the change in the travel time with respect to the change in permeability at certain specified water cut value. So for M grid blocks we need to run the forward model M+1 times to obtain the sensitivity matrix. The amount of perturbation used is 5% of the amount of the permeability at each grid block.

The example shown here to show the comparison between the sensitivity from perturbation and that obtained from adjoint method is quarter five spot of 15x15x2 where we used a homogenous permeability of ($K_x = K_y = 244.7$ md) and $K_z = 15$ md, The porosity is homogenous of 0.22. The producer is at grid block (3,3) and produced from the two layers with constant production rate of 500 STB/D while the injector is at grid block 13,13 and injected water in two layers with constant bottom hole pressure of 6100 psi. The initial reservoir pressure used is 5500 psi and the initial water saturation is 0.2. The simulation time used is 950 days and the water breakthrough at the producer occurs after 300 days of production.

Figs. 4.3 a and **b** show the travel time sensitivity from perturbation and from adjoint method, respectively. It can be easily shown that the travel time sensitivity from adjoint method has the same trend and the locations of the high and low values matches well with the travel time sensitivity from perturbation. It is important to mention that including the injectors and using the formula given by **Eq. B.51** to model it and getting its derivative with respect to pressure and water saturation required by the adjoint system of equations and getting its derivative with respect to permeability required by the sensitivity equations always overestimate the sensitivity values at the injectors compared to the perturbation and by not including the injectors the results getting better as shown in **Fig. 4.3b**. This might be due to the reason that the formula used to model the rate allocation from the injectors use the total mobility not only the water mobility so that if the injectors is placed in grid block that water saturation is at connate water saturation,

(a)   (b)
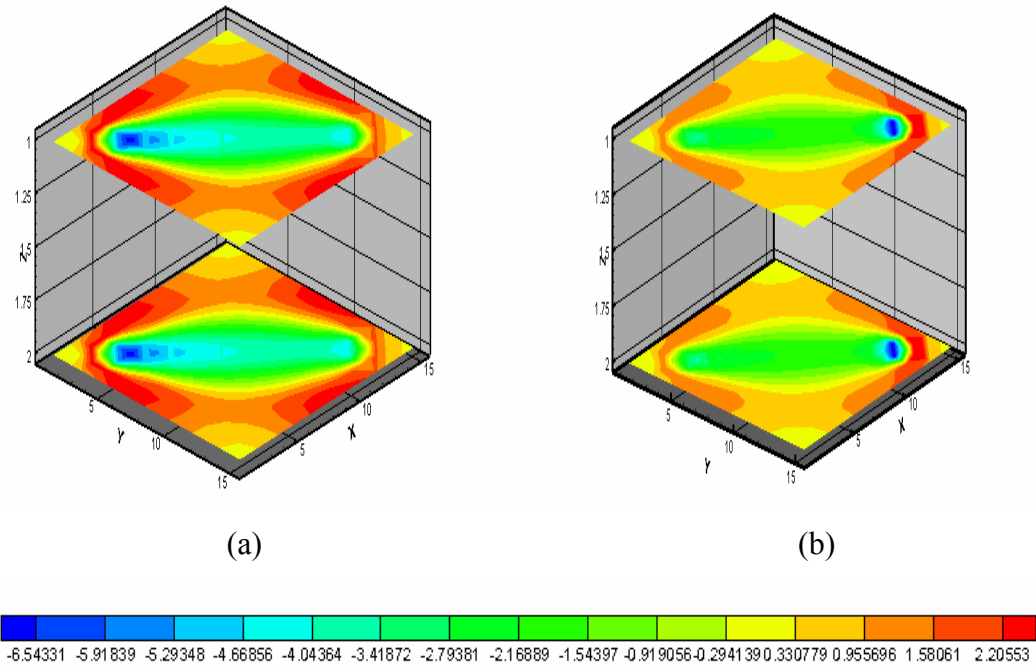
Fig. 4.3−Comparison of travel time sensitivity with respect to horizontal permeability from the two layers, (a) perturbation, (b) Adjoint method

the mobility of water will be zero while the mobility of oil is not, thus using total mobility to model the rate allocation from injectors will ensure that always water is in mobile status even if the water saturation at the injection grid block is still at connate water saturation. In fact in solving the adjoint equation backward in time the water saturation at late time will not be at connate water saturation anymore so using the formula given by **Eq. B.51** with total mobility will definitely overestimate the sensitivity. So, clearly additional work should be done in this area to see how we can include the injectors in the sensitivity calculation so that we can get reasonable result with perturbation.

The comparison of streamline sensitivity with perturbation can be found elsewhere[10,11]

4.2 Synthetic Examples Using Adjoint Method-Based Sensitivity

The objective of these synthetic examples is to test the accuracy of the inversion and to test its practical application using adjoint method-based sensitivity for 3D two phase reservoirs under two different approaches; one is for full covariance with Gauss-Newton, the other is with LSQR with the approximation of the square root of the inverse of the covariance using numerical stencil.

4.2.1   Example 1: Reservoir of Three Permeability Regions

This example is 15x15x2, **Table 4.1** summaries the data used for this example and **Fig. 4.4** shows the relative permeability data. The well pattern is five spot, with four producers at the four corners and one injector at the middle. The four producers produce from the two layers with constant total rate of 100 STB/Day and the injector injects in the two layers with constant bottom hole pressure of 6100 psi. The location of the producing wells 1, 2, 3, and 4 are at grid blocks (3,3), (13,3), (13,13), (3,13), respectively, while the injector is at grid block (8,8).

The true horizontal permeability field is given in **Fig. 4.5a,** where it consists of three different zones of permeability at the upper and lower layer. The prior mean and the initial horizontal permeability are the same and it is given in **Fig. 4.5b** which is a uniform permeability of **244.7 md** at the upper layer and **54.6 md** at the lower layer. The vertical permeability is kept constant and it is equal to **15 md**. The observed water cuts at the four producers are generated by running the simulator for the true permeability distribution given in **Fig. 4.5a.**

   The inversion is done using two different approaches that discussed before in Chapter II, which are "Conventional Bayesian" and "Field-scale Bayesian".

I- For conventional Bayesian approach

Fig. 4.6a shows the final permeability after inversion along with the true permeability in **Fig.4.6b** for comparison purpose. **Fig. 4.7** shows the water cut match from the initial and the MAP estimate for the four producers and **Fig. 4.8** shows the "generalized travel time" misfit and the conventional "amplitude" misfit as function of number of iterations

and it is clearly seen that the reduction in the misfit from both generalized travel time and amplitude are in good consistent with each other.

Table 4.1–Data for example 1	
Grid Blocks	15x15x2
Grid Size	$\Delta x = 40\,ft$, $\Delta y = 40\,ft$, $\Delta z = 30\,ft$
Porosity	0.22
Oil and Water Viscosity	$\mu_o = 0.82cp$, $\mu_w = 1.0cp$
Oil and Water Compressibility	$c_o = 0.00001$, $c_w = 0.000004$
Initial Water Saturation	0.2
Initial Pressure	5500 psi
Oil Formation volume factor, B_o at P_i	1.24 bbl/stb
Water Formation volume factor, B_w at P_i	1.0 bbl/stb
Prior Variogram Model	Exponential
Ranges in x, y, z direction	160 ft
Sill for ln K	$\sigma_{\ln K} = 0.5$
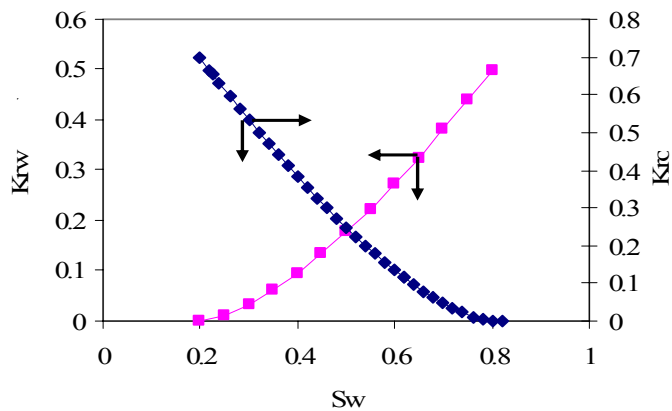
Fig. 4.4–Relative permeability data for example 1
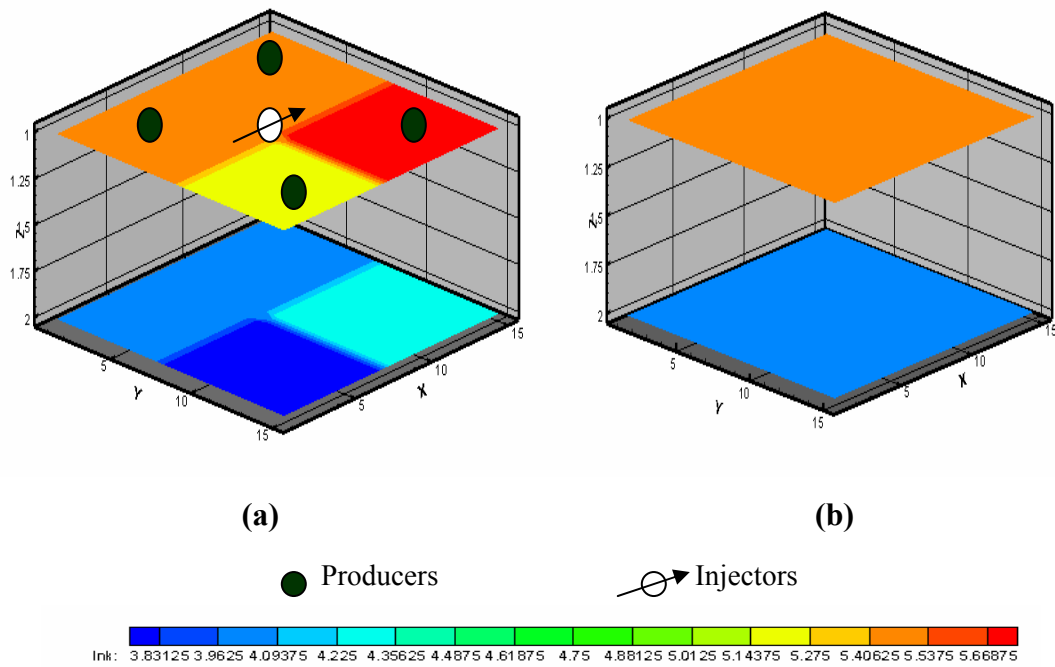
(a)   **(b)**

● Producers   ⊘► Injectors

Ink: 3.83125 3.9625 4.09375 4.225 4.35625 4.4875 4.61875 4.75 4.88125 5.0125 5.14375 5.275 5.40625 5.5375 5.66875

Fig. 4.5−(a)True permeability, (b)Initial and prior mean permeability for example 1

(a)   **(b)**

Ink: 3.83125 3.9625 4.09375 4.225 4.35625 4.4875 4.61875 4.75 4.88125 5.0125 5.14375 5.275 5.40625 5.5375 5.66875

Fig. 4.6−(a) Final permeability using conventional Bayesian approach,
(b) True permeability for example 1
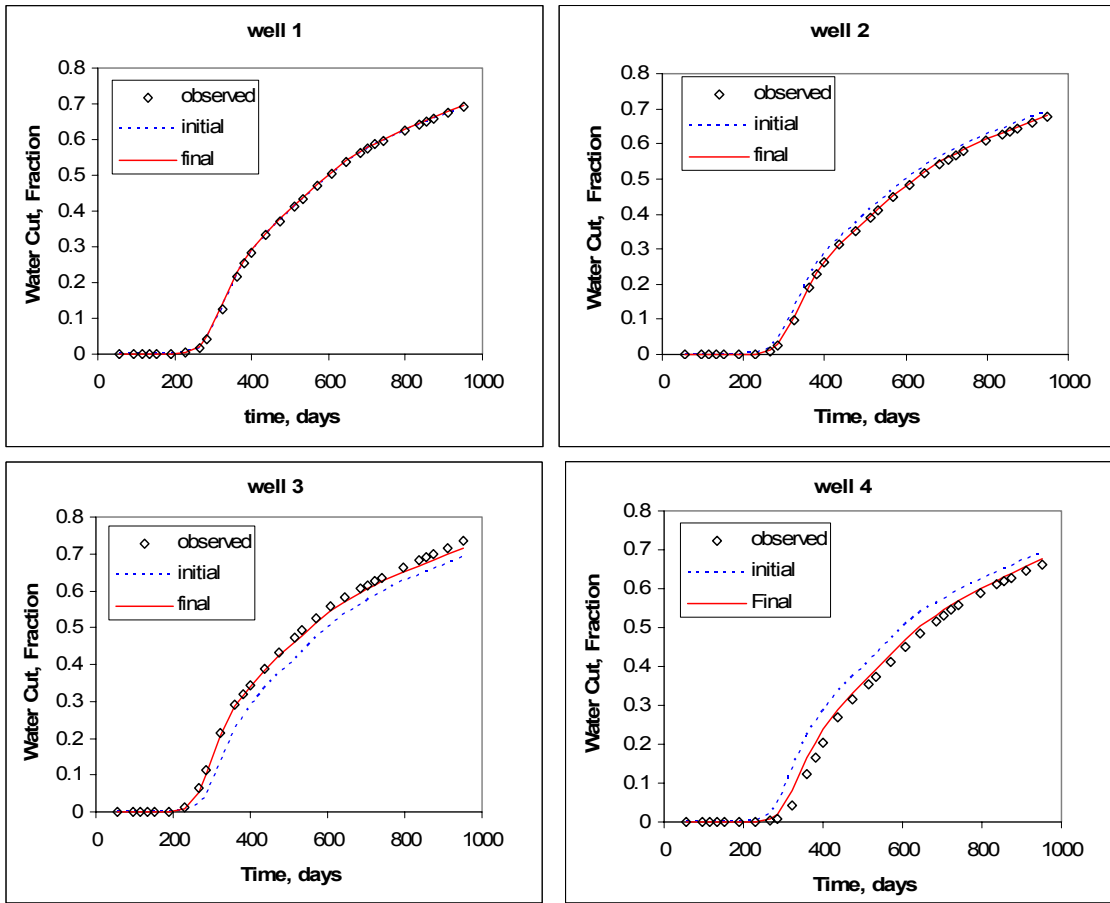
Fig. 4.7–Water cut match for the four producers using conventional Bayesian approach for example 1

Fig. 4.8–Misfit reduction for example 1 using conventional Bayesian approach

It is clear from the final permeability given in **Fig. 4.6a** that most of the changes takes place around well 3 and 4 and this is shown from the water cut match for those wells, where the match has been improved compared to the initial model, while for the other wells, well 1 and 2 the match was quite good from the initial model that is why no changes occurs around those wells.

II-For field-scale Bayesian approach

In this approach we use 5x5x5 stencil to approximate the square root of the inverse of the covariance using numerical stencil as discussed in Chapter II and then use LSQR as a sparse matrix solver for updating the model which is the permeability at each grid block to minimize the objective function given by **Eq. 2.8** or **Eqs. 2.27** and **2.28**.

Fig. 4.9 shows the comparison between the exact covariance which is used in the conventional Bayesian approach and the covariance obtained from using 5x5x5 stencil in order to see how accurate is our choice of 5x5x5 stencil in approximating the full covariance. The covariance from the stencil is obtained using the same way of approximating the square root of the inverse of the covariance, where instead of using column 63 after getting the square root of the inverse of the 125x125 matrix generated form the 5x5x5 stencil, we used directly column 63 from the 125x125 covariance matrix and populate the full matrix using the same technique of the numerical stencil. In **Fig. 4.9** we show the comparison for one of the row of the full siz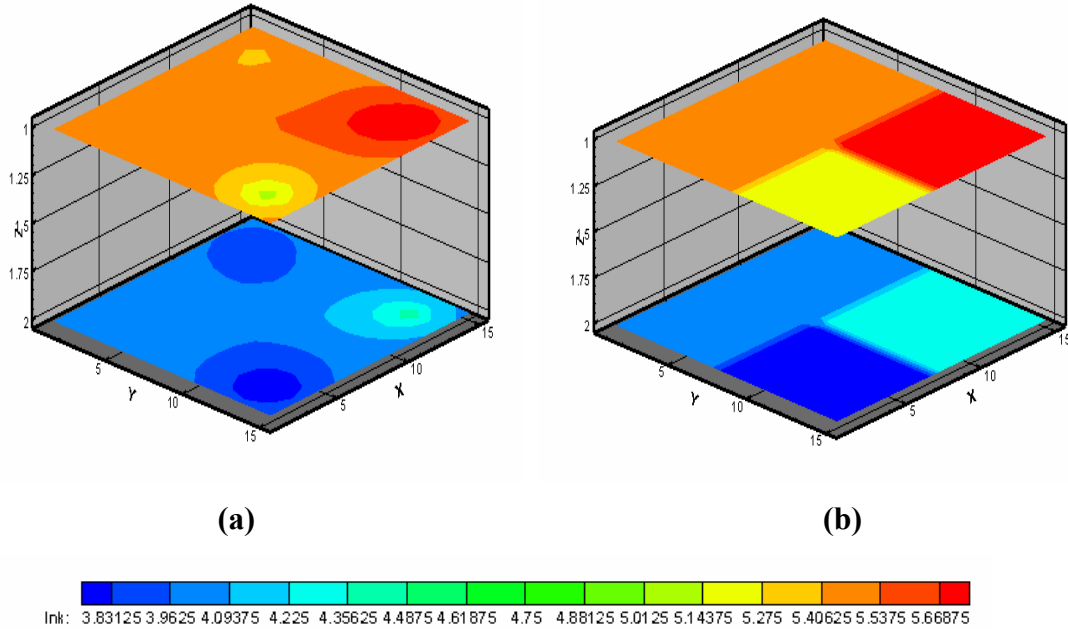e covariance matrix which is the middle row (row 225), as it is difficult to show the comparison row by row for 450 x450 full covariance matrix used in Example 1. As seen from the comparison that however the ranges of variogram used in example 1 extend to four grid blocks in the x, y directions and five grid blocks in the z direction, while the 5x5x5 stencil extend only to two grid blocks in the x, y, and z direction and any correlation between any two permeabilities that has distance more than two grid blocks will be zero, the comparison of the covariance obtained from the stencil shows good agreement with the exact covariance especially for the values of high magnitude.
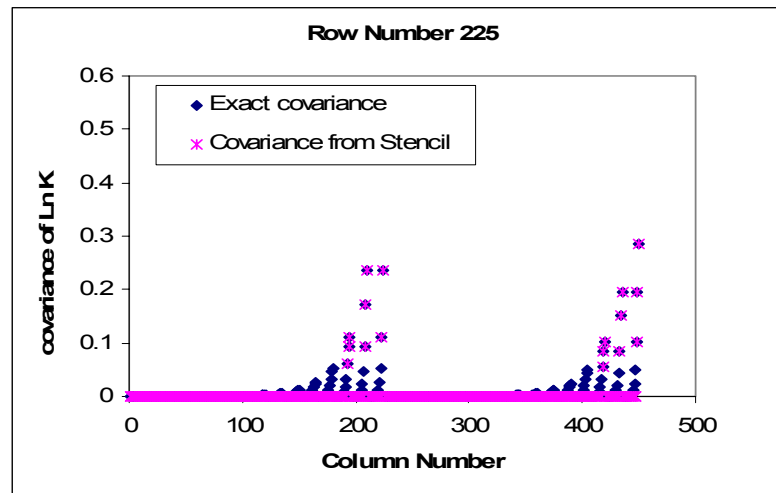
Fig. 4.9−Comparison between the exact covariance and covariance from 5x5x5 stencil at row 225 for example 1

Similarly as done before for the conventional Bayesian, **Fig. 4.10a** shows the final permeability after inversion along with the true permeability in **Fig.4.10b** for comparison purpose. **Fig. 4.11** shows the water cut match from the initial and the MAP estimate for the four producers and **Fig. 4.12** shows the "generalized travel time" misfit and the conventional "amplitude" misfit as function of number of iterations and also both the amplitude and the generalized travel time misfit are in good consistent with each other.

It is clearly seen from the final permeability and the water cut match from the both approaches that there is no big difference, only the filed scale Bayesian approach take more iteration to converge to the same value as obtained by the conventional approach, however the computation time taken for the field-scale Bayesian approach is far less than that for the conventional approach especially for field-scale applications where the number of model size can exceed thousands to millions grid blocks as will be shown in the next part of this chapter that compares the computation time required by the both approaches for field-scale applications.
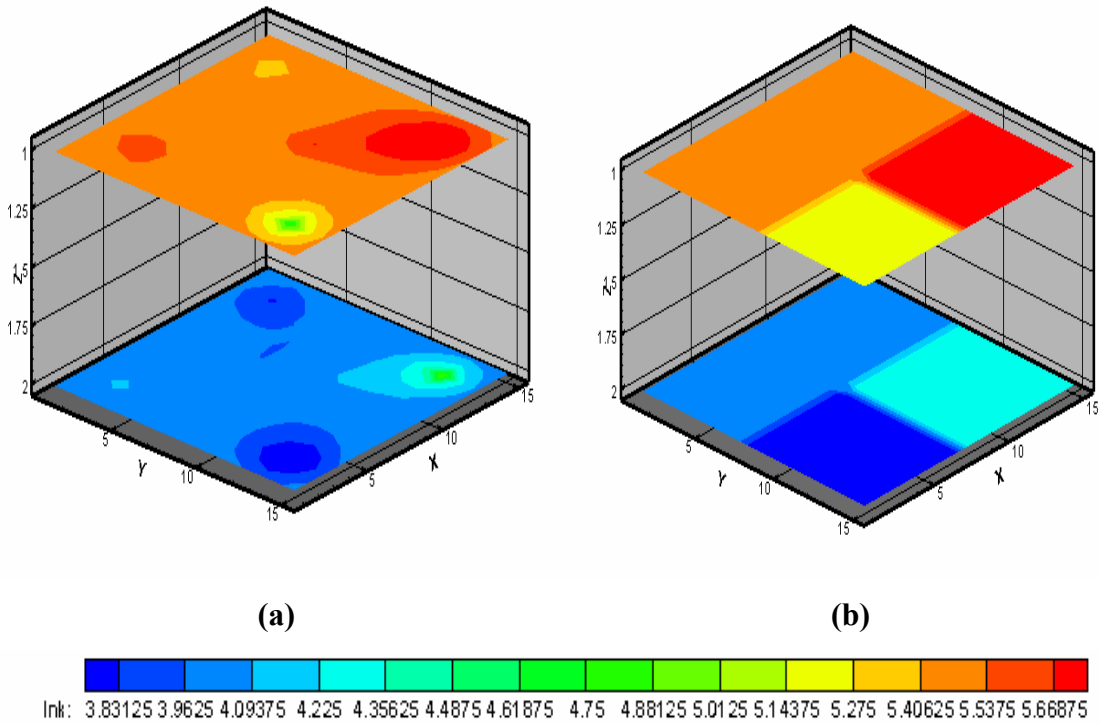
Fig. 4.10−(a) Final permeability using field-scale Bayesian approach,
(b) True permeability for example 1
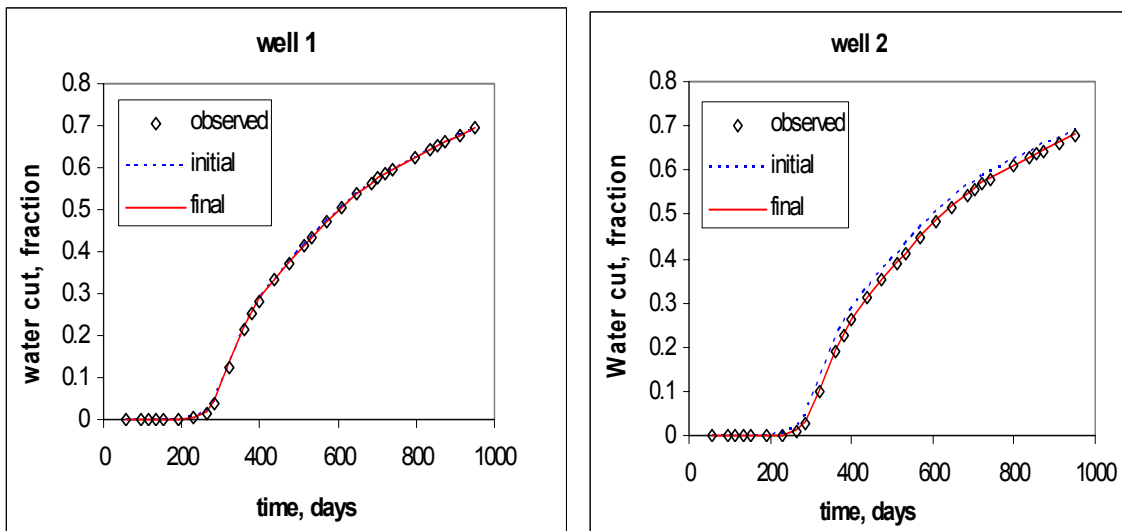
Fig. 4.11−Water cut match for the four producers using field-scale Bayesian
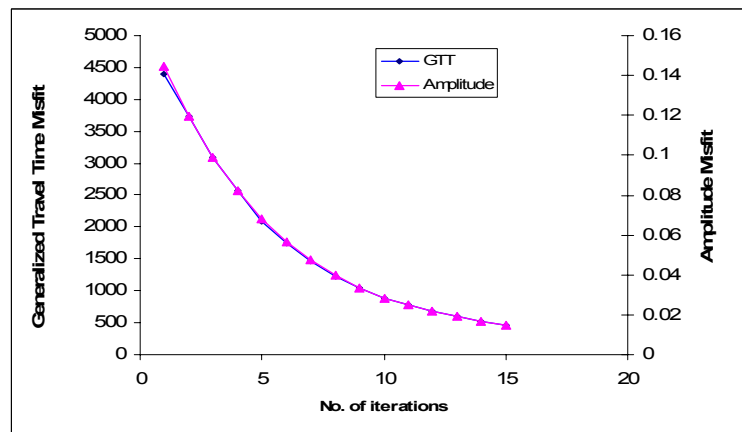approach for example 1
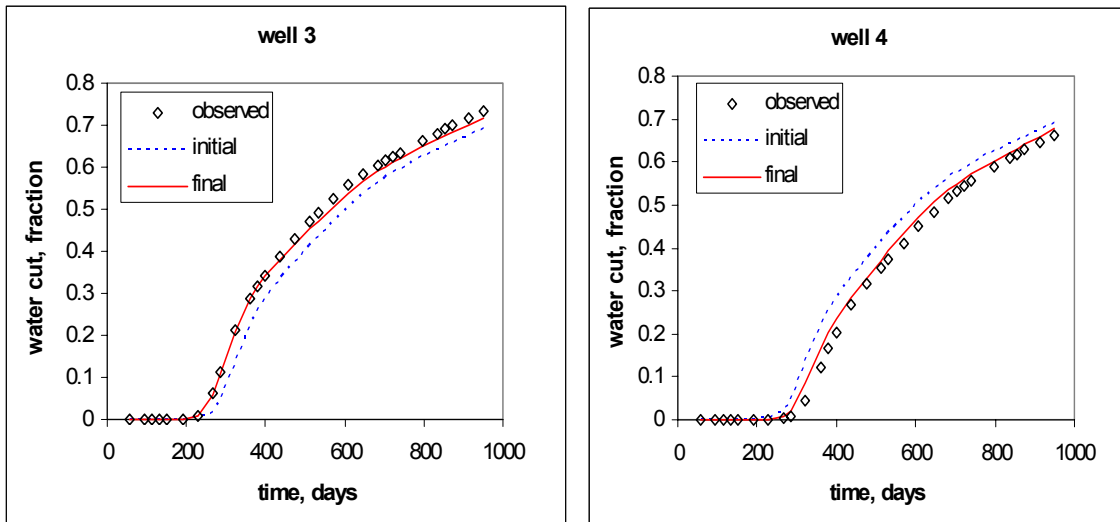
Fig. 4.11−Continued

Fig. 4.12−Misfit reduction for example 1 using field-scale Bayesian approach

4.2.2 Example 2: Heterogeneous Reservoir

The purpose of this example is to test our formulation of the generalized travel time sensitivity using adjoint method for heterogeneous case. The example used is 15x15x3, the data used for this example and the relative permeability is the same as that given for example 1 in **Table 4.1** and **Fig. 4.4.** Also, the well pattern is 5 spot with the location of the producers and the injectors the same as given in Example1 with the only exception that the producers and injectors are completed in the three layers.

The true horizontal permeability field is given in **Fig. 4.13a,** the true permeability is generated using Sequential Gaussian Simulation[52] and characterized by high permeability in the lower right quadrant for both layer 1 and 3, while layer 2 is characterized by its low permeability compared to the upper and lower permeability with high permeability on the right half of the layer and single high permeability streak on the left part of the layer. The prior mean and the initial horizontal permeability are the same and it is given in **Fig. 4.13b** which is a uniform permeability of **387.5 md (ln k = 5.95)** in the upper layer, **119.2 md (ln k = 4.78)** in the middle layer and **419.32 md (ln k = 6.03)** in the lower layer. The vertical permeability is kept constant and it is equal to **15 md**. The observed water cuts at the four producers are generated by running the simulator for the true permeability distribution given in **Fig. 4.13a.**

Similarly, the inversion is done using two different approaches that discussed before in Chapter II, which are "Conventional Bayesian" and "Field-scale Bayesian".

