# RESERVOIR PARAMETER ESTIMATION CONSTRAINED TO PRESSURE TRANSIENTS, PERFORMANCE HISTORY AND DISTRIBUTED SATURATION DATA

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF PETROLEUM ENGINEERING
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

By
Jorge Luis Landa
June 1997

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Dr. Roland N. Horne
(Principal Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Dr. Khalid Aziz

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Dr. Thomas A. Hewett

Approved for the University Committee on Graduate Studies:

# Acknowledgements

I would like to thank my research advisor Professor Roland N. Horne for his constant support, patience and guidance during the course of this study.

Special thanks and recognition are due to my friend Dr. Giovanni da Prat, who encouraged me to come to Stanford and believed I had the conditions to undertake and to succeed in the adventure of returning to school after many years.

My warm recognition to my friend Yan Pan for her constant encouragement and support.

I wish to thank ARCO and Dr. Med Kamal for assisting us to obtain data which was used during the course of this work.

Financial support from Stanford University Petroleum Research Institute (SUPRI–D) and the Department of Petroleum Engineering are gratefully acknowledged.

This dissertation is dedicated to the memory of my parents,
*Santiago and Marta Landa*

# Abstract

This work deals with the problem of estimating the distributions of permeability and porosity in a petroleum reservoir by matching the dynamic behavior. The dynamic data is in the form of field measurements from well testing, production history, interpreted 4–D seismic information, and other data such as correlations between permeability and porosity, geostatistics in the form of a variogram model and the inference of large scale geological structure.

The issue was posed as an inverse problem and solved by using nonlinear parameter estimation. The procedure developed here is capable of processing all the information simultaneously and this results in a fast and efficient method. The procedure is also able to determine the uncertainty associated with the estimated permeability and porosity fields.

The behavior of the reservoir was modeled with a finite difference numerical simulator because of the requirement of a mathematical model that is sufficiently complex to accommodate all the types of the dynamic data we used. This allowed us to apply the approach to heterogeneous reservoirs, multiphase flow and multiple well problems.

The key issue in the procedure is in the efficient computation of the derivatives of the field observations with respect to parameters that define the distributions of permeability and porosity in the reservoir. The algorithms developed here to compute these derivatives, referred to as sensitivity coefficients, were found to be extremely fast, and were generalized to a wide variety of different parameter types. Examples

of different parameter types that may be estimated by this approach include: (a) individual block permeabilities and porosities; (b) geological objects such as channels and faults; (c) pilot points that form the basis of a kriged distribution; and (d) seismic attenuation values from 3–D seismic images.

An important conclusion of this work is that the value of each piece of information does not reside in its isolated use but in the value it adds to integrated analysis of the complete set of information. Thus data that traditionally was considered to be of low information content for reservoir characterization becomes useful and enhances the value of the data set as a whole.

# Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

## 1.1 Statement of the Problem

Devising the optimal strategy for the development of an oil or gas reservoir is an important and difficult task. Many mathematical techniques for optimization can be used to deal with problems in engineering and economics systems. These techniques assume that we have a fairly complete understanding of the problem and also that we can construct a mathematical model that predicts the system's performance accurately in time under different scenarios; this is not a serious concern in most engineering problems since the parameters that define the system may not be very difficult to obtain by direct measurement. Unfortunately this is not the case in reservoir engineering, where the system, that is the oil–gas reservoir, is physically inaccessible many thousands of feet underground. Thus, any serious attempt at optimization of reservoir development first requires the determination of the parameters of the reservoir and the only way to obtain them is through indirect measurement. The process of inferring the parameters from the indirect measurements is an inverse or parameter estimation problem. Such is the focus of this research work.

Permeability and porosity are the parameters that have the largest influence in determining the performance of the reservoir, and thus, this work addresses the problem of estimating permeability and porosity from a variety of measurements that are only indirectly related to them.

Estimating permeability and porosity is difficult for the following reasons:

1. Permeability and porosity have spatial variability.
2. There are very few sampling locations (wells) compared to the areal extent of the reservoir.
3. Information (data) is scarce.
4. Measurements are obtained with different technologies.
5. The mathematical model of the reservoir is very complex, usually consisting of a numerical reservoir simulation.

The data set defining a reservoir is derived from measurements that are related to the permeability and porosity according to different laws or complex relationships. It is common to try to solve the inverse problem by working on each set of measurements independently of the others. In general each individual interpretation may not be fully consistent with the others, as a result the final interpretation comes after a long process of iteration to ensure compatibility. As a final step the interpretations are validated by history matching, since it would not be reasonable to rely on a model that could not predict the past performance. Since data is being collected almost continuously, the process of updating the reservoir model never ends.

During the producing life of a reservoir, data of different nature are always being collected. These data can be classified as *static* or *dynamic* depending on their association with the movement or flow of fluids in the reservoir. Data that have originated from geology, electrical logs, core analysis, fluid properties, seismic and geostatistics can be generally classified as *static*, whereas the information originating from well testing, pressure shut–in surveys, production history, bottom hole pressure from permanent gauges, water–cut, and gas–oil ratio (GOR) can be classified as *dynamic*. A special consideration needs to be applied in the case of 4–D seismic information, which is a relatively new technology being developed in the field of geophysics. With this process it is possible to estimate the areal distribution of change of saturations in the

reservoir due to the production or injection of fluids. Since 4–D seismic information is related to the movement of fluids in the reservoir it can be classified as *dynamic* data. One of the outstanding features of the 4–D seismic information is that it is areally distributed whereas the other dynamic data are available only at the location of the production or injection wells.

The parameter estimation problem would not only be faster but also more reliable if it were performed with a process that uses all or at least most of the information in the reservoir data set simultaneously. The process of handling different data simultaneously is known as *data integration*.

So far, most of the success in data integration has been obtained with *static* information. Remarkably, it has not yet become common to completely or systematically integrate *dynamic* data with *static* data and it is currently the subject of major research effort in several places. This work addresses this specific problem and represents a number of steps in the direction of full integration.

## 1.2 Literature Review

Estimating reservoir permeability and porosity by matching the past performance using a mathematical model has been a commonly used approach in reservoir engineering. One approach has been to minimize an objective function of the form:

$$\text{Objective Function } = E = \sum_{i=1}^{nobs} w_i \left(d_i^{calc} - d_i^{obs}\right)^2 \tag{1.1}$$

where *nobs* is the number of observations (data), $d_i^{obs}$ are the measured data, $d_i^{cal}$ are the data as calculated by the mathematical model of the reservoir, and $w_i$ is a factor used to weight and equalize units. The mathematical model includes a number of unknown parameters, including permeability, porosity and several others.

Before modern numerical simulators become available, reservoir engineers performed history matching by adjusting resistors and capacitances in analog computers [1]. A major breakthrough came in 1962 when Jacquard and Jain [2] developed a method to automate the history matching. The method was derived from variational

analysis applied to an electric network analog to the reservoir. In 1974 Carter, Pierce, Kemp and Williams [3] following the ideas introduced by Jacquard and Jain [2] applied variational analysis to the diffusion equation to derive a method to compute the derivatives of the calculated pressure in the wells with respect to the permeability and porosity as discretized in a numerical reservoir simulator. These derivatives are referred to in this work as *sensitivity coefficients* (explained in Chapter 3). The method of Carter et al. [3] is very efficient from the computational point of view, in that it allows the computation of the sensitivity coefficients with a reduced number of function evaluations (*number of wells + 1* simulation runs). The major limitation of the method is that it is applicable only to linear problems and therefore cannot be used only for multiphase flow.

In 1973 Chen, Gavalas, Seinfeld, and Wasserman [4] and Chavent, Dupuy, and Lemonnier [5] applied optimal control theory to develop an efficient method to compute the gradient of the objective function $E$ (Equation 1.1) with respect to permeability and porosity without needing to compute the sensitivity coefficients. The method is based on computing the gradient of a form of the objective function (Equation 1.1) by solving two differential equations: the equation that mathematically models the reservoir, and its associated *adjoint equation*. The optimal control formulation is not limited to linear cases and can be applied to nonlinear cases as well. In 1979 Watson, Seinfeld, Gavalas and Woo [6] used the optimal control formulation in a two–phase system. The main advantage of this approach resides in its inherently high efficiency from the computational point of view, specially when the number of parameters is large. The main shortcoming of the method is that it is difficult to implement in computer code. In addition, the method can be used only with parameter estimation algorithms that are not as efficient as those that use *sensitivity coefficients*. In 1987 Yang and Watson [7] used the optimal control formulation combined with parameter estimation based on the quasi–Newton algorithms [8] for optimization with the intent of overcoming the slow convergence experienced with first order methods (steepest descent and conjugate gradient). A first attempt at including geostatistical information

was reported in 1986 by Fasanino, Molinard, and de Marsily [9] by combining the optimal control formulation with the concept of kriging and pilot points in the reservoir.

In 1985 Tang, and Chen [10], [11] presented the GPST (Generalized Pulse–Spectrum Technique) as a method for history matching. The GPST formulation does not compute the sensitivity coefficients. In 1993 Oliver [12] used the GPST approach to compute permeability and porosity from pressure transient data and was able to include the geostatistical information in the least square formulation of the objective function (Equation 1.1) . The geostatistical information was included into the definition of the objective function following a method developed in the field of geophysical inverse theory (see Tarantola [13], [14], and Menke [15]). In 1994 Chu, Reynolds, and Oliver [16], [17] modified the GPST in order to compute an approximation of the sensitivity coefficients for use in the Gauss–Newton algorithm for parameter estimation. From 1994 through 1996 Oliver, Reynolds, Chu and He [18], [19], [20], [21], [22] presented a series of papers in which they used the modified GPST method to compute the sensitivity coefficients, the Gauss–Newton algorithm for parameter estimation and the geophysical inverse theory approach to include geostatistical information. In 1996 they reported [23] inaccuracy in the computation of the sensitivity coefficients with respect to the porosity field by the modified GPST formulation and then reverted to the Carter et al. [3] formulation which is restricted to single phase systems. It is important to remark that the works of Oliver, Reynolds, Chu and He were the first to present an approach that can combine *dynamic* and *static* data consistently by using a probabilistic frame.

In 1989 Anterion, Eymard, and Karcher [24] developed a similar procedure to compute the sensitivity coefficients, the method was referred to later as the gradient simulator and an application was shown by Bissell, Sharma, and Killough [25]. In 1991 Tan and Kalogerakis [26] followed the same approach of Anterion et al. [24] to compute the sensitivity coefficients from an implicit numerical reservoir simulator. In 1995 Tan [27] reported improvements in the implementation of the method. In 1996 Bissell [28] and Landa, Kamal, Jenkins and Horne [29] extended this approach to object modeling.

All the approaches mentioned so far are known as *gradient based methods*. Other authors have used other approaches based on direct search.

In 1993 Ounes, Bréfort, Meunier, and Dupéré [30] reported the use of the simulated annealing method for automatic history matching. Sultan, Ounes, and Weiss [31], [32] used parallel computing to improve the performance of the method. The simulated annealing method is applicable to problems in which it is difficult to compute the derivatives, however the approach is computationally inefficient. In 1992 Sen, Datta–Gupta, Stoffa, Lake and Pope [33] investigated the use of the genetic algorithm in reservoir modeling. The genetic algorithm works as an analogy to the natural selection and genetics in biology. Like the simulated annealing method the genetic algorithm has a slow rate of convergence and is computationally inefficient. The main attractions to the simulated annealing and genetic algorithm formulations are in their simple integration of *static* data and in their ability to converge to the global minimum of the objective function (Equation 1.1) whereas in the gradient based methods it is common to become trapped at stationary points (local minima or saddle points).

On a parallel path, researchers in the field of well testing developed their own interpretation methods relying completely on reservoir models that are generally simple but have to be modeled with complex analytical methods. The approach is an excellent way to deal with pressure transient data at single wells. The proper use of this method can provide good insight into the average properties and major geological features in the vicinity of a well but it becomes difficult to use when it is necessary to process data originating from different wells or separated in time.

## 1.3 Outline of Approach

Methods currently available in the literature can be divided into three main groups:

1. *Deterministic.*

   This group of methods focuses on the processing of dynamic data, generally in pursuit of a single defining model. This approach has been used in two areas.

   (a) *Automatic history matching* with numerical reservoir simulators. The constraining data is represented by the production history, including such information as pressure, water cut, gas–oil ratio, composition, etc.

   (b) *Well testing* or pressure transient analysis. The constraining data are a set of pressure measurements of short duration taken during a precisely determined change in the well condition. Nonlinear regression is frequently used to match analytical models to the pressure transient data.

2. *Probabilistic.*

   This approach uses geostatistics to process and integrate static data. The usual procedure is to generate multiple equiprobable realizations of the reservoir which later have to be validated using reservoir simulation.

3. *Emerging.*

   These approaches attempt to integrate static and dynamic data. Both deterministic and probabilistic formulations have been tried.

The approach of this work can be included in the third category.

The objectives of this research were:

1. To combine production history with well test data and also to add two new types of dynamic information: data from permanent bottom hole pressure gauges and changes in the saturation distribution as inferred from 4–D seismic interpretation.

2. To include the large scale geological information in the process of parameter estimation.

3. To include core analysis data and geostatistical information about the spatial correlation of permeability using a variogram model.

The procedures adopted here represent a merger of the techniques of well test interpretation, automated history matching, reservoir simulation, geological modeling,

geostatistics and 4–D seismic modeling. The principal components of the approach were as follows:

1. A numerical reservoir simulator was used as the mathematical model. This allowed the inclusion of multiphase flow as well as the modeling of geologically complex reservoirs.

2. A gradient based algorithm was developed to allow for efficient parameter estimation. A key component of the algorithm was the efficient computation of the sensitivity coefficients.

3. A procedure was developed to handle reservoir objects as a way to preserve the geological information in the reservoir model.

4. Kriging was used as a method to preserve the geostatistical information in the resulting interpretation.

5. The computation of variance of the parameter estimates and resolution of the data were performed by using the theory of singular value decomposition developed for linear systems.

The review of the literature revealed some important areas that were poorly developed and hence were targets for this work. These areas are as follows:

*Extended bottom–hole pressure history.* In the past, the *dynamic* data was limited to well testing, production rates, water–cut, and gas–oil ratio. These data may be useful to estimate properties in the surroundings of the wells. New advances in pressure gauge technology now allow the installation of permanent bottom–hole pressure gauges. In 1987 Schmidt, Stright and Forcade [34] detailed plans to install permanent pressure gauges in the Gullfaks field in the Norwegian North Sea. The purpose for the installation of the gauges was to monitor the well condition and optimize the reservoir development. The pressure transients were analyzed with standard well testing approaches and the long duration behavior was analyzed to understand the compartmentalization of the reservoir using reservoir simulation. In 1988 Gallivan, Kivington

and Shere [35] described the use of permanent bottom–hole pressure gauges in the Beatrice oil field in the North Sea, here the main objective was to monitor the intake pressure at the electrical submersible pumps. As an extra advantage the gauges allowed the gathering of transient data that could be used for drill stem test analysis or interference test analysis. In a more recent paper, 1993, Unneland and Haugland [36] reported the use of 40 permanent gauges installed during 1987–1993 in the Gullfaks and Veslefrikk fields. In this work the continuous pressure data and associated rates are used simultaneously, that is the drill stem test, interference, continuous bottom–hole drawdown or injection pressure and shut–in pressure data were all used at the same time in the process of inversion of the reservoir properties.

*Change of saturation distribution.* During the last 10 years the geophysics field has developed and refined the 3–D and 4–D seismic technology. The application of these technologies in reservoir characterization and monitoring has been summarized in 1997 by Seymour, and Barr [37]. The 3–D seismic data comes from a high resolution seismic survey that allows a more detailed three–dimensional characterization of the subsurface. When two or more successive 3–D surveys are shot over the same area the process is referred to as 4–D seismic as a way to denote the addition of the time dimension. The interpretation of 4–D seismic data allows the detection of changes in the seismic attributes of the reservoir which are due mainly to change in the saturation of the fluid which comes both from the production of fluids from the reservoir (gas out of solution) and/or the injection of fluids (gas or water injection). The sensitivity to the change of saturation allows tracking of the movement of fronts in the reservoir, which is indirectly related to the permeability and porosity. Despite the fact that the interpretation of the 4–D seismic data cannot provide an exact value for the change of saturation it provides qualitative information that can be used in the inversion of reservoir permeability and porosity. The outstanding characteristic of the 4–D seismic data is that it provides areally distributed information, which can therefore provide a higher areal resolution than well data. Currently the 4–D seismic interpretation is considered in isolation from the other dynamic data. In this work the integration of the 4–D seismic data is accomplished through numerical reservoir

simulation and by computing the derivatives of the change of saturation distribution with respect to permeability and porosity.

*Large scale geological information.* The direct application of the current methods for automatic inversion of reservoir permeability and porosity cannot integrate the large scale geological information, which results in descriptions that are difficult to accept from the geological point of view. As a remedy, the geostatistics field developed the concept of *object modeling* as a way to preserve the geological information. In 1992 Tyler, Henriquez, Georgsen, Holden, and Tjelmeland [38] presented a program for modeling a fluvial reservoir by using Boolean mathematics. In 1993 Wietzerbin, and Mallet [39] presented another procedure to generate sharp objects. In 1995 Bratvold, Holden, Tarald, and Tyler [40] presented STORM, a software system for stochastic reservoir modeling that includes the use of object modeling. In 1994 Petit, Biver, Calatayud, Lesuer, and Alabert [41] used stochastic object modeling for the quantification of hydrocarbon in place. In 1996 Deutsch and Wang [42] presented another procedure to generate shapes with stochastic modeling and filling them with a distribution of permeability and porosity. The attraction of the stochastic approach is its simplicity in integrating the large scale geological information with other geostatistical data. The stochastic object modeling can be classified within the *probabilistic* approach mentioned earlier. Like the other approaches in this group the stochastic approach relies on the *simulated annealing* algorithm to integrate the *dynamic* data from reservoir performance which makes the method very inefficient from the computational point of view. This work address this area. The modeling approach first generates parametric objects (like channels or ellipses) which are translated to the simulation grid with soft boundaries and then applies a gradient method to the inverse problem of finding the set of parameters which results in a match of the *dynamic* data. The difference from the other approaches is that the parameters to invert are no longer permeability and porosity but the parameters that define the objects, such as shape and spatial location. The approach presented in this work not only preserves the information about the nature of the objects but also speeds up the process of inversion of reservoir permeability and porosity.

*Resolution of the data and variance of parameter estimates.* This work studied the meaning and the uncertainty of the parameter estimates. Also we evaluated how each type of data contributes to the determination of the parameters. The evaluation is based on the resolution and variance analysis for linear systems using the singular value decomposition of the linear operator ([43] and [15]). The theory of resolution and variance was extended to the nonlinear systems of reservoir parameter estimation by assuming that the reservoir can be modeled as a linear system in the neighborhood of a predetermined field of permeability and porosity. The variance and resolution of the linearized system provide the mathematical basis for a qualitative understanding of the uncertainty associated with the parameter estimates and also provide a method for planning the data collection.

# Chapter 2

# Mathematical Preliminaries

## 2.1 General Procedure for Solution of Inverse Problems

The process of inversion to determine values of reservoir parameters, like permeability and porosity from indirect measurements is referred to as a *parameter estimation problem*. The usual approach to solve the parameter estimation problem is by going through three major steps, not only for the specific case of this work but for any general problem. These steps are as follows:

1. To construct a *mathematical model*.
2. To define an *objective function*.
3. To apply a *minimization algorithm*.

The *minimization algorithm* is also referred to in this work as the *parameter estimation algorithm*.
Each of these steps are described in more detail in the next sections.

Once the *mathematical model* has been constructed, the *objective function* has been defined, and the *minimization algorithm* has been chosen, the procedure for inversion works in the following way:

1. Assign an arbitrary, but reasonable, value to the unknown set of parameters, this is referred to as the first guess.

2. Compute the response of the system with the mathematical model.

3. Compute the objective function, which compares the calculated response of the system to the actual set of measurements. If the objective function is less than a certain predetermined value then *STOP*.

4. Compute with the minimization algorithm a change in the set of parameters. If the change in the set of parameters is less than a certain predetermined value then *STOP*.

5. Return to Step (3).

## 2.2 Mathematical Model

The physical system under study is represented by a mathematical model that is constructed by applying the fundamental physical laws that are relevant to the problem. The purpose of the mathematical model is to predict with reasonable accuracy the behavior of the system under different conditions. The problem of computing the response of the mathematical model to an external perturbation is referred to as the *forward problem*. The physical properties that remain invariant for different problems are referred to as *parameters* of the system. The ones that change are referred to as *variables*. The opposite problem, the *inverse problem* consists of finding a set of parameters for a given model such that the predicted behavior of the system replicates the true behavior (measurements) under the same set of external conditions. These concepts are summarized in Figure 2.1.

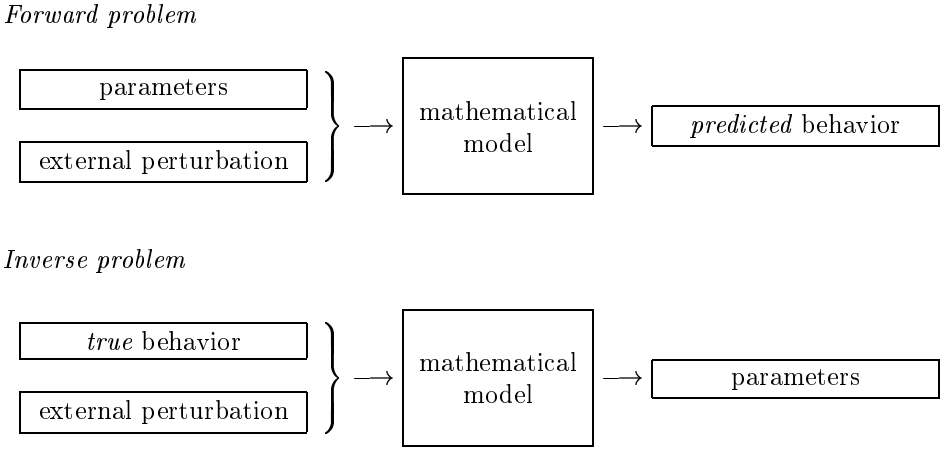



Figure 2.1: Forward and inverse problem.

In this work the physical system under study is a reservoir. The following fundamental laws are relevant to the dynamics of the reservoir:

- Mass conservation law.
- Darcy's law.
- Equation of state.
- Relative permeability and capillary pressure relationships.

The mathematical model is constructed by combining these laws and results in a system of differential equations. In a few cases, such as in traditional well testing theory, it is possible to obtain an explicit solution to the differential equations, but in the general case this is not possible and therefore it is necessary to resort to numerical methods to obtain a solution. In this work the mathematical model consisted of a numerical reservoir simulation code that had the following features:

| | |
|---|---|
| Language: | Fortran–77 |
| Type: | finite differences |
| Fluids: | water–oil |
| Variables: | pressure–water saturation |
| Solution method: | fully implicit |
| Dimensions: | two |
| Grid: | Cartesian |
| Wells: | vertical |
| Capillary pressure: | no |
| Gravity: | no |
| Linear Solver: | Sparse Matrix Direct Solver. |

In case of an *inverse problem* the mathematical model needs to be constructed in such a way that it can reproduce the main features observed in the data that will be used later as input to the process of inversion. For example if the input data is well test pressure with wellbore storage effects then it is necessary to be sure that the numerical simulator has the capability to model the wellbore storage properly. Using the wrong model, that is one that cannot make accurate predictions, may result in the computation of a meaningless set of parameters during the inversion process.

## 2.3 Objective Function

The objective function is a measure of the discrepancy between the information, that is the data, and the response as calculated by the mathematical model using the current set of parameters. Equation 2.1 shows a form of the objective function that is commonly used.

$$\text{Objective Function } = E = \sum_{i=1}^{nobs} w_i \left( d_i^{obs} - d_i^{calc} \right)^2 \tag{2.1}$$

where *nobs* is the number of observations (data), $d_i^{obs}$ are the measured data, $d_i^{cal}$ are the data as calculated by the mathematical model of the reservoir, and $w_i$ are factors used to weight and equalize units.

The use of the vectorial notation results in equations that are compact to write and easy to understand and manipulate. Thus the following notation is used:

- $\vec{\alpha}$ is the vector of the set of parameters of the system. $\vec{\alpha} \in R^{npar}$ where $npar$ is the number of parameters that define the system.

$$\vec{\alpha} = \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_{npar} \end{bmatrix} \tag{2.2}$$

- $\vec{d}^{obs}$ and $\vec{d}^{cal}$ are the vectors of the set of measurements and their values calculated by the mathematical model respectively. $\vec{d}^{obs}$ , $\vec{d}^{cal} \in R^{nobs}$ where $nobs$ is the number of observations.

$$\vec{d} = \begin{bmatrix} d_1 \\ \vdots \\ d_{nobs} \end{bmatrix} \tag{2.3}$$

The objective function can be defined in different ways provided it is a measure of discrepancy. The most common forms found in the literature are briefly described in the following subsections.

### 2.3.1 Least Square Formulation

This is a very well known formulation. $E$ is expressed as:

$$E = \left(\vec{d}^{obs} - \vec{d}^{cal}\right)^T \left(\vec{d}^{obs} - \vec{d}^{cal}\right) \tag{2.4}$$

### 2.3.2 Weighted Least Square Formulation

This is a simple modification to the least square formulation and it is written as follows:

$$E = \left(\vec{d}^{obs} - \vec{d}^{cal}\right)^T \mathbf{W} \left(\vec{d}^{obs} - \vec{d}^{cal}\right) \tag{2.5}$$

$\mathbf{W}$ is a diagonal matrix that assigns individual weights to each measurement. This is done basically to scale the measurements and there are two main reasons to do this: to unify the units in cases where the measurements have different units and to take into consideration the relative importance to be assigned to different measurements. The latter is specially important when the data set is composed of subsets of measurements whose absolute values or number of data are of different order of magnitude, or whose degrees of uncertainty (confidence in the measurement) are not the same.

### 2.3.3 Generalized Least Square Formulation

This formulation is derived from the theory of probability. The objective function $E$ is written as follows:

$$\begin{aligned} E &= \tfrac{1}{2}\left[\left(\vec{d}^{obs} - \vec{d}^{cal}\right)^T \mathbf{C}_d^{-1} \left(\vec{d}^{obs} - \vec{d}^{cal}\right)\right] + \\ &\quad \tfrac{1}{2}\left[\left(\vec{\alpha} - \vec{\alpha}_{prior}\right)^T \mathbf{C}_\alpha^{-1} \left(\vec{\alpha} - \vec{\alpha}_{prior}\right)\right] \end{aligned} \tag{2.6}$$

where $\mathbf{C}_d$ is the covariance matrix of the data. $\mathbf{C}_d$ provides information about the correlation among the data. In general it is assumed that data are independent of each other and thus this matrix becomes diagonal and the nonzero elements are the variance of the data, that is the square of the standard deviation ($\sigma_d^2$). $\mathbf{C}_\alpha$ is the covariance matrix of the parameters of the mathematical model, $\vec{\alpha}_{prior}$ is *a priori* information about the parameters. $\vec{\alpha}_{prior}$ is obtained before the application of the procedure for inversion. It may come as the result of a previous inverse problem.

A brief but clear derivation of this equation can be found in Chapter 5 of Menke [15]. A more complete treatment can be found in Tarantola [14]. Both books deal with inverse problems in the field of geophysics.

This formulation allows us to introduce into the least square formulation both the *a priori* and the *statistical* information about the parameters of the system. This approach was first introduced in reservoir parameter estimation in 1994 by Oliver [12].

The $\mathbf{C}_\alpha$ matrix is obtained from geostatistical information. This probabilistic formulation allows us to integrate the information about the spatial correlation of permeability and porosity directly into the least square formulation.

## 2.4 Parameter Estimation Algorithms

### 2.4.1 Classification

Common to all parameter estimation algorithms is that they try to minimize a discrepancy function (*objective function*). The problem of mathematical minimization is within the field of operations research (*optimization*) where the objective is to minimize a *cost function*. Thus, since parameter estimation is a subproblem of optimization, then any of the already available tools for optimization can be adapted to be used in inverse problem solving.

One of the characteristics of reservoir parameter estimation is that the objective function $E$ to minimize is *nonlinear* with respect to the parameters; consequently all the algorithms rely on iterative procedures that minimize by a succession of changes to a given first set of parameters.

The optimization algorithms can be classified depending on whether they use the gradient of the *objective function* or not, where the gradient of $E$ is defined as:

$$\nabla E \equiv \left(\frac{\partial E}{\partial \vec{\alpha}}\right)^T \tag{2.7}$$

The following is a list of the most commonly used algorithms for each category.

**Methods**

| **Gradient** | **Nongradient** |
|---|---|
| Steepest Descent | Simulated Annealing |
| Gauss–Newton | Genetic Algorithm |
| Singular Value Decomposition | Polytope |
| Conjugate Gradient | |
| Quasi–Newton | |

This work used the Gauss-Newton and Singular Value Decomposition algorithms for parameter estimation. These algorithms will be explained in detail in Sections 2.5.2 and 2.5.3.