I- For conventional Bayesian approach

Fig. 4.14a shows the final permeability after inversion along with the true permeability in **Fig.4.14b** for comparison purpose. **Fig. 4.15** shows the water cut match from the initial and the MAP estimate for the four producers and **Fig. 4.16** shows the "generalized travel time" misfit and the conventional "amplitude" misfit as function of number of iterations and as it is clearly seen that the reduction in the misfit from both generalized travel time and amplitude are in good consistent with each other as shown before for example 1.

(a) **(b)**

Ink: 4.60151 4.76189 4.92227 5.08265 5.24303 5.40341 5.56379 5.72417 5.88455 6.04493 6.20531 6.36569 6.52607 6.68645 6.84683
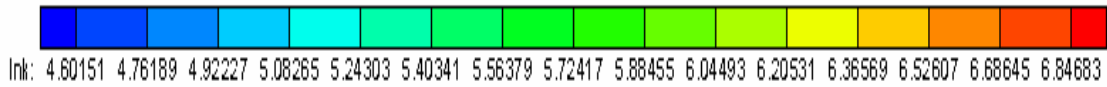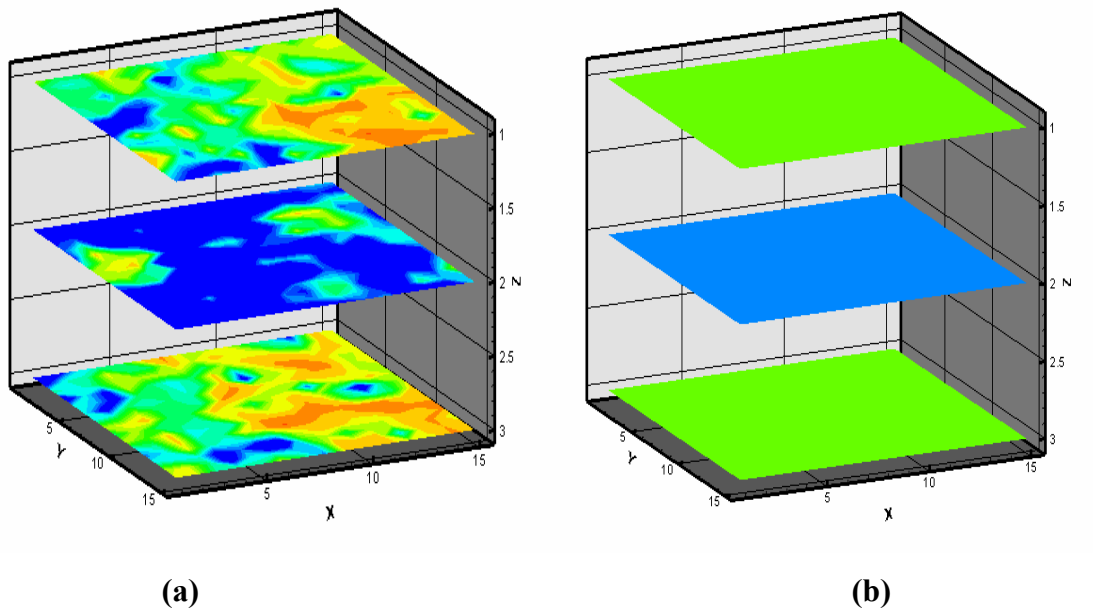
Fig. 4.13–(a) True permeability, (b) Initial and prior mean permeability for example 2

(a) **(b)**

Ink: 4.60151 4.76189 4.92227 5.08265 5.24303 5.40341 5.56379 5.72417 5.88455 6.04493 6.20531 6.36569 6.52607 6.68645 6.84683
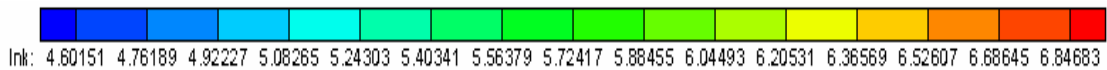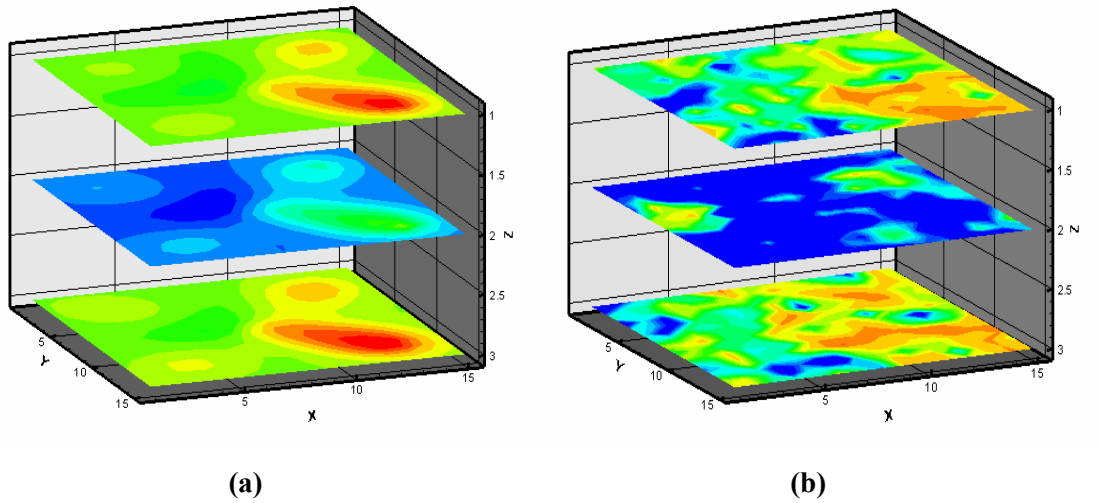
Fig. 4.14–(a) Final permeability using conventional Bayesian approach, (b) True permeability for example 2
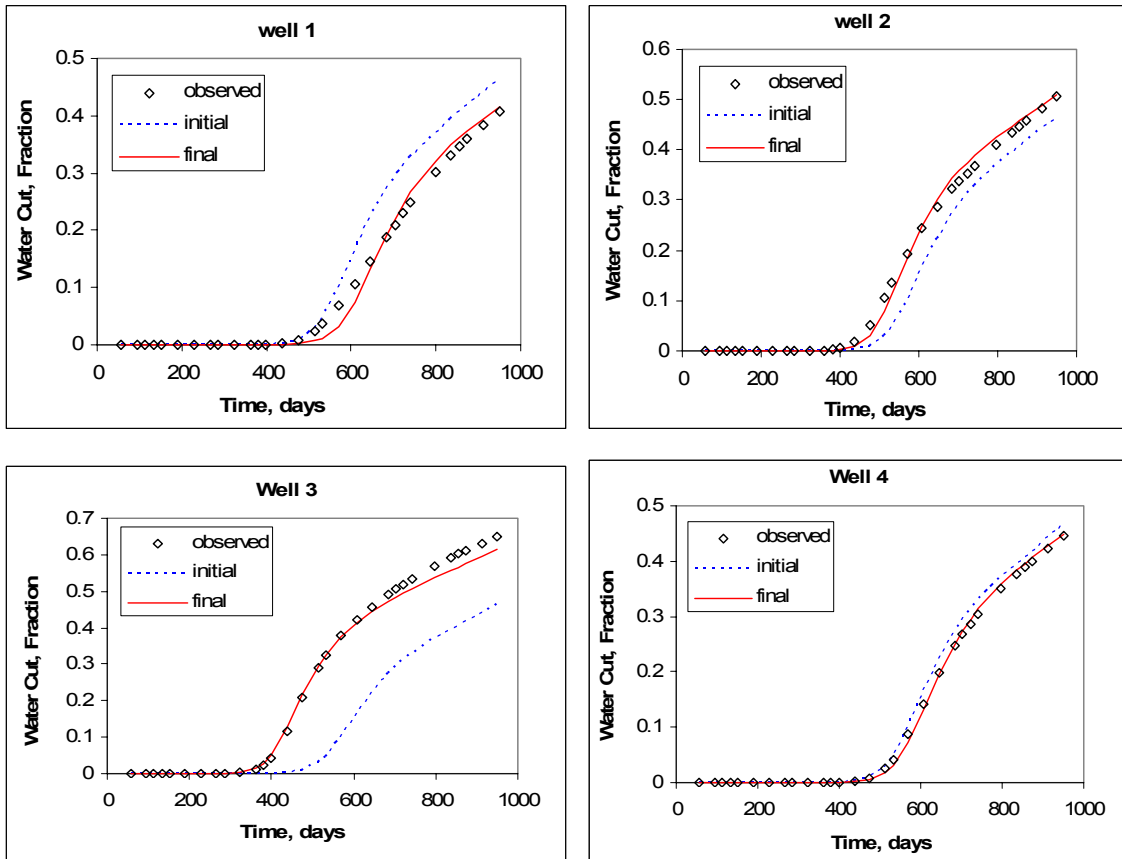
**Fig. 4.15–Water cut match for the four producers using conventional Bayesian approach for example 2**
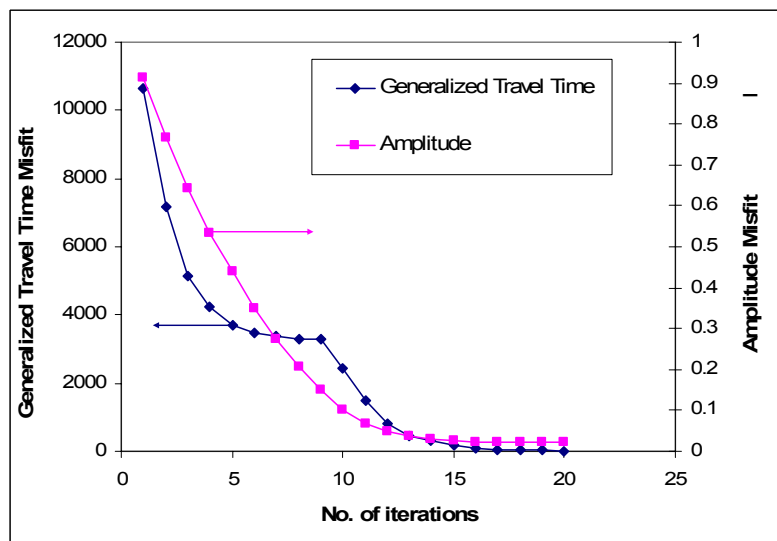
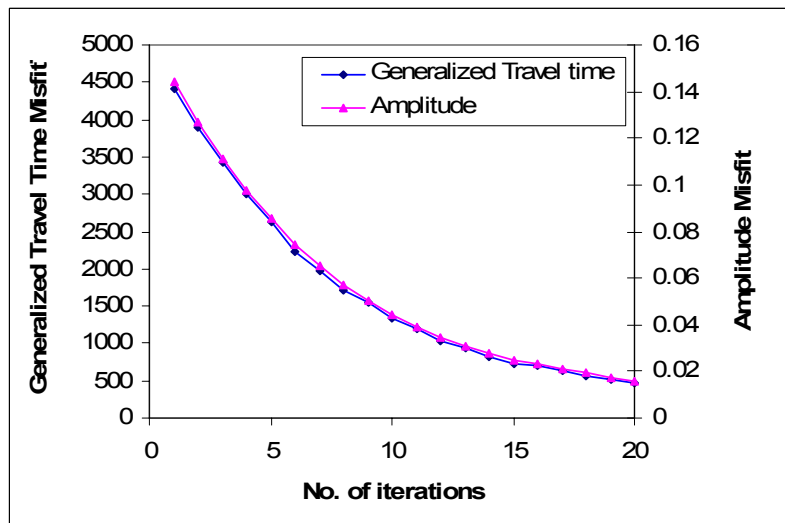**Fig. 4.16–Misfit reduction for example 2 using conventional Bayesian approach**

As It is clear from the final permeability given in **Fig. 4.14a** that most of the changes from the initial takes place around the producing wells 1, 2, and 3 and the injectors to match the water cut data and this shows also in the water cut match shown in **Fig. 4.15** where the water cut has been changed completely for wells 1, 2, and 3 to match the water cut data. However, between the producing well 4 and the injectors, no significant changes take place as the water cut match from the initial model was quite satisfactory.

II-For field-scale Bayesian approach

Similarly, as mentioned before in Example 1, we use 5x5x5 stencil to approximate the square root of the inverse of the covariance using numerical stencil as discussed in Chapter II and then use LSQR as a sparse matrix solver for updating the model which is the permeability at each grid block to minimize the objective function given by **Eq. 2.8** or **Eqs. 2.27** and **2.28**. **Fig. 4.17** shows the comparison between the exact covariance which is used in the conventional Bayesian approach and the covariance obtained from using 5x5x5 stencil in the same manner that was done before in order to test the accuracy of the 5x5x5 stencil in constructing accurate covariance which in turn gives accurate square root of the inverse of the covariance required by the LSQR solver during the minimization.

Fig. 4.17 compares row number 338 of the full covariance which is of order 675x675 for this example with the covariance obtained from the stencil, as it is clear that the covariance from the stencil shows perfect agreement with the exact covariance for the values of higher magnitude, however the ranges used in the stencil covers only 2 grid blocks in the three directions, while the actual ranges used in this example covers four grid blocks in the x, and y direction and five grid blocks in the z direction.
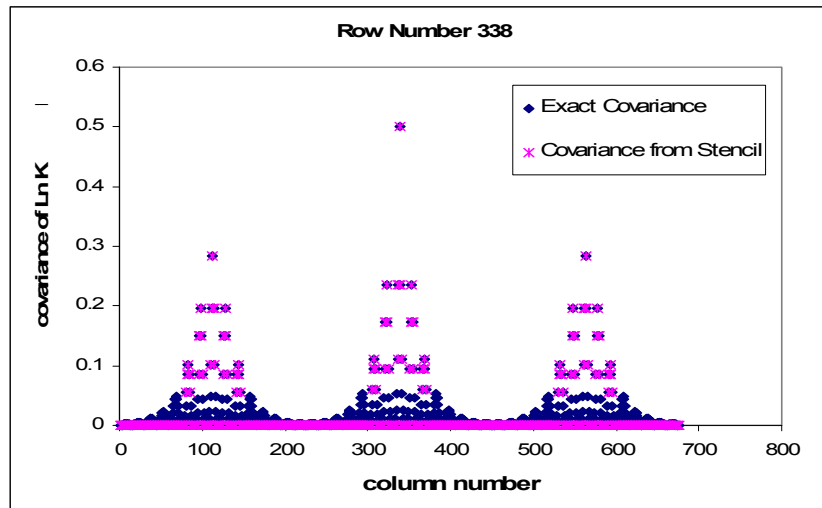
Fig. 4.17−Comparison between the exact covariance and covariance from 5x5x5 stencil at row number 338 for Example 2

Similarly as done before for the conventional Bayesian, **Fig. 4.18a** shows the final permeability after inversion along with the true permeability in **Fig.4.18b** for comparison purpose. **Fig. 4.19** shows the water cut match from the initial and the MAP estimate for the four producers and **Fig. 4.20** shows the "generalized travel time" misfit and the conventional "amplitude" misfit as function of number of iterations and also both the amplitude and the generalized travel time misfit are in good consistent with each other.
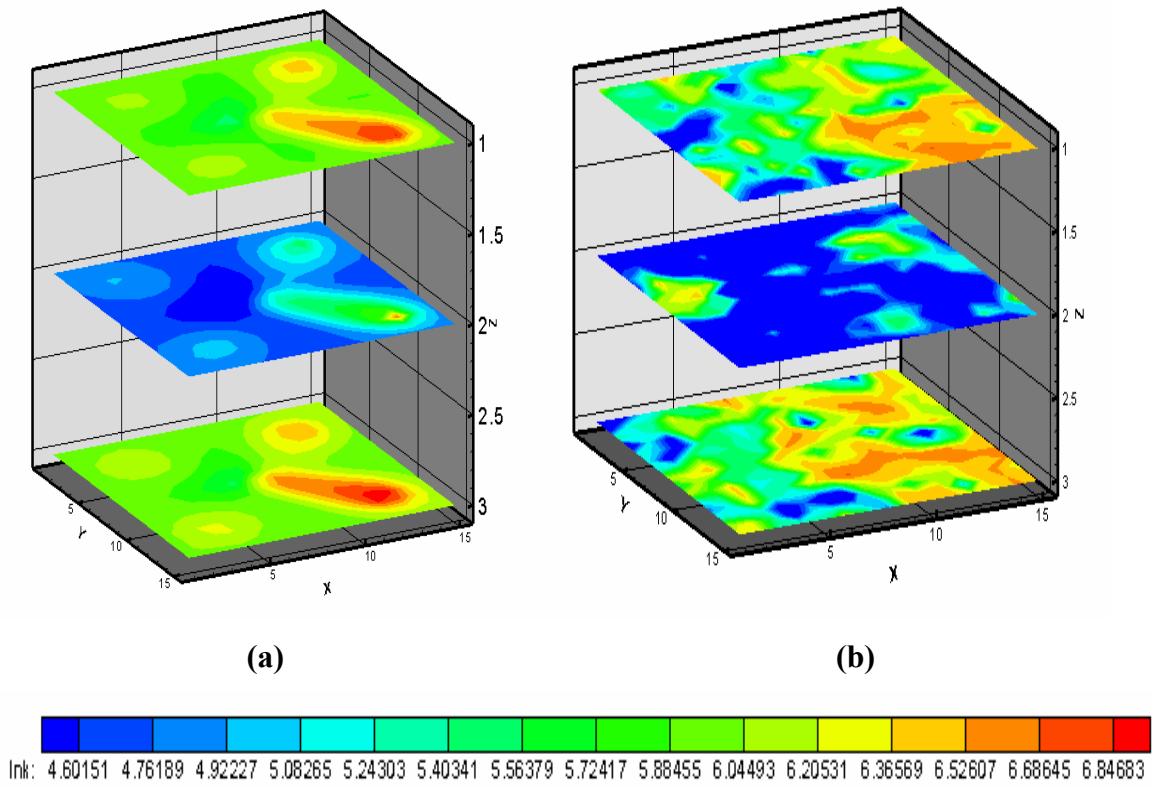
**Fig. 4.18: (a) Final permeability using field-scale Bayesian approach,
(b) True permeability for example 2**
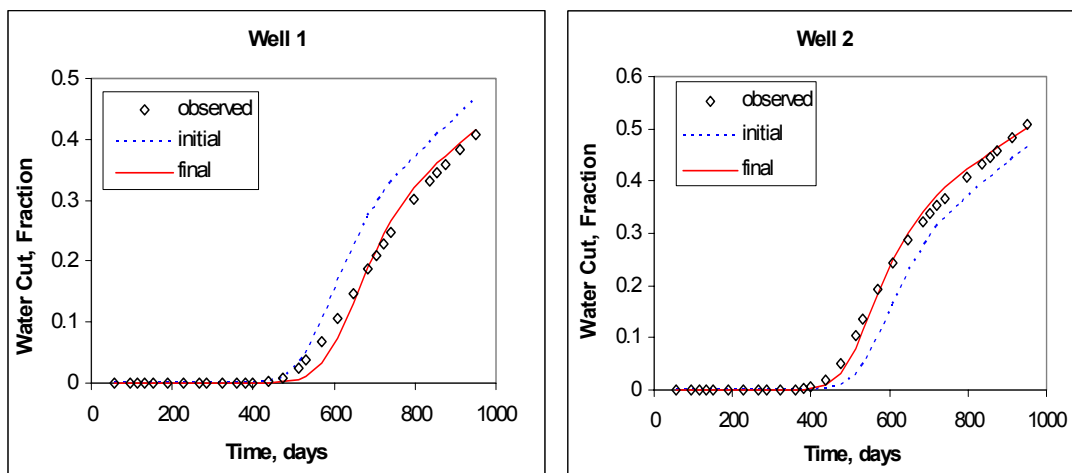
**Fig. 4.19–Water cut match for the four producers using field-scale Bayesian
approach for example 2**
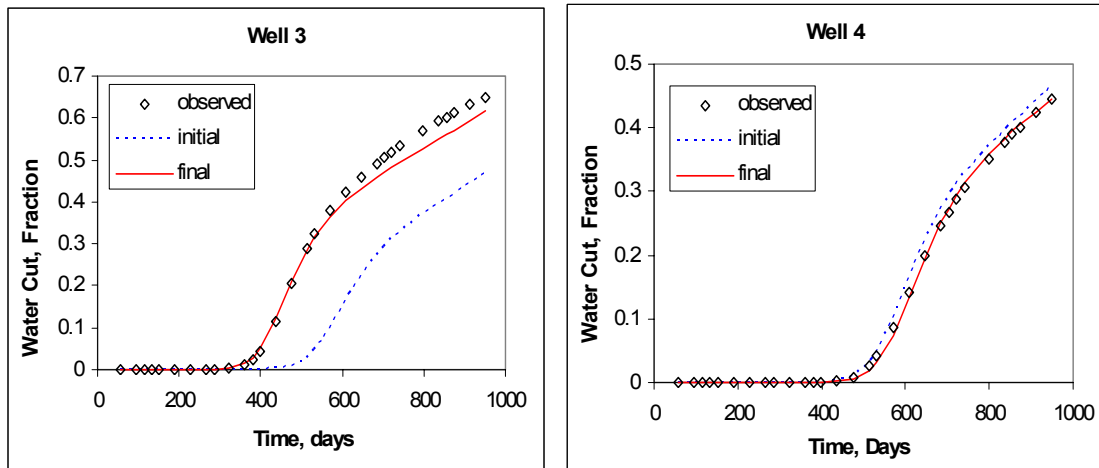
Fig. 4.19‒Continued
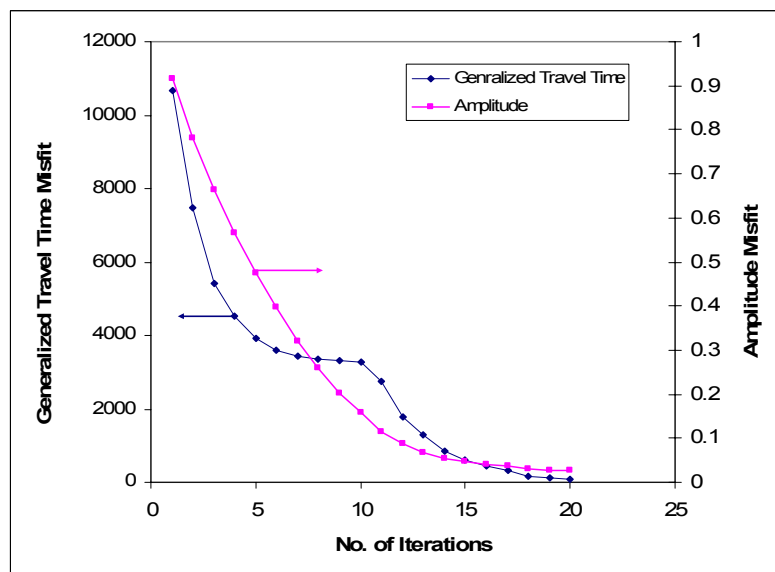
Fig. 4.20‒Misfit reduction for example 2 using field-scale Bayesian approach

From the comparison between the final permeability and the water cut match obtained from the field-scale Bayesian and the conventional Bayesian, it is clearly no big changes due to the good approximation of the 5x5x5 stencil compared to the exact as shown from **Fig. 4.17**. However, using the concept of the stencil reduces the

computation required as mentioned before especially for large sale field applications. The only notice which also takes place in Example 1 is that the reduction of the misfit using the Field-scale Bayesian needs more iteration to converge to the same values used by Gauss-Newton.

4.3 Scaling Comparison

The primary purpose of this section is to examine the CPU time required during the minimization process and updating the permeability using Gauss-Newton with full covariance (Conventional Bayesian) and LSQR with the square root of the inverse of the covariance approximated using numerical stencil (Field-Scale Bayesian) in order to validate the findings in Chapter II, where we found that the number of multiplications, which is directly proportion to the CPU time, for the conventional Bayesian approach scales quadratically compared to the linear scale of the Field-scale Bayesian. The second purpose is to show a comparison between the CPU times for the sensitivity calculation from Adjoint method and streamline method and finally, to show the scaling properties of the conventional and field-scale Bayesian through the whole iteration process using adjoint method-based sensitivity and streamline-based sensitivity.

To achieve those purposes, we generated four 2D synthetic cases using Sequential Gaussian Simulation[52] with grid blocks ranges from 8000 to 80,000. **Table 4.2** summaries the different cases used in this study. The well configuration for the four synthetic cases is multi pattern water flooding as shown in **Fig. 4.21** with **27** producers and **15** injectors and we are integrating the water cut response from **5** producers.

Table 4.2–Grid block size for the synthetic cases			
Cases	**Nx**	**Ny**	**M**
1	128	64	8192
2	150	80	12000
3	200	100	20000
4	400	200	80000
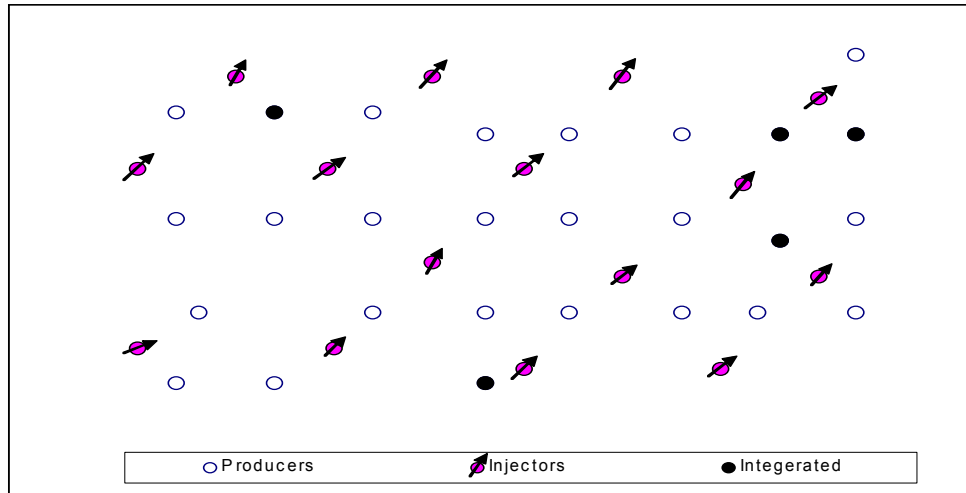
Fig. 4.21−Well configuration for the synthetic examples, showing the location of the producers, the injectors and the producers whose water cut are integrated during the history matching

It is important to mention that in using the conventional Bayesian approach with full covariance, we used to save the covariance matrix row by row in a binary file to overcome the memory allocation problem of the covariance matrix for large model sizes. **Fig. 4.22** shows the CPU time comparison per iteration for the conventional and field-scale Bayesian during the minimization process as function of the number of grid blocks. It is clear that the conventional Bayesian shows a quadratic scaling compared to the linear scaling of the field-scale Bayesian and this confirms the formulation of the number of multiplications for each approach that shown before in Chapter II.

CPU Time Comparison During the Minimization Process

◆ Conventional Bayesian

■ Field Scale Bayesian

$y = 2E\text{-}07x^{2.011}$
$R^2 = 1$

$y = 0.001x^{0.9985}$
$R^2 = 0.9994$

CPU time/iteration, sec

Number of Gridblocks
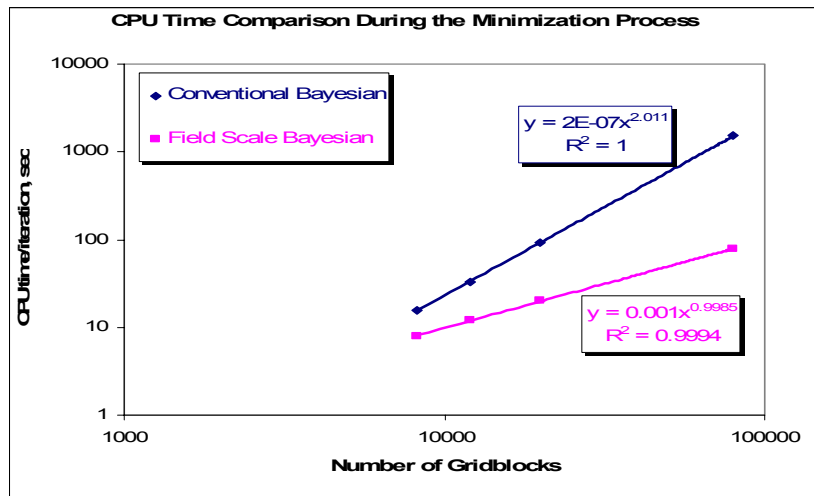
Fig.4.22–CPU time comparison during the minimization process between the conventional and field-scale Bayesian

The second objective of this section is to compare the sensitivity calculation from adjoint method and streamline-based sensitivities. **Fig. 4.23** shows this comparison, where both shows almost linear trend with increasing the model size, however the rate of increase of the CPU time with increasing the model size is small in case of streamline-based sensitivity compared to the adjoint method-based sensitivity. The most important is the CPU time different between the two methods, where the CPU time in case of adjoint method is about two logarithmic cycles more than the streamline-based sensitivities. This is because the adjoint method-based sensitivity using generalized travel time concept depends upon the number of integrated wells which are 5 wells in these synthetic cases and if the number of integrated wells increase, the CPU time different between the both methods will obviously increase.
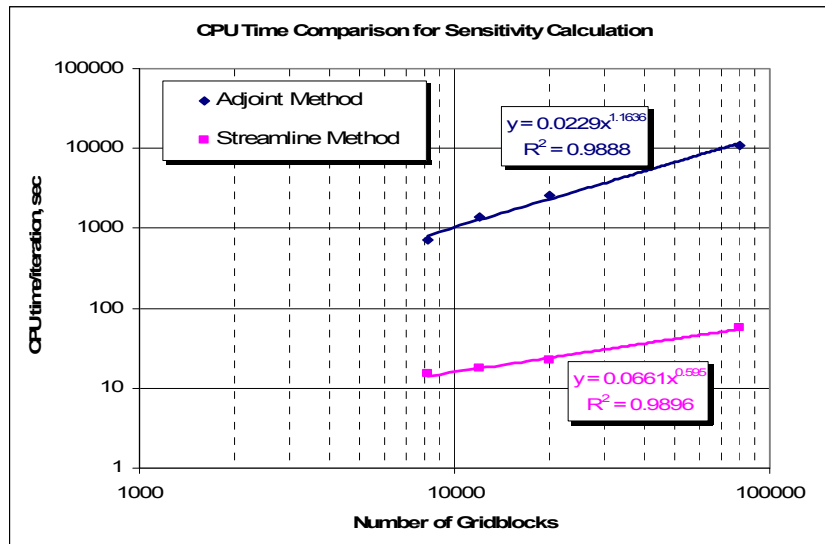
Fig.4.23–CPU time comparison for sensitivity calculation between the adjoint method and streamline-based sensitivity

The third objective of this section is to show the CPU time per iteration for the whole iteration process for both the conventional and field-scale Bayesian using adjoint method-based sensitivity and streamline-based sensitivity, **Fig. 4.24 a**, and **b** show the CPU time comparison for the both sensitivity method, respectively. It is clearly shown that the CPU time per iteration in case of adjoint method-based sensitivity and streamline-based sensitivity gives a linear trend in case of Field-scale Bayesian. While in case of conventional Bayesian the CPU time per iteration in case of streamline-based sensitivity shows a quadratic trend as expected while the adjoint method-based sensitivity shows a nearly linear trend. This is because the CPU timing of the whole process in case of adjoint method is highly affected by the sensitivity calculation timing, this can be noticed form a comparison between the timing taken during the minimization for the conventional Bayesian (**Fig. 4.22**) and that during the sensitivity calculation from the adjoint method (**Fig. 4.23**), thus the total CPU time per iteration for the whole process using adjoint method-based sensitivity is highly affected by the linear trend of the sensitivity calculation, that is why it shows nearly linear trend with increasing the model size. The opposite situation takes place in case of streamline-based sensitivity for

the conventional Bayesian, where most of the operation timing is highly affected by the timing of the minimization process which behaves quadratically with increasing the model size, that is why the streamline-based sensitivity shows quadratic trend in case of conventional Bayesian.

(a) **(b)**
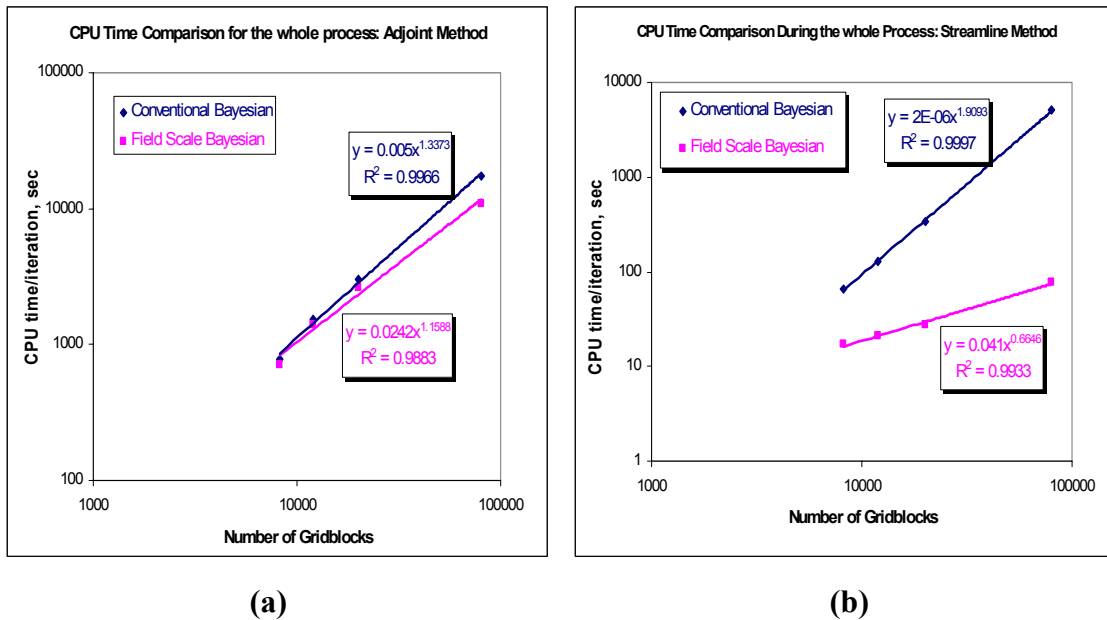
Fig. 4.24–CPU time per iteration for the whole inversion process for (a) adjoint method-based sensitivity; (b) streamline-based sensitivity

4.4 Field Application: Goldsmith San Andreas Unit

In this section we show a field application taken from Goldsmith San Andreas unit in west Texas using the field-scale Bayesian approach with our formulation of the generalized travel time sensitivity from adjoint method and from streamline-based sensitivity.

Goldsmith is a CO_2 pilot project area in the Goldsmith San Andres Unit (GSAU), a dolomite formation in west Texas. The pilot area consists of nine inverted 5-spot patterns covering around 320 acres with average thickness of 100 ft and has over 50 years of production history prior to CO_2 project initiation in Dec 1996. We used the

waterflood production history prior to the CO2 injection. **Fig. 4.25** shows the CO2 pilot project site in the GSAU. The extended study area is shown in **Fig. 4.26** with **11** water injectors and **31** producers. Among the producers within the study area, **9** wells showed significant water-cut response before the initiation of the CO2 injection and are used for data integration. These **9** producers are specified with well name in **Fig. 4.26**. The producing wells have changing productions rates and some producers were shut in and some others are converted to injectors during the production period, a summary of the well schedules indicating infill drilling, well conversions and also well shut in is discussed by He et al.[11] For adjoint method-based sensitivity, it is important to know the well schedules during the production period especially the conversions of the wells from producers to injectors or shut in or for introducing new wells to account for the source and sink term in the adjoint system of equations and in the sensitivity calculation. For streamline-based sensitivity, it is important to account for the changing production rates and different starting times of the injection and production wells by saving the fluxes, pressures and water saturation for sensitivity calculation at the those times from ECLIPSE. We used 11 pressure updates to retrace the streamline and updating the sensitivity calculation from streamline-based sensitivity. The study area is discretized into a 58x53x10 mesh or a total of 30,740 grid cells. The porosity field, **Fig. 4.27**, was generated from log data using sequential Gaussian simulation.[52] It was not allowed to change during the inversion. The initial permeability which is also the prior mean in using the field-scale Bayesian approach is generated via a cloud transform based on porosity-permeability relationship and is given in **Fig. 4.28.**

As we are using the field-scale Bayesian approach with 5x5x5 stencil as an approximation of the square root of the inverse of the covariance during the minimization for both adjoint method-based sensitivity and streamline-based sensitivity, thus it is important to see how accurate is this approximation taken into consideration that the variogram model used is exponential with sill of (Ln k) of 13, and the ranges in the x, y, and z directions are 1000, 1000, and 14 ft respectively. These ranges cover about 10 grid blocks in the x and y direction and about two grid blocks in the z-direction.

Fig. 4.29 shows a comparison between the exact covariance and the covariance obtained from the stencil using the same technique used before in the synthetic cases of the two examples shown in the adjoint method-based sensitivity.
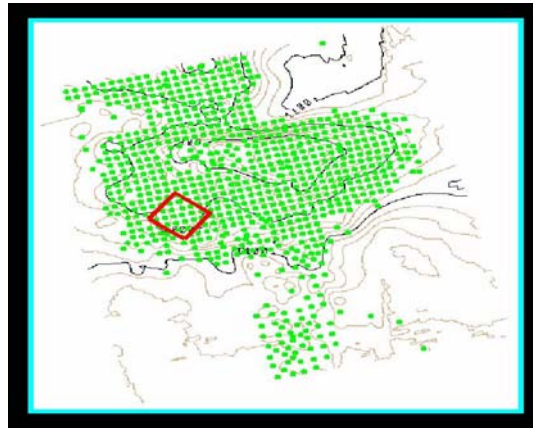
Fig. 4.25–Goldsmith field – Co$_2$ pilot project area
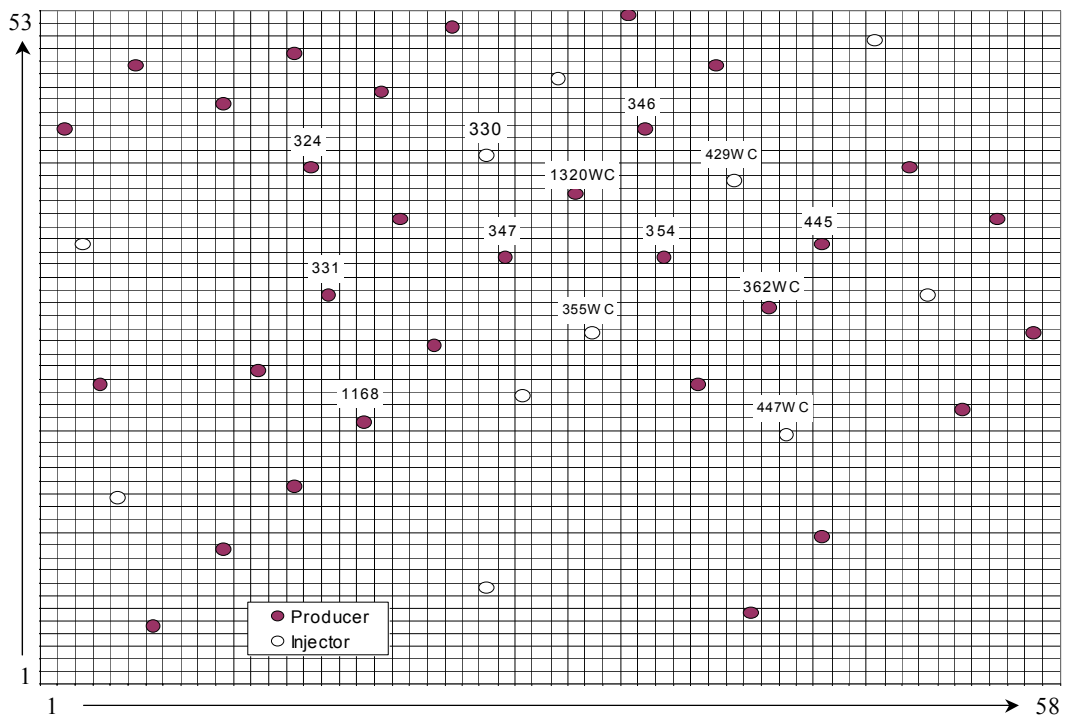
Fig. 4.26–Well configuration of Goldsmith case study area
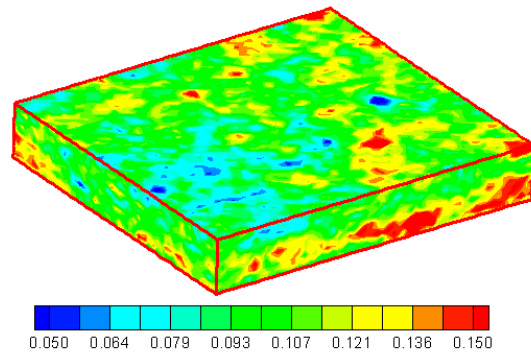
Fig. 4.27–Goldsmith field - porosity distribution
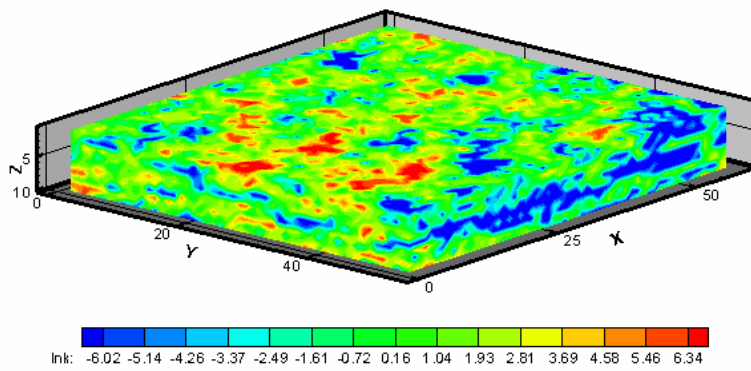
Fig. 4.28–Initial and the prior mean permeability for Goldsmith case in log scale
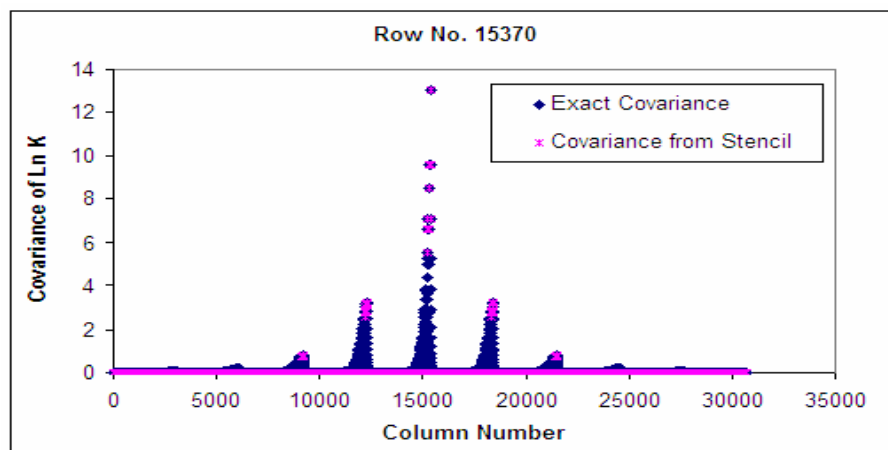
Fig. 4.29–Comparison between exact covariance and covariance from 5x5x5 stencil at row 15370 for Goldsmith case

It can be seen from **Fig. 4.29** that the covariance from the stencil always in a perfect agreement with the covariance of high magnitude, but due to the fact that the 5x5x5 stencil covers only 2 grid blocks in the x, y, z directions only compared to the 10 grid blocks in case of the exact covariance in case of x and y direction, so some of the high values in exact covariance are not in good agreement with the covariance from the stencil. Since the covariance from the stencil covers the same number of grid blocks as that of the exact in the z direction so clearly for better accuracy, the size of the stencil should be increased in the x and y direction to 21x21x5 but this will increase the computation time, so a tradeoff between accuracy and computational efficiency should be taken place to select the most suitable stencil to use. We used 5x5x5 stencil in Goldsmith case during dynamic data integration using both adjoint method-based sensitivity and streamline-based sensitivity.

4.4.1   Field Application: Adjoint Method-Based Sensitivity with Field-Scale Bayesian Approach

Fig. 4.30 a and b shows the water cut response from the initial and final permeability, respectively compared to the observed water cut for the nine producing wells during the **20** years of production history and **Fig. 4.31 a, and b** shows the final and change in the permeability form the initial, respectively. **Fig. 4.32** shows the reduction in the misfits versus the number of iterations during the inversion.

As seen from the water cut match, most of the wells are in good agreement with the observed water cut especially wells **3**, 7, and **9**. From **Fig. 4.31 b,** we can see that most changes in the permeability from the initial are at high **X** and **Y** values which are at the locations of the nine integrated wells.
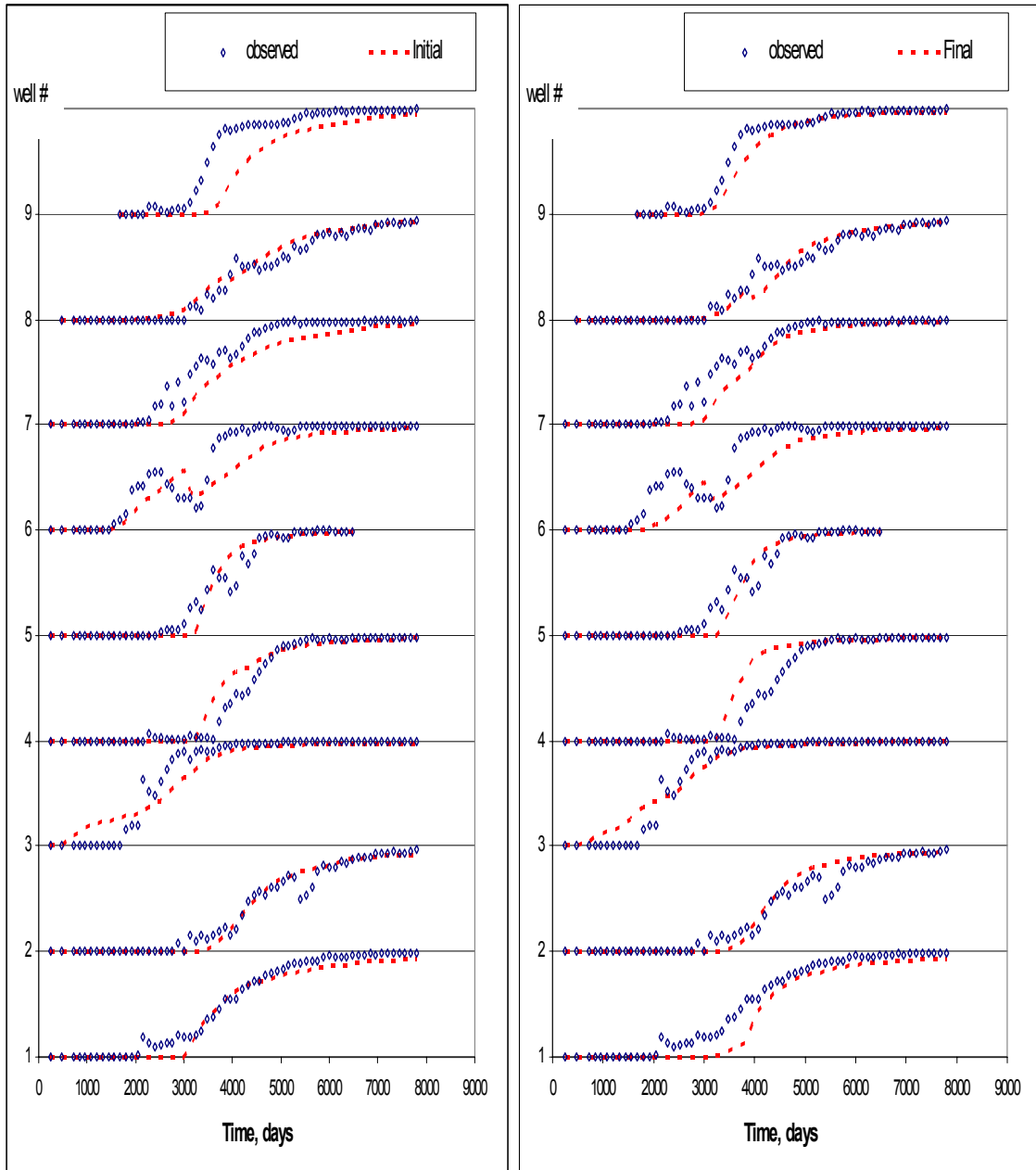
(**a**)   (**b**)

Fig. 4.30–Water cut match for Goldsmith case using adjoint method-based sensitivity, (a) Match from initial permeability model, (b) Match from the final permeability model

(a)   **(b)**
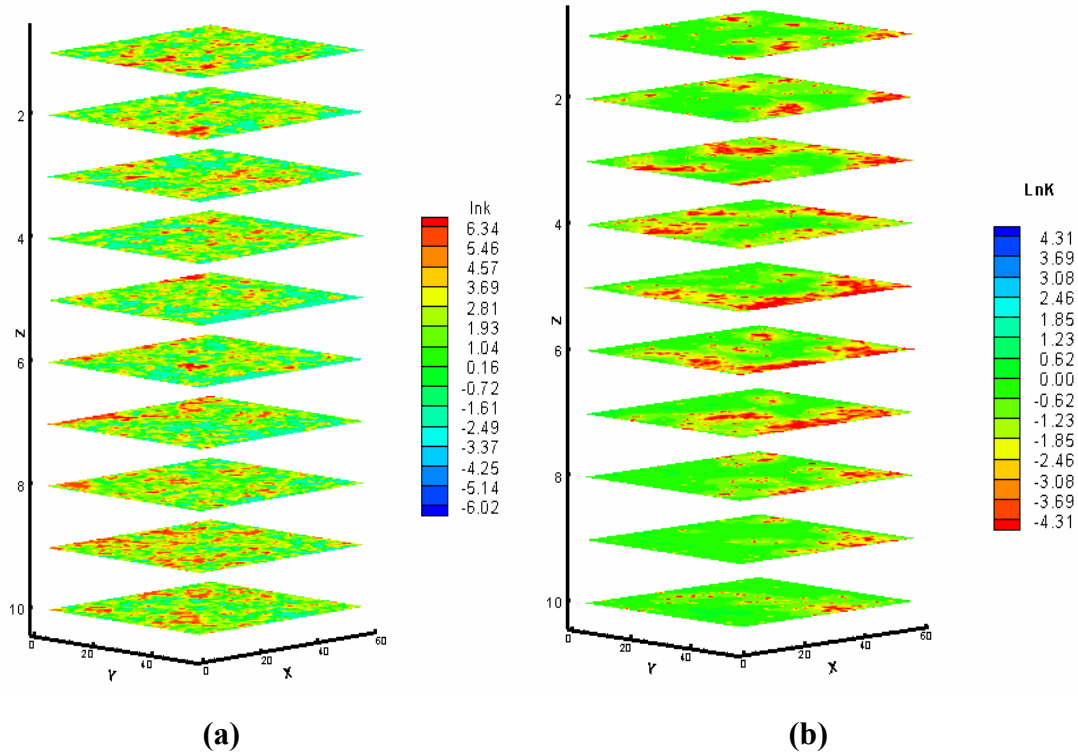
Fig. 4.31−(a) Final permeability, (b) Permeability changes from initial (in Log scale) using adjoint method-based sensitivity
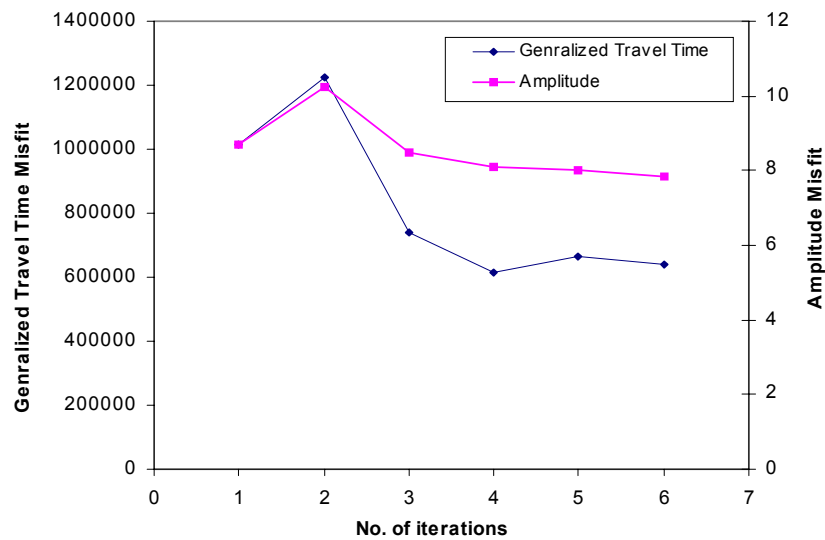
Fig. 4.32−Amplitude and generalized travel time misfit reduction using adjoint method-based sensitivity for Goldsmith case

4.4.2 Field Application: Streamline-Based Sensitivity with Field-Scale Bayesian Approach

Fig. 4.33 a and b show the water cut response from the initial and final permeability, respectively compared to the observed water cut for the nine producing wells during the **20** years of production history and **Fig. 4.34** shows the change in the permeability form the initial, where the initial model used is the same as the initial model used in the adjoint method-based sensitivity and is given in **Fig. 4.28**. **Fig. 4.35** shows the reduction in the misfits versus the number of iterations during the inversion.

As seen from the water cut match, most of the wells are in good agreement with the observed water cut especially wells **1**, **3**, **4**, **7**, and **9**. From a comparison between the changes in the permeability from both adjoint and streamline-based sensitivity (**Fig. 4.31b** and **Fig. 4.34**), we can notice the similarity in the locations of the changes, which indicates that both methods are successful in resolving the changes in the permeability however streamline-based sensitivity is much faster than the adjoint method by several order of magnitude due to the dependency of the adjoint method on the number of integrated wells, where for goldsmith case we need one simulation run and solving 2M by 2M (61480 x 61480) adjoint system of equations N_w (number of integrated wells, 9) times to get the sensitivity per iteration. While for the streamline-based sensitivity, we only need one forward run and analytical sensitivity calculation along the streamline, where all the information required during this calculation can be obtained from one single forward run per iteration, so it is independent on the number of integrated wells and this makes it several orders of magnitude faster than the adjoint method-based sensitivity.
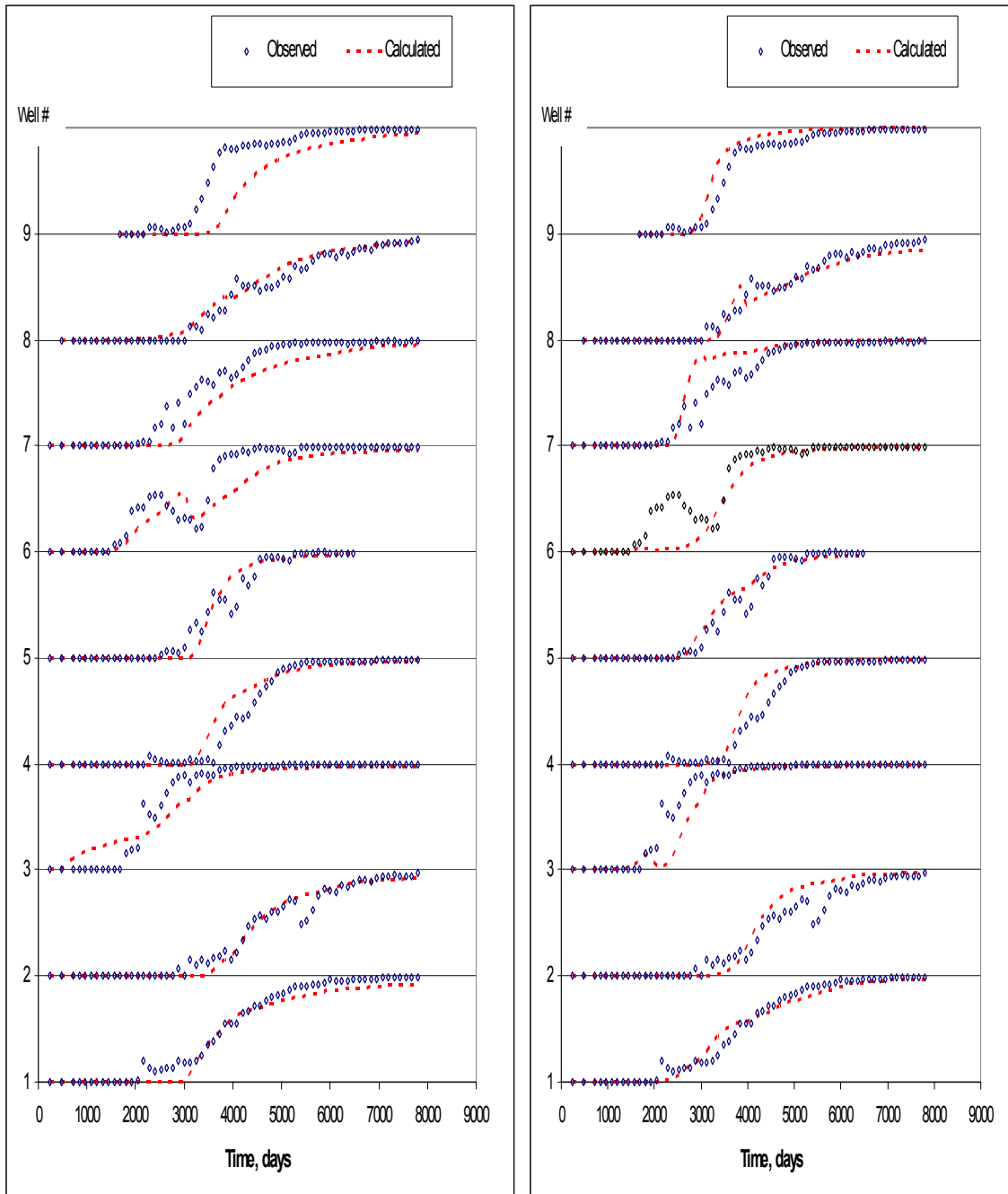
Fig. 4.33–Water cut match for Goldsmith case using streamline-based sensitivity, (a) Match from initial permeability model, (b) Match from the final permeability model
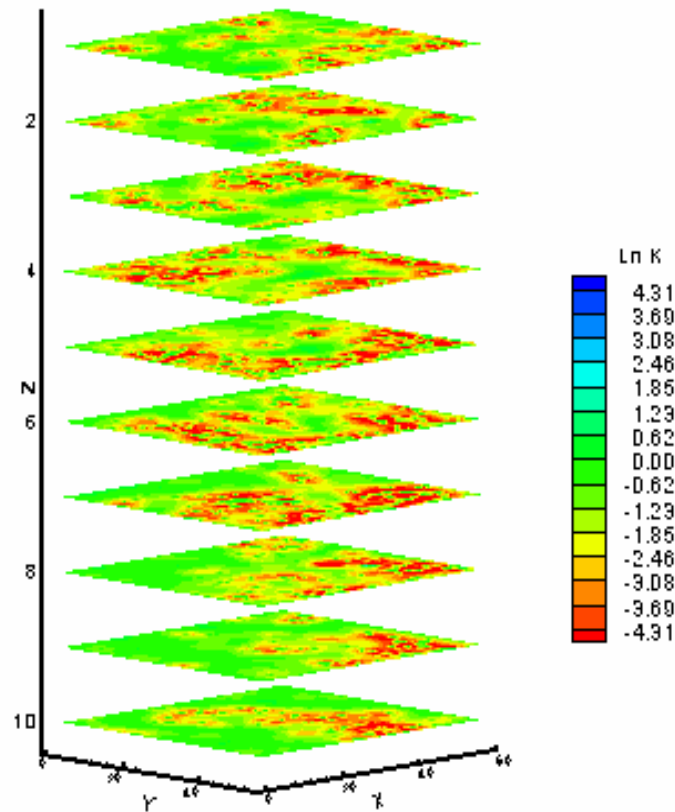
Fig. 4.34–Permeability changes from initial (in Log scale) using streamline-based sensitivity
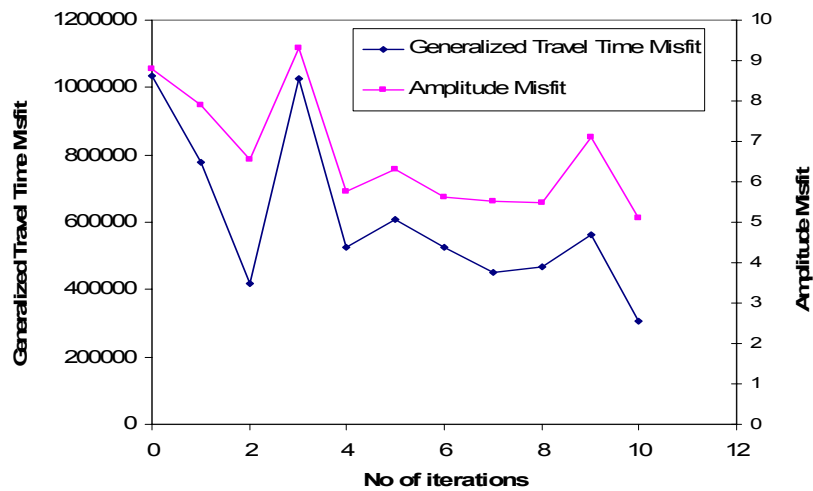
Fig. 4.35–Amplitude and generalized travel time misfit reduction using streamline-based sensitivity for Goldsmith case

4.5 Multiple Realizations from Posterior Distribution for Uncertainty Assessment :
   Goldsmith Field Application

The main objective of this section is to apply our proposed approach that we used in history matching to generate multiple realizations that sample the posterior distribution using one of the approximate sampling methods which is Randomized Maximum Likelihood that discussed before in Chapter II. These realizations can be further used to assess the uncertainty in the production forecast.