### 2.4.2 Optimality Conditions

As was mentioned earlier, parameter estimation is performed by solving a multivariate minimization problem, that is to find an optimal $\vec{\alpha} = \vec{\alpha}^*$ such that,

$$E(\vec{\alpha}^*) = \min_{\vec{\alpha}} E(\vec{\alpha}) \tag{2.8}$$

$\vec{\alpha}^*$ will be a global minimum when,

$$E(\vec{\alpha}^* + \Delta\vec{\alpha}) > E(\vec{\alpha}^*)\,; \forall \parallel \Delta\vec{\alpha} \parallel > 0 \tag{2.9}$$

$\vec{\alpha}^*$ will be a local minimum when,

$$E(\vec{\alpha}^* + \Delta\vec{\alpha}) > E(\vec{\alpha}^*)\,; \forall 0 < \parallel \Delta\vec{\alpha} \parallel \leq \epsilon \tag{2.10}$$

$\vec{\alpha}^*$ is referred to as the *optimal point*. Since the main purpose of the inverse problem is to solve Equation 2.8, it is important to review the properties of $E$ at the optimal point.

For the case of *unconstrained* optimization and *smooth* functions, the necessary conditions for optimality are:

$$\nabla E(\vec{\alpha}^*) = \left.\frac{\partial E^T}{\partial \vec{\alpha}}\right|_{\vec{\alpha}^*} = 0 \tag{2.11}$$

and,

$$\vec{x}^T \mathbf{H}^* \vec{x} > 0\,; \forall \vec{x} \neq 0 \tag{2.12}$$

The matrix $\mathbf{H}^*$ is the *Hessian* matrix of the second derivatives of $E$ evaluated at $\vec{\alpha} = \vec{\alpha}^*$. Equation 2.12 means that the matrix $\mathbf{H}^*$ is *positive definite*.

The *Hessian* of $E$ is defined as follows:

$$\mathbf{H} = \frac{\partial \nabla E}{\partial \vec{\alpha}} \tag{2.13}$$

$$H_{i,j} = \frac{\partial^2 E}{\partial \alpha_i \, \partial \alpha_j} \tag{2.14}$$

$$H_{i,j} = H_{j,i} \Leftrightarrow \mathbf{H} = \mathbf{H}^T \tag{2.15}$$

$$\mathbf{H} = \begin{pmatrix} H_{1,1} & \cdots & H_{1,npar} \\ \vdots & \ddots & \vdots \\ H_{npar,1} & \cdots & H_{npar,npar} \end{pmatrix} \tag{2.16}$$

For the case of *constrained optimization* the necessary conditions for optimality are given by the *Karush–Khun–Tucker conditions*, commonly referred to as the *KKT* conditions. These conditions are, as stated in page 520 of reference [44]:

$\vec{x}^*$ is an optimal point of the following problem,

$$\begin{array}{lll} \vec{x}^* = & \max\limits_{\vec{x}} \; f(\vec{x}), & \vec{x} \in R^n \\ \text{subject to} & g_i(\vec{x}) \leq b_i, \;\; \text{for } i = 1, 2, \ldots, m & \\ \text{and} & \vec{x} \geq 0 & \end{array} \tag{2.17}$$

" assuming that $f(\vec{x})$ and $g_1(\vec{x}), g_2(\vec{x}), \ldots, g_m(\vec{x})$ are differentiable satisfying certain regularity conditions. Then $\vec{x}^* = (x_1^*, x_2^*, \ldots, x_n^*)$ can be an optimal solution for the nonlinear programing problem if there exist $m$ numbers, $u_1, u_2, \ldots, u_m$ such that the following conditions are satisfied:"

$$\left\{\begin{array}{lll}
\left.\begin{array}{ll} 1. & \frac{\partial f}{\partial x_j} - \sum_{i=1}^{m} u_i \frac{\partial g_i}{\partial x_j} \leq 0 \\ 2. & x_j^* \left( \frac{\partial f}{\partial x_j} - \sum_{i=1}^{m} u_i \frac{\partial g_i}{\partial x_j} \right) = 0 \end{array}\right\} & \text{at } \vec{x} = \vec{x}^*, & \text{for } j = 1, 2, \ldots, n \\
\left.\begin{array}{ll} 3. & g_i(\vec{x}^*) - b_i \leq 0 \\ 4. & u_i \left( g_i(\vec{x}^*) - b_i \right) = 0 \end{array}\right\} & & \text{for } i = 1, 2, \ldots, m \\
5. \quad x_j^* \geq 0 & & \text{for } j = 1, 2, \ldots, n \\
6. \quad u_i \geq 0 & & \text{for } i = 1, 2, \ldots, m
\end{array}\right. \tag{2.18}$$

The problems examined in this work involve inequality constraints, so it would be reasonable to look for the points that satisfy the $KKT$ conditions. Unfortunately that approach is very hard to solve. In this work we will solve the constrained problem by solving an equivalent unconstrained problem that is easier to solve than the constrained one. Still the equivalent unconstrained problem will not violate the inequality constraints. The main difference between the constrained and the unconstrained approaches is that in the latter it is not permissible to reach the constraint whereas in the former it is. This difference seems subtle but it makes a lot of difference in the algorithms. Consequently the algorithms we used in this work attempt to satisfy the optimality conditions as stated in Equations 2.11 and 2.12. Equations 2.9 and 2.10 imply Equations 2.11 and 2.12. The reverse only implies 2.10.

Since Equation 2.12 is relevant to the algorithms used in this work, it is worth reviewing a short derivation.

By assuming smoothness, the function $E$ can be approximated in the neighborhood of $\vec{\alpha}$ with a second order Taylor expansion.

$$E\left(\vec{\alpha} + \Delta\vec{\alpha}\right) = E\left(\vec{\alpha}\right) + \nabla E^T \Delta\vec{\alpha} + \frac{1}{2} \Delta\vec{\alpha}^T \mathbf{H} \Delta\vec{\alpha} + O(3) \tag{2.19}$$

where $\nabla E$ and $\mathbf{H}$ are evaluated at $\vec{\alpha}$.

When $\vec{\alpha} \rightarrow \vec{\alpha}^*$ then $\nabla E \rightarrow \nabla E^* = 0$ and $\mathbf{H} \rightarrow \mathbf{H}^*$.

Assuming a small $\Delta\vec{\alpha}$ Equation 2.19 becomes:

$$E(\vec{\alpha}^* + \Delta\vec{\alpha}) - E(\vec{\alpha}^*) \approx \frac{1}{2} \Delta\vec{\alpha}^T \mathbf{H}^* \Delta\vec{\alpha} \tag{2.20}$$

and then $\mathbf{H}^*$ has to be positive definite if $\vec{\alpha}^*$ is a minimum.

Many algorithms for minimization have been developed around the concept that the matrix $\mathbf{H}^*$ has to be positive definite at the minimum.

## 2.5 Gradient Based Algorithms

The main underlying assumptions in most gradient based algorithms are:

- The objective function $E$ is smooth.
- It is possible to compute $\nabla E$ and/or $\nabla d_i$.

The basic concept in this approach is that, assuming that $\nabla E \neq 0$ for a given value of $\vec{\alpha} = \vec{\alpha_0}$, then it is always possible to reduce the value of $E$ from its current value by introducing a change in the parameters (at this time this statement may seem trivial, but it will lead to an important conclusion). That is:

$$E(\vec{\alpha}_0 + \rho\,\vec{p}) < E(\vec{\alpha}_0) \tag{2.21}$$

where $\vec{p}$ is the unit vector that provides a *direction* for descent ( $\| \vec{p} \| = 1$) and $\rho$ is a positive scalar that provides the *step size* ($\rho > 0$). The existence of $\vec{p}$ and $\rho$ are proven as follows.

The function $E$ is approximated by a first order Taylor expansion around an already given point in the paramenter space $\vec{\alpha}_0$.
That is,

$$E(\vec{\alpha}_0 + \rho\,\vec{p}) = E(\vec{\alpha_0}) + \rho\,\nabla E^T \vec{p} + \text{ second order terms} \tag{2.22}$$

From Equation 2.22 we see that it is *always* possible to find a positive value of $\rho$ that results in a reduction of the value of $E$ provided that $\vec{p}$ satisfies the following condition:

$$\nabla E^T \vec{p} < 0 \tag{2.23}$$

When $\vec{p}$ satisfies the condition stated in Equation 2.23, $\vec{p}$ is said to be a *direction of sufficient descent*. Thus the existence of $\rho$ and $\vec{p}$ are guaranteed provided that $\nabla E \neq 0$.

The structure of the gradient algorithms can be synthesized as follows:

- Compute a *sufficient descent direction* ($\vec{p}$).
- Compute an adequate *step size* ($\rho$).

The algorithms will terminate when:

$$\nabla E = 0 \tag{2.24}$$

which is one of the necessary conditions for unconstrained optimality. From a practical point of view the algorithms will be terminated when $\| \nabla E \| < \epsilon$, where $\epsilon$ is an arbitrarily small positive number.

One of the main shortcomings of these methods is that they may become stuck at a local stationary point which in general does not coincide with the global minimum.

The most commonly used algorithms of this group are: the steepest descent, Gauss–Newton, Singular Value Decomposition, quasi–Newton and conjugate gradient.

A thorough mathematical presentation of gradient based algorithms for optimization can be found in references [8] and [45]. The use of these algorithms in nonlinear parameter estimation is detailed in reference [46].

### 2.5.1 Steepest Descent Algorithm

In this approach the direction of sufficient descent is chosen as:

$$\vec{p} = \frac{-\nabla E}{\| \nabla E \|} \tag{2.25}$$

since $\nabla E^T \nabla E < 0$.

This is the simplest approach. Here the change in the parameters is introduced in the direction of $-\nabla E$ which is the natural direction for descent. There are two problems with this approach. First the method does not provide a hint about the appropriate value of $\rho$. Second it has been illustrated in the literature that the direction of descent computed with this method is very inefficient. Also the procedure has only linear convergence at the optimal point.

### 2.5.2 Gauss-Newton Algorithm

This is the main algorithm used in this work. The algorithm is a special case of Newton's method for optimization. The advantages of this method are twofold. First, the algorithm provides both $\vec{p}$ and $\rho$. Second, the procedure converges quadratically at the optimal point. A short derivation follows.

As mentioned earlier the objective is to minimize the function $E$. Assuming smoothness, the function $E$ can be approximated in the neighborhood of $\vec{\alpha}_0$ with a second order Taylor expansion.

$$E(\vec{\alpha}) = E(\vec{\alpha}_0 + \Delta\vec{\alpha}) \approx E(\vec{\alpha}_0) + \nabla E^T \Delta\vec{\alpha} + \frac{1}{2} \Delta\vec{\alpha}^T \mathbf{H} \Delta\vec{\alpha} \tag{2.26}$$

where $\nabla E$ and $\mathbf{H}$ are computed at $\vec{\alpha}_0$.

We use this second order approximation to compute a direction and step size that result in a reduction of $E$, thus we compute the gradient of Equation 2.26 with respect to $\vec{\alpha}$ and set it equal to zero.

$$\nabla E(\vec{\alpha}) = \nabla E + \mathbf{H} \Delta\vec{\alpha} = 0 \tag{2.27}$$

and,

$$\mathbf{H}\,\Delta\vec{\alpha} \;=\; -\nabla E \tag{2.28}$$

The application of Equation 2.28 to compute a direction for descent is known as *Newton's* method. As it will be shown, this method cannot guarantee a direction for descent. To be a good direction for descent the condition set in Equation 2.23 has to be true, that is $\nabla E^T\,\Delta\vec{\alpha} \;<\; 0$. Premultiplying Equation 2.28 by $\Delta\vec{\alpha}^T$ then transposing and changing signs, we obtain,

$$\nabla E^T\,\Delta\vec{\alpha} \;=\; -\Delta\vec{\alpha}^T\,\mathbf{H}\,\Delta\vec{\alpha} \tag{2.29}$$

which does not guarantee a direction of descent unless the matrix $\mathbf{H}$ is positive definite. The conclusion is that the direct application of Newton's method can lead to the wrong direction for descent and the method may fail. Newton's method will work when we are close to the optimal point since, as was shown in Equation 2.12, there $\mathbf{H}$ is positive definite; moreover the rate of convergence will be quadratic, which means fast convergence. From the previous discussion arises an interesting observation, that by using *any* positive definite matrix $\tilde{\mathbf{H}}$ instead of the actual $\mathbf{H}$ it is possible to have a method that guarantees the computation of a direction for descent. Obviously the closer is the approximation to the true $\mathbf{H}$, the better will be the direction that is computed. One concern with Newton's method is that it requires computation not only of the first derivative of $E$ ($\nabla E$) but also of the second derivatives as well ($\mathbf{H}$). This may become expensive from the computational point of view when the mathematical model is complex.

The idea of computing the direction for descent by using a positive definite approximation to $\mathbf{H}$ has suggested many algorithms. For example one trivial approximation for $\mathbf{H}$ is the identity matrix $\mathbf{I}$ which gives rise to the *steepest descent* method described in Section 2.5.1.

The Gauss–Newton algorithm used in this work is derived directly from the Newton approach and exploits the special structure of the function $E$ to compute an

approximation of **H** that is positive definite, is cheap to compute since it requires first derivatives only, and becomes the true **H** at the optimal point, therefore providing quadratic convergence.

The derivation of the Gauss–Newton method for the case of the weighted least squared and the generalized least squared formulation is developed here.

In the case of the weighted least squared formulation, $E$ was defined as:

$$E(\vec{\alpha}) = \left(\vec{d}^{obs} - \vec{d}^{cal}\right)^T \mathbf{W} \left(\vec{d}^{obs} - \vec{d}^{cal}\right) \tag{2.30}$$

and

$$\vec{d}^{cal} = \vec{d}^{cal}(\vec{\alpha}) \tag{2.31}$$

$\vec{\alpha} \in R^{npar}$ and $\vec{d}^{calc}$, $\vec{d}^{obs} \in R^{nobs}$.

By taking the derivative of Equation 2.31 with respect to the vector of parameters $\vec{\alpha}$ we obtain what it is known as the sensitivity matrix **G**, that is:

$$\mathbf{G} \equiv \frac{\partial \vec{d}^{cal}}{\partial \vec{\alpha}} = [\vec{g}_1\, \vec{g}_2\, \cdots\, \vec{g}_i\, \cdots\, \vec{g}_{npar}] \tag{2.32}$$

where $\vec{g}_i$ means:

$$\vec{g}_i \equiv \frac{\partial \vec{d}^{cal}}{\partial \alpha_i} = \begin{bmatrix} \frac{\partial d_1^{cal}}{\partial \alpha_i} \\ \frac{\partial d_2^{cal}}{\partial \alpha_i} \\ \vdots \\ \frac{\partial d_{nobs}^{cal}}{\partial \alpha_i} \end{bmatrix} \tag{2.33}$$

The definition of *sensitivity coefficient* $s_{i,j}$ follows as:

$$s_{i,j} = \frac{\partial d_i^{calc}}{\partial \alpha_j} \tag{2.34}$$

The magnitude of $s_{i,j}$ is an indication of how much $d_i^{calc}$ is affected by a change in $\alpha_j$.

The efficient computation of the sensitivity matrix **G** was one of the central issues in this work.

By assuming that the matrix $\mathbf{W}$ is symmetric with constant coefficients and considering Equation 2.32, the calculation of $\nabla E$ and the Hessian matrix $\mathbf{H}$ of $E$ follows as:

$$\nabla E = -2\mathbf{G}^T\mathbf{W}\left(\vec{d}^{obs} - \vec{d}^{cal}\right) \tag{2.35}$$

$$\mathbf{H} = \frac{\partial \nabla E}{\partial \vec{\alpha}} = 2\mathbf{G}^T\mathbf{W}\,\mathbf{G} - 2\left[\vec{v}_1\,\vec{v}_2\,\cdots\,\vec{v}_i\,\cdots\,\vec{v}_{npar}\right] \tag{2.36}$$

where $\vec{v}_i$ means:

$$\vec{v}_i \equiv \frac{\partial \mathbf{G}^T}{\partial \alpha_i}\mathbf{W}\left(\vec{d}^{obs} - \vec{d}^{cal}\right) \tag{2.37}$$

The Gauss–Newton Hessian matrix $\mathbf{H}_{GN}$ is defined as the first term on the right hand side of Equation 2.36. That is:

$$\mathbf{H}_{GN} \equiv 2\mathbf{G}^T\mathbf{W}\,\mathbf{G} \tag{2.38}$$

If we also impose the condition that the matrix $\mathbf{W}$ has to be positive definite then we guarantee that $\mathbf{H}_{GN}$ will be at least semipositive definite which is not far from the requirement of positive definitiveness. The lack of positive definitiveness can be cured with a further modification of matrix $\mathbf{H}_{GN}$. This is referred as stabilization and will be discussed later. For the time being we can assume that $\mathbf{H}_{GN}$ is positive definite.

The Gauss–Newton formulation is then constructed by computing the descent direction and step size following Equation 2.28 but using $\mathbf{H}_{GN}$ as an approximation to $\mathbf{H}$, that is:

$$\mathbf{H}_{GN}\,\Delta\vec{\alpha} \;=\; -\nabla E \tag{2.39}$$

which becomes after replacing the definitions of $\nabla E$ and $\mathbf{H}_{GN}$:

$$\mathbf{G}^T\,\mathbf{W}\,\mathbf{G}\,\Delta\vec{\alpha} \;=\; \mathbf{G}^T\,\mathbf{W}\,\left(\vec{d}^{obs} - \vec{d}^{cal}\right) \tag{2.40}$$

The main advantages of using the Gauss–Newton formulation are:

- *Right descent direction.* Since the matrix $\mathbf{H}_{GN}$ is guaranteed to be positive–definite the direction computed with Equation 2.39 is always descending.
- *Quadratic convergence.* In the limit, at the optimal point $\mathbf{H}_{GN} \rightarrow \mathbf{H}$ and consequently the convergence becomes quadratic.
- *First derivatives.* To compute $\mathbf{H}_{GN}$ it is necessary to calculate first derivatives only (gradient of the data $\mathbf{G}$). Second derivatives are not required.

The most efficient way to solve Equation 2.39 is by using *Cholesky* factorization. This approach is numerically stable and efficient, and also allows the checking of the requirement of positive definiteness of $\mathbf{H}_{GN}$. The Cholesky factors of a matrix exist only when the matrix is positive definite, if the Cholesky factorization fails then it means that the matrix is not positive definite and then the matrix has to be be stabilized. The process of stabilization is explained later in Section 2.11.

Understanding the reasons for the lack of positive definiteness is important since they provide an insight into the inverse problem. Numerical semipositive definiteness or singularity in $\mathbf{H}_{GN}$ arises directly from the structure of $\mathbf{G}$. For example if:

- Data are insufficient to resolve some parameters.
  - for some $i$, $\frac{\partial \vec{d}^{calc}}{\partial \alpha_i} = 0$ will make $\mathbf{H}_{GN}$ singular.
  - for some $i$, $\| \frac{\partial \vec{d}^{calc}}{\partial \alpha_i} \| \ll \| \frac{\partial \vec{d}^{calc}}{\partial \alpha_j} \| \quad \forall\, j \neq i$ will make $\mathbf{H}_{GN}$ numerically ill–conditioned.
- Linear combination of parameters.
  - For some $i$, $\alpha_i = \sum_{j=1\, j\neq i}^{npar} \lambda_j\, \alpha_j$ will result in columns vectors of $\mathbf{G}$ that are not linearly independent and consequently $\mathbf{H}_{GN}$ becomes singular.
  - For some $i$, $\alpha_i \approx \sum_{j=1\, j\neq i}^{npar} \lambda_j\, \alpha_j$ will create numerically instability in $\mathbf{H}_{GN}$ for the same reasons as above. This is equivalent to saying that for some $i$, $\frac{\partial \vec{d}^{cal}}{\partial \alpha_i} \approx \sum_{j=1\, j\neq i}^{npar} \lambda_j \frac{\partial \vec{d}^{cal}}{\partial \alpha_j}$.

For the simple case of *weighted least squares* the actual computation of the gradient and Hessian of $E$ is performed with the following formulas:

$$(\nabla E)_i = -2 \sum_{k=1}^{nobs} \frac{\partial d_k^{cal}}{\partial \alpha_i} w_k \left( d_k^{obs} - d_k^{cal} \right) \tag{2.41}$$

$$(\mathbf{H}_{GN})_{i,j} = 2 \sum_{k=1}^{nobs} w_k \frac{\partial d_k^{cal}}{\partial \alpha_i} \frac{\partial d_k^{cal}}{\partial \alpha_j} \tag{2.42}$$

In the case of the *generalized least square formulation*, as presented in Equation 2.6, the actual computation of the gradient and Hessian is performed with the following formulas:

$$\begin{aligned} E &= \tfrac{1}{2} \left[ \left( \vec{d}^{obs} - \vec{d}^{cal} \right)^T \mathbf{C}_d^{-1} \left( \vec{d}^{obs} - \vec{d}^{cal} \right) \right] + \\ &\quad \tfrac{1}{2} \left[ (\vec{\alpha} - \vec{\alpha}_{prior})^T \mathbf{C}_\alpha^{-1} (\vec{\alpha} - \vec{\alpha}_{prior}) \right] \end{aligned} \tag{2.43}$$

$$\nabla E = -\mathbf{G}^T \mathbf{C}_d^{-1} \left( \vec{d}^{obs} - \vec{d}^{cal} \right) + \mathbf{C}_\alpha^{-1} (\vec{\alpha} - \vec{\alpha}_{prior}) \tag{2.44}$$

$$\mathbf{H} = \frac{\partial \nabla E}{\partial \vec{\alpha}} = \mathbf{G}^T \mathbf{C}_d^{-1} \mathbf{G} + \mathbf{C}_\alpha^{-1} - [\vec{v}_1 \, \vec{v}_2 \cdots \vec{v}_i \cdots \vec{v}_{npar}] \tag{2.45}$$

and the Gauss–Newton Hessian matrix $\mathbf{H}_{GN}$ is defined as:

$$\mathbf{H}_{GN} \equiv \mathbf{G}^T \mathbf{C}_d^{-1} \mathbf{G} + \mathbf{C}_\alpha^{-1} \tag{2.46}$$

The matrices $\mathbf{C}_d$ and $\mathbf{C}_\alpha$ are always positive definite because they are the covariance of the data and the model respectively and consequently the matrix $\mathbf{H}_{GN}$ is positive definite. It is important to note that the matrix $\mathbf{C}_\alpha$ introduces stabilization by itself for the case of ill–conditioning in $\mathbf{G}^T \mathbf{C}_d^{-1} \mathbf{G}$.

A key step in the implementation of the Gauss–Newton method is in the computation of the sensitivity matrix $\mathbf{G}$, that is $\frac{\partial \vec{d}^{cal}}{\partial \vec{\alpha}}$. For analytical models the computation can be done directly. In this work the mathematical model is a numerical reservoir simulator and consequently the evaluation of $\vec{d}^{cal}$ is computationally intensive from

the numerical point of view and has to be done with care, the details are given in Chapter 3.

### 2.5.3 Singular Value Decomposition Algorithm

This approach is based on the *Singular Value Decomposition* or *Spectral Decomposition* of a matrix. That is, any square or nonsquare $m$ x $n$ matrix $\mathbf{A}$ can be factored as follows:

$$\mathbf{A} = \mathbf{U}\,\mathbf{S}\,\mathbf{V}^T \tag{2.47}$$

where $\mathbf{U}$ is ($m$ x $m$) orthogonal, $\mathbf{S}$ is a diagonal matrix ($m$ x $n$) whose elements are nonnegative, and $\mathbf{V}$ is ($n$ x $n$) orthogonal.

$$\mathbf{U}^T\mathbf{U} = \mathbf{U}\mathbf{U}^T = \mathbf{I}_m \tag{2.48}$$

$$\mathbf{V}^T\mathbf{V} = \mathbf{V}\mathbf{V}^T = \mathbf{I}_n \tag{2.49}$$

The matrices $\mathbf{U}$ and $\mathbf{V}$ and $\mathbf{S}$ can be partitioned as follows:

$$\mathbf{U} = [\mathbf{U}_p\,\mathbf{U}_0] \tag{2.50}$$

where the columns of $\mathbf{U}_p$ are the columns of $\mathbf{U}$ corresponding to the nonzero diagonal elements of $\mathbf{S}$ (referred to as *singular values* of $\mathbf{A}$).

The same partition method is applied to $\mathbf{V}$

$$\mathbf{V} = [\mathbf{V}_p\,\mathbf{V}_0] \tag{2.51}$$

Since $\mathbf{S}$ is diagonal, then it is partitioned as follows:

$$\mathbf{S} = \begin{bmatrix} \mathbf{S}_p & \\ & \mathbf{0} \end{bmatrix} \tag{2.52}$$

the diagonal matrix $\mathbf{S}_p$ is square ($p$ x $p$).

It follows that the matrix $\mathbf{A}$ can be factored in a short form, as:

$$\mathbf{A} = \mathbf{U}_p \, \mathbf{S}_p \, \mathbf{V}_p^T \tag{2.53}$$

and in the long form:

$$\mathbf{A} = [\mathbf{U}_p \, \mathbf{U}_0] \begin{bmatrix} \mathbf{S}_p & \\ & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{V}_p^T \\ \mathbf{V}_0^T \end{bmatrix} \tag{2.54}$$

The generalized inverse of $\mathbf{A}$ is defined as :

$$\mathbf{A}^{-g} = \mathbf{V}_p \, \mathbf{S}_p^{-1} \, \mathbf{U}_p^T \tag{2.55}$$

and $\mathbf{A}^{-g} = \mathbf{A}^{-1}$ when $\mathbf{S} = \mathbf{S}_p$.

The singular value decomposition is used to solve nonsquare linear systems of the type:

$$\mathbf{A} \, \vec{x} \; = \; \vec{b} \; \longrightarrow \; \vec{x}_{SVD} \; = \; \mathbf{A}^{-g} \, \vec{b} \tag{2.56}$$

An interpretation of the meaning of $\mathbf{U}$ and $\mathbf{V}$ follows.

$$\mathbf{A} \, \vec{x} \; = \; \mathbf{U} \, \mathbf{S} \, \mathbf{V}^T \, \vec{x} \; = \; \vec{b} \tag{2.57}$$

then it is said that the columns of $\mathbf{U}$ span the vector space of $\vec{b}$ and the columns of $\mathbf{V}$ span the vector space of $\vec{x}$. This will be important to understand the concepts of resolution of parameters in Section 2.6.

The *singular value decomposition* is used in nonlinear parameter estimation as follows.

A first order approximation of $\vec{d}^{calc}(\vec{\alpha})$ around a given $\vec{\alpha}_0$ will give:

$$\vec{d}^{calc}(\vec{\alpha}_0 + \Delta\vec{\alpha}) = \vec{d}^{calc}(\vec{\alpha}_0) + \frac{\partial \vec{d}^{calc}}{\partial \vec{\alpha}} \Delta\vec{\alpha} \tag{2.58}$$

By setting:

$$\vec{d}^{calc}(\vec{\alpha}_0 + \Delta\vec{\alpha}) = \vec{d}^{obs} \tag{2.59}$$

and recalling $\frac{\partial \vec{d}^{calc}}{\partial \vec{\alpha}} = \mathbf{G}$ we obtain from Equation 2.58:

$$\mathbf{G}\,\Delta\vec{\alpha} \;=\; \left(\vec{d}^{obs} - \vec{d}^{calc}\right) \tag{2.60}$$

Equation 2.60 is then solved using singular value decomposition,

$$\mathbf{G} \;=\; \mathbf{U}_p\,\mathbf{S}_p\,\mathbf{V}_p^T \tag{2.61}$$

$$\Delta\vec{\alpha}_{SVD} \;=\; \mathbf{G}^{-g}\,\left(\vec{d}^{obs} - \vec{d}^{calc}\right) \tag{2.62}$$

$$\Delta\vec{\alpha}_{SVD} \;=\; \mathbf{V}_p\,\mathbf{S}_p^{-1}\,\mathbf{U}_p^T\,\left(\vec{d}^{obs} - \vec{d}^{calc}\right) \tag{2.63}$$

We now verify that $\Delta\vec{\alpha}_{SVD}$ is a *descent direction.*

$$\begin{aligned}
\nabla E^T\,\Delta\vec{\alpha}_{SVD} &= -2\left(\vec{d}^{obs} - \vec{d}^{calc}\right)^T\,\mathbf{W}\,\mathbf{G}\,\mathbf{G}^{-g}\,\left(\vec{d}^{obs} - \vec{d}^{calc}\right) && (2.64)\\
&= -2\left(\vec{d}^{obs} - \vec{d}^{calc}\right)^T\,\mathbf{W}\,\mathbf{U}_p\,\mathbf{U}_p^T\,\left(\vec{d}^{obs} - \vec{d}^{calc}\right) && (2.65)\\
&< \;0\;,\ \text{if}\ \mathbf{U}_p^T\,\left(\vec{d}^{obs} - \vec{d}^{calc}\right) \neq 0 && (2.66)
\end{aligned}$$

since $\mathbf{W}$ and $\mathbf{U}_p\,\mathbf{U}_p^T$ are positive definite.

The algorithm will stop when

$$\mathbf{U}_p^T\,\left(\vec{d}^{obs} - \vec{d}^{calc}\right) \;=\; 0 \tag{2.67}$$

and this condition arises when the $\nabla E = 0$

$$\nabla E = \mathbf{G}^T \left( \vec{d}^{obs} - \vec{d}^{calc} \right) \tag{2.68}$$

$$= \mathbf{V}_p \, \mathbf{S}_p \, \mathbf{U}_p^T \left( \vec{d}^{obs} - \vec{d}^{calc} \right) \tag{2.69}$$

$$\nabla E = 0 \iff \mathbf{U}_p^T \left( \vec{d}^{obs} - \vec{d}^{calc} \right) = 0 \tag{2.70}$$

This means that the *singular value decomposition* method like the Gauss–Newton approach will fail at local minima and saddle points.