In sampling multiple realizations from the posterior distribution we use the field-scale Bayesian approach along with streamline-based sensitivity using generalized travel time inversion concept due to its robustness and less computation time required compared to the adjoint method-based sensitivity. In addition we use the Randomized Maximum Likelihood which accepts all the transition to the new state in the Markov chain, thus reduce the computation time in proposing transitions that have low acceptance rate in the Metropolis Hasting algorithm. We follow the same steps showed before in Chapter II in sampling multiple realizations from the posterior distribution. **The first step**, we used cloud transform based on permeability-porosity relationship to generate four different unconditional realizations of the model parameter. For the four unconditional realization of the permeability field we used the same variogram model used for Goldsmith case, which is the exponential variogram with ranges in the x, y, and z directions, respectively equal to 1000, 1000, 14 ft and with the variance of horizontal log permeability of 13.0 in order to honor the prior information of Goldsmith field. **The second step**, we generated four different realizations of the observed data by adding a randomly generated Gaussian error with a standard deviation of 0.03 to the observed water-cut data. **The last step**, we computed the conditional realization using the same process of history matching the Goldsmith field case that shown in the previous section with the only difference that the regularization is with respect to unconditional realizations of the model and the data instead of the prior model and the observed data. These conditional realizations are considered as a new state in Markov chain, where all are accepted using Randomized Maximum likelihood method.

The resulting unconditioned permeability fields for the four realizations are shown in **Fig. 4.36a** to **Fig. 4.39a**. We used our proposed approach to integrate the entire 7800 days of production history using the four unconditional permeability realizations as the prior means and the four unconditional water cut as the observed data in our proposed field-scale Bayesian approach. **Fig. 4.40 a, and b to Fig. 4.48 a, and b** shows the unconditioned and conditioned water cuts, respectively for each of the four realizations for the nine wells. The corresponding four conditioned permeability fields are shown in **Fig. 4.36 b to Fig. 4.39 b**. **Fig. 4.36 c to Fig. 4.39 c** shows the difference between the unconditional and the conditional realization. Upon comparison of the unconditioned to the conditioned ensemble of water cut realizations, we found that match has been improved compared to the unconditional realizations, however some conditional realizations does not show good match for all the wells this due to the poor unconditional realizations used. So, for better uncertainty assessment in the production forecast, many unconditional realizations should be used so that we can select the best conditional realizations that give good match with the water cut history to be used in quantifying the uncertainty on the production forecast. The other way is to improve the way of generating the unconditional realizations to ensure that many conditional realizations will have good match with the production history. Additional work has been done for improving the unconditional realizations and thus improving the production match from the conditional realizations and uses them for assessing the uncertainty in the production forecast for Goldsmith case, but it is not presented here as the main objective of this section is to test the applicability of our proposed field-scale Bayesian approach along with the generalized travel time sensitivity obtained using streamline-based sensitivity for fast and efficient uncertainty assessment using Randomized Maximum Likelihood.
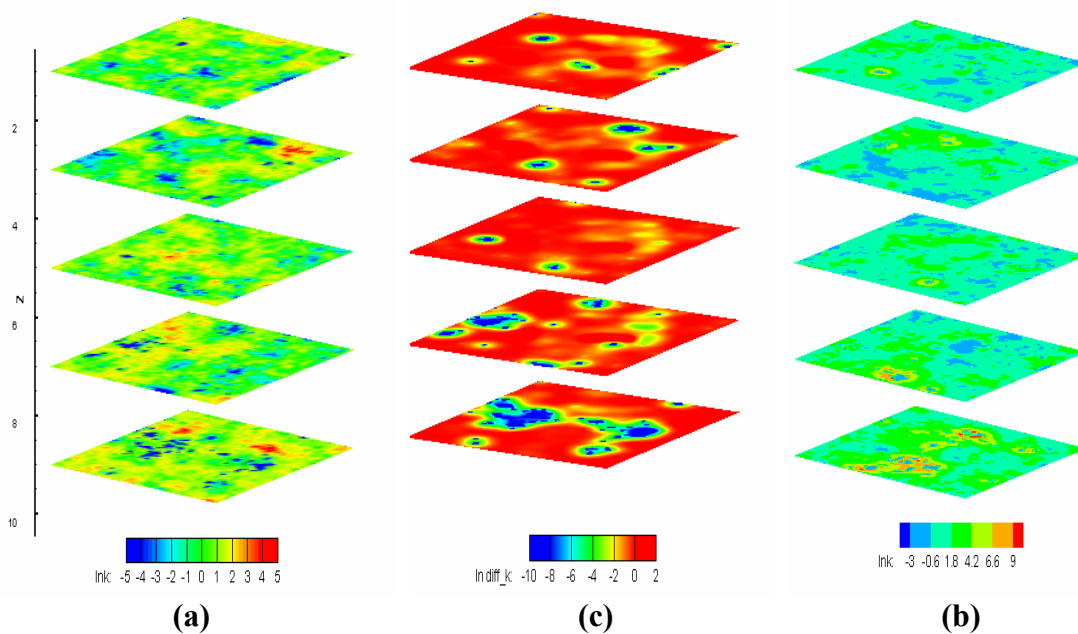
Fig. 4.36–Permeability field, realization 1. (a) Unconditioned, (b) Conditioned, (c) Different. (Five out of ten layers, all in log scale)
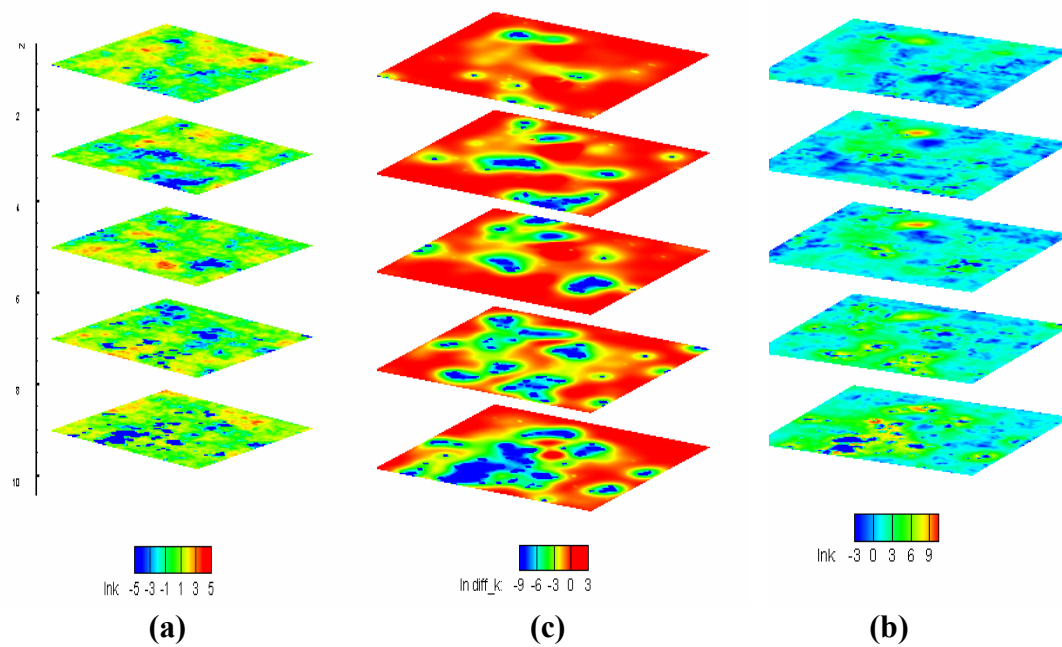
Fig. 4.37–Permeability field, realization 2. (a) Unconditioned, (b) Conditioned, (c) Different. (Five out of ten layers, all in log scale)
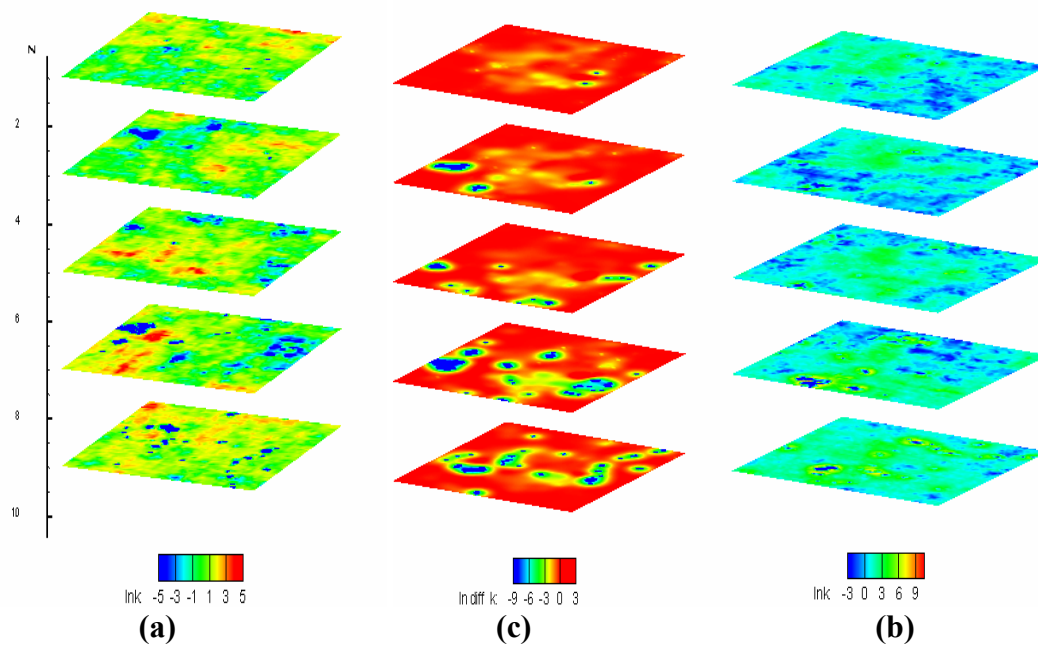
Fig. 4.38–Permeability field, realization 3. (a) Unconditioned, (b) Conditioned, (c) Different. (Five out of ten layers, all in log scale)
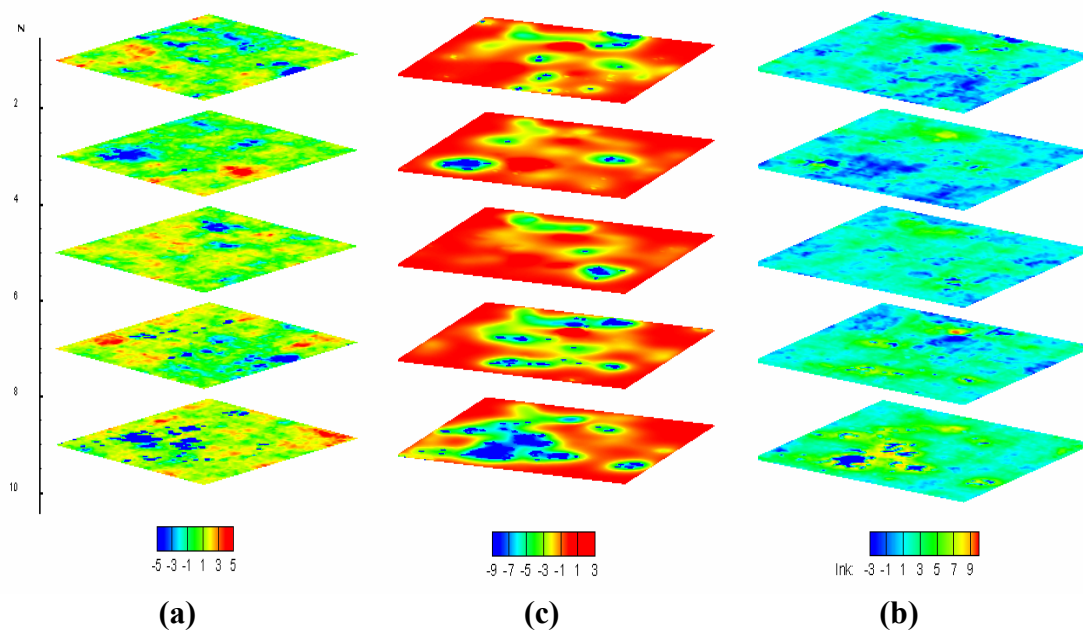
Fig. 4.39–Permeability field, realization 4. (a) Unconditioned, (b) Conditioned, (c) Different. (Five out of ten layers, all in log scale)

(a)   (b)
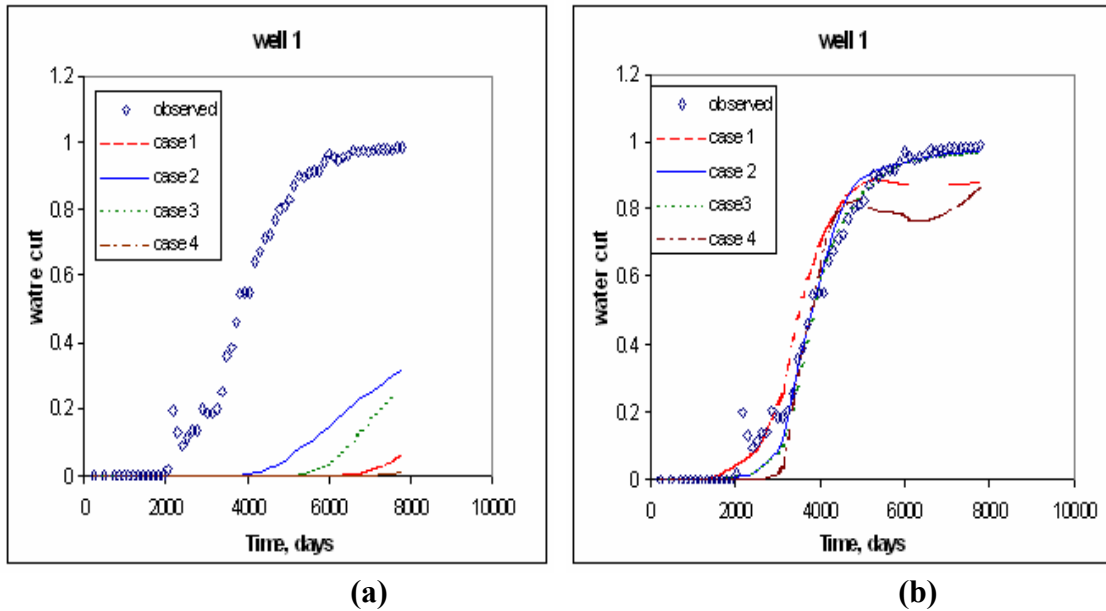
Fig. 4.40–Water cut, well 1. (a) Unconditioned, (b) Conditioned

(a)   (b)
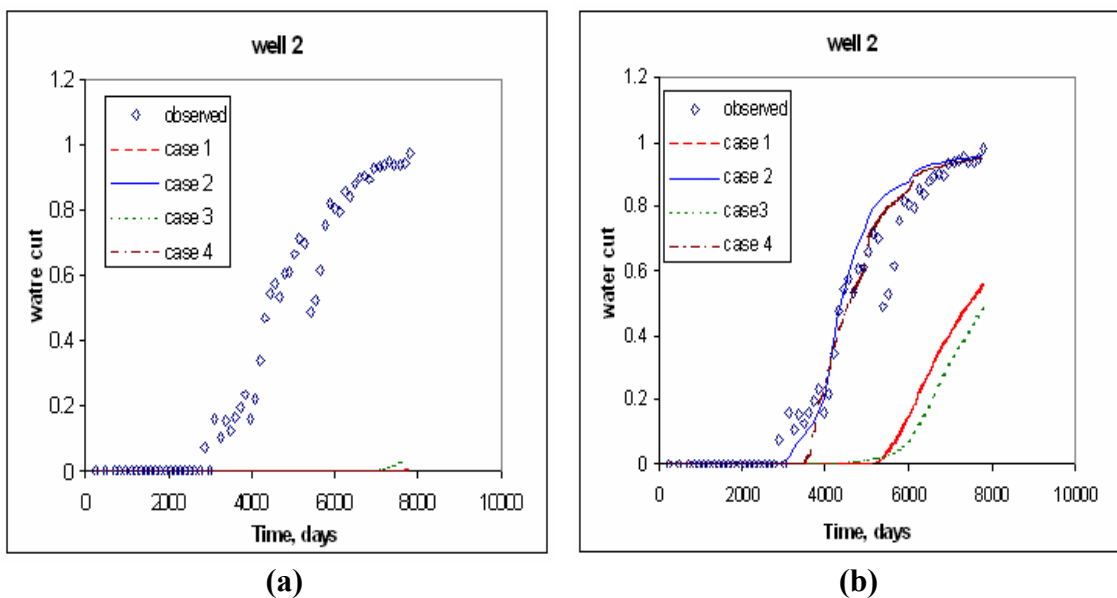
Fig. 4.41–Water cut, well 2. (a) Unconditioned, (b) Conditioned

(a)   (b)
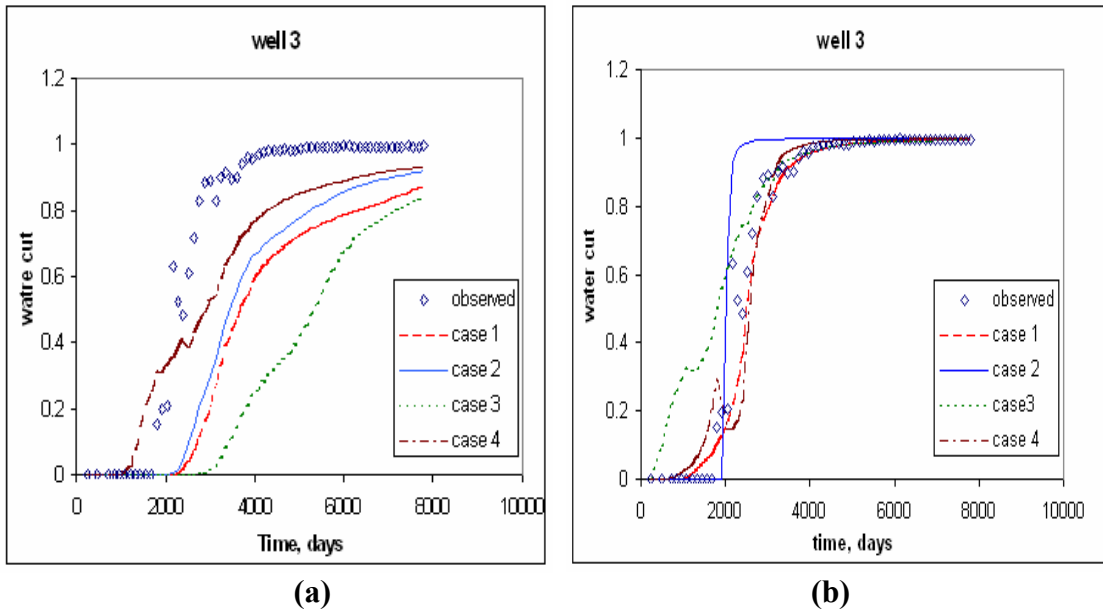
Fig. 4.42−Water cut, well 3. (a) Unconditioned, (b) Conditioned

(a)   (b)
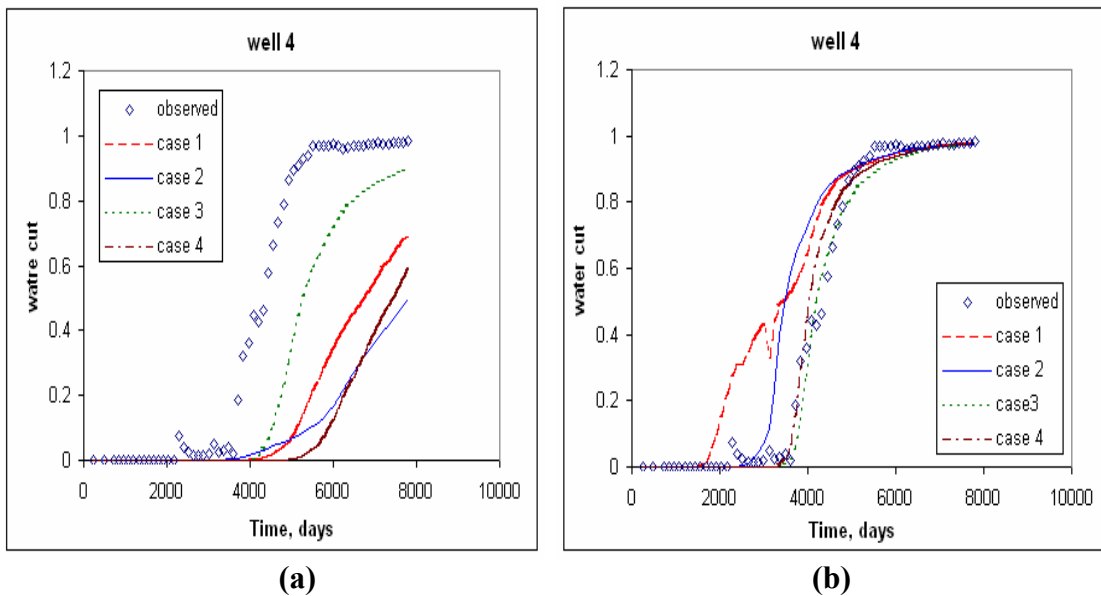
Fig. 4.43−Water cut, well 4. (a) Unconditioned, (b) Conditioned

(a)   **(b)**
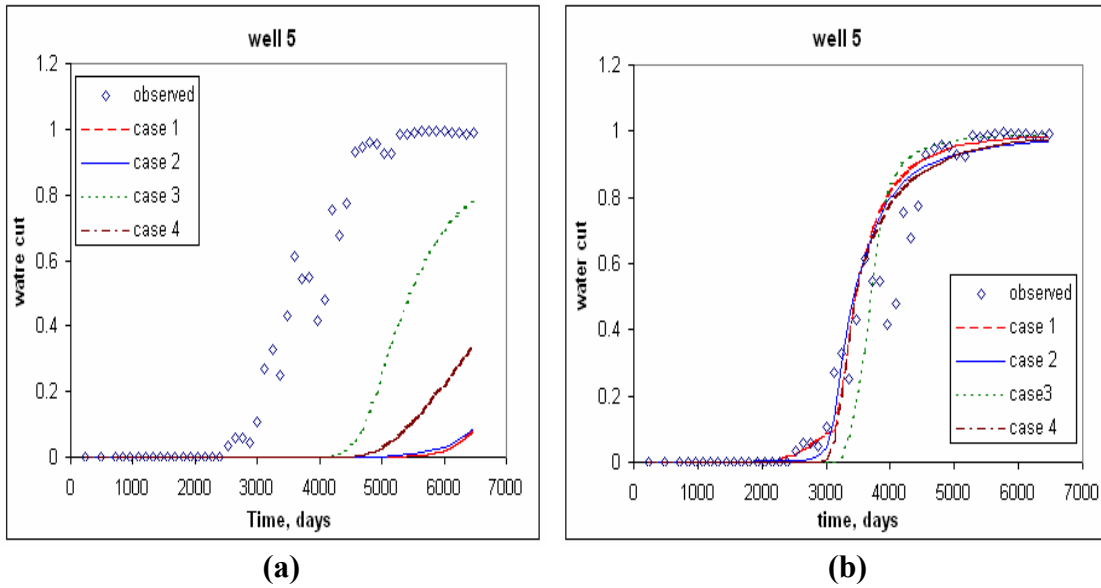
Fig. 4.44–Water cut, well 5. (a) Unconditioned, (b) Conditioned

(a)   **(b)**
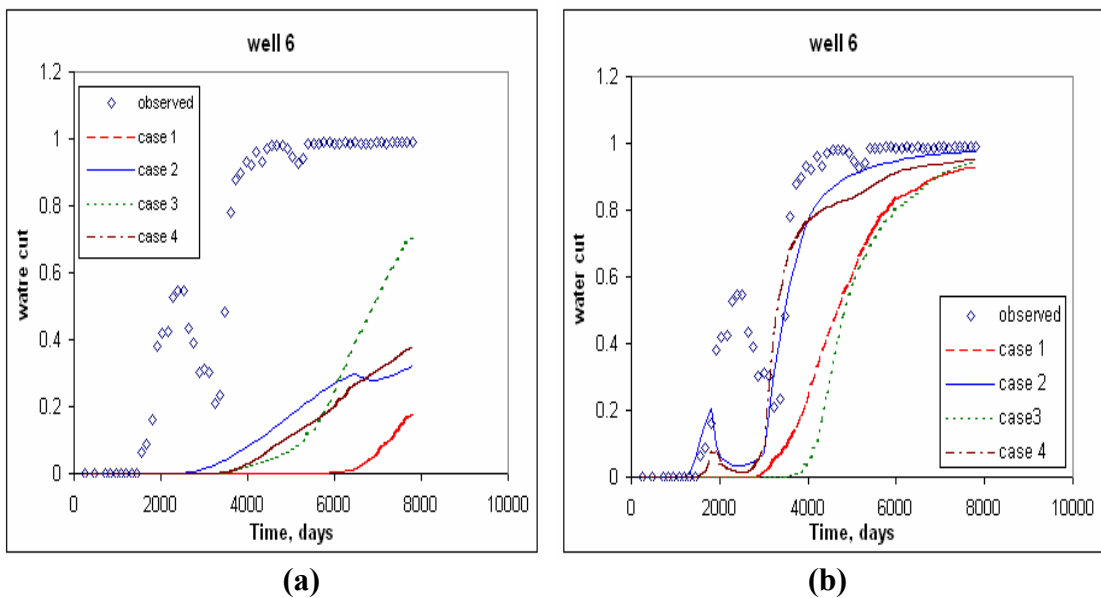
Fig. 4.45–Water cut, well 6. (a) Unconditioned, (b) Conditioned

(a)   (b)
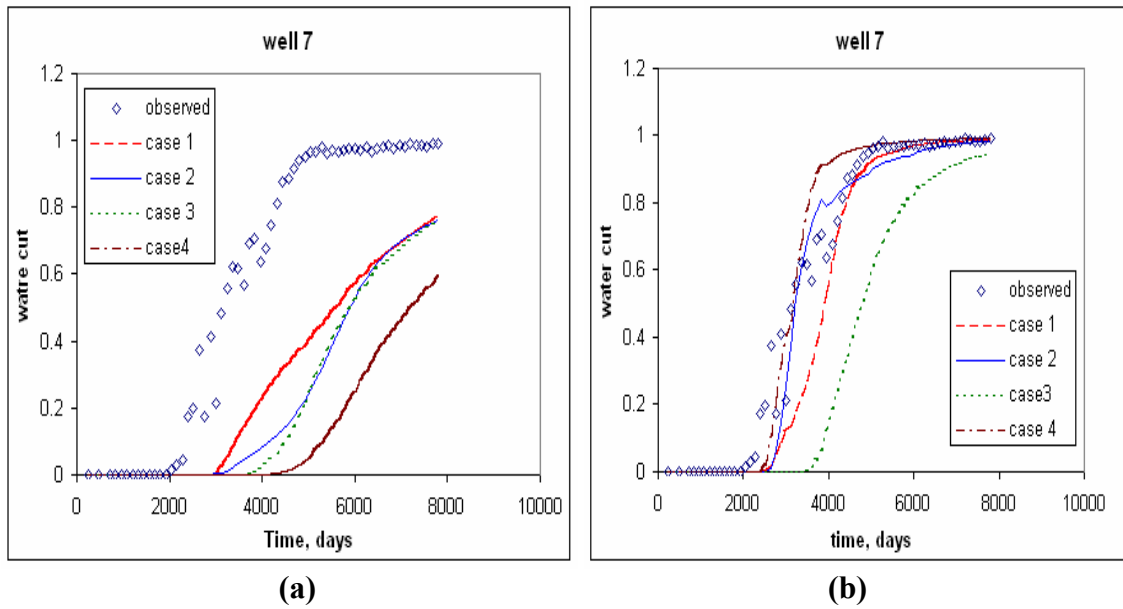
Fig. 4.46–Water cut, well 7. (a) Unconditioned, (b) Conditioned

(a)   (b)
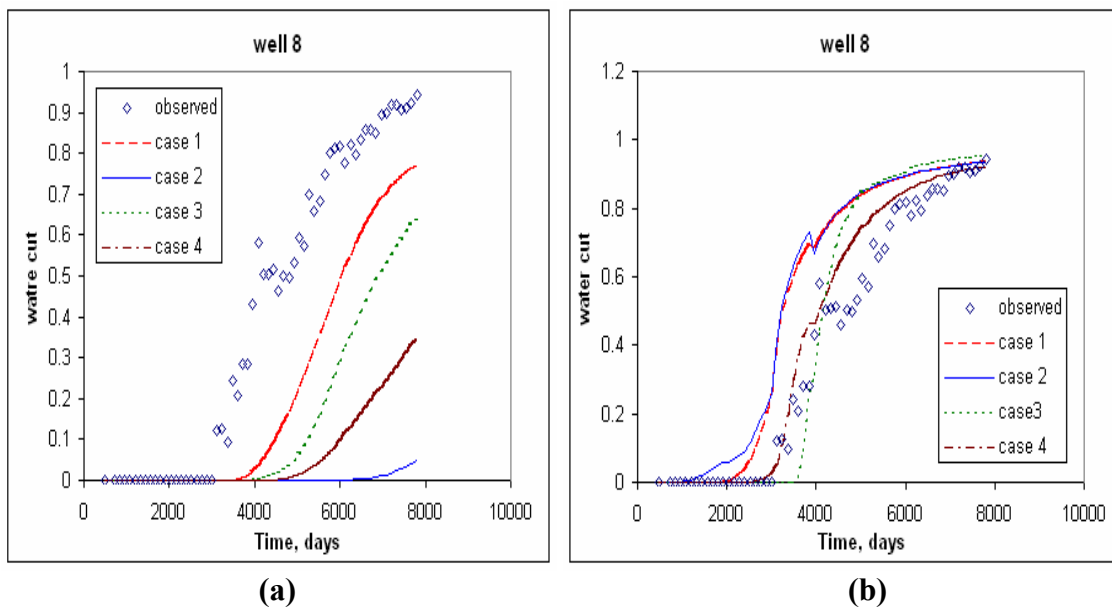
Fig. 4.47–Water cut, well 8. (a) Unconditioned, (b) Conditioned
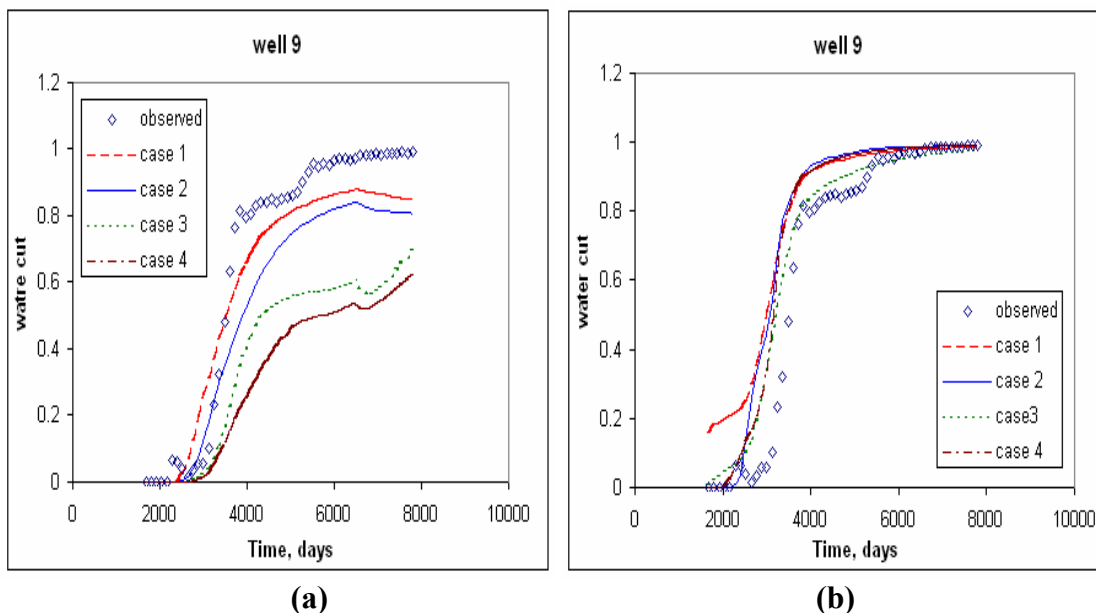
Fig. 4.48–water cut, well 9. (a) Unconditioned, (b) Conditioned

4.6 Chapter Summary

This chapter presents applications from automatic history matching in Bayesian frame work using two different ways of sensitivity calculations, adjoint method-based sensitivity and streamline-based sensitivity, and two different approaches during the minimization process, namely the conventional Bayesian with full covariance and the field-scale Bayesian with an approximation of the square root of the inverse of the covariance required by LSQR using numerical stencil. We first present a comparison of the sensitivity of travel time with respect to permeability form perturbation with our formulation using adjoint method-based sensitivity for 3D two phase flow. Then, we tested the robustness and the utility of our proposed field-scale Bayesian approach along with the conventional Bayesian approach using synthetic cases with a commercial finite difference simulator (ECLIPSE) as forward model and generalized travel time sensitivity using adjoint method. The use of commercial finite difference simulator as a forward model extends the application of generalized travel time sensitivity using adjoint method to more practical applications. The third part of this chapter shows the CPU time scaling comparison between our proposed field-scale Bayesian approach and the conventional

Bayesian approach for increasing the number of model parameters and the results shows the linear scaling of the field-scale Bayesian compared to the quadratic scaling of the conventional Bayesian, this leads the field-scale Bayesian approach to be well-suited for field-scale applications. The fourth part shows field application from Goldsmith San Andreas unit using our proposed field-scale Bayesian approach along with generalized travel time sensitivity using both adjoint and streamline-based sensitivity with commercial finite difference simulator (viz ECLIPSE) as the forward model. To the best of our knowledge, this is the first field case application using our proposed field-scale Bayesian approach with generalized travel time sensitivity obtained from both adjoint method and streamline-based sensitivity and with finite difference simulators as the forward model. The last part of this chapter shows the applicability of our approach for uncertainty assessment by generating multiple realizations using RML to be used for the uncertainty quantification of the production forecast for Goldsmith field case.

CHAPTER V
CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

Automatic history matching in Bayesian framework especially for field-scale applications requires the following: **first,** a stable and general forward model that can handle field applications with complex physical mechanism, **second,** a computationally efficient way for representing the data misfit in the objective function, **third,** an efficient way of including the regularization term in the objective function for stable inversion, and **finally,** a proper optimization method that is well-suited for field-scale applications.

In this study we proposed an approach to improve all the above factors for fast and efficient automatic history matching in a Bayesian framework. **First**, we used a commercial finite difference simulator (viz ECLIPSE) to model fluid flow in the porous media. The simulator is general and can account for complex physical behavior that dominates most of the field applications. **Second**, the data misfit is represented by a single generalized travel time misfit for each well, thus reducing the number of data points into one per well and at the same time ensuring the matching of the entire production history. This saves computation time required during the minimization and makes this approach well-suited for field-scale applications. In addition, using the generalized travel time misfit reduces the computational effort of calculating the sensitivities required by any gradient-based optimization algorithm. We have used both adjoint method-based sensitivity and streamline-based sensitivity during this study. **Third**, we proposed a field-scale Bayesian approach that utilizes an approximation of the square root of the inverse of the covariance using numerical stencil. This leads to large savings in computation time and memory compared to the calculation of the full covariance required by the conventional Bayesian approach. **Finally**, we used the LSQR method as the sparse matrix solver for updating the model parameters during minimization. The approach is stable when dealing with large-scale applications and also using LSQR along with the approximation of the square root of the inverse of the

covariance using numerically derived stencil shows a linear trend with respect to the increase in the model parameters compared to the quadratic scaling of the conventional Bayesian with the full covariance matrix. This makes the field-scale Bayesian approach well-suited for large-scale field applications. We applied our approach on different synthetic cases and field case from Goldsmith San Andreas unit to demonstrate the applicability of our approach for history matching and also for generating multiple realizations for uncertainty assessment.

The major conclusions of this study are summarized as follows:

1- Use of a commercial finite difference simulator as the forward model

The use of commercial finite difference simulator (ECLIPSE) during this study as a forward model helps in obtaining a more general and stable solution especially in dealing with field-scale applications with complex physical mechanisms. For adjoint method-based sensitivity, we save the pressure, water saturation, and bottom hole pressure at each time step, which are required during the sensitivity calculations. For streamline-based sensitivity, we utilize the pressure, water saturation, and fluxes at each pressure update time to retrace the streamline and update the sensitivity calculation.

2- Formulation of the generalized travel time and travel time sensitivity with respect to permeability using adjoint method for 3D, two phase flow

The generalized travel time and travel time inversion provide a unique advantage over the conventional amplitude inversion as it depends only on the number of wells and not the number of data points as in amplitude inversion. This makes the generalized travel time sensitivity more computationally efficient than the conventional amplitude sensitivity especially in using adjoint method-based sensitivity. This makes it well-suited for field-scale applications. The source term in the adjoint system of equation has been formulated to account for the generalized and travel time sensitivity for 3D, two phase flow problems.

3- Formulation of the field-scale Bayesian approach with approximate calculation of the square root of the inverse of the covariance using a numerically-derived stencil

The Bayesian approach is formulated in the form of a system of equations and we use the LSQR method as a sparse matrix solver. The approach is practical for large-scale problems for updating the model parameters by knowing the square root of the inverse of the covariance matrix. The numerically-derived stencil, which is applicable for any covariance model, is used to approximate the square root of the inverse of the covariance using 5x5x5 stencil. The use of 5x5x5 stencil reduces the computation time and memory required compared to the conventional Bayesian with full covariance, thus making the field-scale Bayesian well-suited for field-scale applications.

4- CPU time scaling with respect to the number of grid blocks using both conventional and field-scale Bayesian approach

Two formulas for the number of multiplications required by the conventional and field-scale Bayesian approach during the minimization step have been developed. The conventional Bayesian approach shows a quadratic behavior while the field-scale Bayesian approach shows a linear behavior with respect to the number of grid blocks. Four synthetic cases with number of gridblocks ranging from 8000 to 80,000 are used to validate the above results by comparing the CPU time per iteration during the minimization step for the two approaches. This proves the computational efficiency of the field-scale Bayesian for large-scale applications.

5- Synthetic and field applications

Different 3D synthetic examples used to compare the accuracy of the estimate from both the conventional and the field-scale Bayesian approach and at the same time to test the applicability of our formulation of the generalized travel time sensitivity for 3D, two phase flow using adjoint method-based sensitivity. The results show the success of our sensitivity formulation in directing the objective function towards the minimum and also

the estimate from the both approaches look very similar; however the computation time for the field-scale Bayesian is order of magnitude less than the conventional Bayesian especially for large-scale applications where the model size exceeds thousands to millions gridblocks.

Goldsmith San Andreas unit in west Texas is used to test the applicability of our approach for field-scale applications using both adjoint and streamline-based sensitivity. The results demonstrate the practical feasibility of our approach with significant improvement in the water cut match from most of the producing wells.

6- Sampling multiple realizations from the posterior distribution for uncertainty assessment

One of the practical applications of the Bayesian approach other than history matching is uncertainty assessment. We used our field-scale Bayesian approach along with the generalized travel sensitivity calculated using streamlines with commercial finite difference simulators (ECLIPSE) as forward model. Multiple realizations were generated from the posterior distribution using Randomized Maximum likelihood (RML) for the Goldsmith field case to be further used in the uncertainty assessment of the production forecast. Thus, get the benefit of our proposed approach that is computationally efficient for field-scale applications and use the Randomized Maximum Likelihood that reduces the computational burden by avoid rejecting any proposed state in the Markov chain.

5.2 Recommendations

The following recommendations are suggested to improve our current approach:

1- The current field-scale Bayesian approach suffers from two limitations in assuming that 5x5x5 stencil is a good approximation for the covariance matrix. **First**, the calculation requires getting the square root of the inverse of 5x5x5 stencil, which is a 125x125 matrix using spectral value decomposition, where the eigen values and eigen vectors of the 125x125 matrix have to be obtained to get

the square root of the inverse and this computation might be computationally inefficient for using more than 5x5x5 stencil. **Second**, using LSQR, which is an iterative solver, for model updating depends upon the number of iteration which is subjective and inaccurate estimation might take place if the number of iterations selected is not sufficient enough, in addition to the time it takes during these iterations compared to the exact calculation of the model update using the conventional Bayesian approach with modified Gauss-Newton as the optimization algorithm. To overcome these limitations, we suggest an approach that gets the benefit of using the numerical stencil to get an approximation of the covariance matrix instead of the inverse of the covariance matrix. Thus there is no need to use the spectral value decomposition to get the approximation of the square root of the inverse. **Second**, gets the benefit of the modified Gauss-Newton by exactly calculating the model update using the approximate of the covariance matrix from the numerical stencil. This is computationally efficient if using the generalized travel time inversion concept, where we only need to get the inverse of N_wxN_w matrix as given by **Eq. 2.25**. So, the steps for applying this approach will be as follows:

i-   Use 5x5x5 stencil and calculate the 125x125 covariance matrix

ii-   Use column 63, which is the middle column of the 125x125 covariance matrix to get the stencil. This is similar to use column 63 of the square root of the inverse of the 125x125 matrix

iii-   Populate the exact size of the covariance matrix using the stencil calculated from step 2 and save only the non-zero values of the exact size of the covariance. This is the same way of populating the exact size of the square root of the inverse of the covariance matrix that is used in the current approach

iv-   Use Modified Gauss-Newton with the approximation of the covariance from step 3 to get an exact solution of the model update.

In fact, on checking the accuracy of the 5x5x5 stencil for the synthetic and field cases used in this study, we applied the first three steps above to get the covariance from the stencil and compare it with the exact covariance. So, the only change required is to modify the current code of the Modified Gauss-Newton to get the non-zero values of the covariance from the stencil instead of calculating the full covariance exactly.

2- In our current approach, we used 5x5x5 stencil to approximate the covariance matrix; however selecting the best size of the stencil depends upon the ranges and the grid block size used to model the reservoir under study and this may vary from one reservoir to another. So, sensitivity study should be done to select the best stencil size to represent the reservoir, which is a tradeoff between the accuracy and the computational efficiency and it depends upon personal judgment.

3- In generating multiple realizations for uncertainty assessment of Goldsmith case, some conditional realizations show good match with production history and some do not. So, we suggest either to generate a lot of priors so that we can have variety to select only the conditioned realizations that show better match or to improve the priors used to obtain conditional realizations that most of them will match well the production history to be further used for uncertainty assessment of the production forecast.

4- From a preliminary result of solving the adjoint system of equations at one particular time step using LSQR and Bi-conjugate gradient method which is currently used for large-scale problems, we found that the LSQR gives more stable results within few iterations compared to the Bi-Conjugate gradient. So, we highly suggest using LSQR as a solver in solving the adjoint system of equations backward in time especially in large-scale problems.

5- In comparing the travel time sensitivity from adjoint method with perturbation, we found that including the injectors in the adjoint system of equations and in the sensitivity calculation always overestimate the sensitivity compared to the

perturbation due to the reason mentioned before in Chapter IV. Further study should be done to investigate this problem.

6- During our work with the adjoint method-based sensitivity, it was found that using large time step higher than the time step used in the forward simulator in solving the adjoint system of equations backward in time does not have much effect in the final estimate. Clearly additional study should be done to select the best time step to be used during solving the adjoint system of equations. This will have a tremendous saving in computation time compared to the current approach, where we are solving the adjoint system of equations backward in time using the same time steps used in the forward simulator.

7- Streamline-based sensitivity has unique advantage in term of its fast sensitivity calculation compared to the adjoint method-based sensitivity. However, in highly depleted reservoirs where frequent pressure updates are required for accurate sensitivity calculation, streamline might not be the good candidate. In adjoint method-based sensitivity, however we are using the generalized travel time inversion, which makes the adjoint method-based sensitivity depends only on the number of wells compared to the conventional approach which depends on the number of data points, still its computation time is unsatisfactory especially for large-scale applications. So, we suggest using the same concept of generalized travel time and using the adjoint method to obtain the gradient of objective function as has been done in the past, thus we need only one simulation run and one solution of adjoint system of equations. Using the generalized travel time inversion will reduce the computational burden during the minimization and using the gradient of the objective function will allow us to solve the adjoint system of equation backward in time only once to get the gradient of the objective function. The draw back of this approach is the rate of convergence of the optimization algorithms that used the gradient of the objective function, like conjugate gradient or LBFGS will be small compared to Newton-type of search algorithms like Gauss-Newton or Modified Gauss-Newton used in this study.

NOMENCLATURE

$\Delta\widetilde{t}$	= Vector of Generalized travel time shift
Λ	= Diagonal matrix whose diagonal are the eigen values of the covariance matrix
α	= Damping factor used in Marquardt Levenberg algorithm
τ	= Time of flight
π_i	= Probability of sample state m^i from the posterior
$\alpha_{i,j}$	= Probability of accepting transition in the Markov Chain
σ_j^2	= Error variance of generalized travel time at well j
μ_m	= Viscosity of (m) phase, m stands for oil and water
γ_m	= Specific weight of (m) phase, m stands for oil and water
λ_m	= Lagrange multipliers for phase m, m stands for oil and water
ρ_m	= Density of phase m, m stands for oil and water
ρ_{msc}	= Density of phase m at standard conditions
λ_t	= Total mobility ratio
Δt	= Time step
Δt_j	= Time shift at well j
$\Delta x_i, \Delta y_j, \Delta z_k$	= Cartesian grid block sizes
A_m	= Accumulation term of phase m, m stands for oil and water
B_m	= Formation volume factor, m stands for oil and water
C_d	= Data covariance matrix
C_K	= Prior covariance matrix of permeability
C_M	= Prior covariance matrix of the model parameter
C_ϕ	= Prior covariance matrix of porosity
$C_{\phi,K}$ or $C_{k,\phi}$	= Cross covariance between porosity and permeability
D	= Depth

d_{obs}	=	Column vector with observed data
d_u	=	Unconditional realization of data
e	=	Residual of the objective function $O(m)$
$f(d_{obs})$	=	Marginal probability distribution
$f(d_{obs}/m)$	=	Likelihood probability distribution given the prior distribution
$f(m)$	=	Prior probability distribution
$f(m/d_{obs})$	=	Posterior probability distribution given the observed data
$f^l_{wcal,j}$	=	calculated water cut at well j and at time step index l
f_m	=	flow term of phase m, m stands for oil and water
F_w	=	Fractional flow of water
g	=	Source term in the adjoint system of equations
$g(m)$	=	Column vector with calculated reservoir performance data
G_l	=	Sensitivity matrix
H	=	Hessian of the objective function $O(m)$
I	=	Identity matrix
J	=	Jacobian of the objective function $O(m)$. Gradient of e
J_p	=	Production data misfit
J_{pj}	=	Production data misfit at well j
$J_{\Delta\tilde{t}}$	=	Generalized travel time misfit
K	=	Permeability
K_{rm}	=	Relative permeability to phase (m); m stands for oil or water
K_x, K_y, K_z	=	Permeability in the x, y, and z direction
L	=	Last time step (last data point)
M	=	Number of model parameters
m	=	Column vector of the reservoir parameter
MAP	=	Maximum a Posteriori estimate
MC	=	Markov Chain
$MCMC$	=	Markov Chain Monte Carlo
M_o, M_w	=	Oil and Water Mobility

m_p	= Column vector with prior knowledge of reservoir parameter
m_u	= Unconditional realization of reservoir model parameters
n_d	= Number of data points
N_d	=Number of data points
N_{dj}	= Number of data points at well j
N_{iter}	=Number of iteration inside the LSQR loop
N_s	= Maximum number of stencil
n_w	= Number of wells
N_w	= Number of wells
N_x, N_y, N_z	= Number of grid blocks in the x, y, and z direction
$O(m)$	= Objective function of Bayesian formulation
P	= Pressure
$P^l_{wf,j}$	= Bottom hole pressure at well j and time step index l
q_{ij}	= Probability of proposing transition to another state in the Markov Chain
q_m	= rate of m phase, m is for oil and water
$\hat{q}_m$	= rate per bulk volume, m stands for oil and water
R^2	= Coefficient of determination
RML	= Randomize Maximum Likelihood
s	= Slowness
$S_{k,j}$	= Skin factor at well j and layer k
S_m	= Saturation of m phase, m stands for water and oil
t	= Time
$T_{i,j}$	= travel time at well j and observed point i
T_{mx}	= Transmissibility of m phase in the x-direction, m stands for oil and water
T_{my}	= Transmissibility of m phase in the y-direction, m stands for oil and water

T_{mz}	= Transmissibility of m phase in the z-direction, m stands for oil and water
U	= Matrix whose column are the eigen vectors of the covariance
V_b	= Bulk volume
$w_{i,j}$	= Data weight for each data point (i) and at well (j)
$WI_{k,j}$	= Well index at well j, produced from layer k
y_j^{cal}	= Calculated data at well j
y_j^{obs}	= Observed data at well j
$\bar{y}_j^{obs}$	= Average of observed data
$Z_{fieldscalebayesian}$	= Number of multiplications in field-scale Bayesian formulation
Z_{GN}	= Number of multiplications in Gauss-Newton Iteration
ϕ	= Porosity

REFERENCES

1. Scales, J.A., Smith, M.L., and Treitel, S.: *Introductory Geophysical Inverse Theory,* Samizdat Press, Golden, Colorado (1997).

2. Gavalas, G.R., Shah, P.C., and Seinfeld, J.H.: "Reservoir History Matching by Bayesian Estimation," paper SPE 5740 presented at the SPE-AIME 4th Symposium on Numerical Simulation of Reservoir Performance, Los Angeles, February 19-20, 1976.

3. Oliver, D.S.: "Incorporation of Transient Pressure Data into Reservoir Characterization," *In Situ* (1994) 243.

4. Reynolds, A.C., He, N., and Oliver, D.S.: "Reducing Uncertainty in Geostatistical Description with Well Testing Pressure Data," *Proc.,* International Reservoir Characterization Conference, Houston, March 2-4 (1997).

5. Wu, Z., Reynolds, A. C. and Oliver, D. S.: "Conditioning Geostatistical Models to Two-Phase Production Data," *SPEJ* (June 1999**)** 142.

6. Li, Ruijian, Reynolds, A. C., and Oliver, D. S.: "History Matching of Three-Phase Flow Production Data," paper SPE 66351 presented at the 2001 Reservoir Simulation Symposium, Houston, February 11-14.

7. Zhang, F., Skjervheim, J. A., Reynolds, A. C., and Oliver, D. S.: "Automatic History Matching in a Bayesian Framework, Example Applications," paper SPE 84461 presented at the 2003 SPE Annual Technical Conference and Exhibition, Denver, October 5-8.

8. Wu, Z. and Datta-Gupta, A.: "Rapid History Matching Using a Generalized Travel Time Inversion Method," *SPEJ* (June 2002) 113.

9. Vasco, D.W., and Datta-Gupta, A.: "Integrating Multiphase Production History in Stochastic Reservoir Characterization," *SPEFE* (September 1997) 149.

10. Vasco, D.W., Yoon, S., and Datta-Gupta, A.: "Integrating Dynamic Data Into High-Resolution Reservoir Models Using Streamline-Based Analytic Sensitivity Coefficients," *SPEJ* (December 1999) 389.

11. He, Z., Datta-Gupta, A., and Yoon, S.: "Streamline-Based Production Data Integration with Gravity and Changing Field Conditions*," SPEJ* (December 2002) 423.

12. Cheng, H., Kharghoria, A., He, Z., and Datta-Gupta, A.: "Fast History Matching of Finite-Difference Models Using Streamline-Derived Sensitivities," paper SPE 89447 presented at the 2004 SPE/DOE Fourteenth Symposium on Improved Oil Recovery, Tulsa, April 17-21.

13. Cobenas, R.H., Aprilian, S.S., and Datta-Gupta, A.: "A Closer Look at Non-Uniqueness during Dynamic Data Integration into Reservoir Characterization," paper SPE 39669 presented at 1998 SPE/DOE Improved Oil Recovery Symposium, Tulsa, April 19-22.

14. Vega, L., Rojas, D. and Datta-Gupta, A., "Scalability of the Deterministic and Bayesian Approaches to Production Data Integration into Field-Scale Reservoir Models," paper SPE 79666 presented at the 2003 SPE Reservoir Simulation Symposium, Houston, February 3-5.

15. Nocedel, J., and Wright, S. J.: *Numerical Optimization*, Springer-Verlag, New York (1999).

16. Jacquard, P. and Jain, C.: "Permeability Distribution from Field Pressure Data," *SPEJ* (1965) 281.

17. Jahns, H.O.: "A Rapid Method for Obtaining a Two-Dimensional Reservoir Description from Well Pressure Response Data," *SPEJ* (1966) 315.

18. Carter, R.D., Kemp, Jr., L.F., Pierce, A.C., and Williams, D.L.: "Performance Matching With Constraints," *SPEJ* (1974) 187.

19. He, N., Reynolds, A.C., and Oliver, D.S.: "Three-Dimensional Reservoir Description from Multiwell Pressure Data and Prior Information," *SPEJ* (1997) 312.

20. Peaceman, D.W.: "Interpretation of Well-Block Pressures in Numerical Reservoir Simulation with Non-Square Grid Blocks and Anisotropic Permeability," *SPEJ* (1983) 531.

21. Yeh, W. W-G.: "Review of Parameter Identification Procedures in Groundwater Hydrology: The Inverse Problem," *Water Resources Research* (1986) 95.

22. Anterion, F., Karcher, B., and Eymard, R.: "Use of Parameter Gradients for Reservoir History Matching," paper SPE 18433, presented at the 1989 SPE Reservoir Simulation Symposium, Houston, February 6-8.

23. Tang, Y.N.: *Application of GPST Algorithm to History Matching*, Ph.D. dissertation, State University of New York at Stony Brook, New York, 1985.

24. Tang, Y.N., Chen, Y.M., Chen, W.H., and Wasserman, M.L.: "Generalized Pulse-Spectrum Technique for 2-D and 2-Phase History Matching," *Applied Numerical Mathematics* (1989) 529.

25. Chu, L., Reynolds, A.C., and Oliver, D.S.: "Computation of Sensitivity Coefficients for Conditioning the Permeability Field to Well-Test Pressure Data," *In Situ* (1995) 179.

26. Chen, W.H., Gavalas, G.R., Seinfeld, J.H., and Wasserman, M.L.: "A New Algorithm for Automatic History Matching," *SPEJ* (December 1974) 593.

27. Chavent, G. M., Dupuy, M., and Lemonnier, P.: "History Matching by Use of Optimal-Control Theory," *SPEJ* (February 1975) 74; *Trans.*, AIME, **259**.

28. Wasserman, M.L., Emanuel, A.S., and Seinfeld, J.H.: "Practical Applications of Optimal Control Theory to History Matching Multiphase Simulator Models," *SPEJ* (August 1975) 347.

29. Watson, A.T., Seinfeld, J.H., Gavalas, G.R., and Woo, P.T.: "History Matching in Two-Phase Petroleum Reservoirs," *SPEJ* (December 1980) 521.