The advantage of this method is that it takes direct account of the singularities introduced by $\mathbf{G}$, such as by correlation of parameters or redundancy of data. This is accomplished at the time of choosing the singular values to form $\mathbf{S}_p$. Since only $\mathbf{G}$ is used and not $\mathbf{H}$, this method is also computationally efficient since second derivatives are not required.

### 2.5.4 Other Gradient Methods

References [8], [45] describe other gradient methods for optimization that can be used as an alternative to the Gauss-Newton method used in this work. It is likely that the most important are the *conjugate gradient* and the *quasi–Newton* algorithms. The importance of these methods in the reservoir parameter estimation problem is that these methods provide a method to avoid the computation of the sensitivity coefficients required by the Gauss-Newton algorithm. The *conjugate gradient* and the *quasi–Newton* algorithms do not require the computation of the Hessian matrix; the only requirement is to be able to compute $\nabla E$, which can be achieved without sensitivity coefficients by using optimal theory [5], [4].

The *conjugate gradient* algorithm is very simple and easy to implement after $\nabla E$ has been calculated. The main advantage is that it requires very little storage. The problem with this algorithm is that it is seriously affected by numerical errors which arise from the finite precision arithmetic of computers.

The *quasi–Newton* algorithm also requires $\nabla E$ as input, the algorithm computes an approximation to the Hessian matrix which is updated after each iteration with

the change in $\nabla E$. The most common of the algorithms in this category is the *BFGS (Broyden–Fletcher–Goldfarb–Shanno)* method.

The *conjugate gradient* and *quasi–Newton* algorithms are a large improvement over the *steepest descent* algorithm that also uses $\nabla E$ as the only source of information.

### 2.5.5 Nongradient Methods

A short but clear description of the use of *simulated annealing* and *genetic* algorithms for reservoir description can be found in Reference [47]. The attraction to these methods is that they are relatively simple to implement, do not require the computation of either $\nabla E$ or the sensitivity coefficients and also are better able to reach a global minimum of the objective function $E$. The main disadvantage is that they are very expensive from the numerical point of view since they require a very large number of functions evaluations, and this may become critical when such functions evaluations involve the use of a reservoir numerical simulator.

## 2.6 Resolution of Parameters

The theory for the resolution of parameters in linear problems is described among others by Jackson [43] and Menke [15]. These two references use the theory developed in *singular value* or *spectral* decomposition of a matrix. We will use this theory for the nonlinear case.

### 2.6.1 Development for Linear Case

#### Resolution and Density Information Matrices

Assuming a linear mathematical model for the problem, that is expressed as a matrix operator $\mathbf{A}$ with constant coefficients, we have:

$$\vec{d}^{cal} = \mathbf{A}\,\vec{\alpha} \tag{2.71}$$

We set up the system,

$$\mathbf{A}\,\vec{\alpha} \;=\; \vec{d}^{obs} \tag{2.72}$$

In general the matrix $\mathbf{A}$ is not square, the dimensions are ($nobs$ x $npar$) and thus the existence of $\mathbf{A}^{-1}$ is not guaranteed. We may assume that the linear system represented by Equation 2.72 can be solved by another linear operator $\tilde{\mathbf{H}}$ resulting in:

$$\tilde{\vec{\alpha}} \;=\; \tilde{\mathbf{H}}\,\vec{d}^{obs} \tag{2.73}$$

$$\tilde{\vec{d}}^{cal} \;=\; \tilde{\mathbf{H}}\,\tilde{\vec{\alpha}} \tag{2.74}$$

From Equations 2.73 and 2.71 we obtain:

$$\tilde{\vec{\alpha}} \;=\tilde{\mathbf{H}}\,\mathbf{A}\,\vec{\alpha} \tag{2.75}$$

$$\tilde{\vec{\alpha}} \;=\mathbf{R}\vec{\alpha} \tag{2.76}$$

The matrix $\mathbf{R} \;\equiv\; \tilde{\mathbf{H}}\,\mathbf{A}$ is called the *resolution matrix*. The resolution matrix determines the relationship between the estimated solution $\tilde{\vec{\alpha}}$ and the true solution $\vec{\alpha}$.

From Equations 2.74 and 2.72 we obtain:

$$\tilde{\vec{d}}^{cal} \;=\; \mathbf{A}\,\tilde{\mathbf{H}}\,\vec{d}^{obs} \tag{2.77}$$

The matrix $\mathbf{S}_{inf} \;\equiv\; \mathbf{A}\,\tilde{\mathbf{H}}$ is called the *information density matrix*.

When $\mathbf{A}\,\tilde{\mathbf{H}} \;=\; \mathbf{S}_{inf} \;\approx\; \mathbf{I}_{nobs}$ then $\vec{d}^{calc} \;\approx\; \vec{d}^{obs}$, and a good match of the data vector $\vec{d}^{obs}$ is obtained. From here we can see that the closer $\mathbf{S}_{inf}$ is to $\mathbf{I}_{nobs}$ the better.

When $\tilde{\mathbf{H}}\,\mathbf{A} \;=\; \mathbf{R} \;\approx\; \mathbf{I}_{npar}$ then we have a good resolution of the parameters. The resolution matrix $\mathbf{R}$ can be interpreted as a filter acting over the true $\vec{\alpha}$; that is, $\tilde{\vec{\alpha}}$ is not the true $\vec{\alpha}$ but a linear combination of the elements of $\vec{\alpha}$. From here we see that

the closer $\mathbf{R}$ is to $\mathbf{I}_{npar}$ the better.

It is important to remark that the matrices $\mathbf{R}$ and $\mathbf{S}_{inf}$ are independent of the numerical values of the data vector $\vec{d}$.
The best approximation of $\mathbf{R}$ to $\mathbf{I}$ can be calculated by minimizing the 2–norm of each row of the matrix $\mathbf{\Delta} = \mathbf{R} - \mathbf{I}$ and in Reference [43] it was proved that this can be accomplished by chosing $\tilde{\mathbf{H}} = \mathbf{A}^{-g}$, that is:

$$\mathbf{A} = \mathbf{U}_p \, \mathbf{S}_p \, \mathbf{V}_p^T \tag{2.78}$$

$$\tilde{\mathbf{H}} = \mathbf{A}^{-g} = \mathbf{V}_p \, \mathbf{S}_p^{-1} \, \mathbf{U}_p^T \tag{2.79}$$

When the inversion is computed with $\mathbf{A}^{-g}$ the resolution and information density matrices are calculated from the singular value decomposition of the model represented by $\mathbf{A}$ as follows:

$$\mathbf{R} = \mathbf{V}_p \, \mathbf{V}_p^T \tag{2.80}$$

$$\mathbf{S} = \mathbf{U}_p \, \mathbf{U}_p^T \tag{2.81}$$

Then $\mathbf{R} = \mathbf{I}_{npar}$ only when $\mathbf{V}_0 = \mathbf{0}$, and $\mathbf{S}_{inf} = \mathbf{I}_{nobs}$ only when $\mathbf{U}_0 = \mathbf{0}$. Also it can be seen that $\mathbf{R}$ will be able to resolve $\vec{\alpha}$ when $\mathbf{R}$ is complete since $\mathbf{V}$ spans the vector space of $\vec{\alpha}$. The same reasoning applies to $\mathbf{S}_{inf}$. The data will be matched exactly when $\mathbf{S}_{inf}$ is complete because $\mathbf{U}$ spans the vector space of $\vec{d}$.

### Statistics

The expected value and covariance matrix of a random variable $\vec{y} = \vec{y}(\vec{x})\,;\, \vec{y} \in R^m\,;\, \vec{x} \in R^n$ are defined as:

$$\text{Expected Value of } \vec{y} = \vec{E}\{\vec{y}\} = \int_{-\infty}^{+\infty} \vec{y}(\vec{x}) \, p(\vec{x}) \, dx_1 \, \ldots \, dx_n \tag{2.82}$$

$$\text{Covariance of } \vec{y} \; = \mathbf{C}\{\vec{y}\} \; = \; \mathbf{E}\left\{\left(\vec{y} - \vec{E}\right)\left(\vec{y} - \vec{E}\right)^T\right\} \tag{2.83}$$

$p(\vec{x})$ is the probability density function associated with the random variable $\vec{y}$. $\vec{E}\{\vec{y}\} \in R^m$ and $\mathbf{C}\{\vec{y}\}$ is a ($m$ x $m$) matrix. Also $\mathbf{C} = \mathbf{C}^T$ and $\mathbf{C}$ is positive definite.

The expected value and covariance of a linear transformation of the form:

$$\vec{y} = \mathbf{A}\,\vec{x} \tag{2.84}$$

are calculated, assuming constant coefficients in $\mathbf{A}$, as follows:

$$\vec{E}\{\vec{y}\} \; = \; \mathbf{A}\,\vec{E}\{\vec{x}\} \tag{2.85}$$

$$\mathbf{C}\{\vec{y}\} \; = \mathbf{A}\,\mathbf{C}\{\vec{x}\}\,\mathbf{A}^T \tag{2.86}$$

from Equation 2.86 we see that given a set of $\vec{x}$ and their corresponding covariance $\mathbf{C}\{\vec{x}\}$ it is always possible to find a linear transformation that results in a diagonal covariance. By eigenvalue decomposition of $\mathbf{C}\{\vec{x}\}$ we get:

$$\mathbf{C}\{\vec{x}\} \; = \; \mathbf{V}\,\mathbf{\Lambda}\,\mathbf{V}^T \tag{2.87}$$

Then the following linear transformation will diagonalize the covariance matrix:

$$\vec{y} = \mathbf{V}^T\,\vec{x} \tag{2.88}$$

which results in the following covariance:

$$\mathbf{C}\{\vec{y}\} \; = \; \mathbf{\Lambda} \tag{2.89}$$

Since the variance must be positive, then all $\Lambda_{i,i} \; > \; 0$ which implies that the covariance matrix always has to be *positive definite*. Moreover, $\mathbf{V}^T$ is an orthonormal transformation (rotation) that preserves sizes and angles.
A more interesting transformation would be:

$$\vec{y} = \mathbf{\Lambda}^{-\frac{1}{2}}\,\mathbf{V}^T\,\vec{x} \tag{2.90}$$

and results in a unit covariance matrix, that is,

$$\mathbf{C}\{\vec{y}\} \;=\; \mathbf{I} \tag{2.91}$$

This approach can be used to normalize data from a statistical point of view: the data become independent and have the same variance.

Now we can apply the previous discussion to the case of the linear model as presented in Equations 2.71, 2.72, and 2.77, then we can relate the covariance of the data $\vec{d}^{obs}$ with the parameter estimates $\tilde{\vec{\alpha}}$.
In general we can assume that the data, that is the measurements, are independent of each other and consequently the covariance matrix will be diagonal. If this were not the case then it is always possible to perform a linear transformation of the data in such a way that the new covariance matrix becomes diagonal, as was shown in Equation 2.87. Thus assuming a diagonal matrix as covariance of the measurements we can compute the diagonal elements of the covariance matrix of the parameter estimates. That is from Equation 2.86:

$$\sigma^2_{\alpha_i} = \left(\mathbf{C}\{\vec{\alpha}\}\right)_{i,i} = \left(\tilde{\mathbf{H}}\,\mathbf{C}\{\vec{d}\}\,\tilde{\mathbf{H}}^T\right)_{i,i} = \sum_{j=1}^{nobs} \tilde{H}^2_{i,j}\,\sigma^2_{d_j} \tag{2.92}$$

If we chose $\tilde{\mathbf{H}} = \mathbf{A}^{-g}$ from the singular value decomposition of $\mathbf{A}$ then we obtain:

$$\begin{aligned}
\mathbf{A} &= \mathbf{U}_p\,\mathbf{S}_p\,\mathbf{V}_p^T && (2.93)\\
\tilde{\mathbf{H}} &= \mathbf{A}^{-g} \;=\; \mathbf{V}_p\,\mathbf{S}_p^{-1}\,\mathbf{U}_p^T && (2.94)\\
\tilde{H}_{i,j} &= \sum_{k=1}^{p} V_{p_{i,k}}\,S^{-1}_{p_k}\,U_{p_{j,k}} && (2.95)\\
\sigma^2_{\alpha_i} &= \sum_{j=1}^{nobs}\left(\sigma^2_{d_j}\left(\sum_{k=1}^{p} V_{p_{i,k}}\,S^{-1}_{p_k}\,U_{p_{j,k}}\right)^2\right) && (2.96)
\end{aligned}$$

When $\mathbf{A}$ is square, symmetric and positive definite then $\mathbf{U}_p = \mathbf{V}_p = \mathbf{V}$ and the eigenvalues of $\mathbf{A}$ take the place of the singular values, $S_i \rightarrow \lambda_i$ and $p = nobs = npar \rightarrow n$ then Equation 2.96 becomes:

$$\sigma^2_{\alpha_i} = \sum_{j=1}^{n} \left( \sigma^2_{d_j} \left( \sum_{k=1}^{n} V_{i,k}\, \lambda_k^{-1}\, V_{j,k} \right)^2 \right) \tag{2.97}$$

Since the values of $\sigma^2_{d_j}$ are bounded and $|U_{i,j}| \leq 1$ ; $|V_{i,j}| \leq 1$ then from Equations 2.96 and 2.97 we conclude that $\sigma_{\alpha_i}$ will be a function of the inverse of the singular values of the matrix model $\mathbf{A}$, that is $S^{-1}_{p_k}$ and this provides a way to control the value of the variance of the parameter estimates by choosing $p$ such that $S_{p_k} \geq \epsilon$ , $\forall\, k$, and the $\epsilon$ is chosen to guarantee that $\sigma_{\alpha_i} \leq \sigma^{limit}$. On the other hand, by making $p$ small we deteriorate both the resolution matrix $\mathbf{R} = \mathbf{V}_p\, \mathbf{V}_p^T$ and the density information matrix $\mathbf{S}_{inf} = \mathbf{U}_p\, \mathbf{U}_p^T$ . Thus there is a trade–off between the variance of the parameter estimates and their resolution. Interestingly the above theory shows, for the linear case, that it is possible to calculate the variance and resolution of parameter estimates without having the actual measurements (data), this can be used for planning an optimal gathering of data.

### 2.6.2 Development for Nonlinear Case

Now we can extend the previous theory to the nonlinear case. For the nonlinear case we can approximate the behavior of the system with a first order approximation, that is by linearization.

$$\frac{\partial \vec{d}}{\partial \vec{\alpha}} = \mathbf{G} \tag{2.98}$$

The linearization is obtained by assuming that $\mathbf{G}$ is constant, then the linearized model becomes:

$$\Delta\vec{d} = \mathbf{G}\,\Delta\vec{\alpha} \tag{2.99}$$

$$\vec{d} = \vec{d}_0 - \mathbf{G}\,\vec{\alpha}_0 + \mathbf{G}\,\vec{\alpha} \tag{2.100}$$

$$\vec{d} = \mathbf{G}\,\vec{\alpha} + Constant \tag{2.101}$$

Equation 2.101 is an approximation that works for $\vec{\alpha}$ close to $\vec{\alpha}_0$. By accepting this approximation, we can make an estimate of the covariance and resolution of the parameter estimates by using the theory developed for the linear case.

The first step is to apply an algorithm to minimize the objective function $E$ and thus obtain a parameter estimate, let's say $\vec{\alpha_0}$, then we can compute $\mathbf{G}$ and finally compute the resolution and covariance of the parameter estimates by using Equations 2.80, 2.81, and 2.96. This is summarized as follows:

1. Find $\vec{\alpha}^*$ as:
$$E(\vec{\alpha}^*) = \min_{\vec{\alpha}} E(\vec{\alpha}) \tag{2.102}$$

2. Compute the sensitivity matrix $\mathbf{G}$ at $\vec{\alpha} = \vec{\alpha}^*$ as:
$$\mathbf{G} = \left.\frac{\partial\vec{d}}{\partial\vec{\alpha}}\right|_{\vec{\alpha}^*} \tag{2.103}$$

3. Find the spectral decomposition of $\mathbf{G}$ as:
$$\mathbf{G} = \mathbf{U}_p\,\mathbf{S}_p\,\mathbf{V}_p^T \tag{2.104}$$

4. Compute the resolution matrix $\mathbf{R}$ as:
$$\mathbf{R} = \mathbf{V}_p\,\mathbf{V}_p^T \tag{2.105}$$

5. Compute the information density matrix $\mathbf{S}_{inf}$ as:
$$\mathbf{S}_{inf} = \mathbf{U}_p\,\mathbf{U}_p^T \tag{2.106}$$

6. Compute the variance of the parameter estimates as:

$$\sigma^2_{\alpha_i} = \sum_{j=1}^{nobs} \left( \sigma^2_{d_j} \left( \sum_{k=1}^{p} V_{p_{i,k}} \, S^{-1}_{p_k} \, U_{p_{j,k}} \right)^2 \right) \tag{2.107}$$

From the previous procedure we conclude that given that the problem is nonlinear, then the matrix $\mathbf{G}$ is not constant, consequently we cannot make an *a priori* estimate of the variance and resolution without knowing $\vec{\alpha}^*$. However if we know how the model looks, that is if we have an *a priori* estimate of $\vec{\alpha}^*$, then we can say something about how bad or good the resolution and variance would be. Then we can use those estimates of resolution and variance to plan how to collect the data. Obviously this would be the right plan only when the calculated parameters are not too far from the *a priori* estimates.

### 2.6.3 Alternate Method to Compute Resolution

When the number of observations is very large the computation of $\mathbf{U}_p$ may become a problem from the numerical point of view. This can be inferred by looking at the dimensions of the matrices we are dealing with. Table 2.6.3 summarizes the dimensions of these matrices.

Table 2.1: Matrix dimensions

| **matrix** | **dimension** |
|---|---|
| $\mathbf{G}$ | $nobs$ x $npar$ |
| $\mathbf{U}_p$ | $nobs$ x $p$ |
| $\mathbf{V}_p$ | $npar$ x $p$ |
| $p \leq \mathbf{Min}\{npar, nobs\}$ | |

When $nobs \gg npar$, as will be the case later in this work, the computation of $\mathbf{U}_p$ may become a problem, which will make it difficult to compute the resolution and variance of the parameter estimates. We now show how to compute the resolution matrix and covariances without forming $\mathbf{U}_p$, that is we know that

$$\mathbf{G} \;=\; \mathbf{U}_p\,\mathbf{S}_p\,\mathbf{V}_p^T \tag{2.108}$$

then,

$$\mathbf{H} \;=\; \mathbf{G}^T\,\mathbf{G} \;=\; \mathbf{V}_p\,\mathbf{S}_p^2\,\mathbf{V}_p^T \tag{2.109}$$

That is, we form the matrix $\mathbf{H}$ which is ($npar$ x $npar$) and then by eigenvalue decomposition we find $\mathbf{V}_p$ and $\mathbf{S}_p^2$. In this process we may lose the smallest singular values because of the finite precision in the calculations, but this is not a major problem since we would probably have to discard the smallest singular values because of the variance consideration.

Eventually, we may try to compute $\mathbf{U}_p$ as:

$$\mathbf{U}_p \;=\; \mathbf{G}\,\mathbf{V}_p\,\mathbf{S}_p^{-1} \tag{2.110}$$

In most cases we will not make such a computation, instead we leave $\mathbf{U}_p$ in the form of Equation 2.110.

We compute the resolution matrix as:

$$\mathbf{R} \;=\; \mathbf{V}_p\,\mathbf{V}_p^T \tag{2.111}$$

The covariance matrix of the parameter estimate can be calculated as:

$$\mathbf{C}\{\vec{\alpha}\} \;=\; \mathbf{G}^{-g}\,\mathbf{C}\{\vec{d}\}\,\mathbf{G}^{-g^T} \;=\; \mathbf{V}_p\,\mathbf{S}_p^{-1}\,\mathbf{U}_p^T\,\mathbf{C}\{\vec{d}\}\,\mathbf{U}_p\,\mathbf{S}_p\,\mathbf{V}_p^T \tag{2.112}$$

by replacing $\mathbf{U}_p$ as defined in Equation 2.110 and taking in consideration $\mathbf{H}$ as defined in Equation 2.109, we obtain:

$$\begin{aligned}
\mathbf{C}\{\vec{\alpha}\} &= \mathbf{V}_p\,\mathbf{S}_p^{-1}\,\mathbf{S}_p^{-1}\,\mathbf{V}_p^T\,\mathbf{G}^T\,\mathbf{C}\{\vec{d}\}\,\mathbf{G}\,\mathbf{V}_p\,\mathbf{S}_p^{-1}\,\mathbf{S}_p^{-1}\,\mathbf{V}_p^T && (2.113)\\
&= \mathbf{V}_p\,\mathbf{S}_p^{-2}\,\mathbf{V}_p^T\,\mathbf{G}^T\,\mathbf{C}\{\vec{d}\}\,\mathbf{G}\,\mathbf{V}_p\,\mathbf{S}_p^{-2}\,\mathbf{V}_p^T && (2.114)\\
&= \mathbf{H}^{-g}\,\bar{\mathbf{C}}\,\mathbf{H}^{-g} && (2.115)
\end{aligned}$$

Equation 2.115 allows us to compute the covariance without using $\mathbf{U}_p$ (but we need to form $\bar{\mathbf{C}}$ and $\mathbf{H}^{-g}$). The dimensions of the matrices $\mathbf{H}^{-g}$ and $\bar{\mathbf{C}}$ are (*npar* x *npar*). They are defined as follows:

$$\bar{\mathbf{C}} = \mathbf{G}^T \mathbf{C}\{\vec{d}\} \mathbf{G} \tag{2.116}$$

$$\mathbf{H}^{-g} = \mathbf{V}_p \mathbf{S}_p^{-2} \mathbf{V}_p^T \tag{2.117}$$

from which we compute the diagonal terms of $\mathbf{C}\{\vec{\alpha}\}$ as:

$$\sigma^2_{\alpha_i} = \sum_{k=1}^{npar} \sum_{j=1}^{npar} H^{-g}_{i,j} \bar{C}_{j,k} H^{-g}_{k,i} \tag{2.118}$$

Equation 2.118 is equivalent to Equation 2.107 but without the need of computing $\mathbf{U}_p$.

If $\mathbf{C}\{\vec{d}\} = \sigma_d^2 \mathbf{I}$, then Equations 2.115 and 2.118 become:

$$\mathbf{C}\{\vec{\alpha}\} = \sigma_d^2 \mathbf{H}^{-g} \tag{2.119}$$

$$\sigma^2_{\alpha_i} = \sigma_d^2 H^{-g}_{i,i} = \sigma_d^2 \sum_{k=1}^{npar} \left( \frac{V_{p_{i,k}}}{S_{p_k}} \right)^2 \tag{2.120}$$

It is important to note that so far we have a method to estimate covariances and resolution without having an explicit probabilistic model.

## 2.7 Generation of Realizations of the Reservoir

Due to the nonlinear characteristics of the reservoir inverse problem there is not a unique solution, that is we can consider that the calculated set of parameters (reservoir permeability and porosity) is only one *realization* of a large set of parameters that also match the data within certain tolerance. The singular value decomposition of $\mathbf{H}$ and the covariance matrix analysis performed in the previous section can be used to generate other *realizations* of the reservoir. A common approach developed in the field of geostatistics is to generate such realizations by using the following formulas:

$$\vec{m} = \vec{m}_{\infty} + \mathbf{Z}\,\vec{l} \tag{2.121}$$

where $\vec{m}$ is the *new* vector of the set of parameters, that is a *new realization*; $\vec{m}_{\infty}$ is the vector of the set of parameters that provides a good match of the data, that is the calculated parameters from the parameter estimation algorithm; $\mathbf{Z}$ is the *Cholesky factor* of the calculated covariance matrix of the parameter estimates, that is $\mathbf{C}\{\vec{\alpha}\} = \mathbf{Z}\,\mathbf{Z}^T$; $\vec{l}$ is a vector whose each element is obtained as a draw from a *normal* probability distribution. The validity of Equation 2.121 can be verified by computing the covariance matrix $\mathbf{C}\{\vec{m}\}$, thus by using Equation 2.86 we obtain:

$$\mathbf{C}\{\vec{m}\} = \mathbf{Z}\,\mathbf{C}\{\vec{l}\}\,\mathbf{Z}^T = \mathbf{C}\{\vec{\alpha}\} \tag{2.122}$$

this arises from the fact that $\mathbf{C}\{\vec{l}\} = \mathbf{I}$ from the assumption of normal distribution in $\vec{l}$.

Thus it can be seen that Equation 2.121 allows us to generate a set of *realizations* that would have in common the same covariance matrix calculated by the parameter estimation algorithm. The observation at this point is that with this approach it is not assured that the each individual realization generated with this method will provide a good match of the data unless they are screened by running the mathematical model (reservoir numerical simulator), here we present a way to perform such a screening without using the mathematical model. This can be accomplished by using the information from the singular value decomposition of $\mathbf{H}$. In Section 2.5.2 it was shown that the objective function $E$ can be approximated as:

$$E\left(\vec{\alpha}_0 + \Delta\vec{\alpha}\right) \approx E\left(\vec{\alpha}_0\right) + \nabla E^T \Delta\vec{\alpha} + \frac{1}{2}\,\Delta\vec{\alpha}^T \mathbf{H}\,\Delta\vec{\alpha} \tag{2.123}$$

Then if we assume that $\alpha_0$ is the result of the calculation from the parameter estimation algorithm, that is $\alpha_0$ provides not only a good match of the data but also a gradient that is either zero or very small then we can calculate the increase in the discrepancy of the match (objective function) as:

$$\Delta E = \frac{1}{2}\,\Delta\vec{\alpha}^T \mathbf{H}\,\Delta\vec{\alpha} \tag{2.124}$$

because $\mathbf{H}$ is a semipositive definite matrix $\Delta E \geq 0$. We can write Equation 2.124 taking into consideration the singular value decomposition of $\mathbf{H}$ as:

$$\Delta E = \frac{1}{2} \Delta\vec{\alpha}^T \mathbf{V}\, \mathbf{S}^2\, \mathbf{V}^T\, \Delta\vec{\alpha} \tag{2.125}$$

since $\mathbf{V}$ spans the complete parameter space, that is is the columns vectors of $\mathbf{V}$ can be used as a base, then we write $\Delta\vec{\alpha}$ as a linear combination of the columns of $\mathbf{V}$:

$$\Delta\vec{\alpha} = \mathbf{V}\, \vec{a} \tag{2.126}$$

then Equation 2.125 becomes:

$$\Delta E = \frac{1}{2} \vec{a}^T\, \mathbf{V}^T\, \mathbf{V}\, \mathbf{S}^2\, \mathbf{V}^T\, \mathbf{V}\, \vec{a} \tag{2.127}$$

$$\Delta E = \frac{1}{2} \sum_{i=1}^{npar} a_i^2\, s_i^2 \tag{2.128}$$

From Equation 2.128 we see that we can choose the $\vec{a}$ arbitrarily but under the constraint that $\frac{1}{2} \sum_{i=1}^{npar} a_i^2\, s_i^2 \leq \epsilon^2$ where $\epsilon$ is the maximum increase in the $E$ we are willing to accept, we see that we can allow large numbers in the components associated to the smallest $s_i$. In the limiting case of having singular values equal to zero then any change in the null space of $\mathbf{H}$ will provide a good match of the data. Obviously this analysis is strictly valid only for linear problems which is not the case here, so the realizations computed with this procedure are only a first approximation. We can use this idea to prescreen the realizations generated with Equation 2.121.

## 2.8 Parameterization

The mathematical model used here to represent the physical system, that is the reservoir, was a numerical reservoir simulator. The parameters in the numerical scheme are, among others, permeability and porosity at each cell of the discretization grid. Hence, with the method used in this work, the spatial distribution of permeability and porosity in the reservoir cannot be estimated at a smaller scale than the simulation

grid. The variables computed by the numerical simulator are the pressure and saturation at each cell of the discretization grid. These variables have to be transformed into pressures and water cut values at the wells and in changes of spatial distribution of saturation because these are the input variables for the parameter estimation procedure.

This work explored two approaches to define the parameters for inversion. Both have the same objective, to compute spatial distributions of permeability and porosity in the reservoir. In the simplest approach, the parameters for inversion are simply the permeability and porosity at each cell of the simulation grid. This is referred to as *pixel modeling* in which the pixel is the smallest unit that can be described. A second approach is to consider that complex distributions of permeability and porosity can be described by a geometric or geostatistical model with a relatively small number of parameters. This is referred to as *object modeling*.

Although the interest of this work is in the computation of permeability and porosity it is necessary to add other parameters into the inversion procedure. That is the case of the skin factor in the wells. The skin factor needs to be included as an unknown parameter because the pressures at the wells are highly dependent on the skin at the wellbore. If the skin is disregarded as a parameter then it is possible to obtain a highly distorted permeability field in the surroundings of the wells.

The pixel and object modeling approaches are explained next.