30. Luo, Y. and Schuster, G.T.: "Wave-equation Traveltime Inversion," *Geophysics* (May 1991) 645.

31. King, M.J. and Datta-Gupta, A.: "Streamline Simulation: A Current Perspective," *In Situ* (1998) 91.

32. Batycky, R. P., Blunt, M. J., and Thiele, M. R.: "A 3-D Field Scale Streamline-Based Reservoir Simulator," *SPE Reservoir Engineering* (1997) 246.

33. Kharghoria, A.: *Field Scale History Matching and Assisted History Matching Using Streamline Simulation,* Ph.D. dissertation, Texas A&M University, College Station, August 2003.

34. Tarantola, A.: *Inverse Problem Theory – Methods for Data Fitting and Model Parameters Estimation,* Elsevier, New York, 1987.

35. Cheng, H., Datta-Gupta, A. and He, Z.: "A Comparison of Travel Time and Amplitude Matching for Field-Scale Production Data Integration: Sensitivity, Non-Linearity and Practical Implications," paper SPE 84570 presented at the 2003 SPE Annual Technical Conference and Exhibition, Denver, CO, October 5-8.

36. Wu, Z.: *Conditioning Geostatistical Models to Two-Phase Production Data,* Ph.D. dissertation, University of Tulsa, Oklahoma, 1999.

37. Xu, W., Tran, T.T., Sirvastava, R.M., and Journel, A.G.: "Integrating Seismic Data in Reservoir Modeling: The Collocated Cokriging Approach," paper SPE 24742 presented at 1992 SPE Annual Technical Conference and Exhibition, Washington DC, October 4-7.

38. Reynolds, A.C., He, N., Chu, L., and Oliver, D.S.: "Reparameterization Techniques for Generating Reservoir Descriptions Conditioned to Variograms and Well-Test Pressure Data," paper SPE 30588 presented at the 1995 SPE Annual Technical Conference and Exhibition, Dallas, October 22-25.

39. Oliver, D.S.: "Calculation of the Inverse of the Covariance," *Mathematical Geology* (1998) 911.

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

41. Bi, Z.: *Conditioning 3D Stochastic Channels to Well-Test Pressure Data,* Ph.D. dissertation, University of Tulsa, Oklahoma, 1999.

42. Paige, C.C., Saunders, M.A., "LSQR: An Algorithm for Sparse Linear Equations and Sparse Least Squares," *ACM Transactions on Mathematical Software* (March 1982) 43.

43. Nolet, G.: "Seismic Wave Propagation and Seismic Tomography," *Seismic Tomography,* (1987) 1.

44. Vega, L., *An Efficient Bayesian Formulation for Production Data Integration into Reservoir Models*, Ph.D. dissertation, Texas A&M University, College Station, 2003.

45. Press, W. H., Teukolesky, S. A., Vetterling, W. T., and Flannery, B.P.: *Numerical Recipe in Fortran 77: The Art of Scientific Computing*, second edition, Press Syndicate of University of Cambridge, New York (1992).

46. Higham, N.J.: "Stable Iteration for the Matrix Square Root," Numerical Algorithm (1997) 227.

47. Liu, N., Betancourt, S., and Oliver, D.S.: "Assessment of Uncertainty Assessment Methods," paper SPE 71624 presented at the 2001 SPE Annual Technical Conference and Exhibition, New Orleans, September 30 – October 3.

48. Floris, F.J.T., Bush, M.D., Cuypers, M., Roggero, F., and Syversveen, A-R.: "Methods for Quantifying the Uncertainty of Production Forecasts: A Comparative Study," *Petroleum Geoscience* (2001) S87.

49. Oliver, D.S., He, N., Reynolds, A.C.: "Conditioning Permeability Fields to Pressure Data," 5th European Conference on the Mathematics of Oil Recovery, Leoben, Austria, September 3-6, 1996.

50. Bonet-Cunha, L., Oliver, D.S., Redner, R.A., and Reynolds, A.C.: "A Hybrid Markov Chain Monte Carlo Method for Generating Permeability Fields Conditioned to Multiwell Pressure Data and Prior Information," *SPEJ* (September 1998) 261.

51. Oliver, D.S., Cunha, L.B., and Reynolds, A.C.: "Markov Chain Monte Carlo Methods for Conditioning a Permeability Field to Pressure Data," *Mathematical Geology* (1997) 61.

52. Deutch, C.V. and Journel, A.G.: *GSLIB Geostatistical Software Library and User's Guide*, Oxford University, 1998.

53. Schlumberger GeoQuest: ECLIPSE Reference Manual 2003A.

54. Datta-Gupta, A. and King, M. J.: "A Semianalytic Approach to Tracer Flow Modeling in Heterogeneous Permeable Media," *Advances in Water Resources* (1995) 9.

55. Reynolds, A.C., University of Tulsa, personal communication.

56. Beck, J.V. and Arnold, K.J.: *Parameter Estimation in Engineering and Science*, John Wiley & Sons, Inc., New York, 1977.

57. Eclipse Technical Description, Well Inflow Performance, Geoquest 2003.

58. Fanchi, J.R.: *Principles of Applied Reservoir Simulation*, Second Edition, Gulf Professional Publishing, Houston, 2001.

APPENDIX A

MODIFIED GAUSS-NEWTON AND MATHEMATICAL EQUIVALENT BETWEEN GAUSS-NEWTON AND FIELD-SCALE BAYESIAN FORMULATION

A.1 Modified Gauss-Newton

Gauss- Newton formulation as given by **Eq. 2.24** and is repeated here is as follows:

$$\mathbf{m}^{l+1} = \mathbf{m}^l - \left[G_l^T C_D^{-1} G_l + C_M^{-1} \right]^{-1} \left[G_l^T C_D^{-1} \Delta \tilde{\mathbf{t}}_\mathbf{l} + C_M^{-1} (\mathbf{m}^\mathbf{l} - \mathbf{m}_\mathbf{p}) \right] \qquad \text{…….…..(A.1)}$$

The inverse of the matrix $\left[G_l^T C_D^{-1} G_l + C_M^{-1} \right]$ and the covariance matrix, C_M which are of order of $M \times M$ makes the formulation given by **Eq. A.1** inefficient for field-scale application where M can be of order of thousands or millions. So, the modification to this formulation can be done by using the following matrix inverse lemma[56]:

$$\left[A + B C D \right]^{-1} = A^{-1} - A^{-1} B \left[C^{-1} + D A^{-1} B \right]^{-1} D A^{-1} \qquad \text{…..……(A.2)}$$

Where, A, C, and $\left[C^{-1} + D A^{-1} B \right]$ must be non-singular square matrix.

So, by letting,

$$
\begin{aligned}
A &= C_M^{-1} \\
B &= G_l^T \\
C &= C_D^{-1} \\
D &= G_l
\end{aligned}
\qquad \text{…...……(A.3)}
$$

By substituting **Eq. A.3** in **Eq. A.2**,

$$\left[G_l^T C_D^{-1} G_l + C_M^{-1}\right]^{-1} = C_M - C_M \, G_l^T \left[C_D + G_l C_M G_l^T\right]^{-1} G_l \, C_M \qquad \text{...........(A.4)}$$

Multiplying **Eq. A.4** by C_M^{-1},

$$\left[G_l^T C_D^{-1} G_l + C_M^{-1}\right]^{-1} C_M^{-1} = I - C_M \, G_l^T \left[C_D + G_l C_M G_l^T\right]^{-1} G_l \qquad \text{...........(A.5)}$$

From the following identity[34]:

$$G_l^T + G_l^T C_D^{-1} G_l C_M G_l^T = G_l^T C_D^{-1}\left[C_D + G_l C_M G_l^T\right] = \left[G_l^T C_D^{-1} G_l + C_M^{-1}\right] C_M G_l^T \qquad \text{...........(A.6)}$$

From the matrix identity given by **Eq. A.6**,

$$G_l^T C_D^{-1}\left[C_D + G_l C_M G_l^T\right] = \left[G_l^T C_D^{-1} G_l + C_M^{-1}\right] C_M G_l^T \qquad \text{...........(A.7)}$$

Multiply **Eq. A.7** from left by $\left[G_l^T C_D^{-1} G_l + C_M^{-1}\right]^{-1}$ and from right by $\left[C_D + G_l C_M G_l^T\right]^{-1}$,

$$\left[G_l^T C_D^{-1} G_l + C_M^{-1}\right]^{-1} G_l^T C_D^{-1} = C_M G_l^T \left[C_D + G_l C_M G_l^T\right]^{-1} \qquad \text{...........(A.8)}$$

It is important also to mention that from the identity given by **Eq. A.6**, one can reach to **Eq. A.5** with simple matrix manipulation.

Gauss-Newton equation, **Eq. A.1,** can be written as:

$$\mathbf{m}^{l+1} = \mathbf{m}^l - \left[G_l^T C_D^{-1} G_l + C_M^{-1}\right]^{-1} G_l^T C_D^{-1} \Delta \tilde{\mathbf{t}}_{\mathbf{l}}$$
$$- \left[G_l^T C_D^{-1} G_l + C_M^{-1}\right]^{-1} C_M^{-1} (\mathbf{m}^{\mathbf{l}} - \mathbf{m}_{\mathbf{p}}) \qquad \text{...........(A.9)}$$

Substituting **Eqs. A.5** and **A.8** in **Eq. A.9,** gives the Modified Gauss-Newton formulation:

$$\mathbf{m}^{l+1} = \mathbf{m_p} - \mathbf{C_M}\mathbf{G_l^T}\left[C_D + G_l C_M G_l^T\right]^{-1}\left[\Delta\tilde{\mathbf{t}}_\mathbf{l} - G_l(\mathbf{m^l} - \mathbf{m_p})\right] \qquad \ldots\ldots\ldots(A.10)$$

The Modified Gauss-Newton formula given by **Eq. A.10** requires only the inverse of matrix $\left[C_D + G_l C_M G_l^T\right]$ which is of order N_w x N_w (N_w is the number of wells) in using the generalized travel time as the data misfit which is of order of magnitude less than the number of model parameters. This makes the Modified Gauss-Newton more efficient for field-scale applications.

A.2 Mathematical Equivalent of Gauss-Newton and Field-Scale Bayesian Formulation

In order to prove the mathematical equivalency between Gauss-Newton formulation and the field-scale Bayesian formulation, it is easy to compare the Newton equation for model updating used by the both method as they both used the same Newton equation for model updating which is given as:

$$H\,\delta\mathbf{m} = -\nabla_\mathbf{m}O(\mathbf{m}) \qquad \ldots\ldots\ldots(A.11)$$

Where, the Hessian matrix, H, is equivalent to $G_l^T C_D^{-1} G_l + C_M^{-1}$ given by **Eq. 2.23** in using Gauss-Newton and is equivalent to $J^T J$ given by **Eq. 2.34** using the field-scale Bayesian formulation. While the gradient of the objective function, $\nabla_\mathbf{m}O(\mathbf{m})$, is equivalent to $G_l^T C_D^{-1}\Delta\tilde{\mathbf{t}}_\mathbf{l} + C_M^{-1}(\mathbf{m^l} - \mathbf{m_p})$ given by **Eq. 2.20** in using Gauss-Newton, and is equivalent to $J^T e$ given by **Eq. 2.32** in using the field-scale Bayesian formulation. Thus to prove that Gauss-Newton and field-scale Bayesian formulation are mathematically equivalent is to prove that:

i- $J^T J = G_l^T C_D^{-1} G_l + C_M^{-1}$, and

ii- $J^T e = G_l^T C_D^{-1} \Delta \tilde{t}_l + C_M^{-1} (\mathbf{m^l} - \mathbf{m_p})$

First: J is given by **Eq. 2.31** as follows:

$$J = \begin{bmatrix} C_D^{-1/2}\, G_l \\ C_M^{-1/2} \end{bmatrix}$$ ……..(A.12)

Thus, $J^T J$ is given as follows:

$$J^T J = \begin{bmatrix} G_l^T \left(C_D^{-1/2}\right)^T & \left(C_M^{-1/2}\right)^T \end{bmatrix} \cdot \begin{bmatrix} C_D^{-1/2} G_l \\ C_M^{-1/2} \end{bmatrix}$$ ………….(A.13)

As, C_D and C_M are symmetric matrices, thus,

$$\left(C_D^{-1/2}\right)^T = C_D^{-1/2}$$
$$\left(C_M^{-1/2}\right)^T = C_M^{-1/2}$$ …..……(A.14)

Substitute **Eq. A.14** in **Eq. A.13** and multiply, we get the first requirement.

$$J^T J = G_l^T C_D^{-1} G_l + C_M^{-1}$$ ……..(A.15)

Second: e is given by **Eq. 2.28** as follows:

$$e = \begin{bmatrix} C_D^{-\frac{1}{2}}\, \Delta \tilde{t}_l \\ C_M^{-\frac{1}{2}}\left(\mathbf{m^l} - \mathbf{m_p}\right) \end{bmatrix}$$ …..…..(A.16)

So, $J^T e$ is given as:

$$J^T e = \begin{bmatrix} G_l^T C_D^{-1/2} & C_M^{-1/2} \end{bmatrix} \cdot \begin{bmatrix} C_D^{-\frac{1}{2}} \Delta \widetilde{t}_1 \\ C_M^{-\frac{1}{2}} \left(\mathbf{m}^1 - \mathbf{m_p} \right) \end{bmatrix}$$ ....(A.17)

$$= G_l^T C_D^{-1} \Delta \widetilde{t}_1 + C_M^{-1} (\mathbf{m}^1 - \mathbf{m_p})$$

From **Eqs. A. 15** and **A.17**, it can be easily seen that the Gauss-Newton and the field-scale Bayesian formulation are mathematically equivalent.

APPENDIX B

COMPUTATION OF THE DERIVATIVES IN THE ADJOINT SYSTEM OF EQUATIONS

The adjoint system of equations is given in i,j,k notations by **Eqs. 3.46** and **3.47**, in this appendix, we give the derivative of the flow terms, $f_{m,i-1,j,k}^l$, $f_{m,i,j-1,k}^l$, $f_{m,i,j,k-1}^l$, $f_{m,i,j,k}^l$, $f_{m,i+1,j,k}^l$, $f_{m,i,j+1,k}^l$, $f_{m,i,j,k+1}^l$, the accumulation terms, $A_{m,i,j,k}^l$, the source/sink terms, $q_{m,i,j,k}^l$, and the source term of the adjoint system of equations, $g_{i,j,k}^l$, with respect to pressure and water saturation, where m stands for oil, o, and water, w, phase.

B.1 Derivative of the Flow Terms in the Adjoint System of Equations

The oil and water flow term equations are given by **Eqs. 3.42** and **3.43**, which are repeated here for convenience,

$$
\begin{aligned}
f_{o,i,j,k}^l =\; & T_{ox,i+1/2,j,k}^l (p_{i+1,j,k}^l - p_{i,j,k}^l - \gamma_{oi+1/2,j,k}^l (D_{i+1,j,k} - D_{i,j,k})) \\
& - T_{ox,i-1/2,j,k}^l (p_{i,j,k}^l - p_{i-1,j,k}^l - \gamma_{oi-1/2,j,k}^l (D_{i,j,k} - D_{i-1,j,k})) \\
& + T_{oy,i,j+1/2,k}^l (p_{i,j+1,k}^l - p_{i,j,k}^l - \gamma_{oi,j+1/2,k}^l (D_{i,j+1,k} - D_{i,j,k})) \\
& - T_{oy,i,j-1/2,k}^l (p_{i,j,k}^l - p_{i,j-1,k}^l - \gamma_{oi,j-1/2,k}^l (D_{i,j,k} - D_{i,j-1,k})) \\
& + T_{oz,i,j,k+1/2}^l (p_{i,j,k+1}^l - p_{i,j,k}^l - \gamma_{oi,j,k+1/2}^l (D_{i,j,k+1} - D_{i,j,k})) \\
& - T_{oz,i,j,k-1/2}^l (p_{i,j,k}^l - p_{i,j,k-1}^l - \gamma_{oi,j,k-1/2}^l (D_{i,j,k} - D_{i,j,k-1}))
\end{aligned}
\qquad \ldots\ldots\ldots(B.1)
$$

$$
\begin{aligned}
f_{w,i,j,k}^l =\; & T_{wx,i+1/2,j,k}^l (p_{i+1,j,k}^l - p_{i,j,k}^l - \gamma_{wi+1/2,j,k}^l (D_{i+1,j,k} - D_{i,j,k})) \\
& - T_{wx,i-1/2,j,k}^l (p_{i,j,k}^l - p_{i-1,j,k}^l - \gamma_{wi-1/2,j,k}^l (D_{i,j,k} - D_{i-1,j,k})) \\
& + T_{wy,i,j+1/2,k}^l (p_{i,j+1,k}^l - p_{i,j,k}^l - \gamma_{wi,j+1/2,k}^l (D_{i,j+1,k} - D_{i,j,k})) \\
& - T_{wy,i,j-1/2,k}^l (p_{i,j,k}^l - p_{i,j-1,k}^l - \gamma_{wi,j-1/2,k}^l (D_{i,j,k} - D_{i,j-1,k})) \\
& + T_{wz,i,j,k+1/2}^l (p_{i,j,k+1}^l - p_{i,j,k}^l - \gamma_{wi,j,k+1/2}^l (D_{i,j,k+1} - D_{i,j,k})) \\
& - T_{wz,i,j,k-1/2}^l (p_{i,j,k}^l - p_{i,j,k-1}^l - \gamma_{wi,j,k-1/2}^l (D_{i,j,k} - D_{i,j,k-1}))
\end{aligned}
\qquad \ldots\ldots\ldots(B.2)
$$

this section gives the derivative of the seven stencil flow terms used in the adjoint system of equations, which are $f^l_{m,i-1,j,k}$, $f^l_{m,i,j-1,k}$, $f^l_{m,i,j,k-1}$, $f^l_{m,i,j,k}$, $f^l_{m,i+1,j,k}$, $f^l_{m,i,j+1,k}$, $f^l_{m,i,j,k+1}$ with respect to pressure and water saturation at grid block i,j,k.

1- Derivatives of Flow terms in grid block $(i \pm 1, j, k)$:

From Eqs. B.1 and B.2, the derivatives of the flow terms at grid block $(i \pm 1, j, k)$ with respect to $p^l_{i,j,k}$ (for $m = o, w$) are as follows:

$$\frac{\partial f^l_{m,i\pm1,j,k}}{\partial p^l_{i,j,k}} = \frac{\partial T^l_{mx,i\pm1/2,j,k}}{\partial p^l_{i,j,k}} \left[p^l_{i,j,k} - p^l_{i\pm1,j,k} + \gamma^l_{m,i\pm1/2,j,k} \left(D_{i\pm1,j,k} - D_{i,j,k} \right) \right]$$
$$+ T^l_{mx,i\pm1/2,j,k} \left[1 + \frac{\partial \gamma^l_{m,i\pm1/2,j,k}}{\partial p^l_{i,j,k}} \left(D_{i\pm1,j,k} - D_{i,j,k} \right) \right]$$

.....………(B.3)

Similarly, the derivatives of $f^l_{m,i\pm1,j,k}$ with respect to $S^l_{w,i,j,k}$ (for $m = o, w$) are as follows:

$$\frac{\partial f^l_{m,i\pm1,j,k}}{\partial S^l_{w,i,j,k}} = \frac{\partial T^l_{mx,i\pm1/2,j,k}}{\partial S^l_{w,i,j,k}} \left[p^l_{i,j,k} - p^l_{i\pm1,j,k} + \gamma^l_{m,i\pm1/2,j,k} \left(D_{i\pm1,j,k} - D_{i,j,k} \right) \right]$$

……….(B.4)

2- Derivatives of Flow terms in grid block $(i, j \pm 1,, k)$:

Similarly, the derivatives of oil and water flow terms, $f^l_{m,i,j\pm1,k}$ with respect to $p^l_{i,j,k}$ for $m = o, w$ are as follows:

$$\frac{\partial f^l_{m,i,j\pm1,k}}{\partial p^l_{i,j,k}} = \frac{\partial T^l_{my,i,j\pm1/2,k}}{\partial p^l_{i,j,k}} \left[p^l_{i,j,k} - p^l_{i,j\pm1,k} + \gamma^l_{m,i,j\pm1/2,k} \left(D_{i,j\pm1,k} - D_{i,j,k} \right) \right]$$
$$+ T^l_{my,i,j\pm1/2,k} \left[1 + \frac{\partial \gamma^l_{m,i,j\pm1/2,k}}{\partial p^l_{i,j,k}} \left(D_{i,j\pm1,k} - D_{i,j,k} \right) \right]$$

……….(B.5)

The derivatives of $f^l_{m,i,j\pm1,k}$ with respect to $S^l_{w,i,j,k}$ for $m=o,w$ are:

$$\frac{\partial f^l_{m,i,j\pm1,k}}{\partial S^l_{w,i,j,k}} = \frac{\partial T^l_{my,i,j\pm1/2,k}}{\partial S^l_{w,i,j,k}} \left[p^l_{i,j,k} - p^l_{i,j\pm1,k} + \gamma^l_{m,i,j\pm1/2,k} \left(D_{i,j\pm1,k} - D_{i,j,k} \right) \right] \quad\ldots\ldots\ldots(\text{B.6})$$

3- Derivatives of Flow terms in grid block $(i,j,k\pm1)$:

Similarly, the derivations of the flow terms at grid block $(i,j,k\pm1)$ with respect to $p^l_{i,j,k}$

for $m=o,w$ are as follows:

$$\frac{\partial f^l_{m,i,j,k\pm1}}{\partial p^l_{i,j,k}} = \frac{\partial T^l_{mz,i,j,k\pm1/2}}{\partial p^l_{i,j,k}} \left[p^l_{i,j,k} - p^l_{i,j,k\pm1} + \gamma^l_{m,i,j,k\pm1/2} \left(D_{i,j,k\pm1} - D_{i,j,k} \right) \right]$$

$$+ T^l_{mz,i,j,k\pm1/2} \left[1 + \frac{\partial \gamma^l_{m,i,j,k\pm1/2}}{\partial p^l_{i,j,k}} \left(D_{i,j,k\pm1} - D_{i,j,k} \right) \right] \quad\ldots\ldots\ldots(\text{B.7})$$

The derivatives of oil and water flow terms $f^l_{m,i,j,k\pm1}$ with respect to water saturation $S^l_{w,i,j,k}$ are:

$$\frac{\partial f^l_{m,i,j,k\pm1}}{\partial S^l_{w,i,j,k}} = \frac{\partial T^l_{mz,i,j,k\pm1/2}}{\partial S^l_{w,i,j,k}} \left[p^l_{i,j,k} - p^l_{i,j,k\pm1} + \gamma^l_{m,i,j,k\pm1/2} \left(D_{i,j,k\pm1} - D_{i,j,k} \right) \right] \quad\ldots\ldots\ldots(\text{B.8})$$

4- Derivatives of flow terms in grid block (i,j,k)

From **Eqs. B.1, B.2** and from the derivatives of the other flow terms, it can be easily noticed that the derivatives of the flow terms at grid block (i,j,k) for $m=o,w$ are:

$$\frac{\partial f^l_{m,i,j,k}}{\partial p^l_{i,j,k}} = -\left(\frac{\partial f^l_{m,i+1,j,k}}{\partial p^l_{i,j,k}} + \frac{\partial f^l_{m,i-1,j,k}}{\partial p^l_{i,j,k}} + \frac{\partial f^l_{m,i,j+1,k}}{\partial p^l_{i,j,k}} + \frac{\partial f^l_{m,i,j-1,k}}{\partial p^l_{i,j,k}} + \frac{\partial f^l_{m,i,j,k+1}}{\partial p^l_{i,j,k}} + \frac{\partial f^l_{m,i,j,k-1}}{\partial p^l_{i,j,k}} \right) \ldots(\text{B.9})$$

$$\frac{\partial f^l_{m,i,j,k}}{\partial S^l_{w,i,j,k}} = -\left(\frac{\partial f^l_{m,i+1,j,k}}{\partial S^l_{w,i,j,k}} + \frac{\partial f^l_{m,i-1,j,k}}{\partial S^l_{w,i,j,k}} + \frac{\partial f^l_{m,i,j+1,k}}{\partial S^l_{w,i,j,k}} + \frac{\partial f^l_{m,i,j-1,k}}{\partial S^l_{w,i,j,k}} + \frac{\partial f^l_{m,i,j,k+1}}{\partial S^l_{w,i,j,k}} + \frac{\partial f^l_{m,i,j,k-1}}{\partial S^l_{w,i,j,k}} \right)..(\text{B.10})$$

As seen from **Eqs. B.3** to **B.10**, to calculate the derivatives of flow terms with respect to state variables, $p_{i,j,k}$, and $S_{w,i,j,k}$, we need to calculate the derivatives of the transmissibility terms (T) & gravity terms (γ) with respect to these state variables. The following provides the details of the calculation of these derivatives.

5- Derivatives of Transmissibility, T :

The transmissibility can be written as:

$$T^l_{mx,i+1/2,j,k} = \frac{c_1 \Delta y_j \Delta z_k K_{x,i+1/2,j,k}}{x_{i+1} - x_i} \cdot \frac{K^l_{rm,i+1/2,j,k}}{\mu^l_{m,i+1/2,j,k} B^l_{m,i+1/2,j,k}} \qquad \text{.........(B.11)}$$

For $m = o, w$, $i = 1, 2, \ldots \ldots n_x - 1$, and c_1 is 1.127×10^{-3}.

For no-flow boundary conditions, the transmissibilities at the boundaries are

$$T^l_{mx,1/2,j,k} = T^l_{mx,n_x+1/2,j,k} = 0 \qquad \text{.........(B.12)}$$

Similarly,

$$T^l_{my,i,j+1/2,k} = \frac{c_1 \Delta x_i \Delta z_k K_{y,i,j+1/2,k}}{y_{j+1} - y_j} \cdot \frac{K^l_{rm,i,j+1/2,k}}{\mu^l_{m,i,j+1/2,k} B^l_{m,i,j+1/2,k}} \qquad \text{.........(B.13)}$$

For $m = o, w$ and $j = 1, 2, \ldots \ldots n_y - 1$ and

$$T^l_{my,i,1/2,k} = T^l_{my,i,n_y+1/2,k} = 0 \qquad \text{.........(B.14)}$$

Also, transmissibilities in the vertical direction are given by

$$T_{mz,i,j,k+1/2}^{l} = \frac{c_1 \Delta y_j \Delta x_i \; K_{z,i,j,k+1/2}}{z_{k+1} - z_k} \cdot \frac{K_{rm,i,j,k+1/2}^{l}}{\mu_{m,i,j,k+1/2}^{l} B_{m,i,j,k+1/2}^{l}}$$ ………..(B.15)

For $m = o, w$ and k $=1,2,\ldots\ldots n_z - 1$ and

$$T_{mz,i,j,1/2}^{l} = T_{my,i,j,n_z+1/2}^{l} = 0$$ ……..(B.16)

During this work, the $K_{rm,i+1/2,j,k}$, $K_{rm,i,j+1/2,k}$, $K_{rm,i,j,k+1/2}$, $B_{m,i+1/2,j,k}$, $B_{m,i,j+1/2,k}$, $B_{m,i,j,k+1/2}$, $\mu_{m,i+1/2,j,k}$, $\mu_{m,i,j+1/2,k}$, $\mu_{m,i,j,k+1/2}$ are evaluated by upstream weighting , for example , for k_{rm}^{l} ,

$$K_{rm,i+1/2,j,k}^{l} = \begin{cases} K_{rm,i+1,j,k}^{l} & \text{if } (i+1,j,k) \text{ is upstream} \\ K_{rm,i,j,k}^{l} & \text{if } (i,j,k) \text{ is upstream} \end{cases}$$ ..……..(B.17)

$$K_{rm,i,j+1/2,k}^{l} = \begin{cases} K_{rm,i,j+1,k} & \text{if } (i,j+1,k) \text{ is upstream} \\ K_{rm,i,j,k}^{l} & \text{if } (i,j,k) \text{ is upstream} \end{cases}$$ ..……..(B.18)

$$K_{rm,i,j,k+1/2}^{l} = \begin{cases} k_{rm,i,j,k+1}^{l} & \text{if } (i,j,k+1) \text{ is upstream} \\ k_{rm,i,j,k}^{l} & \text{if } (i,j,k) \text{ is upstream} \end{cases}$$ ..……..(B.19)

The derivatives of the transmissibilities with respect to pressure are

$$\frac{\partial T_{mx,i+1/2,j,k}^{l}}{\partial p_{i,j,k}^{l}} = -\frac{T_{mx,i+1/2,j,k}^{l}}{\mu_{m,i+1/2,j,k}^{l} B_{m,i+1/2,j,k}^{l}} \times$$

$$\left[\frac{\partial B_{m,i+1/2,j,k}^{l}}{\partial p_{i,j,k}^{l}} \mu_{m,i+1/2,j,k}^{l} + \cdot \frac{\partial \mu_{m,i+1/2,j,k}^{l}}{\partial p_{i,j,k}^{l}} B_{m,i+1/2,j,k}^{l} \right]$$ ..……..(B.20)

$$\frac{\partial T^l_{my,i,j+1/2,k}}{\partial p^l_{i,j,k}} = -\frac{T^l_{my,i,j+1/2,k}}{\mu^l_{m,i,j+1/2,k} B^l_{m,i,j+1/2,k}} \times$$

$$\left[\frac{\partial B^l_{m,i,j+1/2,k}}{\partial p^l_{i,j,k}} \mu^l_{m,i,j+1/2,k} + \cdot \frac{\partial \mu^l_{m,i,j+1/2,k}}{\partial p^l_{i,j,k}} B^l_{m,i,j+1/2,k} \right]$$

………(B.21)

$$\frac{\partial T^l_{mz,i,j,k+1/2}}{\partial p^l_{i,j,k}} = -\frac{T^l_{mz,i,j,k+1/2}}{\mu^l_{m,i,j,k+1/2} B^l_{m,i,j,k+1/2}} \times$$

$$\left[\frac{\partial B^l_{m,i,j,k+1/2}}{\partial p^l_{i,j,k}} \mu^l_{m,i,j,k+1/2} + \frac{\partial \mu^l_{m,i,j,k+1/2}}{\partial p^l_{i,j,k}} B^l_{m,i,j,k+1/2} \right]$$

.………..(B.22)

The derivatives of transmissibilities at boundaries are all equal to 0.

The formation volume factor (B) and viscosity (μ) are evaluated by upstream weighting, so the derivatives of these terms are given by:

$$\frac{\partial B^l_{m,i+1/2,j,k}}{\partial p^l_{i,j,k}} = \frac{\partial B^l_{m,i,j+1/2,k}}{\partial p^l_{i,j,k}} = \frac{\partial B^l_{m,i,j,k+1/2}}{\partial p^l_{i,j,k}} = \begin{cases} \dfrac{\partial B^l_{m,i,j,k}}{\partial p^l_{i,j,k}} & \text{if } (i,j,k) \text{ is upstream} \\ 0 & \text{if } (i,j,k) \text{ is not upstream} \end{cases}$$

…..(B.23)

$$\frac{\partial \mu^l_{m,i+1/2,j,k}}{\partial p^l_{i,j,k}} = \frac{\partial \mu^l_{m,i,j+1/2,k}}{\partial p^l_{i,j,k}} = \frac{\partial \mu^l_{m,i,j,k+1/2}}{\partial p^l_{i,j,k}} = \begin{cases} \dfrac{\partial \mu^l_{m,i,j,k}}{\partial p^l_{i,j,k}} & \text{if } (i,j,k) \text{ is upstream} \\ 0 & \text{if } (i,j,k) \text{ is not upstream} \end{cases}$$

…..(B.24)

The derivatives, $\dfrac{\partial B^l_{m,i,j,k}}{\partial p_{i,j,k}}$ and $\dfrac{\partial \mu^l_{m,i,j,k}}{\partial p_{i,j,k}}$ can be evaluated from PVT correlations or table.

Similarly, the derivatives of the transmissibilities with respect to water saturation are evaluated as follows:

$$\frac{\partial T^l_{mx,i+1/2,j,k}}{\partial S^l_{w,i,j,k}} = \begin{cases} \dfrac{c_1 \Delta y_j \Delta z_k K_{x,i+1/2,j,k}}{x_{i+1} - x_i} \cdot \dfrac{1}{\left(\mu^l_m B^l_m\right)_{i,j,k}} \cdot \dfrac{\partial K^l_{rm,i,j,k}}{\partial S^l_{w,i,j,k}} & \text{if } (i,j,k) \text{ is upstream} \\ 0 & \text{if } (i,j,k) \text{ is not upstream} \end{cases} \quad ..\text{(B. 25)}$$

$$\frac{\partial T^l_{my,i,j+1/2,k}}{\partial S^l_{w,i,j,k}} = \begin{cases} \dfrac{c_1 \Delta x_i \Delta z_k K_{y,i,j+1/2,k}}{y_{j+1} - y_j} \cdot \dfrac{1}{\left(\mu^l_m B^l_m\right)_{i,j,k}} \cdot \dfrac{\partial K^l_{rm,i,j,k}}{\partial S^l_{w,i,j,k}} & \text{if } (i,j,k) \text{ is upstream} \\ 0 & \text{if } (i,j,k) \text{ is not upstream} \end{cases} \quad ...\text{(B.26)}$$

$$\frac{\partial T^l_{mz,i,j,k+1/2}}{\partial S^l_{w,i,j,k}} = \begin{cases} \dfrac{c_1 \Delta x_i \Delta y_j K_{z,i,j,k+1/2}}{z_{k+1} - z_k} \cdot \dfrac{1}{\left(\mu^l_m B^l_m\right)_{i,j,k}} \cdot \dfrac{\partial K^l_{rm,i,j,k}}{\partial S^l_{w,i,j,k}} & \text{if } (i,j,k) \text{ is upstream} \\ 0 & \text{if } (i,j,k) \text{ is not upstream} \end{cases} \quad ...\text{(B.27)}$$

The derivative is evaluated at upstream grid block as according to **Eqs. B.17** to **B.19**, $K^l_{rm,i+1/2,j,k}$, $K^l_{rm,i,j+1/2,k}$, and $K^l_{rm,i,j,k+1/2}$ are evaluated by the upstream weighting. The derivatives of $\dfrac{\partial K^l_{rm,i,j,k}}{\partial S^l_{w,i,j,k}}$ are computed by using correlation or from the relative permeability table.

6- Derivatives of gravity terms

The specific weights γ^l_m are defined as follows:

$$\gamma^l_{m,i\pm1/2,j,k} = \frac{g}{144 g_c} \frac{\left(\rho^l_{m,i,j,k} + \rho^l_{m,i\pm1,j,k}\right)}{2} \qquad \ldots\ldots\ldots\text{(B.28)}$$

Similarly, for $\gamma^l_{m,i,j\pm1/2,k}$ and $\gamma^l_{m,i,j,k\pm1/2}$. The phase density $\rho^l_{m,i,j,k}$ terms for two phase flow problems are given as:

$$\rho_{m,i,j,k}^{l} = \frac{\rho_{m,sc}^{l}}{B_{m}^{l}} \qquad\qquad ....(B.29)$$

Where, $\rho_{m,sc}^{l}$ is the density of oil and water at standard conditions, the density unit is lbm/ft^3.

From **Eqs. B.28** and **B.29**, the derivatives of specific weight terms are given as follows:

$$\frac{\partial \gamma_{m,i\pm1/2,j,k}^{l}}{\partial p_{i,j,k}^{l}} = \frac{g}{288g_{c}} \frac{\partial \rho_{m,i,j,k}^{l}}{\partial p_{i,j,k}^{l}} \qquad\qquad ....(B.30)$$

Similarly, $\dfrac{\partial \gamma_{m,i,j\pm1/2,k}^{l}}{\partial p_{i,j,k}^{1}}$ and $\dfrac{\partial \gamma_{m,i,j,k\pm1/2}^{l}}{\partial p_{i,j,k}^{l}}$ are obtained using **Eq. B.30,** where, $\dfrac{\partial \rho_{m,i,j,k}^{l}}{\partial p_{i,j,k}^{l}}$ for oil and water phase are obtained as follows:

$$\frac{\partial \rho_{m,i,j,k}^{l}}{\partial p_{i,j,k}^{l}} = -\frac{\rho_{m,sc}}{(B_{m}^{l})^{2}} \cdot \frac{\partial B_{m,i,j,k}^{l}}{\partial p_{i,j,k}^{l}} \qquad\qquad ....(B.31)$$

As mentioned before the derivatives of $\dfrac{\partial B_{m,i,j,k}^{l}}{\partial p_{i,j,k}}$ for $m = o, w$ can be evaluated from PVT correlations or table.

B.2 Derivative of the Accumulation Terms in the Adjoint System of Equations

The definition of accumulation terms are given in **Eqs. 3.14, 3.15, and 3.17**, and it is repeated here as follows:

$$A_{w,i,j,k}^{l} = \frac{V_{b,i,j,k}\phi_{i,j,k}}{C_{2}\Delta t^{l}} \cdot \frac{S_{w,i,j,k}^{l}}{B_{w,i,j,k}^{l}} \qquad\qquad ....(B.32)$$

$$A^l_{o,i,j,k} = \frac{V_{b,i,j,k}\phi_{i,j,k}}{C_2 \Delta t^l} \cdot \frac{\left(1 - S^l_{w,i,j,k}\right)}{B^l_{o,i,j,k}} \qquad \ldots\ldots\ldots\text{(B.33)}$$

Accordingly, the derivatives of water accumulation term with respect to pressure and water saturation are given as follows:

$$\frac{\partial A^l_{w,i,j,k}}{\partial p^l_{i,j,k}} = -\frac{V_{b,i,j,k}\phi_{i,j,k}S^l_{w,i,j,k}}{C_2 \Delta t^{l+1}(B^l_{w,i,j,k})^2} \frac{\partial B^l_{w,i,j,k}}{\partial p^l_{i,j,k}} \qquad \ldots\ldots\ldots\text{(B.34)}$$

$$\frac{\partial A^l_{w,i,j,k}}{\partial S^l_{w,i,j,k}} = \frac{V_{b,i,j,k}\phi_{i,j,k}}{C_2 \Delta t^l B^l_{w,i,j,k}} \qquad \ldots\ldots\ldots\text{(B.35)}$$

Similarly, the derivatives of oil accumulation term with respect to pressure and water saturation are given as follows:

$$\frac{\partial A^l_{o,i,j,k}}{\partial p^l_{i,j,k}} = -\frac{V_{b,i,j,k}\phi_{i,j,k}S^l_{o,i,j,k}}{C_2 \Delta t^l (B^l_{o,i,j,k})^2} \frac{\partial B^l_{o,i,j,k}}{\partial p^l_{i,j,k}} \qquad \ldots\ldots\ldots\text{(B.36)}$$

$$\frac{\partial A^l_{o,i,j,k}}{\partial S^l_{w,i,j,k}} = -\frac{V_{b,i,j,k}\phi_{i,j,k}}{C_2 \Delta t^l B^l_{o,i,j,k}} \qquad \ldots\ldots\ldots\text{(B.37)}$$

As mentioned before, $\dfrac{\partial B^l_{w,i,j,k}}{\partial p^l_{i,j,k}}$ and $\dfrac{\partial B^l_{o,i,j,k}}{\partial p^l_{i,j,k}}$ are obtained from the PVT correlation or table.

B.3 Derivative of the Source/Sink Terms in the Adjoint System of Equations

In this section, we compute the derivative of the source and sink terms with respect to the state variables (pressure and water saturation).

1- Production wells

We assume the producing wells are completed in a total of K connections, so the rate allocation modeling for oil and water is given by[57]:

$$q^l_{o,j,k_o} = WI_{k_o,j} \cdot \frac{K^l_{ro,j,k_o}}{\mu^l_{o,j,k_o} B^l_{o,j,k_o}} \cdot \left(P^l_{j,k_o} - P^l_{wf,j}\right) = WI_{k_o,j} \cdot M^l_{o,j,k_0} \cdot \left(P^l_{j,k_o} - P^l_{wf,j}\right) \ldots\ldots\ldots\text{(B.38)}$$

$$q^l_{w,j,k_o} = WI_{k_o,j} \cdot \frac{K^l_{rw,j,k_o}}{\mu^l_{w,j,k_o} B^l_{w,j,k_o}} \cdot \left(P^l_{j,k_o} - P^l_{wf,j}\right) = WI_{k_o,j} \cdot M^l_{w,j,k_o} \cdot \left(P^l_{j,k_o} - P^l_{wf,j}\right) \ldots\ldots\ldots\text{(B.39)}$$

Where, q^l_{o,j,k_o} is the oil rate from layer k_o at well j at time step l, q^l_{w,j,k_o} is the water rate from layer k_o at well j at time step l, $WI_{k_o,j}$ is the well index of layer k_o for well j, M^l_{o,j,k_o} and M^l_{w,j,k_o} are the water and oil mobility ratio at layer k_o for well j at time step l, P^l_{j,k_o} is the pressure at well j and layer k_o at time step l, and $P^l_{wf,j}$ is the bottom hole pressure at well j and it is assumed constant throughout the producing intervals by neglecting the friction loss of tubing through the producing intervals.

It is important to mention here, that by using commercial simulator as the forward model, we can get the $P^l_{wf,j}$ directly from the simulator at each time step regardless the wells are producing with constant rate or constant pressure or switched between both during the simulation run. Thus, help us to prevent the tedious way of calculating the $P^l_{wf,j}$ when the wells are produced with constant rate or switched from constant rates to constant bottom hole pressures and the opposite during the simulation run. This is considered one of the major advantages of adjoining the adjoint method-based sensitivity with a commercial simulator as a forward model.

The well index term, $WI_{k_o,j}$ is given as follows:

$$WI_{k,j} = \frac{0.00708\,\Delta z_k \sqrt{K_{x,i,j,k} K_{y,i,j,k}}}{\ln\left(r_{o,k,j}/r_{w,k,j}\right) + s_{k,j}}$$ ....(B.40)

Where, the equivalent radius, $r_{o,k,j}$, obtained from Peaceman's equation as follows:

$$r_{o,k,j} = \frac{0.28073\,\Delta x_i \sqrt{1 + \dfrac{K_{x,i,j,k}}{K_{y,i,j,k}} \cdot \dfrac{\Delta y_j}{\Delta x_i}}}{1 + \sqrt{K_{x,i,j,k}/K_{y,i,j,k}}}$$ ....(B.41)

$r_{w,k,j}$ is the well bore radius of well j at layer k and the $s_{k,j}$ is the skin factor for well j at layer k. It is worth to mention that $WI_{k,j}$ is independent of pressure and water saturation.

The derivatives of q_{o,j,k_o}^l and q_{w,j,k_o}^l with respect to pressure and water saturation at well j and layer k are as follows:

At $k = k_o$,

$$\frac{\partial q_{o,j,k_o}^l}{\partial p_{j,k_o}^l} = WI_{k_o,j} \cdot M_{o,j,k_o}^l + WI_{k_o,j} \cdot \left(P_{j,k_o}^l - P_{wf,j}^l\right) \cdot \frac{\partial M_{o,j,k_o}^l}{\partial p_{j,k_o}^l}$$ ....(B.42)

$$\frac{\partial q_{w,j,k_o}^l}{\partial p_{j,k_o}^l} = WI_{k_o,j} \cdot M_{w,j,k_o}^l + WI_{k_o,j} \cdot \left(P_{j,k_o}^l - P_{wf,j}^l\right) \cdot \frac{\partial M_{w,j,k_o}^l}{\partial p_{j,k_o}^l}$$ ....(B.43)

$$\frac{\partial q_{o,j,k_o}^l}{\partial S_{w,j,k_o}^l} = WI_{k_o,j} \cdot \left(P_{j,k_o}^l - P_{wf,j}^l\right) \cdot \frac{\partial M_{o,j,k_o}^l}{\partial S_{w,j,k_o}^l} \qquad \text{..........(B.44)}$$

$$\frac{\partial q_{w,j,k_o}^l}{\partial S_{w,j,k_o}^l} = WI_{k_o,j} \cdot \left(P_{j,k_o}^l - P_{wf,j}^l\right) \cdot \frac{\partial M_{w,j,k_o}^l}{\partial S_{w,j,k_o}^l} \qquad \text{..........(B.45)}$$

Where,

$$\frac{\partial M_{o,j,k_o}^l}{\partial p_{j,k_o}^l} = \frac{-K_{ro,j,k_o}^l}{\left(\mu_{o,j,k_o}^l B_{o,j,k_o}^l\right)^2} \cdot \left[\mu_{o,j,k_o}^l \cdot \frac{\partial B_{o,j,k_o}^l}{\partial p_{j,k_o}^l} + B_{o,j,k_o}^l \cdot \frac{\partial \mu_{o,j,k_o}^l}{\partial p_{j,k_o}^l}\right] \qquad \text{..........(B.46)}$$

$$\frac{\partial M_{w,j,k_o}^l}{\partial p_{j,k_o}^l} = \frac{-K_{rw,j,k_o}^l}{\left(\mu_{w,j,k_o}^l B_{w,j,k_o}^l\right)^2} \cdot \left[\mu_{w,j,k_o}^l \cdot \frac{\partial B_{w,j,k_o}^l}{\partial p_{j,k_o}^l} + B_{w,j,k_o}^l \cdot \frac{\partial \mu_{w,j,k_o}^l}{\partial p_{j,k_o}^l}\right] \qquad \text{..........(B.47)}$$

$$\frac{\partial M_{o,j,k_o}^l}{\partial S_{w,j,k_o}^l} = \frac{1}{\mu_{o,j,k_o}^l B_{o,j,k_o}^l} \cdot \frac{\partial K_{ro,j,k_o}^l}{\partial S_{w,j,k_o}^l} \qquad \text{..........(B.48)}$$

$$\frac{\partial M_{w,j,k_o}^l}{\partial S_{w,j,k_o}^l} = \frac{1}{\mu_{w,j,k_o}^l B_{w,j,k_o}^l} \cdot \frac{\partial K_{rw,j,k_o}^l}{\partial S_{w,j,k_o}^l} \qquad \text{..........(B.49)}$$

At $k \neq k_o$,

$$\frac{\partial q_{o,j,k_o}^l}{\partial p_{j,k}^l} = \frac{\partial q_{o,j,k_o}^l}{\partial S_{w,j,k}^l} = \frac{\partial q_{w,j,k_o}^l}{\partial p_{j,k}^l} = \frac{\partial q_{w,j,k_o}^l}{\partial S_{w,j,k}^l} = 0 \qquad \text{.....……(B.50)}$$

The derivatives of B^l_{o,j,k_o}, B^l_{w,j,k_o}, K^l_{ro,j,k_o}, K^l_{rw,j,k_o} with respect to pressure and water saturation can be obtained as mentioned before by knowing the correlation or from PVT and relative permeability table.

2- Injection wells

We assume the water injection wells are completed in a total of K connections, so the injection rate allocation modeling for water is given by[57, 58]:

$$q^l_{winj,j,k_o} = WI_{k_o,j} \cdot \left[\frac{\dfrac{K^l_{ro,j,k_o}}{\mu^l_{o,j,k_o}} + \dfrac{K^l_{rw,j,k_o}}{\mu^l_{w,j,k_o}}}{B^l_{w,j,k_o}} \right] \left[p^l_{j,k_o} - p^l_{wf,j} \right] \qquad \text{..........(B.51)}$$

Where, q^l_{winj,j,k_o} is the water injection rate to layer k_o, $p^l_{wf,j}$ is the bottom hole pressure for well j at time step l, which is assumed constant by neglecting the friction loss as discussed before in case of producing wells. It is important also to mention that by using the commercial simulator as the forward model, $p^l_{wf,j}$ is obtained directly from the simulator at each time step without any tedious way of calculating the bottom hole pressure for the injectors of constant injection rates.

The derivatives of q^l_{winj,j,k_o} with respect to pressure and water saturation at well j and layer k are as follows:

At $k = k_o$,

$$\frac{\partial q_{winj,j,k_o}^l}{\partial p_{j,k_o}^l} = WI_{k_o,j} \cdot \left[p_{j,k_o}^l - p_{wf,j}^l \right] \cdot$$

$$\left\{ \begin{bmatrix} \dfrac{-K_{ro,j,k_o}^l}{\left(\mu_{o,j,k_o}^l B_{w,j,k_o}^l\right)^2} \cdot \left[\mu_{o,j,k_o}^l \cdot \dfrac{\partial B_{w,j,k_o}^l}{\partial p_{j,k_o}^l} + B_{w,j,k_o}^l \cdot \dfrac{\partial \mu_{o,j,k_o}^l}{\partial p_{j,k_o}^l} \right] \\ -\dfrac{K_{rw,j,k_o}^l}{\left(\mu_{w,j,k_o}^l B_{w,j,k_o}^l\right)^2} \cdot \left[\mu_{w,j,k_o}^l \cdot \dfrac{\partial B_{w,j,k_o}^l}{\partial p_{j,k_o}^l} + B_{w,j,k_o}^l \cdot \dfrac{\partial \mu_{w,j,k_o}^l}{\partial p_{j,k_o}^l} \right] \end{bmatrix} \right\} \qquad \ldots\ldots\ldots(B.52)$$

$$+ WI_{k_o,j} \cdot \left[\dfrac{\dfrac{K_{ro,j,k_o}^l}{\mu_{o,j,k_o}^l} + \dfrac{K_{rw,j,k_o}^l}{\mu_{w,j,k_o}^l}}{B_{w,j,k_o}^l} \right]$$

$$\frac{\partial q_{winj,j,k_o}^l}{\partial S_{w,j,k_o}^l} = WI_{k_o,j} \cdot \left[p_{j,k_o}^l - p_{wf,j}^l \right] \cdot$$

$$\qquad\qquad \left\{ \dfrac{1}{\mu_{o,j,k_o}^l B_{w,j,k_o}^l} \cdot \dfrac{\partial K_{ro,j,k_o}^l}{\partial S_{w,j,k_o}^l} + \dfrac{1}{\mu_{w,j,k_o}^l B_{w,j,k_o}^l} \cdot \dfrac{\partial K_{rw,j,k_o}^l}{\partial S_{w,j,k_o}^l} \right\} \qquad \ldots\ldots\ldots(B.53)$$

At $k \neq k_o$,

$$\frac{\partial q_{winj,j,k_o}^l}{\partial p_{j,k}^l} = \frac{\partial q_{winj,j,k_o}^l}{\partial S_{w,j,k}^l} = 0 \qquad\qquad\qquad\qquad\qquad \ldots\ldots\ldots(B.54)$$

The derivatives of B_{w,j,k_o}^l, K_{ro,j,k_o}^l, K_{rw,j,k_o}^l with respect to pressure and water saturation can be obtained as mentioned before from correlations or from PVT and relative permeability table.

B.4 Derivative of the Source Term in the Adjoint System of Equations

As shown from **Eqs. 3.32 – 3.38**, the derivative of the source term, g with respect to pressure and water saturation is reduced to getting only the derivative of the water cut with respect to the state variables at observed time, t^i equivalent to simulation time step,

l. In this section, we give the derivative of the water cut with respect to pressure and water saturation for producing wells.

The water cut for producing well j is obtained as follows:

$$f_{wcal,j}^l = \frac{\sum_{k=1}^{K} q_{w,j,k}^l}{\sum_{k=1}^{K} q_{w,j,k}^l + \sum_{k=1}^{K} q_{o,j,k}^l} \qquad \text{.........(B.55)}$$

By substituting **Eqs. B.38** and **B.39** in **Eq. B.60**, the water cut will be as follows:

$$f_{wcal,j}^l = \frac{\left[\sum_{k=1}^{K} \left[\frac{K_{rw}^l}{\mu_w^l B_w^l} \right]_{k,j} \cdot WI_{k,j} \cdot \left(P_{j,k}^l - P_{wf,j}^l \right) \right]}{\left[\sum_{k=1}^{K} \left[\frac{K_{rw}^l}{\mu_w^l B_w^l} \right]_{k,j} \cdot WI_{k,j} \cdot \left(P_{j,k}^l - P_{wf,j}^l \right) + \sum_{k=1}^{K} \left[\frac{K_{ro}^l}{\mu_o^l B_o^l} \right]_{k,j} \cdot WI_{k,j} \cdot \left(P_{j,k}^l - P_{wf,j}^l \right) \right]}$$

$$= \frac{\sum_{k=1}^{K} M_{w,j,k}^l \cdot WI_{k,j} \cdot \left(P_{j,k}^l - P_{wf,j}^l \right)}{\sum_{k=1}^{K} M_{w,j,k}^l \cdot WI_{k,j} \cdot \left(P_{j,k}^l - P_{wf,j}^l \right) + \sum_{k=1}^{K} M_{o,j,k}^l \cdot WI_{k,j} \cdot \left(P_{j,k}^l - P_{wf,j}^l \right)}$$

$$\text{...…….(B.56)}$$

The derivatives of water cut given by **Eq. B.56** with respect to pressure and water saturation at well j and layer k_o, where k_o is one of the layers completed for well j, are as follows:

$$\frac{\partial f_{wcal,j}^{l}}{\partial p_{j,k_o}^{l}} = \frac{1}{\left[\sum_{k=1}^{K} M_{w,j,k}^{l} \cdot WI_{k,j} \cdot \left(P_{j,k}^{l} - P_{wf,j}^{l}\right) + \sum_{k=1}^{K} M_{o,j,k}^{l} \cdot WI_{k,j} \cdot \left(P_{j,k}^{l} - P_{wf,j}^{l}\right)\right]^2} \times$$

$$\left\{ \begin{array}{l} \left[WI_{k_o,j} \cdot M_{w,j,k_o}^{l} + WI_{k_o,j} \cdot \dfrac{\partial M_{w,j,k_o}^{l}}{\partial p_{j,k_o}^{l}} \cdot \left(P_{j,k_o}^{l} - P_{wf,j}^{l}\right)\right] \cdot \\[3mm] \left[\sum_{k=1}^{K} M_{w,j,k}^{l} \cdot WI_{k,j} \cdot \left(P_{j,k}^{l} - P_{wf,j}^{l}\right) + \sum_{k=1}^{K} M_{o,j,k}^{l} \cdot WI_{k,j} \cdot \left(P_{j,k}^{l} - P_{wf,j}^{l}\right)\right] \\[3mm] -\left[\sum_{k=1}^{K} M_{w,j,k}^{l} \cdot WI_{k,j} \cdot \left(P_{j,k}^{l} - P_{wf,j}^{l}\right)\right] \cdot \\[3mm] \left[\begin{array}{l} WI_{k_o,j} \cdot M_{w,j,k_o}^{l} + WI_{k_o,j} \cdot \dfrac{\partial M_{w,j,k_o}^{l}}{\partial p_{j,k_o}^{l}} \cdot \left(P_{j,k}^{l} - P_{wf,j}^{l}\right) + \\[3mm] WI_{k_o,j} \cdot M_{o,j,k_o}^{l} + WI_{k_o,j} \cdot \dfrac{\partial M_{o,j,k_o}^{l}}{\partial p_{j,k_o}^{l}} \cdot \left(P_{j,k}^{l} - P_{wf,j}^{l}\right) \end{array}\right] \end{array} \right\}$$ .....(B.57)

$$\frac{\partial f_{wcal,j}^{l}}{\partial S_{w,j,k_o}^{l}} = \frac{1}{\left[\sum_{k=1}^{K} M_{w,j,k}^{l} \cdot WI_{k,j} \cdot \left(P_{j,k}^{l} - P_{wf,j}^{l}\right) + \sum_{k=1}^{K} M_{o,j,k}^{l} \cdot WI_{k,j} \cdot \left(P_{j,k}^{l} - P_{wf,j}^{l}\right)\right]^2} \times$$

$$\left\{ \begin{array}{l} WI_{k_o,j} \cdot \dfrac{\partial M_{w,j,k_o}^{l}}{\partial S_{w,j,k_o}^{l}} \cdot \left(P_{j,k}^{l} - P_{wf,j}^{l}\right) \left[\begin{array}{l} \sum_{k=1}^{K} M_{w,j,k}^{l} \cdot WI_{k,j} \cdot \left(P_{j,k}^{l} - P_{wf,j}^{l}\right) \\[3mm] + \sum_{k=1}^{K} M_{o,j,k}^{l} \cdot WI_{k,j} \cdot \left(P_{j,k}^{l} - P_{wf,j}^{l}\right) \end{array}\right] \\[6mm] -\left[\sum_{k=1}^{K} M_{w,j,k}^{l} \cdot WI_{k,j} \cdot \left(P_{j,k}^{l} - P_{wf,j}^{l}\right)\right] \cdot \left[\begin{array}{l} WI_{k_o,j} \cdot \dfrac{\partial M_{w,j,k_o}^{l}}{\partial S_{w,j,k_o}^{l}} \cdot \left(P_{j,k}^{l} - P_{wf,j}^{l}\right) \\[3mm] + WI_{k_o,j} \cdot \dfrac{\partial M_{o,j,k_o}^{l}}{\partial S_{w,j,k_o}^{l}} \cdot \left(P_{j,k}^{l} - P_{wf,j}^{l}\right) \end{array}\right] \end{array} \right\}$$

……...(B.58)

The derivatives of $\dfrac{\partial M_{o,j,k_o}^{l}}{\partial p_{j,k_o}^{l}}$, $\dfrac{\partial M_{w,j,k_o}^{l}}{\partial p_{j,k_o}^{l}}$, $\dfrac{\partial M_{o,j,k_o}^{l}}{\partial S_{w,j,k_o}^{l}}$, and $\dfrac{\partial M_{w,j,k_o}^{l}}{\partial S_{w,j,k_o}^{l}}$ are given before in **Eqs.**

B.46 - B.49.