### 2.8.1 Pixel Modeling

Figure 2.2 is an example of a reservoir that has been parameterized with the pixel modeling approach. Pixel modeling may seem an attractive approach since, from the theoretical point of view, it would permit the computation of distributions of permeability and porosity at the smallest level of the simulation grid. The problem with this approach is that it requires the computation of a large number of parameters, that is two times the number of simulation cells, which in general is in the order of

thousands. From the point of view of the resolution analysis presented earlier, we may anticipate a reasonable *information density matrix*, that is the data may be matched quite well, but also we expect a poorly conditioned sensitivity matrix $\mathbf{G}$ which results in very small *singular values* which then translate to large and unacceptable variances of the parameter estimates. The inverse problem is in general highly undetermined because the data are insufficient to resolve the parameters. As a result, the application of this approach may result in the computation of distributions of permeability and porosity that may match the dynamic data but that cannot be accepted from the geological point of view. Although the inversion can be stabilized by introducing *a priori* information in the definition of the objective function, as was presented in Equation 2.6, the procedure still requires a lot of computing power. Also, this approach cannot preserve the large scale information.

When the pixel modeling approach is used along with gradient based parameter estimation algorithms it is necessary to compute the sensitivity coefficients with respect to the permeability and porosity at each simulation block and with respect to the skin factor at each well. The sensitivity coefficients are:

$$\vec{s}_{k_i} = \frac{\partial \vec{d}}{\partial k_i} \tag{2.129}$$

$$\vec{s}_{\phi_i} = \frac{\partial \vec{d}}{\partial \phi_i} \tag{2.130}$$

$$\vec{s}_{s_j} = \frac{\partial \vec{d}}{\partial s_j} \tag{2.131}$$

$s_{k_i}$ is the sensitivity with respect to the permeability in the simulation block $i$. $s_{\phi_i}$ is the sensitivity with respect to the porosity in block $i$. $s_{s_j}$ is the sensitivity with respect to the skin in well $j$.

The method to compute these sensitivity coefficients is explained in detail in Chapter 3.

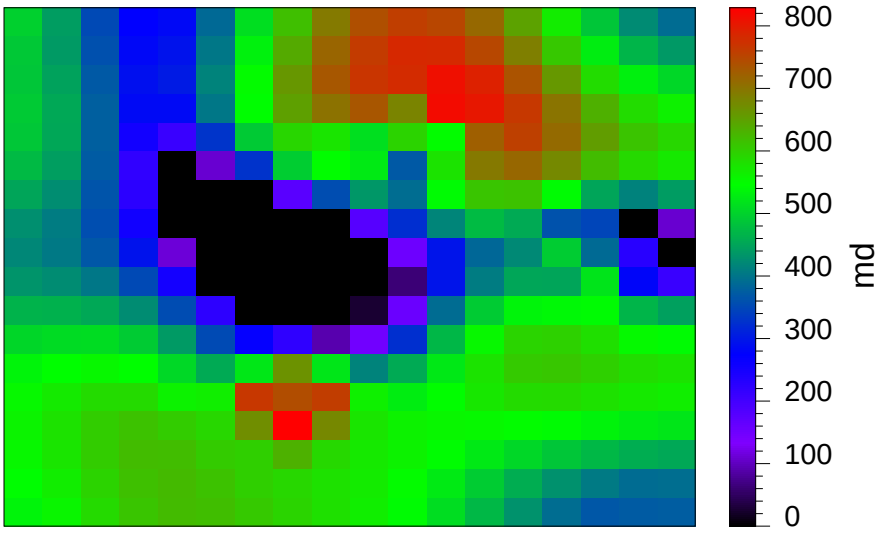



Figure 2.2: Pixel model.

### 2.8.2 Object Modeling

This approach was developed as a way to reduce the number of parameters and also to preserve the large scale information. Figure 2.3 shows a very simple model of a reservoir. In this case the reservoir can be characterized with a relatively simple geometric object that represents a channel. In the case of *pixel* modeling the parameters were permeability and porosity, in the case of *object* modeling the parameters are of a different nature. The parameters for inversion are a mix that define the geometry of the permeability and porosity distributions. For example in the case of a channel the object can be defined by the coordinates of its boundaries. Thus the complete permeability field can be described as follows:

$$y_1 = \alpha_1 \sin(\alpha_2 x + \alpha_3) + \alpha_4 \tag{2.132}$$

$$y_2 = \alpha_1 \sin(\alpha_2 x + \alpha_3) + \alpha_5 \tag{2.133}$$

$$k(x,y) = \begin{cases} \alpha_6 & \text{if } y_2 < y < y_1 \\ \alpha_7 & \text{otherwise} \end{cases} \tag{2.134}$$

Each of the parameters has the following physical meaning:

- $\alpha_1$ is the "amplitude" of the channel.
- $\alpha_2$ is the "sinuosity" of the channel.
- $\alpha_3$ is the displacement of the channel in the $x$ direction.
- $\alpha_4 - \alpha_5$ is the "width" of the channel.
- $\alpha_4 + \alpha_5$ is the displacement of the center of the channel in the $y$ direction.
- $\alpha_6$ is the permeability *inside* the channel.
- $\alpha_7$ is the permeability *outside* the channel.

The description is completed by adding geometrical and physical constraints, for example:

$$y_2 < y_1 \Longrightarrow \alpha_5 < \alpha_4 \tag{2.135}$$

$$\alpha_6 > 0 \tag{2.136}$$

$$\alpha_7 > 0 \tag{2.137}$$

Any change in the geometric parameters result in changes of shape and displacement of the object "channel". This is the main concept in the use of objects, to control a large number of simulation cells with few parameters. For example in the case of the channel, if it were modeled with the *pixel* approach it would have required thousands of parameters (one parameter for each simulation cell), but with the *object* approach the complete problem is characterized with only seven parameters. The problem not only becomes smaller but also better determined. There is also a more important advantage that is the preservation of the large scale geological information. None of the pixel based approaches can guarantee this point. If the geological object is a channel and it is modeled with the *object* approach then the outcome of the parameter inversion algorithm will *always* be a channel. This could not be guaranteed with the pixel based approach, and actually would be unlikely.

The objects for reservoir parameterization can be classified as dynamic or static.

- *Static.* When a parameter that defines the object does not have an effect in the geometry it is referred to as a *static object.* In this case the functions that link the parameters to the permeability and porosity in the simulation grid are very simple. The permeability of a channel is an example of a static object:

$$k_i = \sum_{j=1}^{npar} \lambda_j^i \, \alpha_i \tag{2.138}$$

- *Dynamic.* When the parameters that define the object are such that they have an effect on the geometry of the object, then the object it is referred to as a *dynamic object.* This means that the object is allowed to "float" in the reservoir, that is a change in the parameters result in change of shape, displacement or rotation of the object. Examples of dynamic objects are the channel and the ellipse depicted in Figures 2.3 and 5.9 respectively.

To apply the gradient based methods for parameter estimation requires the computation of the sensitivity coefficients with respect to the parameters that define the object. For example in the case of the channel it will be necessary to compute the sensitivity of pressure, water cut and distribution of saturation with respect to the "sinuosity" of the channel. A key step to the computation is in the process of linking the continuous characteristics of the boundary of the object to the discrete environment of the simulation grid. The method to compute such a sinuosity effect is described in detail in Chapter 3.

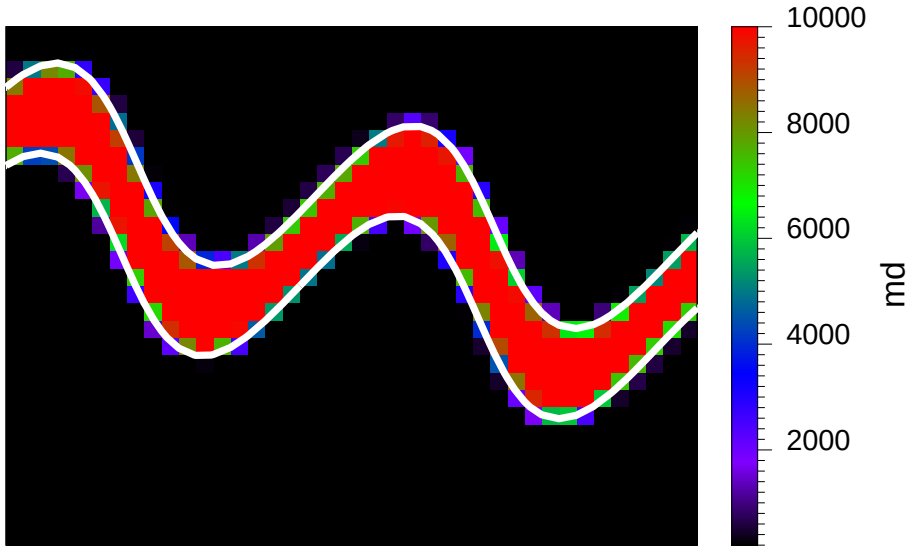



Figure 2.3: Object model.

## 2.9 Constraints

In reservoir parameter estimation the parameters for inversion are not completely arbitrary, for example permeability and porosity cannot be negative, also they cannot take very high values. In general the parameters for inversion are constrained to a certain range, which is referred to as the feasibility region. It would be then be reasonable to use algorithms that are designed to work with constraints. References [8] and [45] provide a good description of these algorithms. The algorithms for "true" constrained optimization are complicated and difficult to implement in an efficient way. One alternative that simplifies the solution is not to solve the actual constrained problem but another "equivalent" problem that can be solved with algorithms for unconstrained optimization. The algorithms for unconstrained optimization are much simpler and more robust than the ones for true constrained optimization. In this work the constrained problem is solved by using the *penalty function* method (sometimes also referred to as the *barrier function* method). The method is explained next.

In general the constraints on the parameters are expressed in the following form:

$$\vec{c}(\vec{\alpha}) \leq \vec{0} \tag{2.139}$$

where $\vec{c} \in R^{ncons}$ and $ncons$ is the number of constraints.

Equation 2.139 represents several simultaneous constraints that apply to the problem. Each individual constraint can be *linear* or *nonlinear* with respect to $\vec{\alpha}$. The constraints can also involve equality and inequality.

A common set of constraints consists of lower and upper bounds on the parameters. When the parameters are permeability and porosity a trivial lower bound is zero, but in general it is possible to to set up more reasonable bounds. For example when permeability is the parameter for the inverse problem then it is possible to set up the following lower and upper bounds:

$$k^{min} \; < \; \alpha_i \; < \; k^{max} \; ; \forall \, i \tag{2.140}$$

where $\alpha_i = k_i$.

Another type of upper and lower constraint arises naturally when dealing with *object* modeling. In this case the parameters are indirectly related to the permeability and porosity field through functions of the type:

$$\begin{cases} k_i &= k_i(\vec{\alpha}) \\ \phi_i &= \phi_i(\vec{\alpha}) \end{cases} \tag{2.141}$$

and the lower and upper constraints would be:

$$\begin{cases} k^{min} &< k_i(\vec{\alpha}) &< k^{max} \\ \phi^{min} &< \phi_i(\vec{\alpha}) &< \phi^{max} \end{cases} \tag{2.142}$$

The functions $k_i(\vec{\alpha})$ and $\phi_i(\vec{\alpha})$ may be linear or nonlinear with respect to $\vec{\alpha}$.

The *penalty function* method works by adding a penalty term. The penalty function is defined to become very large when the parameters approach the boundary of the feasible region and to become negligible otherwise.

In the penalty function method the new objective function for unconstrained minimization, $\hat{E}$ is defined as:

$$\hat{E} = E + \sum_{i=1}^{ncons} \frac{\varepsilon_i}{c_i(\vec{\alpha})} \tag{2.143}$$

The term $\varepsilon_i$ is a weighting factor. This weighting factor is adjusted after each iteration of the parameter in such a way that in the limit:

$$\varepsilon_i \rightarrow 0 \text{ when } E \rightarrow 0 \tag{2.144}$$

The purpose of having a variable coefficient is to allow the minimization algorithm to converge to an optimum close to the true optimal point, even if the true optimal point is located close to the constraints.

One procedure to satisfy the condition stated in Equation 2.144 is by defining $\varepsilon_i$ as follows:

$$\varepsilon_i = \omega_i \, E_{previous} \tag{2.145}$$

where $\omega_i$ is a constant factor that has to be preset. $E_{previous}$ is the value of $E$ in the previous iteration of the minimization algorithm.
In summary the constrained problem has been converted to an unconstrained problem by adding the penalty function. In general the optimal point obtained with this approach would not coincide with the true one, but it is expected to be close. The convergence will be slow when the true optimal point is close to one of the constraints.

### 2.9.1 Implementation of Constraints in the Gauss–Newton Method

With the introduction of the constraints in the problem and the addition of the penalty function into the definition of the objective function it becomes necessary to recompute the gradient and Hessian. Thus the new set of basic equations become:

$$\hat{E} = E + \sum_{i=1}^{ncons} \frac{\varepsilon_i}{c_i(\vec{\alpha})} \tag{2.146}$$

$$\nabla\hat{E} = \nabla E - \sum_{j=1}^{ncons} \frac{\varepsilon_j}{c_j^2} \nabla c_j \tag{2.147}$$

$$\hat{\mathbf{H}} = \mathbf{H} + 2 \sum_{j=1}^{ncons} \frac{\varepsilon_j}{c_j^3} \nabla c_j \, \nabla c_j^T - \sum_{j=1}^{ncons} \frac{\varepsilon_j}{c_j^2} \frac{\partial \nabla c_j}{\partial \vec{\alpha}} \tag{2.148}$$

$$\hat{\mathbf{H}}_{GN} = \mathbf{H}_{GN} + 2 \sum_{j=1}^{ncons} \frac{\varepsilon_j}{c_j^3} \nabla c_j \, \nabla c_j^T \tag{2.149}$$

$\frac{\partial \nabla c_j}{\partial \vec{\alpha}}$ is the Hessian matrix of the $j^{th}$ constraint, $c_j(\vec{\alpha})$.
The new $\hat{\mathbf{H}}_{GN}$ is guaranteed to be positive definite, as is required, since $\nabla c_j \, \nabla c_j^T$ is semipositive definite and $c_j > 0$. In case of linear constraints $\frac{\partial \nabla c_j}{\partial \vec{\alpha}} = 0$.

## 2.10 Linear Search

Even though the Gauss–Newton algorithm computes not only a *sufficient descent direction* but also a step size, the computed step size may only be appropriate when

the algorithm is close to the minimum or when the second order approximation of $E$ is accurate. This condition is not the case most of the time, therefore the most adequate step size is better found by a process of linear search. Thus the linear search procedure consists of finding the most suitable step size in the given direction.

The linear search involves only function evaluations (computing the value of $E$). The design of the most efficient linear search has to take into consideration the difference between the cost of evaluating the direction for descent against the cost for the plain function evaluation. There are many algorithms for linear search in the literature. In this work the linear search was constructed following the algorithm described by Bard [46]. This algorithm works by using a quadratic approximation of the objective function in the descent direction as it is computed by the Gauss–Newton algorithm. The optimal step size is found by optimizing the quadratic form.

The quadratic approximation is of the form:

$$\tilde{E}(\rho) = a\,\rho^2 + b\,\rho + c \tag{2.150}$$

where $a$, $b$ and $c$ can be calculated as:

$$c = E(0) = \text{ value of } E \text{ from previous iteration} \tag{2.151}$$

$$b = \vec{p}^T\,\nabla E \tag{2.152}$$

$$a = \frac{E(\rho_{GN}) - b\,\rho_{GN} - c}{\rho_{GN}^2} \tag{2.153}$$

$\vec{p}$ and $\rho_{GN}$ are the direction and the step size as they are calculated by the Gauss–Newton algorithm.

The optimal step size $\rho^*$ is obtained from Equation 2.150 as:

$$\rho^* = \frac{-b}{2\,a} \tag{2.154}$$

The actual algorithm is more complex since the linear search has to take account of the constraints, that is to stay within the feasible region, and not to a maximum

for a minimum in the quadratic form.

The overall performance of the parameter estimation algorithm is greatly enhanced by a properly designed linear search when the cost of computing a new direction for descent is expensive compared to the plain function evaluation. This is likely to be the case in the reservoir problems encountered here.

## 2.11 Stabilization of the Hessian Matrix

In Section 2.5.2 it was shown that at each iteration of the Gauss–Newton algorithm it is necessary to solve a linear system of equations represented by the equation:

$$\mathbf{H}_{GN}\,\Delta\vec{\alpha} \;=\; -\nabla E \tag{2.155}$$

Also in Section 2.5.2 it was indicated that the matrix $\mathbf{H}_{GN}$ is often ill–conditioned or singular. When the matrix $\mathbf{H}_{GN}$ is singular it is not possible to solve Equation 2.155, and when it is ill–conditioned the solution vector $\Delta\vec{\alpha}$ is very large. It is important at this point to recall that our main interest is not in finding a value of $\Delta\vec{\alpha}$ but in finding a direction of descent, thus we can introduce changes to $\mathbf{H}_{GN}$ in such a way that it is possible to solve Equation 2.155 and still provide a direction of descent. Obviously we want to change $\mathbf{H}_{GN}$ only slightly since we do not want to lose the quadratic convergence of the method close to the optimum.

In this work the stabilization is performed in two stages. In the first stage the matrix $\mathbf{H}_{GN}$ is scaled in such a way that all the diagonal elements become unity, this is accomplished by defining a diagonal matrix $\mathbf{F}$ whose elements are the inverse of the square root of the diagonal elements of $\mathbf{H}_{GN}$, that is:

$$F_{i,i} = (H_{i,i})^{-\frac{1}{2}} \tag{2.156}$$

Then instead of Equation 2.155 we solve an equivalent system given by:

$$(\mathbf{F}\mathbf{H}_{GN}\mathbf{F})\,\mathbf{F}^{-1}\,\Delta\vec{\alpha} \;=\; -\mathbf{F}\,\nabla E \tag{2.157}$$

The scaled matrix $\hat{\mathbf{H}} = \mathbf{F}\mathbf{H}_{GN}\mathbf{F}$ has all its diagonal elements equal to unity and all the off–diagonal elements have absolute value less than unity because of the semipositive definitess in $\mathbf{H}_{GN}$. The scaled system is more stable than the original one.

The second stage in the stabilization process is applied when the scaled matrix $\hat{\mathbf{H}}$ is singular or ill–conditioned. The singularity is detected at the time of performing the Cholesky factorization. The ill condition is detected by checking that the calculated direction of descent satisfies the *sufficient descent* condition, that is:

$$\frac{\nabla E^T \, \Delta\vec{\alpha}}{\| \nabla E \| \, \| \Delta\vec{\alpha} \|} < -a \tag{2.158}$$

where $a$ is a predetermined positive number, that is $0 < a < 1$.

The second stage in the stabilization consist in applying the *Marquardt* method for stabilization.

### 2.11.1 Marquardt Method

The *Marquardt* method consists of a modification to $\hat{\mathbf{H}}$ as follows.

$$\hat{\mathbf{H}} = \hat{\mathbf{H}} + \lambda\mathbf{I} \tag{2.159}$$

where $\lambda$ is a small positive number. The value of $\lambda$ is chosen to satisfy Equation 2.158. The value of $\lambda$ is reduced in each successive iteration that results in a downhill step, or increased when the computed direction appears to be uphill.

### 2.11.2 Modified Cholesky Factorization Method

Another alternative to stabilize $\hat{\mathbf{H}}$ is the *modified Cholesky factorization* method. This method is described in Ref. [8], [45] and was also implemented here as an alternative method to the Marquardt method. The method introduces changes in the diagonal elements of $\hat{\mathbf{H}}$ at the time of performing the Cholesky factorization.

# Chapter 3

# Computation of Sensitivity Coefficients

## 3.1 General Case for Discrete Systems

In general, many physical systems can be modeled by a differential equation of the type:

$$F(y, \vec{x}, t, \vec{\alpha}, \frac{\partial^n y}{\partial \vec{x}^n}, \frac{\partial y}{\partial t}) = 0 \tag{3.1}$$

Where $y = y(\vec{x}, t)$ is the response of the system for a given set of initial and boundary conditions, $\vec{x}$ is the spatial coordinate vector, $\vec{x} \in R^3$, $t$ is time and $\vec{\alpha}$ is the vector of the parameters of the system, $\vec{\alpha} \in R^{npar}$. $npar$ is the finite number of discrete parameters, which are considered to be constant with respect to time, but may have spatial variation, that is:

$$\vec{\alpha} = \vec{\alpha}(\vec{x}) = \begin{bmatrix} \alpha_1(\vec{x}) \\ \vdots \\ \alpha_{npar}(\vec{x}) \end{bmatrix} \tag{3.2}$$

The behavior or response of the system can be found by solving Equation 3.1 with appropriate initial and boundary conditions. In some cases it is possible to compute the solution explicitly with an analytical expression. However most real problems are

complex and it is not possible to obtain an explicit solution, therefore Equation 3.1 is solved with numerical methods.

The *forward problem* consists of solving Equation 3.1 that is, finding $y = y(\vec{x}, t)$ given a set of $\vec{\alpha} = \vec{\alpha}(\vec{x})$.

The *inverse problem* consists of finding the set of parameters $\vec{\alpha} = \vec{\alpha}(\vec{x})$ such that the calculated $y = y(\vec{x}, t)$ from Equation 3.1 matches a set of discrete observations from the actual physical system. The inverse problem is also referred to in the literature as a *parameter estimation* problem.

The gradient of $y$ with respect to $\vec{\alpha}$, is referred to as the vector of *sensitivity coefficients*. Mathematically it is defined as:

$$\nabla_\alpha y = \left(\frac{\partial y}{\partial \vec{\alpha}}\right)^T = \begin{bmatrix} \frac{\partial y(\vec{x},t,\vec{\alpha})}{\partial \alpha_1} \\ \vdots \\ \frac{\partial y(\vec{x},t,\vec{\alpha})}{\partial \alpha_{npar}} \end{bmatrix} \tag{3.3}$$

From a physical point of view Equation 3.3 can be seen as a measure of how the response of the system $y$ varies when the parameter $\vec{\alpha}$ is changed. The first order approximation to $y$ by Taylor series is:

$$y(\vec{x}, t, \vec{\alpha} + \Delta\vec{\alpha}) \approx y(\vec{x}, t, \vec{\alpha}) + \frac{\partial y}{\partial \vec{\alpha}} \Delta\vec{\alpha} \tag{3.4}$$

Gradient based parameter estimation algorithms work by using these sensitivity coefficients as described in Chapter 2, therefore it is our interest to have an efficient procedure to compute Equation 3.3. When Equation 3.1 is solved explicitly with an analytical expression, the procedure to compute the gradients is straightforward. A more complex case arises when Equation 3.1 is solved with a numerical method. One approach to calculate the sensitivity coefficients is to compute a first order approximation of $\frac{\partial y}{\partial \vec{\alpha}}|_{\vec{\alpha}=\vec{\alpha}_0}$ with a finite difference procedure, this is referred as the *substitution method*, that is:

**Substitution Method**

1. Set $\vec{\alpha} = \vec{\alpha}_0$

2. Solve $y_0 = y(\vec{x}, t, \vec{\alpha_0})$

3. For $i = 1 : npar$

   (a) $\alpha_i = \alpha_{0_i} + \epsilon_i$ where $\epsilon_i$ are small numbers.

   (b) Solve $y = y(\vec{x}, t, \vec{\alpha})$.

   (c) $\frac{\partial y}{\partial \alpha_i}|_{\vec{\alpha}=\vec{\alpha_0}} \approx \frac{y-y_0}{\epsilon_i}$

4. End

This method is very simple and easy to implement, the obvious drawback is that it is very expensive from the computational point of view when the number of parameters *npar* is large. More efficient methods can be derived by looking at the internal structure of the numerical method used to solve Equation 3.1.

In general the numerical methods used to solve Equation 3.1 solve a discretized version of the problem, that is we solve for the values of $y = y(\vec{x}, t, \vec{\alpha})$ at $\vec{x} = \vec{x}_j$ ($j = 1 : m,\ m =$ nodes of the discretization grid) and $t = t^k$ (discretization in time). The solution is obtained by solving at each time step a system of nonlinear algebraic equations of the type:

$$\vec{f}(y^{k+1}, y^k, \vec{x}_j, \Delta t, \vec{\alpha}) = 0 = \begin{bmatrix} f_1(y^{k+1}, y^k, \vec{x}_j, \Delta t, \vec{\alpha}) \\ \vdots \\ f_m(y^{k+1}, y^k, \vec{x}_j, \Delta t, \vec{\alpha}) \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \tag{3.5}$$

The initial and boundary conditions of the problem are included in the construction of the system of equations 3.5.

$k$ is the index for the time step in the numerical scheme for solving Equation 3.1
$t^k$ is the time corresponding to the time step $k$
$\Delta t = t^{k+1} - t^k$
$\vec{x}_j \quad j = 1 : m;\ \ m =$ number of nodes in the spatial grid
$y^k$ value of $y$ at time $= t$; $y \ \in \ R^m$

In Equation 3.5 it is assumed that the values of $y^k$ are known, and the unknown is $y^{k+1}$ which needs to be solved from the nonlinear system of equations. The solution

of Equations 3.5 is frequently obtained by an iterative process known as the *Newton–Raphson* method. At each iteration of this method a linear system of equations is solved as:

$$\mathbf{J}^{(i)}_{y^{k+1}} \Delta^{(i+1)} \vec{y}^{k+1} = -\vec{f}^{(i)} \tag{3.6}$$

where $\mathbf{J}_{y^{k+1}}$ is the *Jacobian* matrix of the system of equations 3.5, and $(i)$ is the iteration index. The iterative procedure continues until a preset convergence criterion is met. The *Jacobian* matrix is defined as:

$$\mathbf{J}_{y^{k+1}} \equiv \frac{\partial \vec{f}}{\partial \vec{y}^{k+1}} \tag{3.7}$$

We are looking to compute the change in the response of the system when one of the parameters is changed, let's say $\alpha_i$.

$$\tilde{\alpha}_i = \alpha_i + \delta\alpha_i \tag{3.8}$$

$$\tilde{\vec{\alpha}} = \vec{\alpha} + \delta\vec{\alpha}_i \tag{3.9}$$

and the response of the system will become,

$$\tilde{y} = y + \delta y_i \tag{3.10}$$

and since we are solving a discretized version of the problem,

$$\begin{cases} \tilde{\vec{y}}^{k} &= \vec{y}^{k} + \delta\vec{y}^{k}_i \\ \tilde{\vec{y}}^{k+1} &= \vec{y}^{k+1} + \delta\vec{y}^{k+1}_i \end{cases} \tag{3.11}$$

we can write,

$$\begin{cases} \delta\vec{y}^{k}_i &= \frac{\partial\vec{y}^{k}}{\partial\alpha_i}\delta\alpha_i = \vec{S}^{k}_i\,\delta\alpha_i \\ \delta\vec{y}^{k+1}_i &= \frac{\partial\vec{y}^{k+1}}{\partial\alpha_i}\delta\alpha_i = \vec{S}^{k+1}_i\,\delta\alpha_i \end{cases} \tag{3.12}$$

where,

$$\vec{S}_i^k \equiv \frac{\partial \vec{y}^k}{\partial \alpha_i} \; ; \quad \vec{S}_i^k \; \epsilon \; R^m \tag{3.13}$$

If we had to solve for $\tilde{y}$ we would have to solve a system of equations similar to Equation 3.5; that is:

$$\vec{f}(\tilde{\vec{y}}^{k+1}, \tilde{\vec{y}}^k, \vec{x}_j, \Delta t, \tilde{\vec{\alpha}}) = 0 \tag{3.14}$$

Our interest is not in solving Equations 3.14 but in finding the sensitivity coefficients $S_i$. We expand Equations 3.14 in a Taylor series around the unperturbed solution at $(y^{k+1}, y^k, \vec{x}_j, \Delta t, \vec{\alpha})$, then:

$$\begin{aligned} \vec{f}(\tilde{\vec{y}}^{k+1}, \tilde{\vec{y}}^k, \vec{x}_j, \Delta t, \tilde{\vec{\alpha}}) = \vec{f}(y^{k+1}, y^k, \vec{x}_j, \Delta t, \vec{\alpha}) + \\ \mathbf{J}_{y^{k+1}} \, \delta \vec{y}^{k+1} + \mathbf{J}_{y^k} \, \delta \vec{y}^k + \frac{\partial \vec{f}}{\partial \alpha_i} \, \delta \alpha_i + 0(\delta^2) = \vec{0} \end{aligned} \tag{3.15}$$

where,

$$\begin{aligned} \mathbf{J}_{y^{k+1}} &= \text{Jacobian of Equations 3.5 with respect to } y^{k+1} \\ \mathbf{J}_{y^k} &= \text{Jacobian of Equations 3.5 with respect to } y^k \end{aligned} \tag{3.16}$$

the first term in the right side of Equation 3.15 is equal to zero since $(\vec{y}^{k+1}, \vec{y}^k, \vec{x}_j, \Delta t, \alpha)$ is a solution to the system represented by Equation 3.14; by replacing the $\delta \vec{y}$ terms with the expressions in Equation 3.12 and neglecting the second order terms we obtain:

$$\left( \mathbf{J}_{y^{k+1}} \, \vec{S}_i^{k+1} + \mathbf{J}_{y^k} \, \vec{S}_i^k + \frac{\partial \vec{f}}{\partial \alpha_i} \right) \, \delta \alpha_i = 0 \tag{3.17}$$

which becomes:

$$\mathbf{J}_{y^{k+1}} \, \vec{S}_i^{k+1} = -\mathbf{J}_{y^k} \, \vec{S}_i^k \; - \frac{\partial \vec{f}}{\partial \alpha_i} \tag{3.18}$$

The implications of Equation 3.18 are important and are:

(a) To compute the sensitivity coefficients we do not need to solve a new system of equations (we need only to rerun the numerical solver).

(b) To compute the current sensitivity we need to have the value at the previous time step.

(c) It is necessary to solve Equation 3.5 *npar* times, where *npar* is the number of parameters of the problem.

(d) The vector $\frac{\partial \vec{f}}{\partial \alpha_i}$ in Equation 3.18 is computed within the numerical simulator.

## 3.2 Sensitivity Coefficients for Discrete Reservoirs

In Section 3.1 we showed a procedure to compute the sensitivity coefficients for a general physical system. Now we will be more explicit and will apply the procedure to a hydrocarbon reservoir. The basic laws that are used to characterize the behavior of the system are: *material balance, Darcy's law,* and an *equation of state*. Because of the nonlinearities and complexities the reservoir is modeled with a numerical reservoir simulator. For a simple black–oil model such as the one used this work, the application of the basic laws results in a system of equations. Each equation is constructed by applying mass conservation at each cell of the discretized reservoir. For example, for a water–oil system, there will be two equations for each cell. The system is completely determined by knowing the variables *pressure* and *water saturation* at each cell. The reservoir parameters are among others, the *permeability* and *porosity* at each cell.

The interest is in the computation of the sensitivity of the pressure and saturation field with respect to the permeability and porosity fields. This computation can be performed very efficiently by applying the general procedure summarized in Equation 3.18. The correspondence between the nomenclature for the general case and the reservoir case follows:

$$\vec{\alpha} = \begin{bmatrix} k_1 \\ \phi_1^0 \\ \vdots \\ k_i \\ \phi_i^0 \\ \vdots \\ k_{nblocks} \\ \phi_{nblocks}^0 \end{bmatrix} \; ; \; \vec{y} = \begin{bmatrix} p_1 \\ S_{w_1} \\ \vdots \\ p_i \\ S_{w_i} \\ \vdots \\ p_{nblocks} \\ S_{w_{nblocks}} \end{bmatrix} \; ; \; \vec{f} = \begin{bmatrix} f_{o_1} \\ f_{w_1} \\ \vdots \\ f_{o_i} \\ f_{w_i} \\ \vdots \\ f_{o_{nblocks}} \\ f_{w_{nblocks}} \end{bmatrix} \tag{3.19}$$

Where $nblocks$ is the number of blocks in the discretized reservoir, $f_{o_i}$ and $f_{w_i}$ are the material balance equations for the phase oil and water for the $i^{th}$ block respectively. Since the porosity $\phi$ is usually a function of the pore pressure it cannot be taken as a parameter, instead we can define a parameter $\phi^0$ as the porosity at a certain predetermined pressure, usually at the initial reservoir conditions; thus $\phi_i = \phi_i(\phi_i^0, p_i)$.

The structure of the mass conservation equations is illustrated (following reference [48]) next for the one–dimensional case ($x$ direction), for the oil phase in a water–oil system with no capillary pressure. For the block $i$ we obtain:

$$\begin{aligned} f_{o_i} &= T_{o_{i-\frac{1}{2}}} (p_{i-1} - p_i) - T_{o_{i+\frac{1}{2}}} (p_i - p_{i+1}) - q_o - \\ & \frac{(\Delta x \, \Delta y \, \Delta z)_i}{\Delta t} \left( \left( \frac{\phi \, S_o}{B_o} \right)_i^{k+1} - \left( \frac{\phi \, S_o}{B_o} \right)_i^{k} \right) \end{aligned} \tag{3.20}$$

where

$$T_{o_{i\pm\frac{1}{2}}} = \left( \frac{\bar{k} \, k_{ro} \, \Delta z \, \Delta y}{B_o \, \mu_o \, \Delta x} \right)_{i\pm\frac{1}{2}} \tag{3.21}$$

$$\bar{k}_{i\pm\frac{1}{2}} = \frac{k_i \, k_{i\pm1} (\Delta x_i + \Delta x_{i\pm1})}{k_{i\pm1} \, \Delta x_i + k_i \Delta x_{i\pm1}} \tag{3.22}$$

$$k_{ro_{i\pm\frac{1}{2}}} = k_{ro}(S_w^{ups\pm\frac{1}{2}})\,;\; S_w^{ups} = \text{ upstream saturation} \tag{3.23}$$

$$\phi_i = \phi_i(\phi_i^0, p_i) \tag{3.24}$$

$$q_o = \text{source term: oil production from a well} \tag{3.25}$$

The formulation for two and three dimensions is straightforward. Reference [48] explains how to construct these equations for other more general cases.

The vectors of sensitivities become:

$$\vec{S}_{k_i}^k = \frac{\partial \vec{y}^k}{\partial k_i} = \begin{bmatrix} \frac{\partial p_1}{\partial k_i} \\ \frac{\partial S_{w_1}}{\partial k_i} \\ \vdots \\ \frac{\partial p_{nblocks}}{\partial k_i} \\ \frac{\partial S_{w_{nblocks}}}{\partial k_i} \end{bmatrix} \tag{3.26}$$

$$\vec{S}_{\phi_i}^k = \frac{\partial \vec{y}^k}{\partial \phi_i^0} = \begin{bmatrix} \frac{\partial p_1}{\partial \phi_i^0} \\ \frac{\partial S_{w_1}}{\partial \phi_i^0} \\ \vdots \\ \frac{\partial p_{nblocks}}{\partial \phi_i^0} \\ \frac{\partial S_{w_{nblocks}}}{\partial \phi_i^0} \end{bmatrix} \tag{3.27}$$

The first Jacobian in the left hand side of Equation 3.18 becomes,

$$\mathbf{J}_{y^{k+1}} \equiv \mathbf{J} = \begin{pmatrix} \frac{\partial f_{o_1}}{\partial p_1} & \frac{\partial f_{o_1}}{\partial S_{w_1}} & \frac{\partial f_{o_1}}{\partial p_2} & \frac{\partial f_{o_1}}{\partial S_{w_2}} & \cdots & \frac{\partial f_{o_1}}{\partial S_{w_{nblocks}}} \\ \frac{\partial f_{w_1}}{\partial p_1} & \frac{\partial f_{w_1}}{\partial S_{w_1}} & \frac{\partial f_{w_1}}{\partial p_2} & \frac{\partial f_{w_1}}{\partial S_{w_2}} & \cdots & \frac{\partial f_{w_1}}{\partial S_{w_{nblocks}}} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_{o_{nblocks}}}{\partial p_1} & \frac{\partial f_{o_{nblocks}}}{\partial S_{w_1}} & \frac{\partial f_{o_{nblocks}}}{\partial p_2} & \frac{\partial f_{o_{nblocks}}}{\partial S_{w_2}} & \cdots & \frac{\partial f_{o_{nblocks}}}{\partial S_{w_{nblocks}}} \\ \frac{\partial f_{w_{nblocks}}}{\partial p_1} & \frac{\partial f_{w_{nblocks}}}{\partial S_{w_1}} & \frac{\partial f_{w_{nblocks}}}{\partial p_2} & \frac{\partial f_{w_{nblocks}}}{\partial S_{w_2}} & \cdots & \frac{\partial f_{w_{nblocks}}}{\partial S_{w_{nblocks}}} \end{pmatrix} \tag{3.28}$$

**J** is the *last Jacobian* solved by the Newton–Raphson method (Equation 3.6) in the numerical simulator for the current *time step k+1*, provided the system of equations corresponds to a fully implicit formulation.

$\mathbf{J}_{y^k}$ becomes a tridiagonal since $\left(\frac{\phi\, S_o}{B_o}\right)_i^k$ is the only term that depends on $y^k$ in Equation 3.20. Thus the structure of $\mathbf{J}_{y^k}$ is:

$$\mathbf{J}_{y^k} \equiv -\mathbf{D} = \begin{pmatrix} -D_{1,1} & -D_{1,2} & & & & & & \\ -D_{2,1} & -D_{2,1} & & & & & & \\ & & x & x & & & & \\ & & x & x & & & & \\ & & & & \ddots & \ddots & & \\ & & & & \ddots & \ddots & & \\ & & & & & & x & x \\ & & & & & & x & x \end{pmatrix} \tag{3.29}$$

The structure of **D** is very simple as a sparse matrix. As an example, for the first four elements of **D**, assuming that the structure of the material balance equations are in the form of Equation 3.20, would be:

$$D_{1,1} = \frac{\partial \left(\frac{\phi S_o}{B_o}\right)_1^k}{\partial p_1^k} \tag{3.30}$$

$$D_{1,2} = -\left(\frac{\phi}{B_o}\right)_1^k \tag{3.31}$$

$$D_{2,1} = \frac{\partial \left(\frac{\phi S_w}{B_w}\right)_1^k}{\partial p_1^k} \tag{3.32}$$

$$D_{1,2} = \left(\frac{\phi}{B_w}\right)_1^k \tag{3.33}$$

$$\tag{3.34}$$

To facilitate notation we define $\vec{V}_{\alpha_i}$ as:

$$\vec{V}_{\alpha_i} \equiv -\frac{\partial \vec{f}}{\partial \alpha_i} \tag{3.35}$$

After Equations 3.28, 3.29 and 3.35, Equation 3.18 becomes:

$$\mathbf{J}\, \vec{S}_{\alpha_i}^{k+1} = \mathbf{D}\, \vec{S}_{\alpha_i}^{k} + \vec{V}_{\alpha_i} \tag{3.36}$$

Where $\alpha_i$ is a short notation for $k$ or $\phi^0$.
Equation 3.36 is a key point in this work. A brief analysis of this equation follows:

- If the pressure and saturation fields are coupled in the forward model (fully implicit simulation) then the sensitivity of the pressure and saturations are also coupled, that is, it is not possible to compute them separately.
- To compute the current sensitivity, $\vec{S}_{\alpha_i}^{k+1}$ it is necessary to have computed the sensitivity corresponding to the previous time step, $\vec{S}_{\alpha_i}^{k}$.
- $\vec{S}_{\alpha_i}^{0} = \vec{0}$ since the numerical simulation is always started with the reservoir in static equilibrium, and this is the starting point in the computation of Equation 3.36.
- The matrices **J** and **D** are constant for each time step ($k$) and are computed at the time of solving the *forward* problem.
- Equation 3.36 has to be solved (2 x $nblocks$) times at the end of each time step, provided that permeability and porosity at *all* the simulation blocks are the unknown parameters to be estimated.
- Except for $\vec{V}_{\alpha_i}$, all of the other terms in Equation 3.36 are computed within the numerical simulator.

The term $\vec{V}_{\alpha_i}$ is obtained by taking derivatives of the material balance equations $\vec{f}$. $\vec{V}_{\alpha_i}$ is a very sparse vector and this can be noticed by inspection of the sample material balance equation (Equation 3.20). For a two–dimensional and single–phase reservoir $\vec{V}_{k_i}$ has no more than five nonzero components, $\vec{V}_{\phi_i^0}$ has only one nonzero

component; for water–oil systems $\vec{V}_{k_i}$ has no more than ten nonzero components, $\vec{V}_{\phi_i^0}$ has only two nonzero components. The calculation is straightforward. For example, for the one–dimensional, single–phase case:

$$\frac{\partial f_{o_i}}{\partial k_{i-1}} = \frac{T_{o_{i-\frac{1}{2}}}}{\bar{k}_{i-\frac{1}{2}}} \frac{\partial \bar{k}_{i-\frac{1}{2}}}{k_{i-1}} (p_{i-1} - p_i) \tag{3.37}$$

$$\frac{\partial f_{o_i}}{\partial k_i} = \frac{T_{o_{i-\frac{1}{2}}}}{\bar{k}_{i-\frac{1}{2}}} \frac{\partial \bar{k}_{i-\frac{1}{2}}}{k_i} (p_{i-1} - p_i) - \frac{T_{o_{i+\frac{1}{2}}}}{\bar{k}_{i+\frac{1}{2}}} \frac{\partial \bar{k}_{i+\frac{1}{2}}}{k_i} (p_i - p_{i+1}) \tag{3.38}$$

$$\frac{\partial f_{o_i}}{\partial k_{i+1}} = -\frac{T_{o_{i+\frac{1}{2}}}}{\bar{k}_{i+\frac{1}{2}}} \frac{\partial \bar{k}_{i+\frac{1}{2}}}{k_{i+1}} (p_i - p_{i+1}) \tag{3.39}$$

$$\frac{\partial f_{o_i}}{\partial \phi_i^0} = \frac{(\Delta x\, \Delta y\, \Delta z)_i}{\Delta t} \left( \left( \frac{S_o}{B_o} \frac{\partial \phi}{\partial \phi^0} \right)_i^{k+1} - \left( \frac{S_o}{B_o} \frac{\partial \phi}{\partial \phi^0} \right)_i^{k} \right) \tag{3.40}$$

and explicitly,

$$\vec{V}_{k_i} = \begin{bmatrix} 0 \\ \vdots \\ \frac{\partial f_{o_{i-1}}}{\partial k_i} \\ \frac{\partial f_{o_i}}{\partial k_i} \\ \frac{\partial f_{o_{i+1}}}{\partial k_i} \\ 0 \\ \vdots \end{bmatrix} \tag{3.41}$$

and

$$\vec{V}_{\phi_i^0} = \begin{bmatrix} 0 \\ \vdots \\ \frac{\partial f_{o_i}}{\partial \phi_i^0} \\ 0 \\ \vdots \end{bmatrix} \tag{3.42}$$

In the general case of a three–dimensional and multiphase reservoir the zero–nonzero structure of $\vec{V}_{k_i}$ is always the same as the structure of the column of the Jacobian matrix corresponding to the $i$ cell.

Equation 3.36 allows us to compute the sensitivity of the pressure and saturation at each block with respect to the permeability and porosity field. However our interest is in computing the sensitivity of the *observed data* with respect to the parameters of the inverse problems. These sensitivities can be computed from the sensitivities computed with Equation 3.36 since, as was mentioned earlier, by knowing pressure and saturation in the reservoir we can completely determine the state of the secondary variables such as pressure and water cut in the wells. The computation of the sensitivities in the form required by the parameter estimation algorithm is described in the next sections.

In this work the field observations, or data for the parameter estimation problem are:

- Bottom hole pressure at the wells.
- Water cut in the wells.
- Change of saturation distribution in the reservoir in a given time interval.

### 3.2.1 Pixel Approach

As described in Section 2.8.1, in the pixel approach the parameters of the inverse problem are the permeability and porosity at each block of the simulation grid. The computation of the sensitivity coefficients for parameter estimation is explained next.

### Pressure in the Wells

The relationship between the pressure in a well, $p_w$ and the pressure in the simulation block where the well is located, $p_{block}$ is usually calculated using the Peaceman formula [49]. For a well that is producing $q_o$ at standard conditions, we have:

$$p_w = p_{block} - q_o W_f \tag{3.43}$$

$$W_f = \left(\frac{B_o}{2\,\pi k\,\lambda_o\,\Delta z}\right)_{block} \ln\left(\frac{r_0}{r_w} + s_k\right) \tag{3.44}$$

$$r_0 = 0.28\,\frac{\left(\sqrt{\Delta x^2 + \Delta y^2}\right)_{block}}{2} \tag{3.45}$$

$$\lambda_o \equiv \frac{k_{ro}}{\mu_o} \tag{3.46}$$

Equation 3.45 only applies for isotropic reservoirs, that is when $k_x = k_y = k$. For injection wells Equation 3.44 is computed using the total mobility, that is $\lambda_T = \lambda_o + \lambda_w$.

The sensitivity is computed as follows:

$$\frac{\partial p_w}{\partial k_j} = \frac{\partial p_{block}}{\partial k_j} - q_o\,\frac{\partial W_f}{\partial k_j} \tag{3.47}$$

$$\frac{\partial p_w}{\partial k_j} = \frac{\partial p_{block}}{\partial k_j} - q_o\,\frac{\partial W_f}{\partial p_{block}}\,\frac{\partial p_{block}}{\partial k_j} - q_o\,\frac{\partial W_f}{\partial S_{w_{block}}}\,\frac{\partial S_{w_{block}}}{\partial k_j} - q_o\,\frac{d\,W_f}{d\,k_j} \tag{3.48}$$

$$\frac{d\,W_f}{d\,k_j} = \begin{cases} -\frac{W_f}{k} & \text{if } j = block \\ 0 & \text{otherwise} \end{cases} \tag{3.49}$$

$$\frac{\partial p_w}{\partial \phi_j^0} = \frac{\partial p_{block}}{\partial \phi_j^0} - q_o\,\frac{\partial W_f}{\partial \phi_j^0} \tag{3.50}$$

$$\frac{\partial p_w}{\partial \phi_j^0} = \frac{\partial p_{block}}{\partial \phi_j^0} - q_o \frac{\partial W_f}{\partial p_{block}} \frac{\partial p_{block}}{\partial \phi_j^0} - q_o \frac{\partial W_f}{\partial S_{w_{block}}} \frac{\partial S_{w_{block}}}{\partial \phi_j^0} \tag{3.51}$$

$$\frac{\partial p_w}{\partial s_k} = \frac{1}{\left(\frac{r_0}{r_w} + s_k\right)} \left(\frac{B_o}{2\,\pi k\,\lambda_o\,\Delta z}\right)_{block} \tag{3.52}$$

Equations 3.48, 3.51, and 3.52 allow us to compute the sensitivity of the pressure at each at the *end* of each time step of the simulation. The sensitivities of the pressure in the block where the well is located are computed using Equation 3.36.

### Water Cut

The water cut at standard conditions is defined as:

$$wc \equiv \frac{q_w}{q_w + q_o} \tag{3.53}$$

In reservoir simulation the water cut in the producing wells is approximated by using the pressure and saturation where the well is located, as follows:

$$wc = \left(\frac{\frac{\lambda_w}{B_w}}{\frac{\lambda_w}{B_w} + \frac{\lambda_o}{B_o}}\right)_{block} \tag{3.54}$$

$$\lambda_o \equiv \frac{k_{ro}}{\mu_o} \qquad \lambda_w \equiv \frac{k_{rw}}{\mu_w} \tag{3.55}$$

The sensitivity is computed as:

$$\frac{\partial wc}{\partial k_j} = \frac{\partial wc}{\partial p_{block}} \frac{\partial p_{block}}{\partial k_j} + \frac{\partial wc}{\partial S_{w_{block}}} \frac{\partial S_{w_{block}}}{\partial k_j} \tag{3.56}$$

$$\frac{\partial wc}{\partial \phi_j^0} = \frac{\partial wc}{\partial p_{block}} \frac{\partial p_{block}}{\partial \phi_j^0} + \frac{\partial wc}{\partial S_{w_{block}}} \frac{\partial S_{w_{block}}}{\partial \phi_j^0} \tag{3.57}$$

$$\frac{\partial wc}{\partial s_k} = 0\,;\,\forall\, s_k \tag{3.58}$$

Equations 3.56, 3.57, and 3.58 allow us to compute the sensitivity of the water cut at each well at the *end* of each time step. The sensitivities of the pressure and saturation in the block where the well is located are calculated with Equation 3.36.

**Change of Saturation Distribution**

The change of saturation $dsat$ at the $i^{th}$ simulation block in the time interval $t_2 - t_1$ is defined as:

$$dsat_i = S_{w_i}^{t_2} - S_{w_i}^{t_1} \tag{3.59}$$

The sensitivity is calculated as:

$$\frac{\partial dsat_i}{\partial k_j} = \frac{\partial S_{w_{block}}^{t_2}}{\partial k_j} - \frac{\partial S_{w_{block}}^{t_1}}{\partial k_j} \tag{3.60}$$

$$\frac{\partial dsat_i}{\partial \phi_j^0} = \frac{\partial S_{w_{block}}^{t_2}}{\partial \phi_j^0} - \frac{\partial S_{w_{block}}^{t_1}}{\partial \phi_j^0} \tag{3.61}$$

$$\frac{\partial dsat_i}{\partial s_k} = 0 \; ; \; \forall \, s_k \tag{3.62}$$

Equation 3.62 is equal to zero when the production and injection rates in the wells are fixed (known). This is the case in reservoir parameter estimation, since the rates are specified to replicate the known history.

Equations 3.60, 3.61, and 3.62 allow us to compute the sensitivity to the change of saturation at each block of the simulation grid. The sensitivities to the saturation are calculated with Equation 3.36.

### 3.2.2 Object Approach

As indicated in Section 2.8.2, the parameters in the object approach to the inverse problem are no longer the permeability and porosity at each cell of the simulation

grid. The permeability and porosity are functions of the parameter vector $\vec{\alpha}$. That is:

$$k_i = k_i(\vec{\alpha}) \tag{3.63}$$

$$\phi_i^0 = \phi_i^0(\vec{\alpha}) \tag{3.64}$$

The interest is in computing the sensitivity of the pressure and water cut in the wells and the change of saturation distribution in the reservoir with respect to the parameters in $\vec{\alpha}$. We define the sensitivity with respect to $\alpha_i$ as $\vec{Z}_{\alpha_i}$ to differentiate from the sensitivity with respect to permeability and porosity, $\vec{S}$. That is:

$$\vec{Z}_{\alpha_j} \equiv \frac{\partial \vec{y}}{\partial \alpha_j}\,;\, \vec{S}_{k_i} \equiv \frac{\partial \vec{y}}{\partial k_i}\,;\, \vec{S}_{\phi_i} \equiv \frac{\partial \vec{y}}{\partial \phi_i^0} \tag{3.65}$$

One alternative to compute the sensitivity would be to use the chain rule to compute derivatives and thus use the basic sensitivities as they can be computed with Equation 3.36, that is for example:

$$\frac{\partial p_i}{\partial \alpha_j} = \sum_{m=1}^{nblocks} \frac{\partial p_i}{\partial k_m} \frac{\partial k_m}{\partial \alpha_j} \tag{3.66}$$

which is generalized as:

$$\vec{Z}_{\alpha_j} = \sum_{m=1}^{nblocks} \left( \vec{S}_{k_m} \frac{\partial k_m}{\partial \alpha_j} + \vec{S}_{\phi_m} \frac{\partial \phi_m}{\partial \alpha_j} \right) \tag{3.67}$$

The problem with the chain rule approach shown in Equation 3.67 is that it would be very inefficient since it requires the computation of all $\frac{\partial p_i}{\partial k_m}$ which means to solve Equation 3.36 2 x $nblocks$ times, hence it would be faster to use the substitution method when $nblocks \gg npar$. Another alternative would be to rewrite the mathematical model as a function of $\vec{\alpha}$ instead of $k$ and $\phi^0$, this would allow us to compute the sensitivities very quickly since it would be necessary to solve Equation 3.18 only $npar$ times; but rewriting the mathematical model means to recode the numerical

reservoir simulator, which is not practical.

We derived a procedure to compute the sensitivities with respect to $\vec{\alpha}$ which does not require a rewrite of the numerical simulator and which requires the solution to an equation similar to 3.36 only *npar* times. By this procedure it is possible to compute the sensitivities in a very efficient way. A description of the procedure follows. A description of the procedure follows.

If we multiply every term in Equation 3.36 for permeability by $\frac{\partial k_m}{\partial \alpha_j}$ and the same for porosity by $\frac{\partial \phi_m}{\partial \alpha_j}$ and then sum over all the blocks in the reservoir, we obtain:

$$\mathbf{J} \sum_{m=1}^{nblocks} \left( \vec{S}_{k_m} \frac{\partial k_m}{\partial \alpha_j} + \vec{S}_{\phi_m} \frac{\partial \phi_m}{\partial \alpha_j} \right)^{k+1} = \mathbf{D} \sum_{m=1}^{nblocks} \left( \vec{S}_{k_m} \frac{\partial k_m}{\partial \alpha_j} + \vec{S}_{\phi_m} \frac{\partial \phi_m}{\partial \alpha_j} \right)^{k} + \sum_{m=1}^{nblocks} \left( \vec{V}_{k_m} \frac{\partial k_m}{\partial \alpha_j} + \vec{V}_{\phi_m} \frac{\partial \phi_m}{\partial \alpha_j} \right) \tag{3.68}$$

By inserting Equation 3.67 into Equation 3.68 we obtain:

$$\mathbf{J}\, \vec{Z}_{\alpha_j}^{k+1} = \mathbf{D}\, \vec{Z}_{\alpha_j}^{k} + \vec{W}_{\alpha_j} \tag{3.69}$$

$$\text{with} \quad \vec{Z}_{\alpha_j}^{0} = \vec{0} \tag{3.70}$$

$\vec{W}_{\alpha_j}$ is defined as:

$$\vec{W}_{\alpha_j} \equiv \sum_{m=1}^{nblocks} \left( \vec{V}_{k_m} \frac{\partial k_m}{\partial \alpha_j} + \vec{V}_{\phi_m} \frac{\partial \phi_m}{\partial \alpha_j} \right) = \frac{\partial \vec{f}}{\partial \alpha_j} \tag{3.71}$$

Equation 3.69 allow us to compute the sensitivities in a very efficient way. Because of the importance of this equation a brief analysis follows:

- The matrices **J** and **D** are the same as in Equation 3.36. That is, they are calculated from within the numerical simulator; no special work is needed to form these matrices.

- Computing the vector $\vec{W}$ is the only "extra" work.
- It is necessary to solve Equation 3.69 only *npar* times, and this is independent of *nblocks*, the discretization in the simulation grid.

The last item has a critical impact in the time performance of the inverse problem. In general, with object modeling, it is possible to characterize a reservoir with a relatively small number of parameters, that is $npar \ll nblocks$ and thus the speed is of the same order of magnitude as the simulation time. This will make the reservoir parameter estimation a practical procedure.

The computation of the sensitivity of the pressure and water cut in the wells and the change of saturation distribution in the reservoir is computed in a way similar to that shown for the case of *pixel modeling* in the previous section, that is:

$$\frac{\partial p_w}{\partial \alpha_j} = \left(1 - q_o \frac{\partial W_f}{\partial p_{block}}\right) \frac{\partial p_{block}}{\partial \alpha_j} - q_o \frac{\partial W_f}{\partial S_{w_{block}}} \frac{\partial S_{w_{block}}}{\partial \alpha_j} - q_o \frac{d\, W_f}{d\, k_j} \frac{\partial k_{block}}{\partial \alpha_j} \tag{3.72}$$

$$\frac{\partial wc}{\partial \alpha_j} = \frac{\partial wc}{\partial p_{block}} \frac{\partial p_{block}}{\partial \alpha_j} + \frac{\partial wc}{\partial S_{w_{block}}} \frac{\partial S_{w_{block}}}{\partial \alpha_j} \tag{3.73}$$

$$\frac{\partial dsat_i}{\partial \alpha_j} = \frac{\partial S^{t_2}_{w_{block}}}{\partial \alpha_j} - \frac{\partial S^{t_1}_{w_{block}}}{\partial \alpha_j} \tag{3.74}$$

$$\frac{\partial p_w}{\partial s_k} = \frac{1}{\left(\frac{r_0}{r_w} + s_k\right)} \left(\frac{B_o}{2\,\pi k\,\lambda_o\,\Delta z}\right)_{block} \tag{3.75}$$

We see that when the parameters in object modeling become the permeability and porosity in each block of the simulation grid, then all the equations developed for object modeling become the equations for pixel modeling. This may seem trivial but it is not. The implications is that the same computer code can be used for object modeling or pixel modeling. That is, the pixel modeling can be considered as a special case of the more general object modeling, where the objects are static (cannot "float" in the reservoir), and the shape of each object is the shape of the simulation block.

## 3.3 Implementation

In this section the implementation of the computation of the sensitivity coefficients by using the formulas developed in the previous section is presented. A special treatment is considered for the case of *dynamic objects*, where it is necessary to convert the continuous nature of the object boundary to a discrete distribution of permeability and porosity in the discrete domain of the simulation grid.

### 3.3.1 Linear Solver

The computation of the sensitivity coefficients with either pixel or object modeling, Equations 3.36 and 3.69, is a multiple right hand side problem. That is, at the end of each time step it is necessary to solve a linear system of the type:

$$\mathbf{J}\,\vec{x}_i = \vec{b}_i\,; \qquad i = 1, 2, \ldots, npar \tag{3.76}$$

The numerical simulator solves a similar equation during the iterations of the Newton–Raphson method to solve the nonlinear material balance equations, $\vec{f} = 0$. The difference is that equations are solved only once, and for this approach, most commercial simulators use iterative solvers that are faster and require less memory than direct solvers, such as $LU$ and direct sparse matrix solvers.

Solving Equation 3.76 many times, specially in the case of pixel modeling where $npar$ is up to twice the dimension of $\mathbf{J}$, $nblocks$, has a critical impact on the performance of the inverse process.

Choosing the best strategy to solve the multiple right hand problem of Equation 3.76 is problem and hardware dependent. If the computer memory is the limiting factor, then the option is to use the same iterative solver used by the numerical simulator, this approach is not efficient because it does not account for the fact that the matrix $\mathbf{J}$ is the same during the sequential solution of Equation 3.76, a better approach would be to use a sort of parallel processing to solve the multiple right hand problem. In this work the the multiple right hand problem was solved using a direct sparse matrix solver, which produces the $\mathbf{LU}$ factors of the matrix $\mathbf{J}$ taking into account the sparse characteristics of the latter. The multiple right hand problem

is solved sequentially by using such factors. Solving triangular systems is very fast and efficient. To solve a system such as $\mathbf{L}\,\mathbf{U}\,\vec{x} = \vec{b}$ requires up to $nblocks^2$ operations, this is in the worst case when the matrices $\mathbf{L}$ and $\mathbf{U}$ being fully populated, but this is not the case with sparse matrices and thus the actual number of operations would be less than $nblocks^2$.

The direct sparse matrix solver used in this work was the *Yale Sparse Matrix Package* [50].