APPENDIX C

COMPUTATION OF THE DERIVATIVES IN THE SENSITIVITY COEFFICIENTS EQUATIONS

The sensitivity of generalized travel time with respect to permeability in the x, y, and z directions is given by **Eqs. 3.63, 3.64,** and **3.65**. In this appendix, we show the computation of the derivative of the flow terms $f_{m,i-1,j,k}^{l+1}, f_{m,i,j,k}^{l+1}, f_{m,i+1,j,k}^{l+1}$, with respect to $K_{x,i,j,k}$, the derivative of $f_{m,i,j-1,k}^{l+1}, f_{m,i,j,k}^{l+1}$, and $f_{m,i,j+1,k}^{l+1}$ with respect to $K_{y,i,j,k}$, the derivative of $f_{m,i,j,k-1}^{l+1}, f_{m,i,j,k}^{l+1}, f_{m,i,j,k+1}^{l+1}$ with respect to $K_{z,i,j,k}$, and the derivative of the source/sink terms, $q_{m,i,j,k}^{l+1}$ with respect to $K_{x,i,j,k}, K_{y,i,j,k}$, and $K_{z,i,j,k}$ for $m = o, w$

C.1 Derivatives of the Flow Terms in the Sensitivity Coefficients Equations

The oil and water flow term equations are given by **Eqs. 3.42** and **3.43**, which are repeated here as follows:

$$
\begin{aligned}
f_{o,i,j,k}^{l} = & T_{ox,i+1/2,j,k}^{l}(p_{i+1,j,k}^{l} - p_{i,j,k}^{l} - \gamma_{oi+1/2,j,k}^{l}(D_{i+1,j,k} - D_{i,j,k})) \\
& - T_{ox,i-1/2,j,k}^{l}(p_{i,j,k}^{l} - p_{i-1,j,k}^{l} - \gamma_{oi-1/2,j,k}^{l}(D_{i,j,k} - D_{i-1,j,k})) \\
& + T_{oy,i,j+1/2,k}^{l}(p_{i,j+1,k}^{l} - p_{i,j,k}^{l} - \gamma_{oi,j+1/2,k}^{l}(D_{i,j+1,k} - D_{i,j,k})) \\
& - T_{oy,i,j-1/2,k}^{l}(p_{i,j,k}^{l} - p_{i,j-1,k}^{l} - \gamma_{oi,j-1/2,k}^{l}(D_{i,j,k} - D_{i,j-1,k})) \qquad\qquad..........(C.1) \\
& + T_{oz,i,j,k+1/2}^{l}(p_{i,j,k+1}^{l} - p_{i,j,k}^{l} - \gamma_{oi,j,k+1/2}^{l}(D_{i,j,k+1} - D_{i,j,k})) \\
& - T_{oz,i,j,k-1/2}^{l}(p_{i,j,k}^{l} - p_{i,j,k-1}^{l} - \gamma_{oi,j,k-1/2}^{l}(D_{i,j,k} - D_{i,j,k-1}))
\end{aligned}
$$

$$f^l_{w,i,j,k} = T^l_{wx,i+1/2,j,k}(p^l_{i+1,j,k} - p^l_{i,j,k} - \gamma^l_{wi+1/2,j,k}(D_{i+1,j,k} - D_{i,j,k}))$$

$$- T^l_{wx,i-1/2,j,k}(p^l_{i,j,k} - p^l_{i-1,j,k} - \gamma^l_{wi-1/2,j,k}(D_{i,j,k} - D_{i-1,j,k}))$$

$$+ T^l_{wy,i,j+1/2,k}(p^l_{i,j+1,k} - p^l_{i,j,k} - \gamma^l_{wi,j+1/2,k}(D_{i,j+1,k} - D_{i,j,k}))$$

$$- T^l_{wy,i,j-1/2,k}(p^l_{i,j,k} - p^l_{i,j-1,k} - \gamma^l_{wi,j-1/2,k}(D_{i,j,k} - D_{i,j-1,k})) \qquad \ldots\ldots\ldots(\text{C.2})$$

$$+ T^l_{wz,i,j,k+1/2}(p^l_{i,j,k+1} - p^l_{i,j,k} - \gamma^l_{wi,j,k+1/2}(D_{i,j,k+1} - D_{i,j,k}))$$

$$- T^l_{wz,i,j,k-1/2}(p^l_{i,j,k} - p^l_{i,j,k-1} - \gamma^l_{wi,j,k-1/2}(D_{i,j,k} - D_{i,j,k-1}))$$

As shown from **Eqs. C.1 and C.2**, the only terms that depend on $k_{x,i,j,k}$ are

$f^{l+1}_{m,i-1,j,k}, f^{l+1}_{m,i,j,k}, f^{l+1}_{m,i+1,j,k}$, those that depend on $K_{y,i,j,k}$ are $f^{l+1}_{m,i,j-1,k}, f^{l+1}_{m,i,j,k}, f^{l+1}_{m,i,j+1,k}$, and

finally, those that depend on $K_{z,i,j,k}$ are $f^{l+1}_{m,i,j,k-1}, f^{l+1}_{m,i,j,k}, f^{l+1}_{m,i,j,k+1}$. In this section we give

the derivatives of those terms with respect to permeability in the x, y, and z directions.

1- Derivatives of flow terms in grid block $(i \pm 1, j, k)$ with respect to $k_{x,i,j,k}$

From **Eqs. C.1** and **C.2**, the derivatives of the flow terms at grid block $(i \pm 1, j, k)$ with

respect to $k_{x,i,j,k}$ (for $m = o, w$) are as follows:

$$\frac{\partial f^l_{o,i\pm1,j,k}}{\partial k_{x,i,j,k}} = \frac{\partial T^l_{ox,i\pm1/2,j,k}}{\partial k_{x,i,j,k}}\left[p^l_{i,j,k} - p^l_{i\pm1,j,k} + \gamma^l_{o,i\pm1/2,j,k}(D_{i\pm1,j,k} - D_{i,j,k})\right] \qquad \ldots\ldots\ldots\ldots(\text{C.3})$$

$$\frac{\partial f^l_{w,i\pm1,j,k}}{\partial k_{x,i,j,k}} = \frac{\partial T^l_{wx,i\pm1/2,j,k}}{\partial k_{x,i,j,k}}\left[p^l_{i,j,k} - p^l_{i\pm1,j,k} + \gamma^l_{w,i\pm1/2,j,k}(D_{i\pm1,j,k} - D_{i,j,k})\right] \qquad \ldots\ldots\ldots(\text{C.4})$$

2- The derivatives of flow terms in grid block $(i, j \pm 1, k)$ with respect to $k_{y,i,j,k}$

Similarly, the derivatives of the flow terms with respect to $k_{y,i,j,k}$ are as follows:

$$\frac{\partial f^l_{o,i,j\pm1,k}}{\partial k_{y,i,j,k}} = \frac{\partial T^l_{oy,i,j\pm1/2,k}}{\partial k_{y,i,j,k}}\left[p^l_{i,j,k} - p^l_{i,j\pm1,k} + \gamma^l_{o,i,j\pm1/2,k}(D_{i,j\pm1,k} - D_{i,j,k})\right] \qquad \ldots\ldots\ldots(\text{C.5})$$

$$\frac{\partial f_{w,i,j\pm1,k}^{l}}{\partial k_{y,i,j,k}} = \frac{\partial T_{wy,i,j\pm1/2,k}^{l}}{\partial k_{y,i,j,k}}\left[p_{i,j,k}^{l} - p_{i,j\pm1,k}^{l} + \gamma_{w,i,j\pm1/2,k}^{l}(D_{i,j\pm1,k} - D_{i,j,k})\right] \qquad \ldots\ldots\ldots(C.6)$$

3- The derivatives of flow terms in grid block $(i,j,k\pm1)$ with respect to $K_{z,i,j,k}$

$$\frac{\partial f_{o,i,j,k\pm1}^{l}}{\partial k_{z,i,j,k}} = \frac{\partial T_{oz,i,j,k\pm1/2}^{l}}{\partial k_{z,i,j,k}}\left[p_{i,j,k}^{l} - p_{i,j,k\pm1}^{l} + \gamma_{o,i,j,k\pm1/2}^{l}(D_{i,j,k\pm1} - D_{i,j,k})\right] \qquad \ldots\ldots\ldots(C.7)$$

$$\frac{\partial f_{w,i,j,k\pm1}^{l}}{\partial k_{z,i,j,k}} = \frac{\partial T_{wz,i,j,k\pm1/2}^{l}}{\partial k_{z,i,j,k}}\left[p_{i,j,k}^{l} - p_{i,j,k\pm1}^{l} + \gamma_{w,i,j,k\pm1/2}^{l}(D_{i,j,k\pm1} - D_{i,j,k})\right] \qquad \ldots\ldots\ldots(C.8)$$

4- The derivatives of flow terms in grid block (i,j,k) with respect to $k_{x,i,j,k}$, $k_{y,i,j,k}$,

   and $K_{z,i,j,k}$

The derivative of **Eqs. C.1,** and **C.2** with respect to $k_{x,i,j,k}$, $k_{y,i,j,k}$, and $K_{z,i,j,k}$ are as follows:

$$\frac{\partial f_{m,i,j,k}^{l}}{\partial k_{x,i,j,k}} = -\left(\frac{\partial f_{m,i+1,j,k}^{l}}{\partial k_{x,i,j,k}} + \frac{\partial f_{m,i-1,j,k}^{l}}{\partial k_{x,i,j,k}}\right) \qquad \ldots\ldots\ldots(C.9)$$

$$\frac{\partial f_{m,i,j,k}^{l}}{\partial k_{y,i,j,k}} = -\left(\frac{\partial f_{m,i,j+1,k}^{l}}{\partial k_{y,i,j,k}} + \frac{\partial f_{m,i,j-1,k}^{l}}{\partial k_{y,i,j,k}}\right) \qquad \ldots\ldots\ldots(C.10)$$

$$\frac{\partial f_{m,i,j,k}^{l}}{\partial k_{z,i,j,k}} = -\left(\frac{\partial f_{m,i,j,k+1}^{l}}{\partial k_{z,i,j,k}} + \frac{\partial f_{m,i,j,k-1}^{l}}{\partial k_{z,i,j,k}}\right) \qquad \ldots\ldots\ldots(C.11)$$

for $m = o,w$, where all the derivatives in Eqs. C.9 – C.11 are obtained before from **Eqs. C.3 – C.8**.

As noticed from Eqs. C3 – C11, all the derivatives depend upon the derivatives of the transmissibilities with respect to permeability in x, y, z direction. These derivatives are obtained as follows:

5- Derivatives of the transmissibility with respect to $k_{x,i,j,k}$, $k_{y,i,j,k}$, and $K_{z,i,j,k}$

The transmissibility equations are given before in Appendix B, Eqs. B.11–B.16 and are repeated here as follows:

$$T^l_{mx,i+1/2,j,k} = \frac{c_1 \Delta y_j \Delta z_k K_{x,i+1/2,j,k}}{x_{i+1} - x_i} \cdot \frac{K^l_{rm,i+1/2,j,k}}{\mu^l_{m,i+1/2,j,k} B^l_{m,i+1/2,j,k}} \qquad \ldots\ldots\ldots(C.12)$$

For $m = o, w$, i =1,2,……. n_x −1, and c_1 is 1.127×10^{-3}.

For no-flow boundary conditions, the transmissibilities at the boundaries are

$$T^l_{mx,1/2,j,k} = T^l_{mx,n_x+1/2,j,k} = 0 \qquad \ldots\ldots\ldots(C.13)$$

Similarly,

$$T^l_{my,i,j+1/2,k} = \frac{c_1 \Delta x_i \Delta z_k K_{y,i,j+1/2,k}}{y_{j+1} - y_j} \cdot \frac{K^l_{rm,i,j+1/2,k}}{\mu^l_{m,i,j+1/2,k} B^l_{m,i,j+1/2,k}} \qquad \ldots\ldots\ldots(C.14)$$

For $m = o, w$ and j =1,2,……. n_y −1 and

$$T^l_{my,i,1/2,k} = T^l_{my,i,n_y+1/2,k} = 0 \qquad \ldots\ldots\ldots(C.15)$$

Also, transmissibilities in the vertical direction are given by

$$T_{mz,i,j,k+1/2}^{l} = \frac{c_1 \Delta y_j \Delta x_i \, K_{z,i,j,k+1/2}}{z_{k+1} - z_k} \cdot \frac{K_{rm,i,j,k+1/2}^{l}}{\mu_{m,i,j,k+1/2}^{l} B_{m,i,j,k+1/2}^{l}} \qquad \ldots\ldots\ldots(C.16)$$

For $m = o, w$ and $k = 1, 2, \ldots\ldots n_z - 1$ and

$$T_{mz,i,j,1/2}^{l} = T_{my,i,j,n_z+1/2}^{l} = 0 \qquad \ldots\ldots\ldots(C.17)$$

The permeabilities at grid block interfaces are computed as harmonic averages as follows:

$$K_{x,i+1/2,j,k} = \frac{(\Delta x_i + \Delta x_{i+1}) K_{x,i,j,k} K_{x,i+1,j,k}}{\Delta x_i K_{x,i+1,j,k} + \Delta x_{i+1} K_{x,i,j,k}} \qquad \ldots\ldots\ldots(C.18)$$

At the boundary,

$$K_{x,1/2,j,k} = K_{x,1,j,k} \qquad \ldots\ldots\ldots(C.19)$$

$$K_{x,n_x+1/2,j,k} = K_{x,n_x,j,k} \qquad \ldots\ldots\ldots(C.20)$$

Thus, the derivative of Eqs. C.18-C.20 with respect to $K_{x,i,j,k}$ are as follows:

For $i = 1,2,3,.., n_x - 1$,

$$\frac{\partial K_{x,i+1/2,j,k}}{\partial K_{x,i,j,k}} = \frac{\Delta x_i K_{x,i+1,j,k}}{K_{x,i,j,k} \left(\Delta x_i K_{x,i+1,j,k} + \Delta x_{i+1} K_{x,i,j,k} \right)} K_{x,i+1/2,j,k} \qquad \ldots\ldots\ldots(C.21)$$

While at the boundary,

$$\frac{\partial K_{x,1/2,j,k}}{\partial K_{x,1,j,k}} = \frac{\partial K_{x,n_x+1/2,j,k}}{\partial K_{x,n_x,j,k}} = 1 \qquad \ldots\ldots\ldots\text{(C.22)}$$

Similarly, $K_{x,i-1/2,j,k}$ is given as follows:

$$K_{x,i-1/2,j,k} = \frac{(\Delta x_{i-1} + \Delta x_i)K_{x,i-1,j,k}K_{x,i,j,k}}{\Delta x_{i-1}K_{x,i,j,k} + \Delta x_i K_{x,i-1,j,k}} \qquad \ldots\ldots\ldots\text{(C.23)}$$

The derivative of **Eq. C.23** with respect to $K_{x,i,j,k}$ is as follows:

For i = 2,3,.., n_x,

$$\frac{\partial K_{x,i-1/2,j,k}}{\partial K_{x,i,j,k}} = \frac{\Delta x_i K_{x,i-1,j,k}}{K_{x,i,j,k}\left(\Delta x_{i-1}K_{x,i,j,k} + \Delta x_1 K_{x,i-1,j,k}\right)}K_{x,i-1/2,j,k} \qquad \ldots\ldots\ldots\text{(C.24)}$$

At the boundary the derivatives are the same as that given by **Eq. C.22**.

Similarly, the derivatives of $K_{y,i,j+1/2,k}$ and $K_{y,i,j-1/2,k}$ with respect to $K_{y,i,j,k}$ are:

$$\frac{\partial K_{y,i,j+1/2,k}}{\partial K_{x,i,j,k}} = \frac{\Delta y_j K_{y,i,j+1,k}}{K_{y,i,j,k}\left(\Delta y_j K_{y,i,j+1,k} + \Delta y_{j+1}K_{y,i,j,k}\right)}K_{y,i,j+1/2,k} \qquad \ldots\ldots\ldots\text{(C.25)}$$

$$\frac{\partial K_{y,i,j-1/2,k}}{\partial K_{x,i,j,k}} = \frac{\Delta y_j K_{y,i,j-1,k}}{K_{y,i,j,k}\left(\Delta y_{j-1}K_{y,i,j,k} + \Delta y_{j+1}K_{y,i,j-1,k}\right)}K_{y,i,j-1/2,k} \qquad \ldots\ldots\ldots\text{(C.26)}$$

Similarly, the derivatives of $K_{z,i,j,k+1/2}$ and $K_{z,i,j,k-1/2}$ with respect to $K_{z,i,j,k}$ are:

$$\frac{\partial K_{z,i,j,k+1/2}}{\partial K_{x,i,j,k}} = \frac{\Delta z_k K_{y,i,j,k+1}}{K_{z,i,j,k}\left(\Delta z_k K_{z,i,j,k+1} + \Delta z_{k+1} K_{z,i,j,k}\right)} K_{z,i,j,k+1/2} \qquad \ldots\ldots\ldots.(C.27)$$

$$\frac{\partial K_{z,i,j,k-1/2}}{\partial K_{x,i,j,k}} = \frac{\Delta z_k K_{z,i,j,k-1}}{K_{z,i,j,k}\left(\Delta z_{k-1} K_{z,i,j,k} + \Delta z_{k+1} K_{z,i,j,k-1}\right)} K_{z,i,j,k-1/2} \qquad \ldots\ldots\ldots.(C.28)$$

The derivative of **Eq. C.12** with respect to $K_{x,i,j,k}$ is as follows:

$$\frac{\partial T_{mx,i+1/2,j,k}^{l}}{\partial K_{x,i,j,k}} = \frac{\Delta x_i K_{x,i+1,j,k}}{K_{x,i,j,k}\left(\Delta x_i K_{x,i+1,j,k} + \Delta x_{i+1} K_{x,i,j,k}\right)} \cdot T_{mx,i+1/2,j,k}^{l} \qquad \ldots\ldots\ldots.(C.29)$$

Similarly,

$$\frac{\partial T_{mx,i-1/2,j,k}^{l}}{\partial K_{x,i,j,k}} = \frac{\Delta x_i K_{x,i-1,j,k}}{K_{x,i,j,k}\left(\Delta x_{i-1} K_{x,i,j,k} + \Delta x_i K_{x,i-1,j,k}\right)} \cdot T_{mx,i-1/2,j,k}^{l} \qquad \ldots\ldots\ldots.(C.30)$$

$$\frac{\partial T_{my,i,j+1/2,k}^{l}}{\partial K_{y,i,j,k}} = \frac{\Delta y_j K_{y,i,j+1,k}}{K_{y,i,j,k}\left(\Delta y_j K_{y,i,j+1,k} + \Delta y_{j+1} K_{y,i,j,k}\right)} \cdot T_{my,i,j+1/2,k}^{l} \qquad \ldots\ldots\ldots.(C.31)$$

$$\frac{\partial T_{my,i,j-1/2,k}^{l}}{\partial K_{y,i,j,k}} = \frac{\Delta y_j K_{y,i,j-1,k}}{K_{y,i,j,k}\left(\Delta y_{j-1} K_{y,i,j,k} + \Delta y_j K_{y,i,j-1,k}\right)} \cdot T_{my,i,j-1/2,k}^{l} \qquad \ldots\ldots\ldots.(C.32)$$

$$\frac{\partial T_{mz,i,j,k+1/2}^{l}}{\partial K_{z,i,j,k}} = \frac{\Delta z_k K_{z,i,j,k+1}}{K_{z,i,j,k}\left(\Delta z_k K_{z,i,j,k+1} + \Delta z_{k+1} K_{z,i,j,k}\right)} \cdot T_{mz,i,j,k+1/2}^{l} \qquad \ldots\ldots\ldots.(C.33)$$

$$\frac{\partial T^l_{mz,i,j,k-1/2}}{\partial K_{z,i,j,k}} = \frac{\Delta z_k K_{z,i,j,k-1}}{K_{z,i,j,k}\left(\Delta z_{k-1} K_{z,i,j,k} + \Delta z_k K_{z,i,j,k-1}\right)} \cdot T^l_{mz,i,j,k-1/2} \qquad \text{..........(C.34)}$$

At the boundaries, the transmissibilities are zeros, thus the derivatives of the transmissibilities are zeros:

$$\frac{\partial T^l_{mx,1/2,j,k}}{\partial K_{x,i,j,k}} = \frac{\partial T^l_{mx,n_x+1/2,j,k}}{\partial K_{x,i,j,k}} = \frac{\partial T^l_{my,i,1/2,k}}{\partial K_{y,i,j,k}} = \frac{\partial T^l_{my,i,n_y+1/2,k}}{\partial K_{y,i,j,k}}$$

$$= \frac{\partial T^l_{mz,i,j,1/2}}{\partial K_{z,i,j,k}} = \frac{\partial T^l_{mz,i,j,n_z+1/2}}{\partial K_{z,i,j,k}} = 0 \qquad \text{..........(C.35)}$$

C.2 Derivatives of the Source/Sink Terms in the Sensitivity Coefficients Equations

In this section, we compute the derivative of the source and sink terms with respect to the control variables (permeabilities in x, y, z direction).

3- Production wells

The oil and water rates allocation modeling from layer k_o produced from well j are given before by **Eqs. B.38**, and **B.39** and repeated here as follows:

$$q^l_{o,j,k_o} = WI_{k_o,j} \cdot \frac{K^l_{ro,j,k_o}}{\mu^l_{o,j,k_o} B^l_{o,j,k_o}} \cdot \left(P^l_{j,k_o} - P^l_{wf,j}\right) = WI_{k_o,j} \cdot M^l_{o,j,k_0} \cdot \left(P^l_{j,k_o} - P^l_{wf,j}\right) \qquad \text{..........(C.36)}$$

$$q^l_{w,j,k_o} = WI_{k_o,j} \cdot \frac{K^l_{rw,j,k_o}}{\mu^l_{w,j,k_o} B^l_{w,j,k_o}} \cdot \left(P^l_{j,k_o} - P^l_{wf,j}\right) = WI_{k_o,j} \cdot M^l_{w,j,k_o} \cdot \left(P^l_{j,k_o} - P^l_{wf,j}\right) \qquad \text{..........(C.37)}$$

The derivatives of q^l_{o,j,k_o} and q^l_{w,j,k_o} with respect to permeability at well j and layer k are as follows:

At $k = k_o$,

$$\frac{\partial q^l_{o,j,k_o}}{\partial K_{x,j,k_o}} = M^l_{o,j,k_o} \cdot \left(P^l_{j,k_o} - P^l_{wf,j}\right) \cdot \frac{\partial WI_{k_o,j}}{\partial K_{x,j,k_o}} \qquad \text{.........(C.38)}$$

$$\frac{\partial q^l_{w,j,k_o}}{\partial K_{x,j,k_o}} = M^l_{w,j,k_o} \cdot \left(P^l_{j,k_o} - P^l_{wf,j}\right) \cdot \frac{\partial WI_{k_o,j}}{\partial K_{x,j,k_o}} \qquad \text{.........(C.39)}$$

$$\frac{\partial q^l_{o,j,k_o}}{\partial K_{y,j,k_o}} = M^l_{o,j,k_o} \cdot \left(P^l_{j,k_o} - P^l_{wf,j}\right) \cdot \frac{\partial WI_{k_o,j}}{\partial K_{y,j,k_o}} \qquad \text{.........(C.40)}$$

$$\frac{\partial q^l_{w,j,k_o}}{\partial K_{y,j,k_o}} = M^l_{w,j,k_o} \cdot \left(P^l_{j,k_o} - P^l_{wf,j}\right) \cdot \frac{\partial WI_{k_o,j}}{\partial K_{y,j,k_o}} \qquad \text{.........(C.41)}$$

As the layer production does not explicitly depend on $K_{z,i,j,k}$[40], so the derivative with respect to $K_{z,i,j,k}$ is as follows:

$$\frac{\partial q^l_{o,j,k_o}}{\partial K_{z,j,k_o}} = \frac{\partial q^l_{w,j,k_o}}{\partial K_{z,j,k_o}} = 0 \qquad \text{.........(C.42)}$$

Where, the derivatives of the well index with respect to permeabilities are obtained from **Eqs. B.40** and **B.41** as follows[40]:

$$\frac{\partial WI_{k_o,j}}{\partial K_{x,j,k_o}} = \frac{0.00708\Delta z_{j,k_o}}{2\left[\ln\left(r_{o,k_o,j}/r_{w,k_o,j}\right) + s_{k_o,j}\right]} \times$$

$$\left[\begin{array}{l} \left[\dfrac{\sqrt{K_{y,j,k_o}}}{\sqrt{K_{x,j,k_o}}} - \left(\dfrac{1}{\ln\left(r_{o,k_o,j}/r_{w,k_o,j}\right) + s_{k_o,j}}\right) \cdot \right. \\ \left. \left(\dfrac{\Delta y^2_{j,k_o}\sqrt{K_{x,j,k_o}K_{y,j,k_o}}}{\Delta x^2_{j,k_o}K_{y,j,k_o} + \Delta y^2_{j,k_o}K_{x,j,k_o}} - \dfrac{\sqrt{K_{y,j,k_o}}}{\sqrt{K_{x,j,k_o}} + \sqrt{K_{y,j,k_o}}}\right) \right] \end{array} \right] \qquad \text{.........(C.43)}$$

$$\frac{\partial WI_{k_o,j}}{\partial K_{y,j,k_o}} = \frac{0.00708\Delta z_{j,k_o}}{2\left[\ln\left(r_{o,k_o,j}/r_{w,k_o,j}\right)+s_{k_o,j}\right]} \times$$

$$\left[\frac{\sqrt{K_{x,j,k_o}}}{\sqrt{K_{y,j,k_o}}} + \left(\frac{1}{\ln\left(r_{o,k_o,j}/r_{w,k_o,j}\right)+s_{k_o,j}}\right)\cdot\right.$$

$$\left.\left(\frac{\Delta y^2_{j,k_o} K_{x,j,k_o}\sqrt{K_{x,j,k_o}}}{\Delta x^2_{j,k_o} K_{y,j,k_o}\sqrt{K_{y,j,k_o}} + \Delta y^2_{j,k_o} K_{x,j,k_o}\sqrt{K_{y,j,k_o}}} - \frac{K_{x,j,k_o}}{K_{y,j,k_o}+\sqrt{K_{x,j,k_o}K_{y,j,k_o}}}\right)\right] \quad ...(C.44)$$

At $k \neq k_o$,

$$\frac{\partial q^l_{o,j,k_o}}{\partial K_{x,j,k}} = \frac{\partial q^l_{o,j,k_o}}{\partial K_{y,j,k}} = \frac{\partial q^l_{o,j,k_o}}{\partial K_{z,j,k}} = \frac{\partial q^l_{w,j,k_o}}{\partial K_{x,j,k}} = \frac{\partial q^l_{w,j,k_o}}{\partial K_{y,j,k}} = \frac{\partial q^l_{w,j,k_o}}{\partial K_{z,j,k}} = 0 \qquad ....(C.45)$$

4- Injection wells

The injection rate allocation modeling for water is given before in **Eq. B.51** for well j injected in layer k_o and is repeated here as follows:

$$q^l_{winj,j,k_o} = WI_{k_o,j} \cdot \left[\frac{\dfrac{K^l_{ro,j,k_o}}{\mu^l_{o,j,k_o}} + \dfrac{K^l_{rw,j,k_o}}{\mu^l_{w,j,k_o}}}{B^l_{w,j,k_o}}\right]\left[p^l_{j,k_o} - p^l_{wf,j}\right] \qquad ....(C.46)$$

The derivatives of q^l_{winj,j,k_o} with respect to permeability at well j and layer k are as follows:

At $k = k_o$,

$$\frac{\partial q^l_{winj,j,k_o}}{\partial K_{x,j,k_o}} = \left[\frac{\dfrac{K^l_{ro,j,k_o}}{\mu^l_{o,j,k_o}} + \dfrac{K^l_{rw,j,k_o}}{\mu^l_{w,j,k_o}}}{B^l_{w,j,k_o}}\right][p^l_{j,k_o} - p^l_{wf,j}] \cdot \frac{\partial WI_{k_o,j}}{\partial K_{x,j,k_o}} \qquad \text{.........(C.47)}$$

$$\frac{\partial q^l_{winj,j,k_o}}{\partial K_{y,j,k_o}} = \left[\frac{\dfrac{K^l_{ro,j,k_o}}{\mu^l_{o,j,k_o}} + \dfrac{K^l_{rw,j,k_o}}{\mu^l_{w,j,k_o}}}{B^l_{w,j,k_o}}\right][p^l_{j,k_o} - p^l_{wf,j}] \cdot \frac{\partial WI_{k_o,j}}{\partial K_{y,j,k_o}} \qquad \text{.........(C.48)}$$

The term q^l_{winj,j,k_o} does not explicitly depends on $K_{z,i,j,k}$ [40], thus its derivative is:

$$\frac{\partial q^l_{winj,j,k_o}}{\partial K_{z,j,k_o}} = 0 \qquad \text{.........(C.49)}$$

Where, $\dfrac{\partial WI_{k_o,j}}{\partial K_{x,j,k_o}}$ and $\dfrac{\partial WI_{k_o,j}}{\partial K_{y,j,k_o}}$ are obtained from **Eqs. C.43** and **C.44,** respectively.

At $k \neq k_o$ **,**

$$\frac{\partial q^l_{winj,j,k_o}}{\partial K_{x,j,k}} = \frac{\partial q^l_{winj,j,k_o}}{\partial K_{y,j,k}} = \frac{\partial q^l_{winj,j,k_o}}{\partial K_{z,j,k}} = 0 \qquad \text{..........(C.50)}$$

VITA

Ahmed Mohamed Ibrahim Daoud was born in Egypt on July 23, 1972. He received Bachelor of Science and Master of Science degrees in petroleum engineering from Cairo University, Giza, Egypt, in 1995 and 1999, respectively. Since 1995 he worked as a Teacher Assistant in Cairo University, in the Petroleum Engineering Department, where he assisted in teaching courses related to well logging, reservoir engineering and reservoir simulation. He began the doctoral program at Texas A&M University in September 2000. His U.S. work experience includes two internships with Schlumberger in Houston, Texas, during his graduate studies. One internship was in the formation evaluation section and the other was in the well completion and productivity section. His permanent address is:

9 El-Atrby Basha Abo El-Ezz St., Heliopolois, Cairo, Egypt.

This dissertation was typed by the author.