### 3.3.2 Computation of Sensitivity for Pixel Models

The accuracy of the calculation of the sensitivities as they are computed with Equation 3.36 was tested by comparing to their calculation by the substitution method.

For this purpose a numerical experiment was prepared for a two–dimensional, water–oil system.

The reservoir was modeled as depicted in Fig. 3.1. There are four wells, well #4 is injecting water, the others are producing. The rates were fixed: water injection rate was fixed at standard conditions, the production rates were fixed at constant total production (water+oil) at bottom hole conditions. The permeability and porosity were set constant throughout the reservoir to $k = 500$ md and $\phi^0 = 0.193$. The dimensions of the reservoir were set to be small enough in order to investigate all the possible scenarios in a short period of time. The reservoir was discretized with a 40 x 30 grid (1200 blocks). At early time, a DST at each well was simulated. Fig. 3.2 depicts the water saturation distribution at $time = 200$ days. The parameters for the inverse problem are the permeability and porosity at each block of the simulation grid plus the skin factor at each well, thus the total number of parameters is 2404. In order to check the accuracy of the calculation of the sensitivity coefficients the permeability and porosity at block #997 were perturbed and the sensitivity of the pressure and water cut at well #3 and reservoir water saturation distribution at $time = 200$ days were computed with the substitution method. The effect of the skin was computed in well #3. The DST pressure at well #3 is depicted in Fig. 3.3, there is one flow

period of 0.5 days followed by a shut–in of 1.5 days. The long term pressure and water cut at well #3 are depicted in Fig. 3.4; water arrives at approximately $time = 85$ days. The apparently odd behavior of the well at the time of the water arrival is due to the type of constraint in the well (*total rate constant at bottom hole conditions*). Figures 3.5 through 3.16 show the sensitivities computed both by the substitution method (central differences) and by using the equations derived from Equation 3.36. It is noted that both computations give almost the same values. The wiggles observed in the computation of the sensitivity of the DST pressure with respect to porosity are due to numerical rounding, Fig. 3.6.

The effect of the position of the control block with respect to the water front was investigated by computing the sensitivity of the water saturation distribution in the reservoir with respect to the permeability and porosity of block #308, which is located ahead of the water front at $time = 200$ days. This is shown in Figures 3.17 and 3.18.

The following are observations with respect to the results shown in Figures 3.5 through 3.16:

- The method used in this work, Equation 3.36, is accurate (assuming the substitution method as the benchmark).
- Porosity is as important as permeability. Most of the work in the literature review concentrated on permeability as the parameter having the highest influence on the reservoir behavior and left the porosity as a constant. From the specific case shown here it is possible to assert that porosity must be included as a parameter in the reservoir parameter estimation.
- The sensitivity of the pressure is higher in the DST than in the long term pressure history.
- The sensitivity of the long term pressure and water cut reach a peak at the time of water arrival. The conclusion is that it is very important to record pressure

and water cut at wells shortly after water arrives, since these data are rich in information.

- Sensitivities to permeability and porosity have opposite signs in the case of water cut. The effect is that, if for example it is necessary to change the permeability and porosity in block #997 to increase the water cut in well #3, the natural adjustment, from the sensitivity coefficient analysis, would be to simultaneously increase permeability and decrease porosity. This is not the right action since permeability and porosity have a positive correlation, that is higher porosity means higher permeability. The conclusion is that permeability and porosity cannot be left as independent variables, it is important to include a correlation between these two parameters.

- The sensitivity of the water saturation in the reservoir with respect to permeability, Fig. 3.11 is higher *all* along the water front. The porosity seems to have an effect also along the water front but it is more localized close to the block being evaluated.

- The effect of permeability and porosity in blocks ahead of the water front have a minor effect on the water saturation distribution in the reservoir from the sensitivity point of view. The conclusion is that the water front position, (obtained from 4–D seismic) will be richer in information from upstream than from downstream.

The actual performance of the application of the method to compute the sensitivity coefficients with Equation 3.36 was evaluated by comparing the actual CPU time to the CPU time required to run a plain simulation with the same code and in the same hardware (a DEC Alpha workstation running at 125 MHz). The results follow:

| *Mode* | *User time (seconds)* |
|---|---|
| To run simulation + 2400 sensitivities | 4255.33 |
| To run simulation only | 136.30 |

From the previous table we can evaluate the efficiency of the application of Equation 3.36 against the substitution method, which is used as a benchmark. The results follow:

| *Mode* | *User time (seconds)* |
|---|---|
| Use Equation 3.36 to compute 2400 sensitivities | 4255.33 |
| Use Substitution method to compute 2400 sensitivities | 327256.30 |

As a conclusion the method used in this work, for the specific case shown here, is 77 times faster than the substitution method. This order of magnitude does not mean anything by itself without specifying the general context in which it is being used. As was presented in Chapter 2, solving reservoir inverse problems involves many function evaluations, that is simulation runs. In a case such as the one shown here this may be of the order of hundreds of runs. Thus the order of magnitude of the computing time will be days. For example, trying to solve a problem like the example here may take two days of computing time when using Equation 3.36 and five months to perform the same work using the substitution method, which is clearly impractical. In other words, the 77 times efficiency increase makes the inversion a viable technique.

Another conclusion from the previous numerical experiment is that the performance of the parameter estimation algorithm will be greatly improved by the application of the linear search technique. The expensive work is in the computation of the *descent direction* which requires sensitivity coefficients, the linear search requires only plain simulation runs, which are much less expensive from the computation point of view.

The improvements introduced by the use of the method developed in Equation 3.36 still are not enough to handle very large problems, that is when the simulation grid is of the order of several thousand cells. Thus the method of object modeling was an important development in this work.

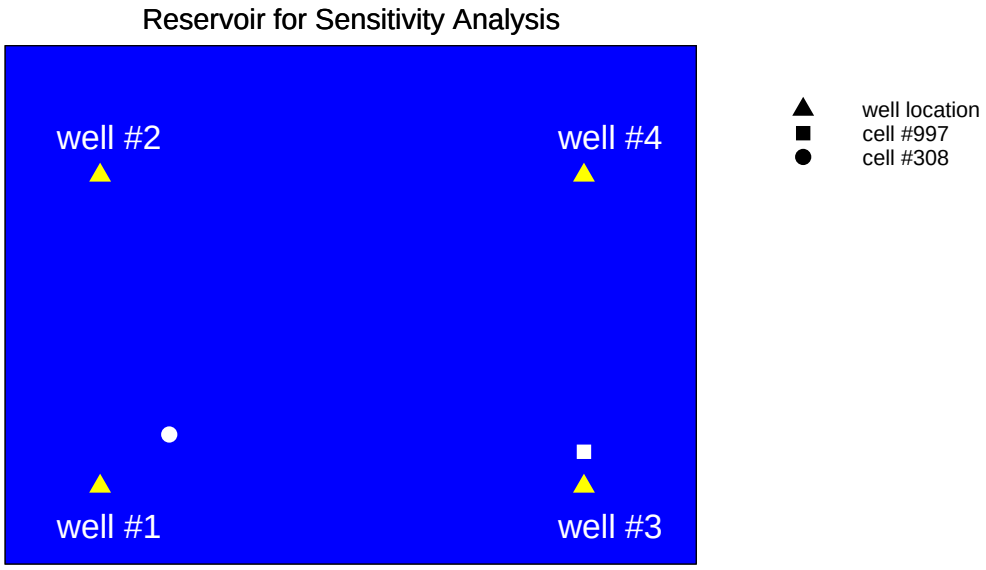



Figure 3.1: Pixel method – reservoir for analysis of the method to compute the sensitivity coefficients.

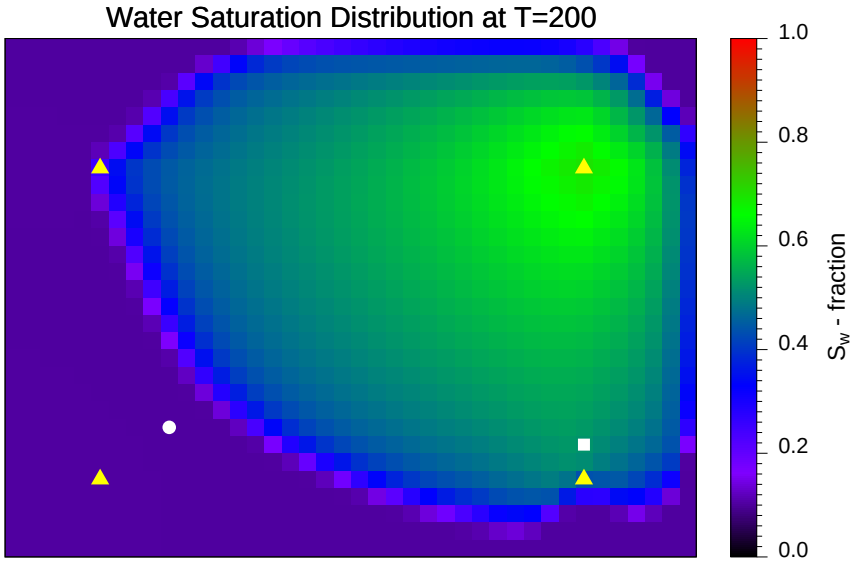



Figure 3.2: Water saturation distribution in the reservoir at time T=200.

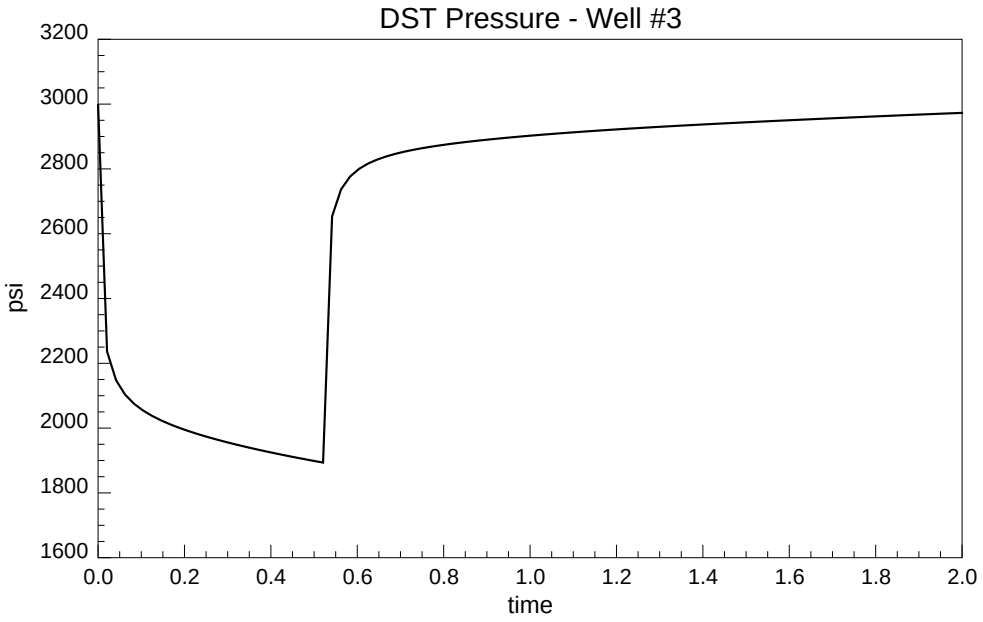



Figure 3.3: Well #3 – DST pressure.

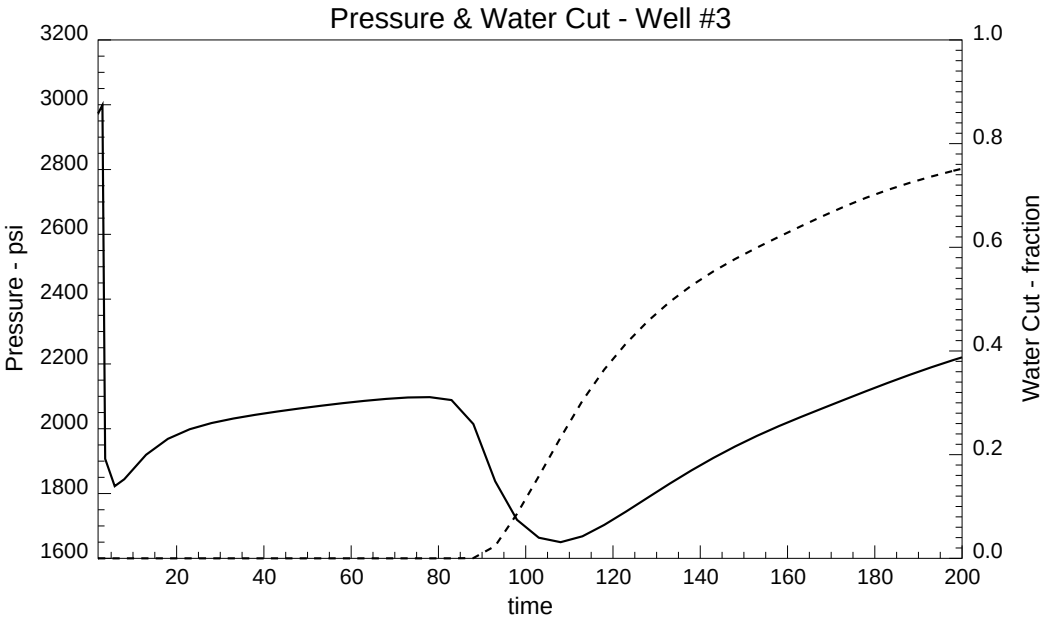



Figure 3.4: Well #3 – Long term pressure and water cut.

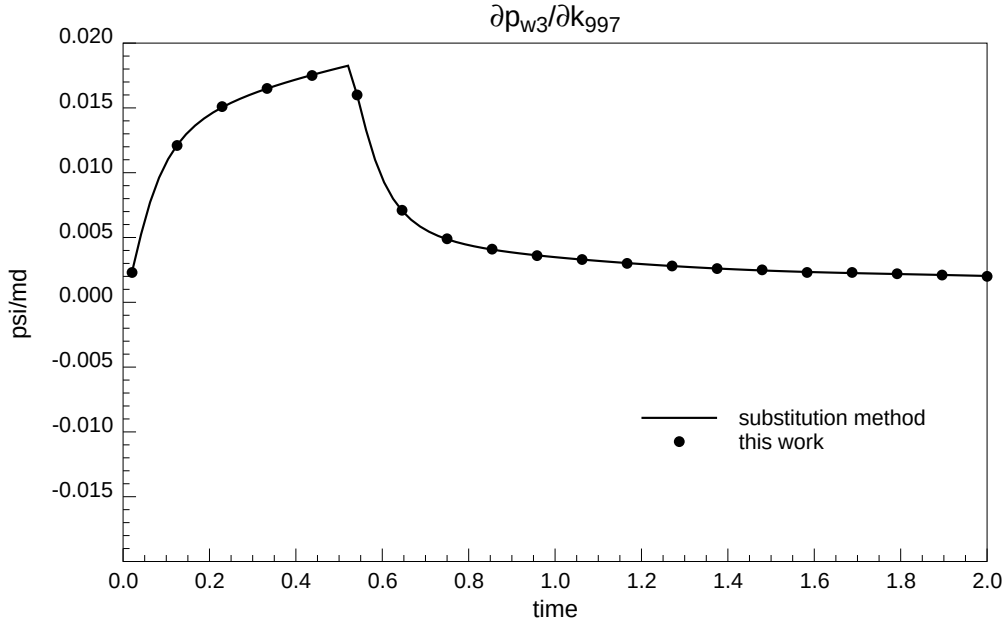



Figure 3.5: Sensitivity of DST–pressure in well #3 with respect to the permeability in block #997.

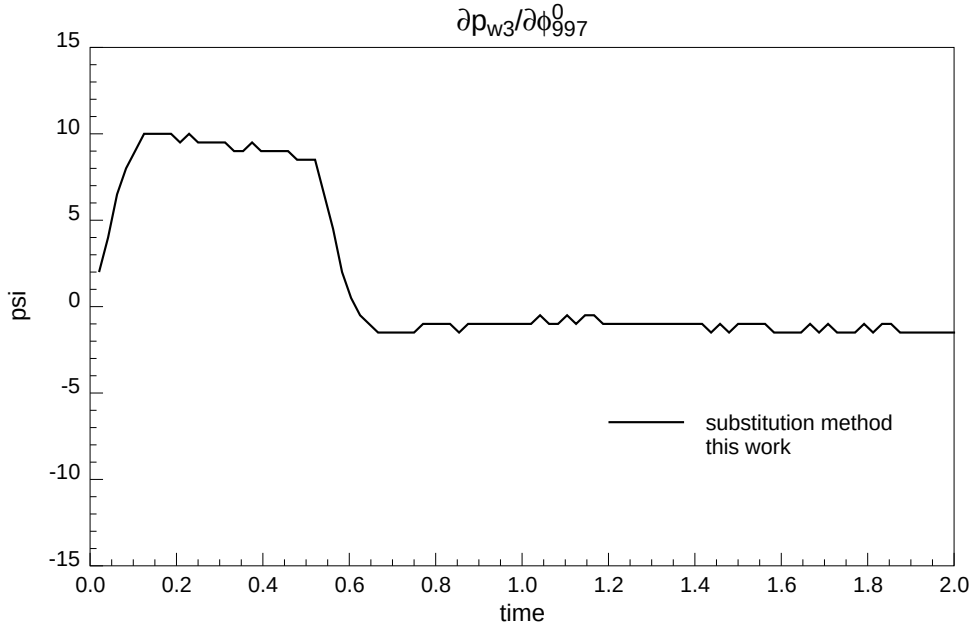



Figure 3.6: Sensitivity of DST–pressure in well #3 with respect to the porosity in block #997.

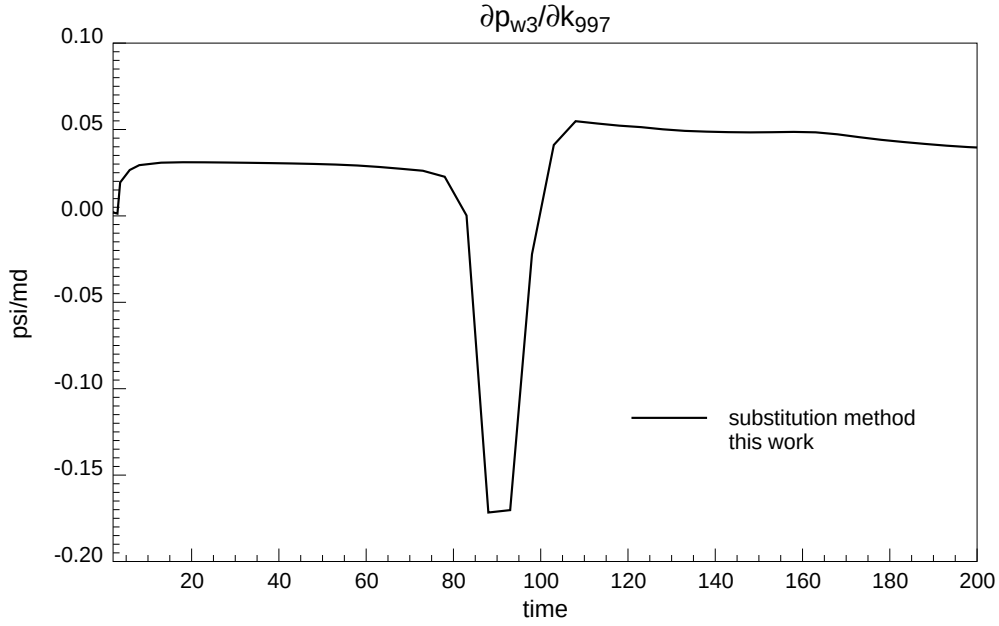



Figure 3.7: Sensitivity of long term pressure in well #3 with respect to the permeability in block #997.

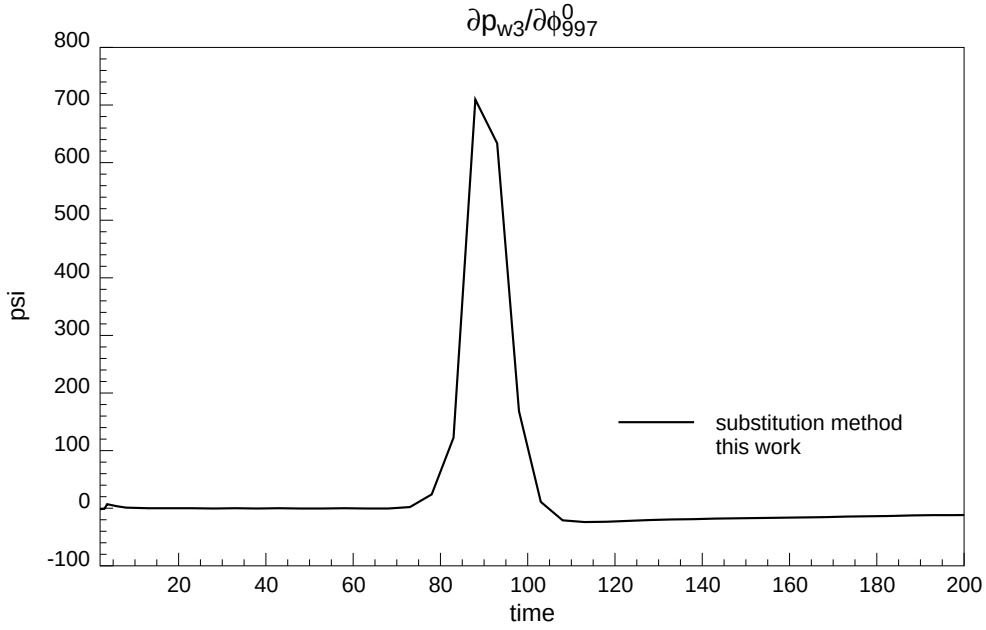



Figure 3.8: Sensitivity of long term pressure in well #3 with respect to the porosity in block #997.

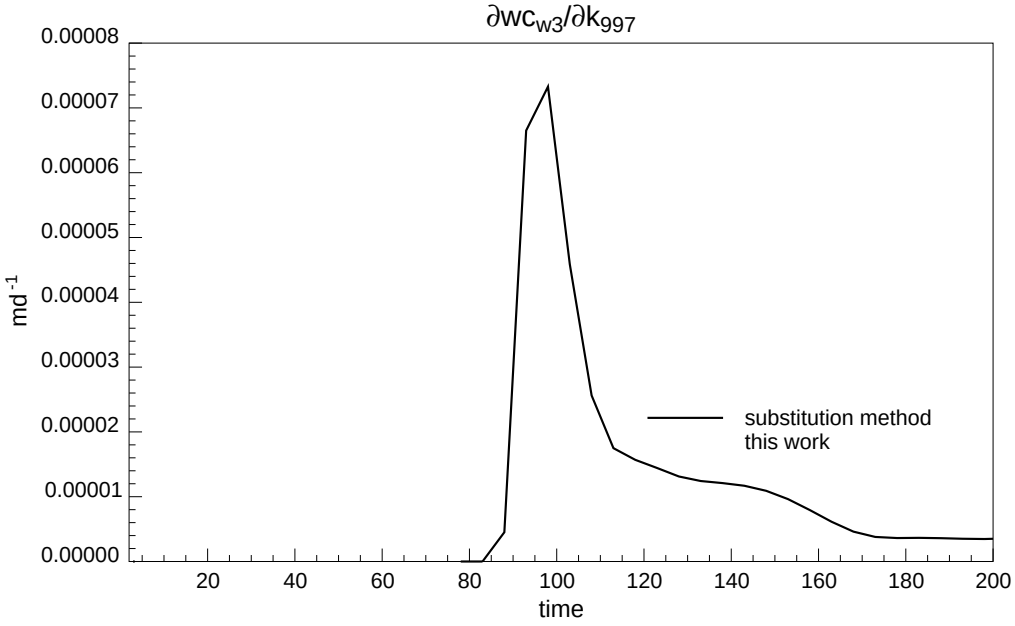



Figure 3.9: Sensitivity of the water cut in well #3 with respect to the permeability in block #997.

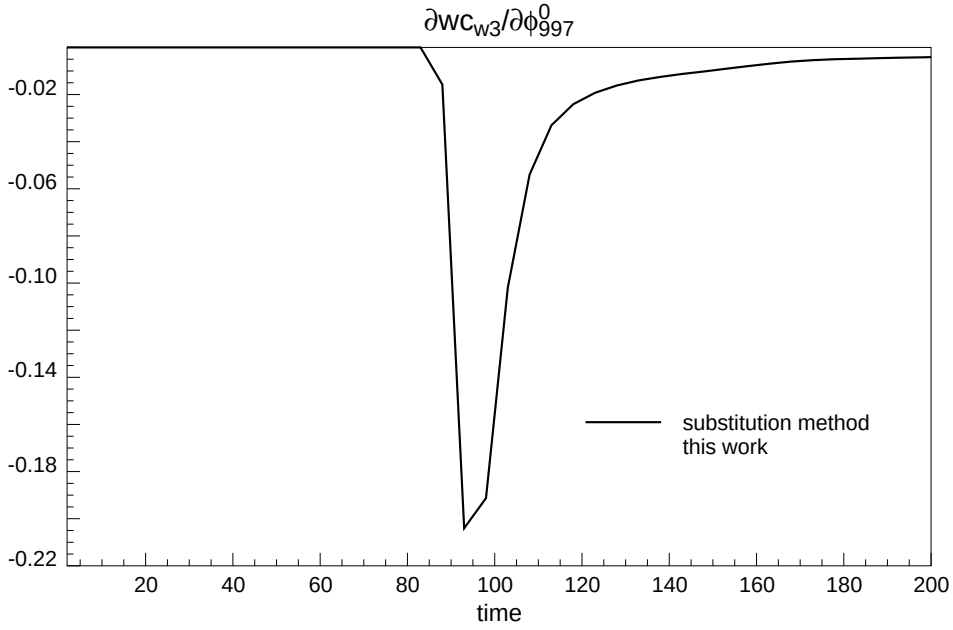



Figure 3.10: Sensitivity of the water cut in well #3 with respect to the porosity in block #997.

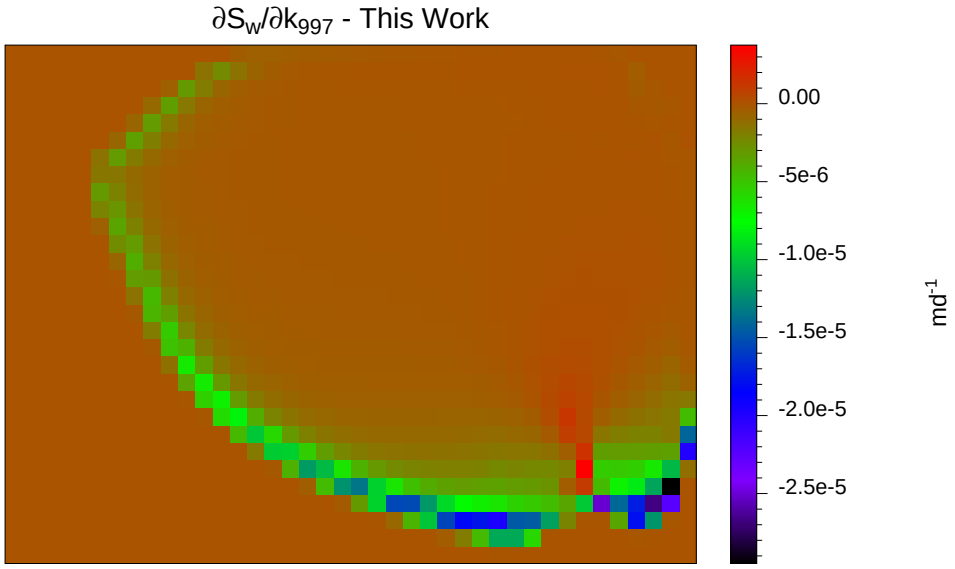



Figure 3.11: Sensitivity of water saturation distribution with respect to the permeability in block #997, as calculated by this work.

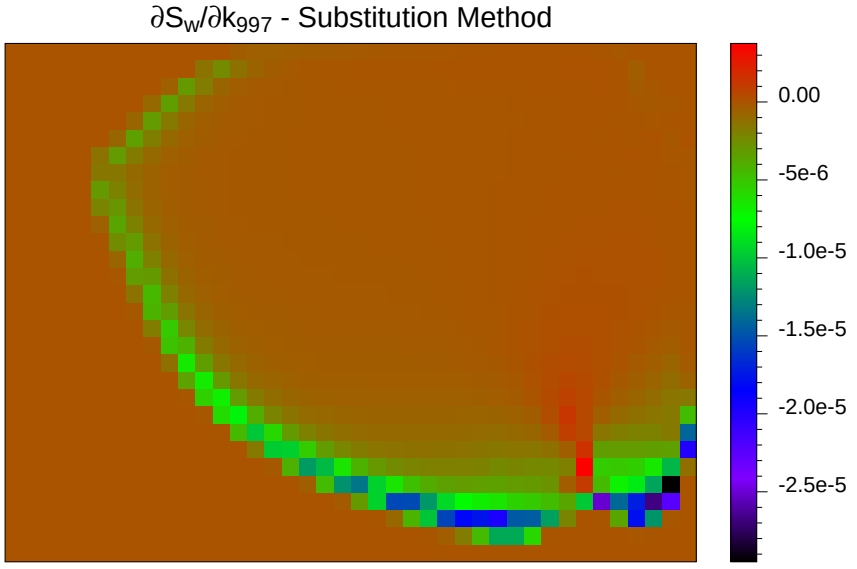



Figure 3.12: Sensitivity of water saturation distribution with respect to the permeability in block #997, as calculated by substitution method.

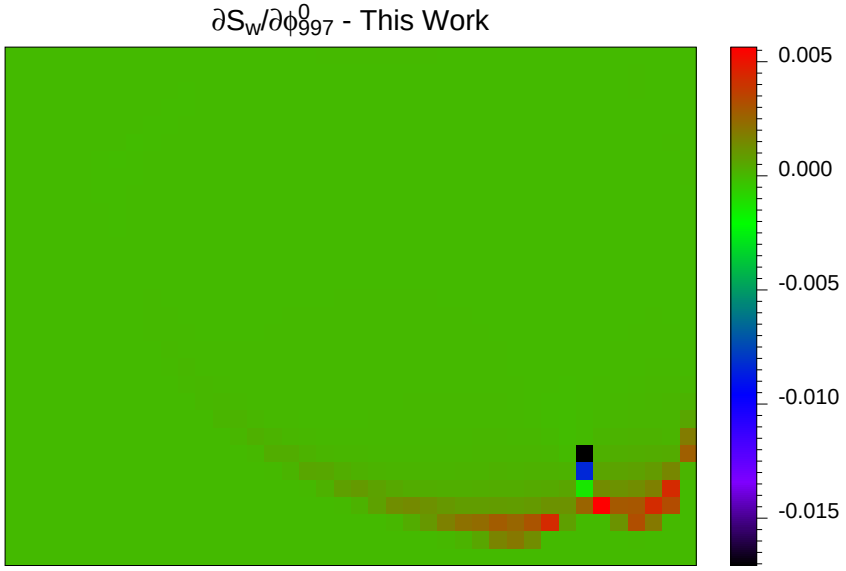



Figure 3.13: Sensitivity of water saturation distribution with respect to the porosity in block #997, as calculated by this work.

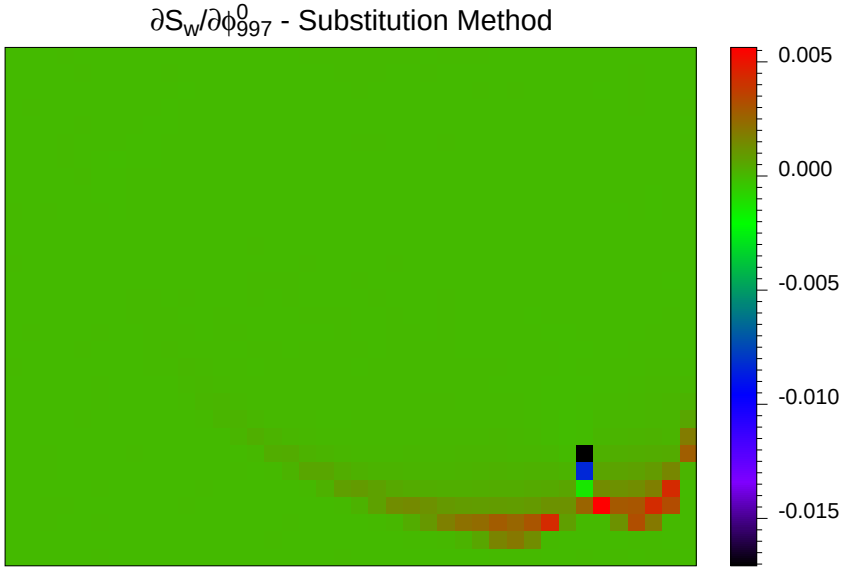



Figure 3.14: Sensitivity of water saturation distribution with respect to the porosity in block #997, as calculated by substitution method.

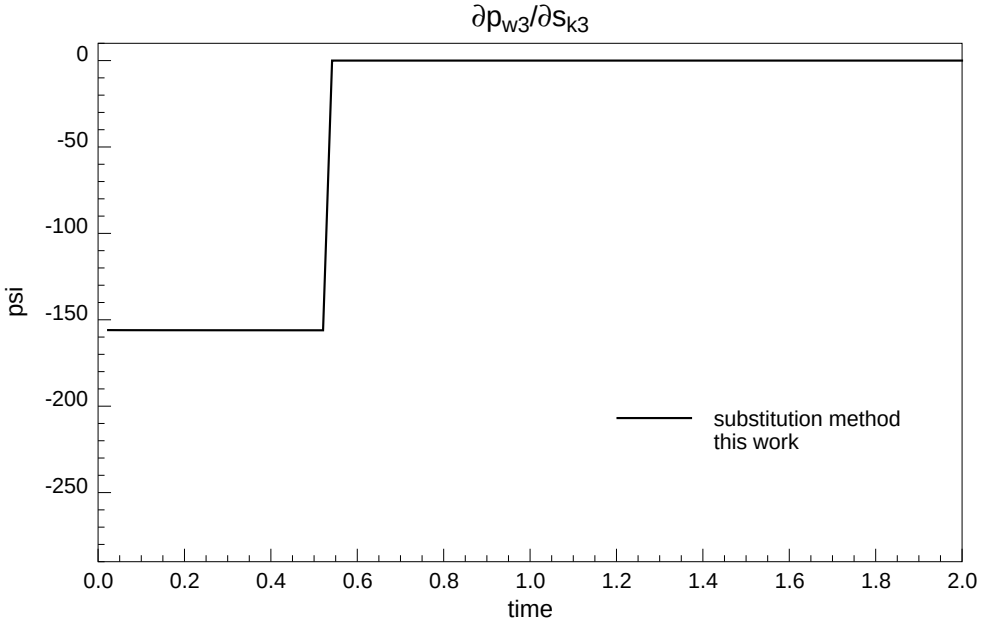



Figure 3.15: Sensitivity of DST – pressure in well # 3 with respect to the skin factor in the same well.

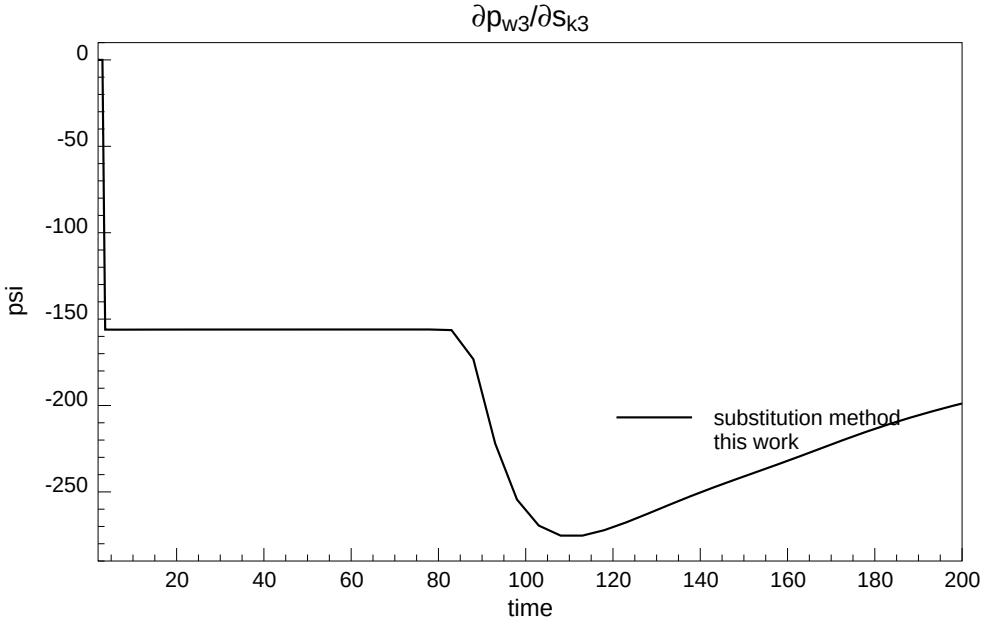



Figure 3.16: Sensitivity of long term pressure in well # 3 with respect to the skin factor in the same well.

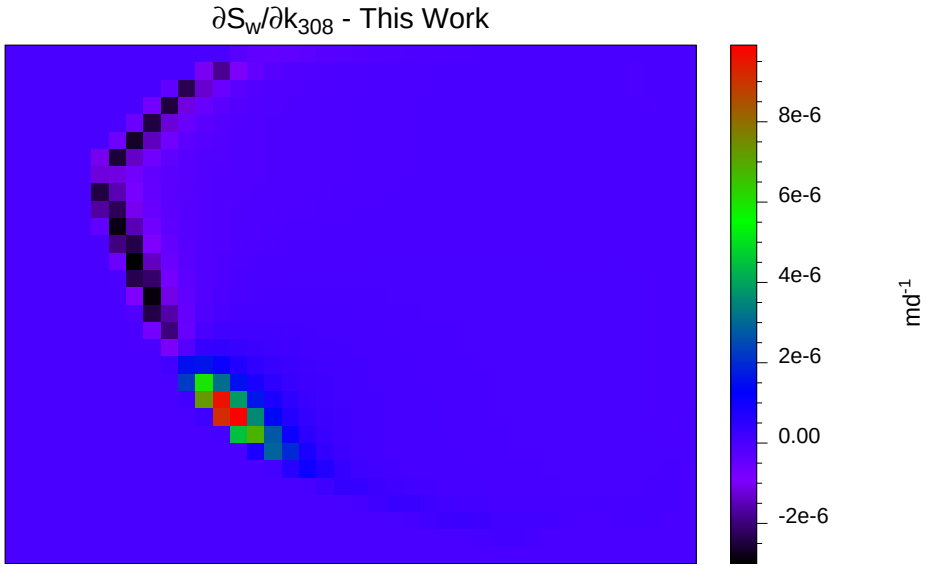



Figure 3.17: Sensitivity of water saturation distribution with respect to the permeability in block #308, as calculated by this work.

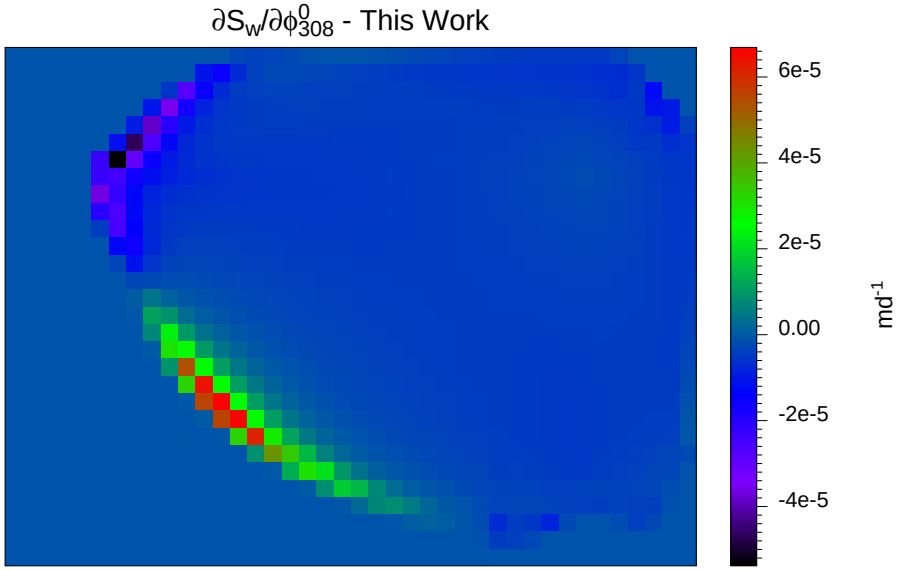



Figure 3.18: Sensitivity of water saturation distribution with respect to the porosity in block #308, as calculated by this work.

### 3.3.3 Computation of Sensitivity Coefficients for Object Models

The most complex case arises when dealing with the computation of the sensitivity coefficients with respect to the parameters that define dynamic objects, that is when the objects can "float" in the reservoir. The difficulty arises when it is necessary to translate a continuous characteristic, like a movement, a rotation or a change of shape to a discrete distribution of permeability and porosity in the simulation grid. Thus the first step is in creating the object in a way that allows a continuous variation of properties along the object boundaries.

One way to construct an object is by defining its boundary through a continuous curve. Such a curve can be defined in many ways, one very convenient way is to define it as a parametric curve, such as:

$$\begin{cases} y &= y(t) \\ x &= x(t) \end{cases} \tag{3.77}$$

where $x$ and $y$ are the spatial coordinates in two dimensions, $t$ is the parameter of the curve. $t$ can vary continuously in a fixed range, let's say in the interval $[0, 1]$. The condition we set to define the object is that the initial and last point are coincident, that is:

$$\begin{cases} y(1) &= y(0) \\ x(1) &= x(0) \end{cases} \tag{3.78}$$

This condition defines a closed area in the plane. Once the curve has been defined it is possible to apply rotations and translation as a sequence of events.

The case of an ellipse is an example of an object that can be worked out easily as a parametric curve. In this case we define the basic coordinates of the object as:

$$\begin{cases} y_0 &= \alpha_1 \sin(2\,\pi t) \\ x_0 &= \alpha_2 \cos(2\,\pi t) \end{cases} \tag{3.79}$$

Where $t \in [0,1]$, $\alpha_1$ and $\alpha_2$ are the parameters that define the main axis of the ellipse.

The rotation and translation of the curve, as defined by Equation 3.79 are calculated as follows, and shown in Fig. 3.19:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos\alpha_3 & -\sin\alpha_3 \\ \sin\alpha_3 & \cos\alpha_3 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + \begin{bmatrix} \alpha_4 \\ \alpha_5 \end{bmatrix} \tag{3.80}$$

Where $\alpha_3$ is the parameter that defines the rotation of the object, and $\alpha_4$ and $\alpha_5$ define translations in the $x$ and $y$ direction respectively.

The parameters $\alpha_1$ though $\alpha_5$ completely define the *geometry* of the object *ellipse*. A change of these parameters results in changes in the shape, and rigid rotations and translations in two–dimensional space.

The definition of the object is determined by setting the permeabilities inside and outside the ellipse as parameters $\alpha_6$ and $\alpha_7$ respectively. The porosity can be assumed to be directly related to permeability through a function $\phi^0 = \phi^0(k)$. Thus only *seven* parameters are necessary to define the object from the point of view of reservoir inverse problem.

The key point arises at the time of converting the continuous reservoir description to the discrete reservoir simulation grid. It is observed that minute changes in a geometric parameter, let's say $\alpha_1$, results in minute changes in the shape of the object that have an effect only in the reservoir properties along the boundary of the object. $\frac{\partial k_i}{\partial \alpha_1}$ as required by Equation 3.69 will be zero everywhere except at the object boundary. Thus the key step is in finding a way to compute this derivative in the simulation grid. In this work, this was accomplished by assigning a continuous range of permeability to the simulation cells that are intercepted by the parametric curve that defines the object. The basic idea is that the permeability to assign to such cells

must be somewhere between the permeabilities inside and outside the object and that the greater the area of the cell inside the object the more the cell permeability must be weighted to the inside permeability value. The simplest way to do this is by assigning the permeability proportional to the area intercepted. This is not the only way, a more sophisticated approach would be to assign different permeability in each direction, or even to try a grid refinement. Since the interest of this work is in pursuing the large scale description, then the simplest approach was applied, that is the one assigning an isotropic permeability proportional to the area intercepted. This methods allows us to compute the needed $\frac{\partial k_i}{\partial \alpha_1}$ in a practical and efficient way. The process of constructing the object and computing $\frac{\partial k_i}{\partial \alpha_1}$ is summarized as follows:

1. Generate the basic object, using Equation 3.79.
2. Apply translations and rotations to the basic object using Equation 3.80. See Fig. 3.19.
3. Find the intercept of the object boundary curve with the grid. See Fig. 3.20.
4. Calculate the surface area inside the object for each simulation block in the boundary. See Fig. 3.21.
5. Assign a weighted permeability value to the blocks. See Fig. 3.23.
6. Compute $\frac{\partial k_i}{\partial \alpha_j}$ for $j = 1, \ldots, 8$. Use either analytical or numerical methods to compute the derivatives.

Figure 3.22 depicts the actual object as an ellipse. Figure 3.23 depicts how the permeability is assigned to a simulation grid to represent the ellipse. Since the boundary of the object is represented as a distribution of permeability intermediate between the permeability inside and outside the object then it is possible to compute the $\frac{\partial k_i}{\partial \alpha_j}$ by first computing the derivatives of the coordinates of the points where the parametric curve crosses the grid.

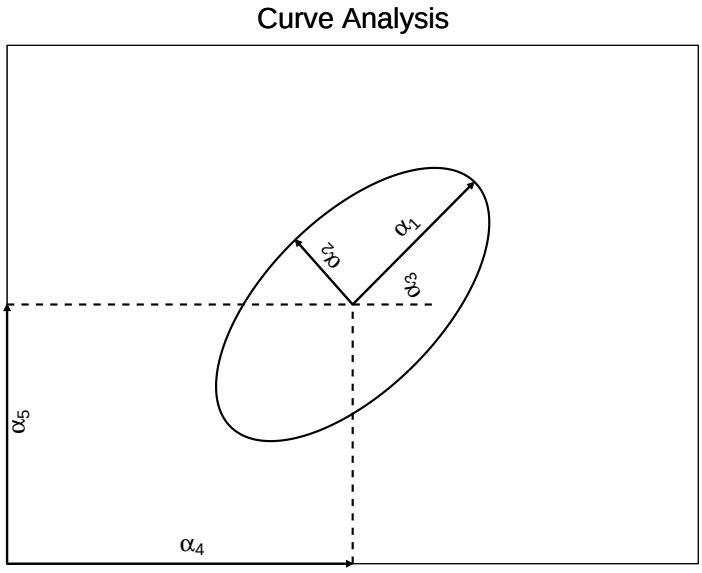



Figure 3.19: Construction of ellipse object with parametric curve.

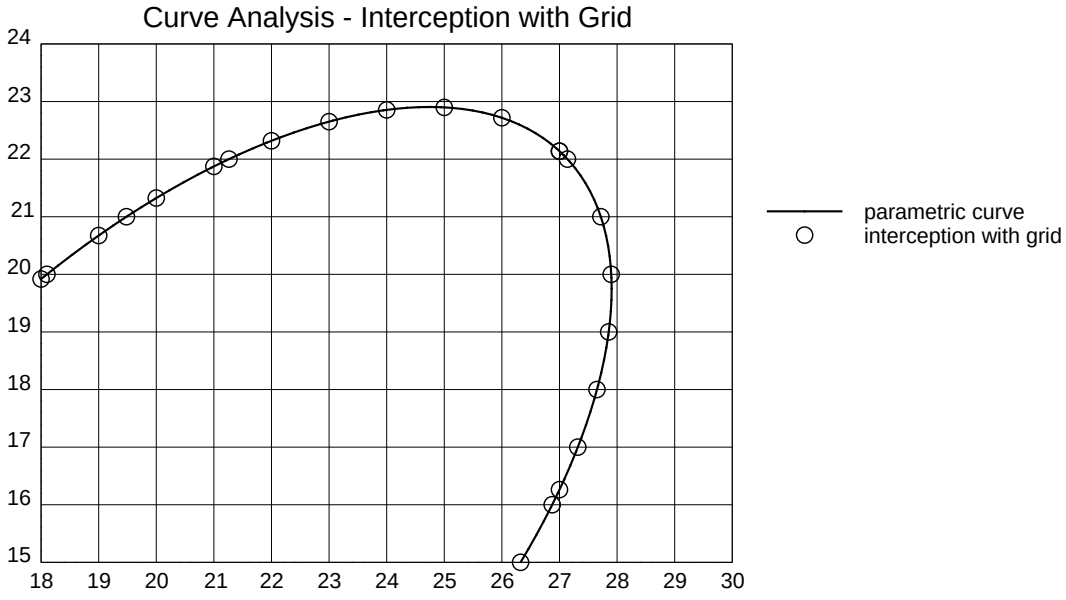



Figure 3.20: Interception of parametric curve with simulation grid

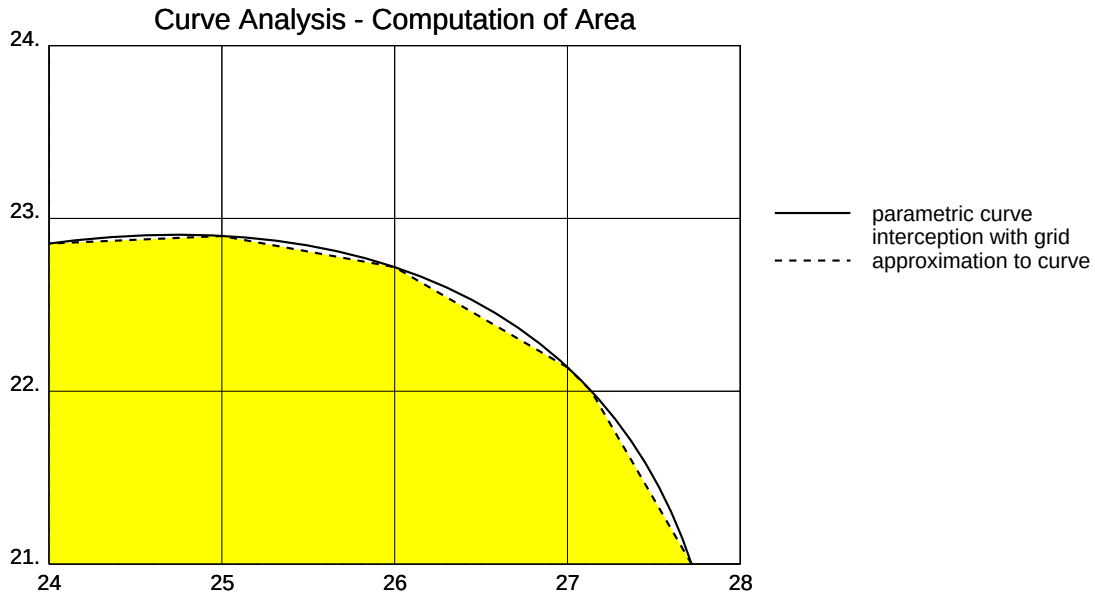



Figure 3.21: Calculation of approximated area of object.

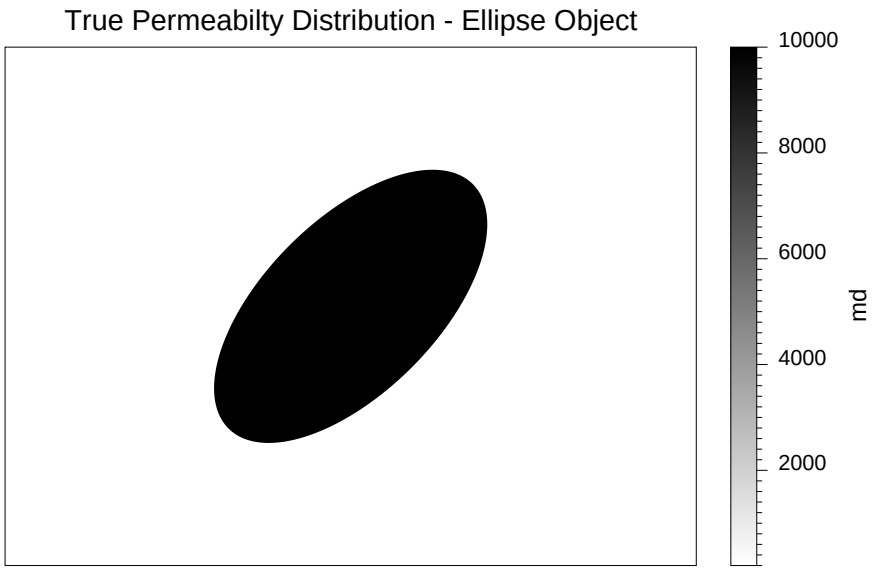



Figure 3.22: True permeability distribution for ellipse object

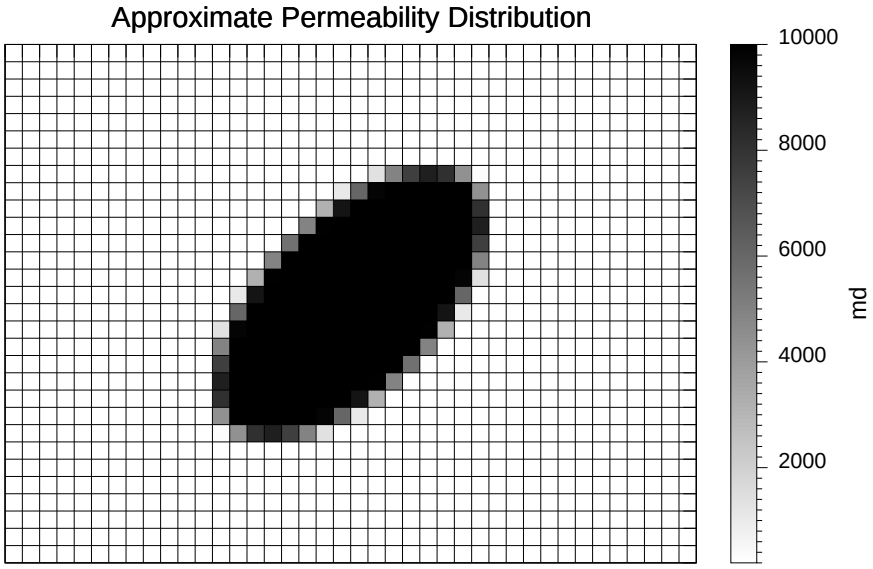



Figure 3.23: Permeability distribution for simulation and computation of $\frac{\partial k}{\partial \alpha_j}$.

The procedure used to construct the ellipse model can be used to construct an object of any shape, in this work the objects were generated by parametric curves that can be described with relatively simple analytical relationships, but in general the method can be used for more complicated cases.

One of the models that may be of interest is the representations of channels which can be used to model fluvial systems. The *channel* model can be generated using the guidelines given above. Fig. 3.24 shows an example of a procedure to parameterize the object.

The parameters that define the object channel are:

$\alpha_1$ = Permeability inside the channel.
$\alpha_2$ = Permeability outside the channel.
$\alpha_3$ = "Amplitude" of the channel
$\alpha_4$ = "Sinuosity" of the channel
$\alpha_5$ = "Width" of the channel
$\alpha_6$ = Translation in x–direction of the channel.
$\alpha_7$ = Translation in y–direction of the channel.
$\alpha_8$ = Rotation of the channel.

The discretized distribution of permeability in the simulation grid is shown in Fig. 3.25

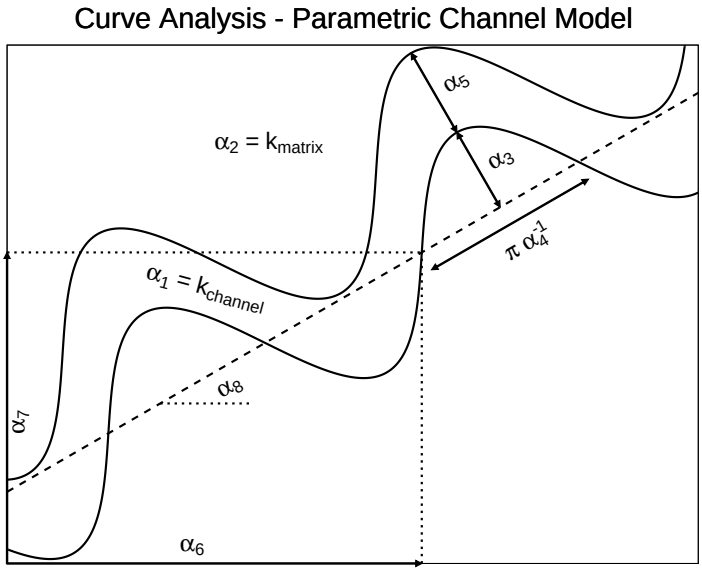



Figure 3.24: Construction of the channel model using parametric curve.

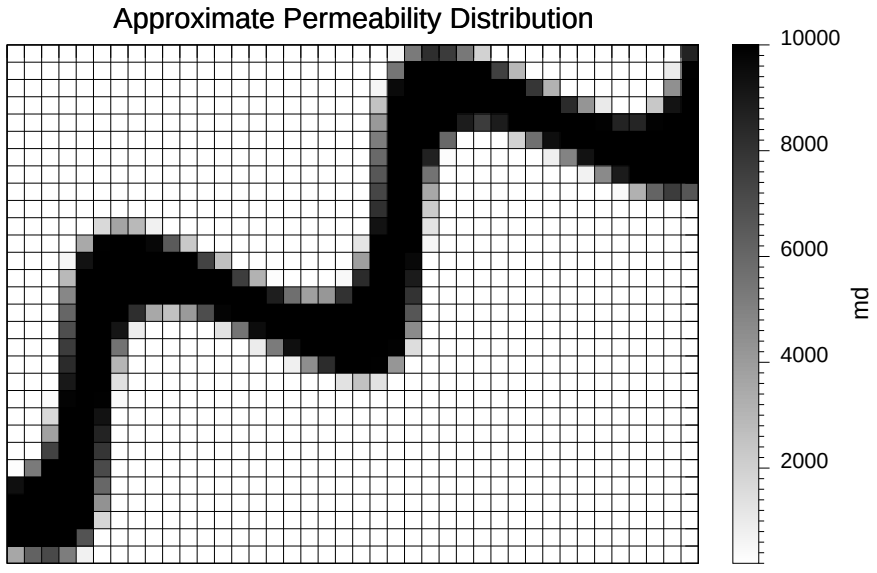



Figure 3.25: Permeability distribution for simulation and computation of $\frac{\partial k}{\partial \alpha_j}$ for channel model.

The calculation of the sensitivity coefficients, as they are computed using Equation 3.69 were compared with the calculation by the substitution method. For this purpose a numerical experiment was prepared. A two–dimensional, water–oil reservoir was modeled with the object approach, the object in this case was a channel. Fig. 3.26 depicts the geometry of the reservoir and the channel. There were three producing wells and one water injection well, well #4. The volume of the reservoir and rates of injection and production were set in such a way that allowed the water injection front to reach all the producing wells in a short period of time. This was done with the purpose of having the reservoir going through all the possible conditions. DST's test at each well were simulated at early time.

The analysis of the sensitivity was restricted to the pressure and water cut at well #2 and to the water distribution in the reservoir at $time = 200$ days. Figures 3.28 and 3.29 show the DST pressure and the long term pressure and water cut at well #2. Fig. 3.27 shows the water saturation distribution after 200 days. The water front reached well #2 at time 120 days.

The channel was parameterized as described in Fig. 3.24, that is:

$\alpha_1$ = Permeability inside the channel.
$\alpha_2$ = Permeability outside the channel.
$\alpha_3$ = "Amplitude" of the channel
$\alpha_4$ = "Sinuosity" of the channel
$\alpha_5$ = "Width" of the channel
$\alpha_6$ = Translation in x–direction of the channel.
$\alpha_7$ = Translation in y–direction of the channel.
$\alpha_8$ = Rotation of the channel.
$\alpha_9$ = Skin factor at well #1.
$\alpha_{10}$ = Skin factor at well #2.
$\alpha_{11}$ = Skin factor at well #3.
$\alpha_{12}$ = Skin factor at well #4.
$\phi_i^0$ = $\phi_i^0(k_i)$

Figures 3.30 through 3.69 show the comparison of the sensitivity coefficients as calculated with the method derived from Equation 3.69 and with the substitution method, which was used as a benchmark. The analysis was performed for the case of the parameters $\alpha_1 - \alpha_8$. The analysis for the case of the skin factor is not shown since this was done earlier in Section 3.2.1. The magnitude of the sensitivity of the long term pressure and water cut was investigated as shown in Figures 3.70 and 3.71. Observations, remarks and conclusions follow:

1. The method used in this work, Equation 3.69, computed the sensitivity coefficients accurately.

2. The wiggles in some plots corresponding to the substitution method are due to numerical rounding.

3. The sensitivity of the water cut with respect to all the parameters peaked shortly after the arrival of water front.

4. The largest sensitivity coefficients are the ones corresponding to $\alpha_4$, the "sinuosity" of the channel. This is to be expected since small variations in $\alpha_4$ result in large movements of the channel.

As in the case of the pixel model, the efficiency of the method to compute sensitivity coefficients was compared with the substitution method. The results follow:

| *Mode* | *User time (seconds)* |
|---|---|
| To run simulation + 8 sensitivities | 213.00 |
| To run simulation only | 146.62 |

From these results we can evaluate the efficiency of the application of Equation 3.36 against the substitution method, which is used as a benchmark. The results follow:

| *Mode* | *User time (seconds)* |
|---|---|
| Use Equation 3.69 to compute 8 sensitivities | 213.00 |
| Use substitution method to compute 8 sensitivities | 1319.58 |

Thus we can conclude that, for this specific case, the method used in this work is approximately 6.2 times faster than the substitution method. As noted in the analysis of the computation of the sensitivity coefficients with the pixel method, the 6.2 efficiency has to be looked in the context of the problem, that is the work of 6.2 days can be done in one day. Another observation is that the linear search may have a negative effect since the computation of a new direction for descent is of the same order of magnitude as the function evaluation (simulation run). Also the choice of solver for solving the multiple right hand side problem (Equation 3.69) has an impact in the overall efficiency. The most remarkable aspect is the fact that computation of the sensitivity requires only the same order of magnitude of computer time as the simulation run, which translates to a very fast process for solving the inverse problem.

### 3.3.4 Comparison of Pixel and Object Methods

The advantage, from the speed point of view, of the object approach over the pixel approach can be evaluated as follows. Let's say we try to solve a problem such as the reservoir with a channel with the pixel method and then compare to the same problem solved with the object approach. Assuming that the reservoir in both approaches is discretized with 1200 cells and that the porosity is related to permeability through a known relationship, then we will have the following computer time comparison for the two approaches to compute the sensitivity coefficients *only*:

| *Mode* | *User time (seconds)* |
|---|---|
| Object Model (8 parameters) | 213.00 |
| Pixel Model (1200 parameters) | 2679.40 |

We see that the object model is approximately 12 times faster than the pixel model approach in computing the sensitivity coefficients. A fast computation of a descent direction is not the only advantage of the *object* model approach. The number of iterations in the pixel approach is of the same order of magnitude as the number of parameters so the overall performance of the object approach will be of the order 1000 times faster than the pixel model approach. As was mentioned earlier the object

model approach also does a better job in preserving large scale geological information.

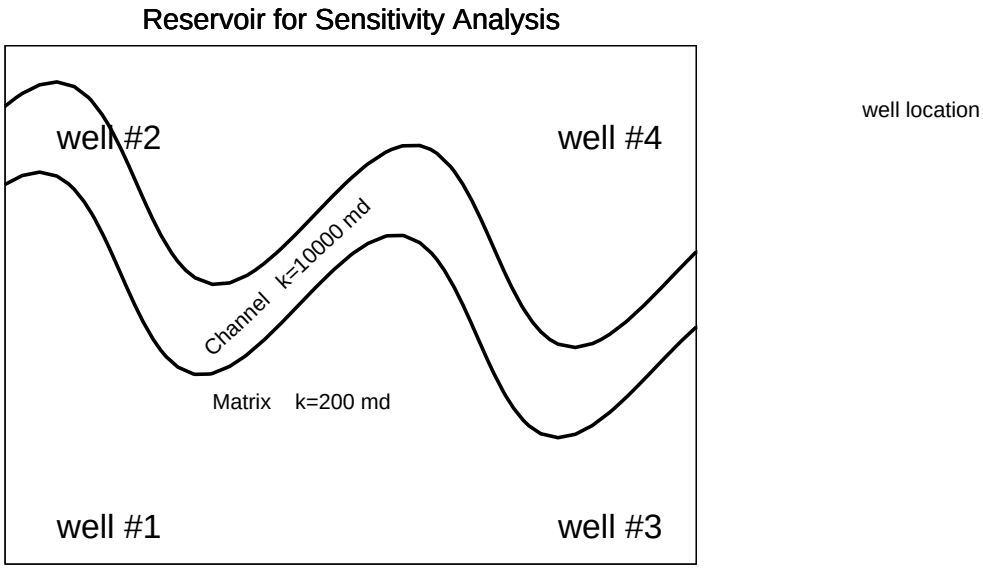



Figure 3.26: Object Method - Reservoir for analysis of the method to compute the sensitivity coefficients.

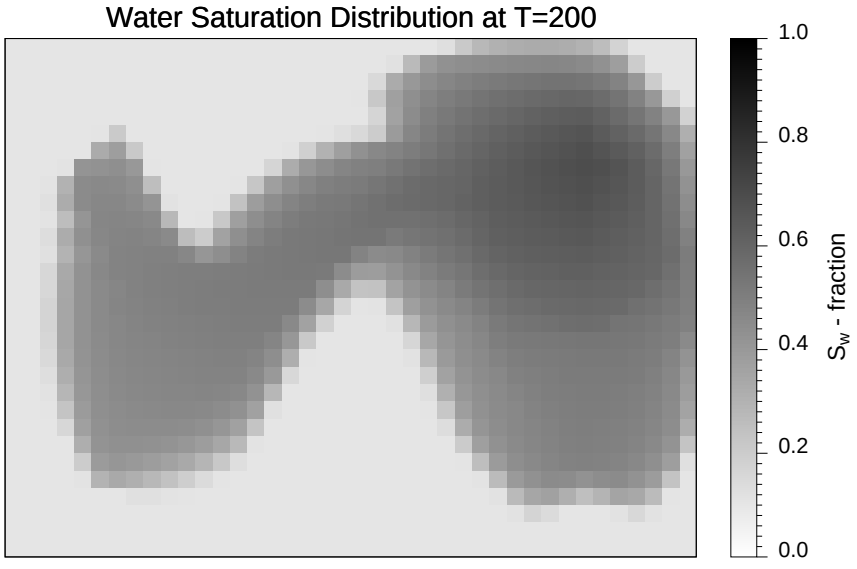



Figure 3.27: Water saturation distribution in the reservoir at time T=200.

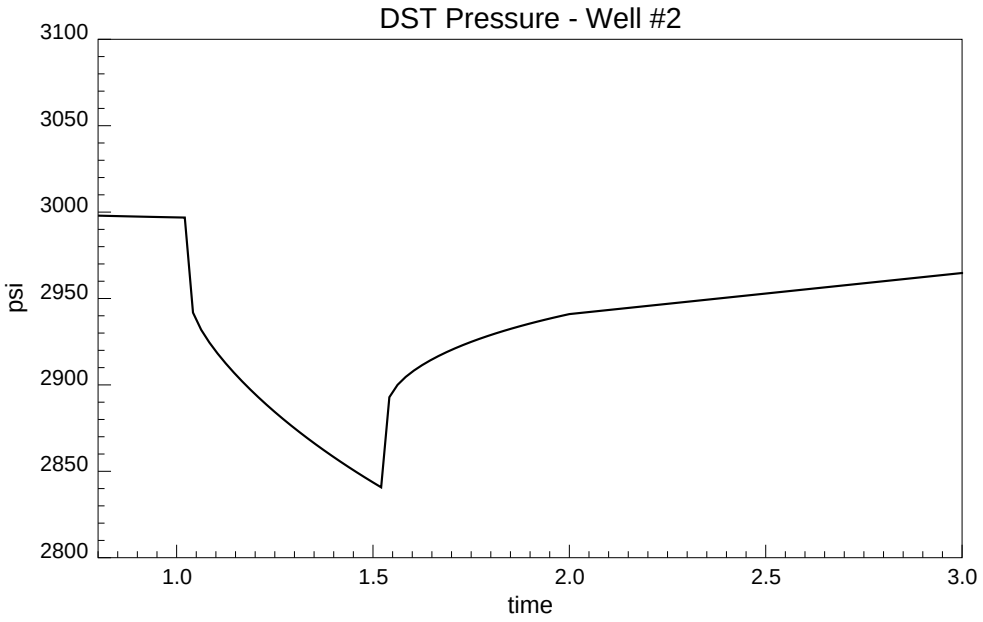



Figure 3.28: Well #2 – DST pressure.

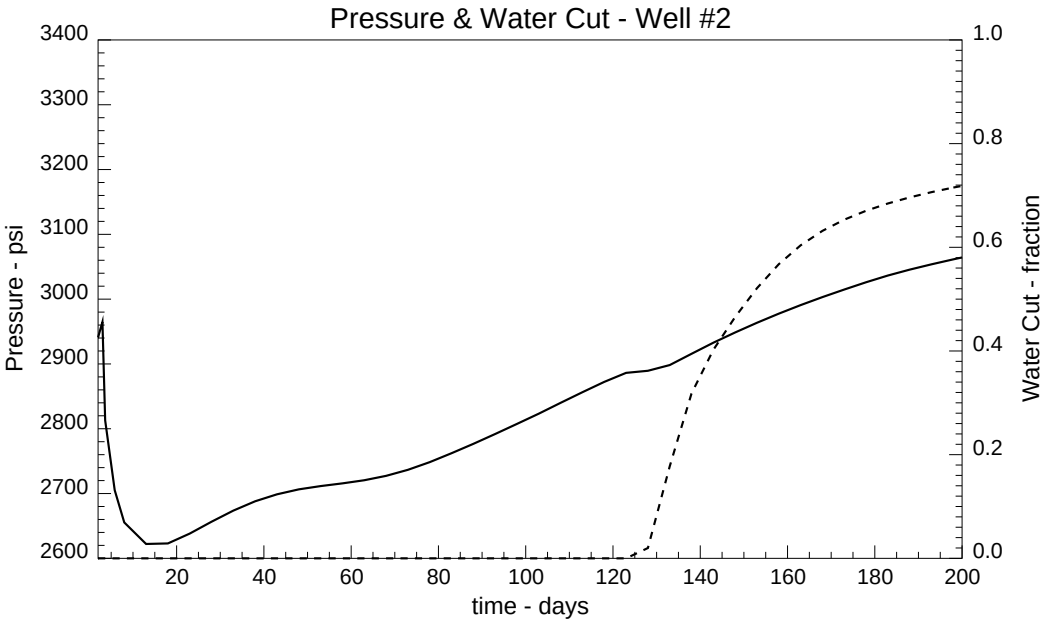



Figure 3.29: Well #2 – Long term pressure and water cut.

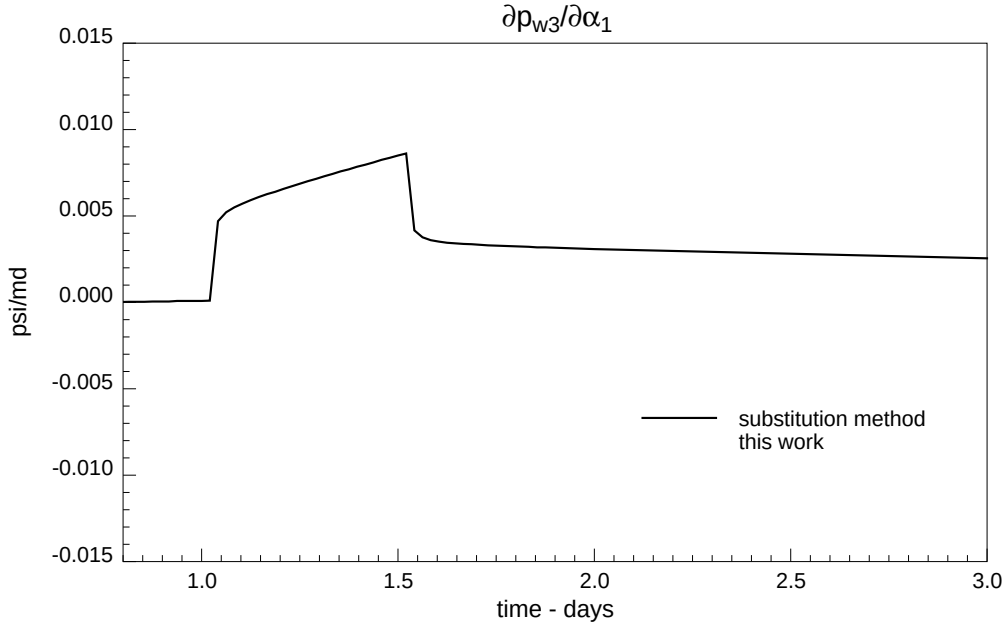



Figure 3.30: Sensitivity of DST–pressure in well #2 with respect to $\alpha_1$ (Permeability inside the channel).

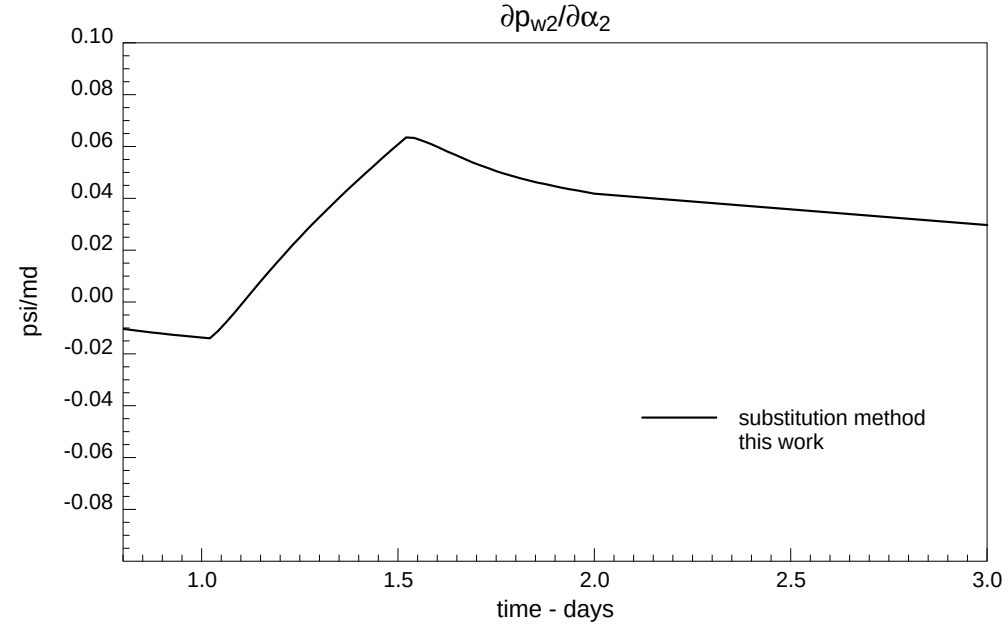



Figure 3.31: Sensitivity of DST–pressure in well #2 with respect to $\alpha_2$ (Permeability outside the channel).

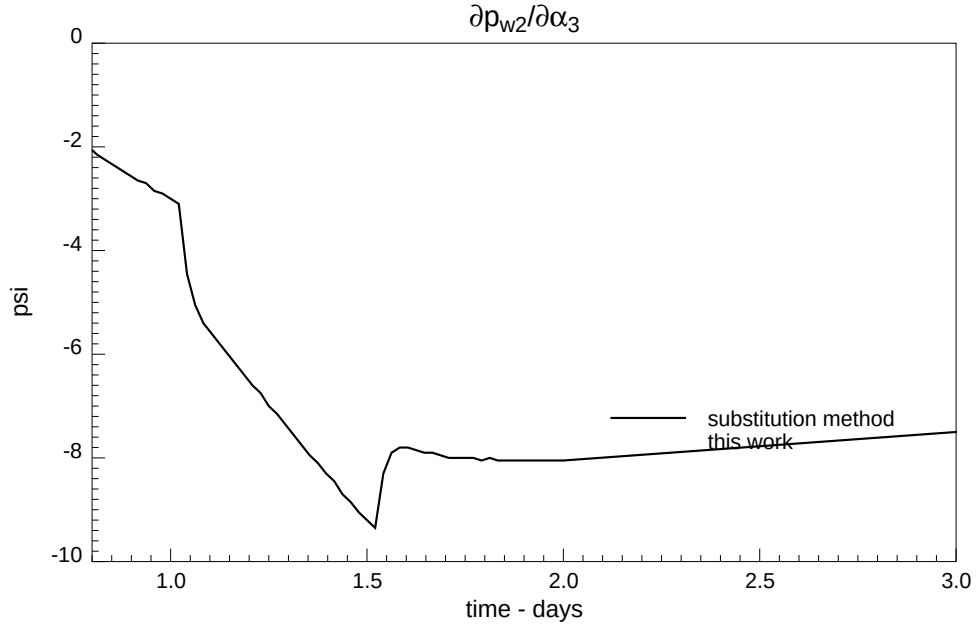



Figure 3.32: Sensitivity of DST–pressure in well #2 with respect to $\alpha_3$ ("Amplitude" of the channel).

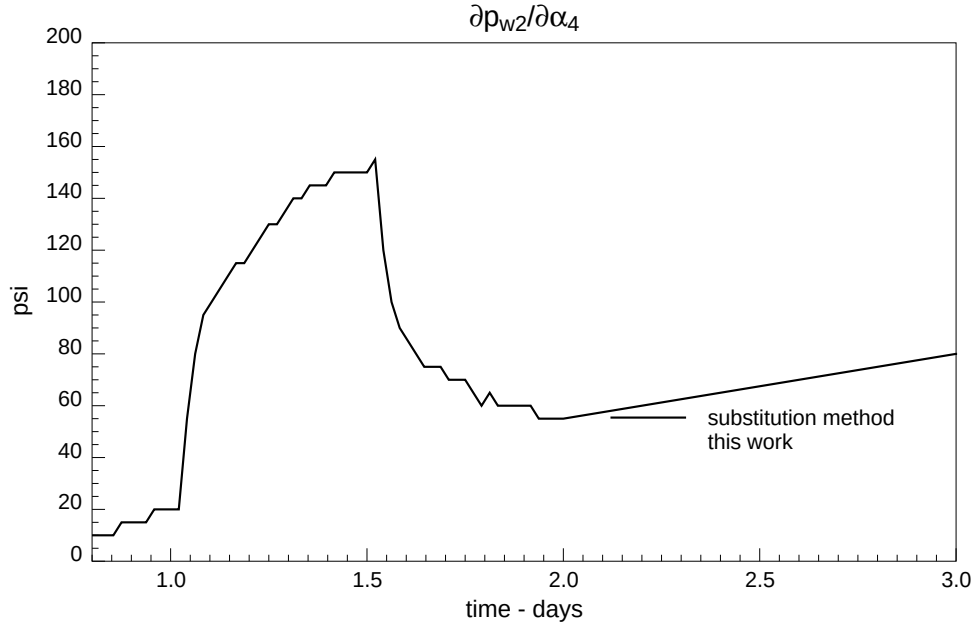



Figure 3.33: Sensitivity of DST–pressure in well #2 with respect to $\alpha_3$ ("Sinuosity" of the channel).

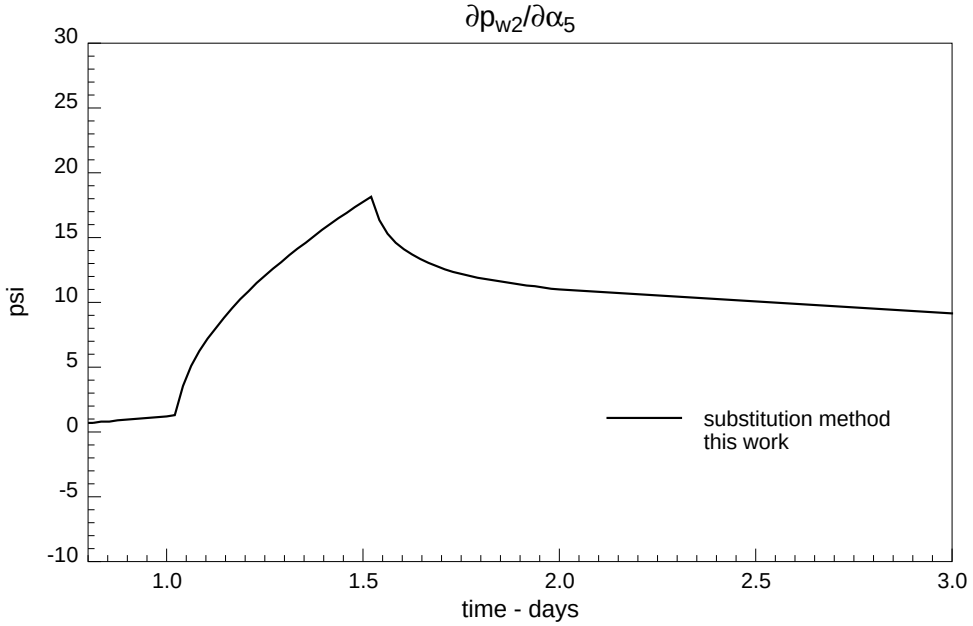



Figure 3.34: Sensitivity of DST–pressure in well #2 with respect to $\alpha_5$ ("Width" of the channel).

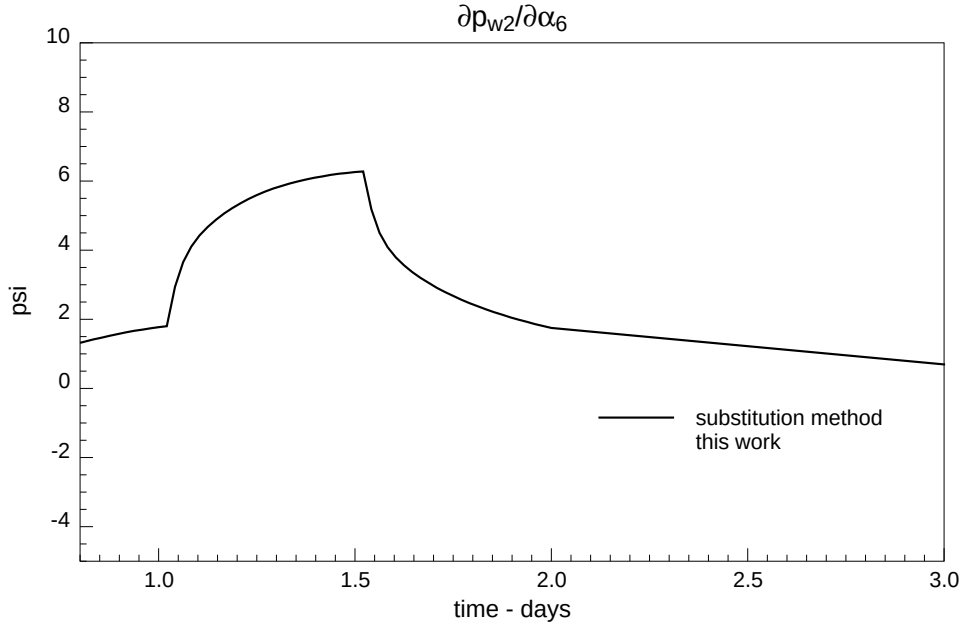



Figure 3.35: Sensitivity of DST–pressure in well #2 with respect to $\alpha_6$ ("Translation x–direction" of the channel).

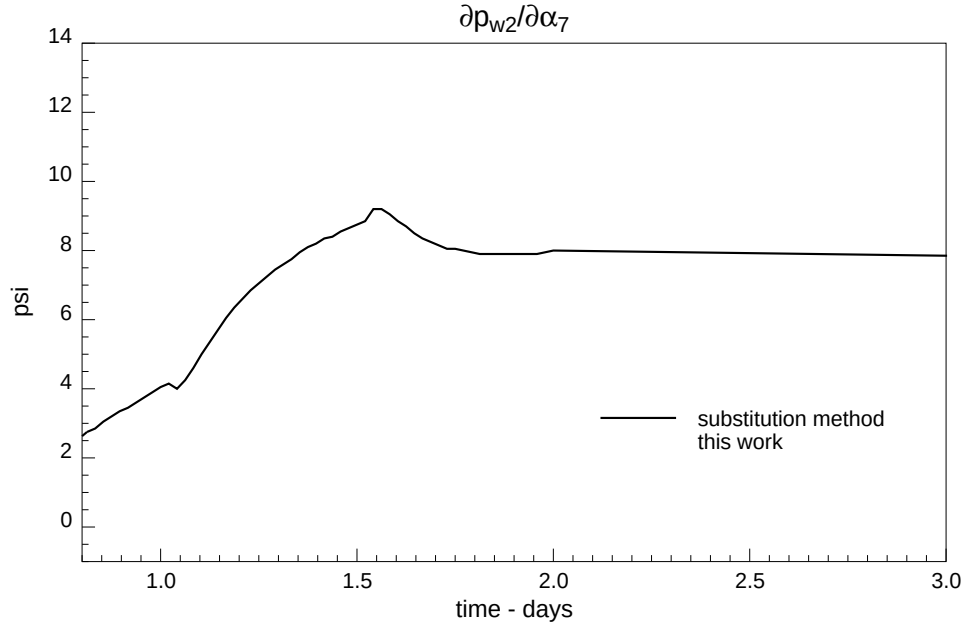



Figure 3.36: Sensitivity of DST–pressure in well #2 with respect to $\alpha_7$ ("Translation y–direction" of the channel).

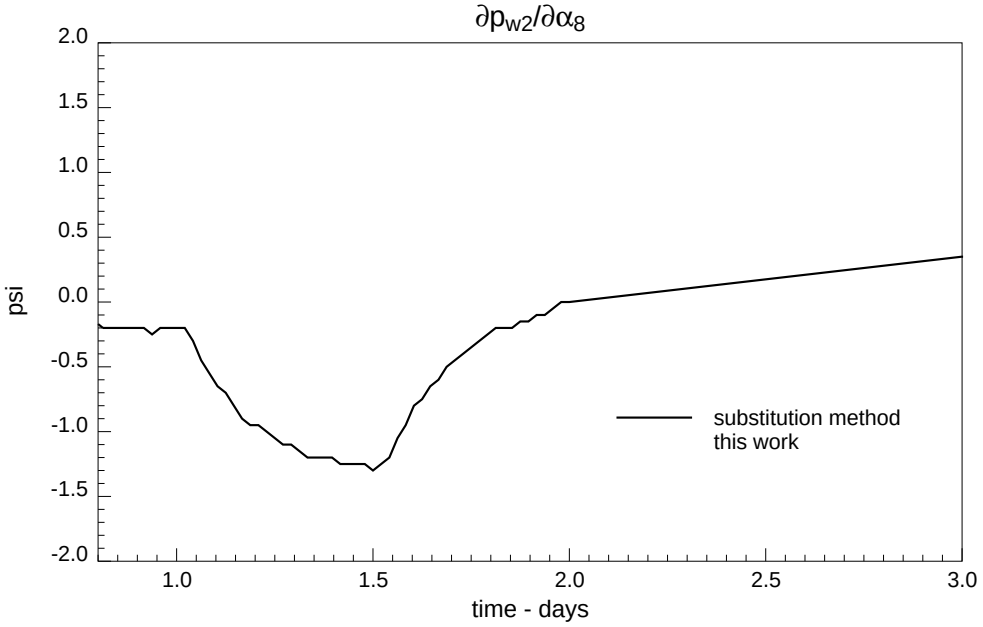



Figure 3.37: Sensitivity of DST–pressure in well #2 with respect to $\alpha_8$ ("Rotation" of the channel).

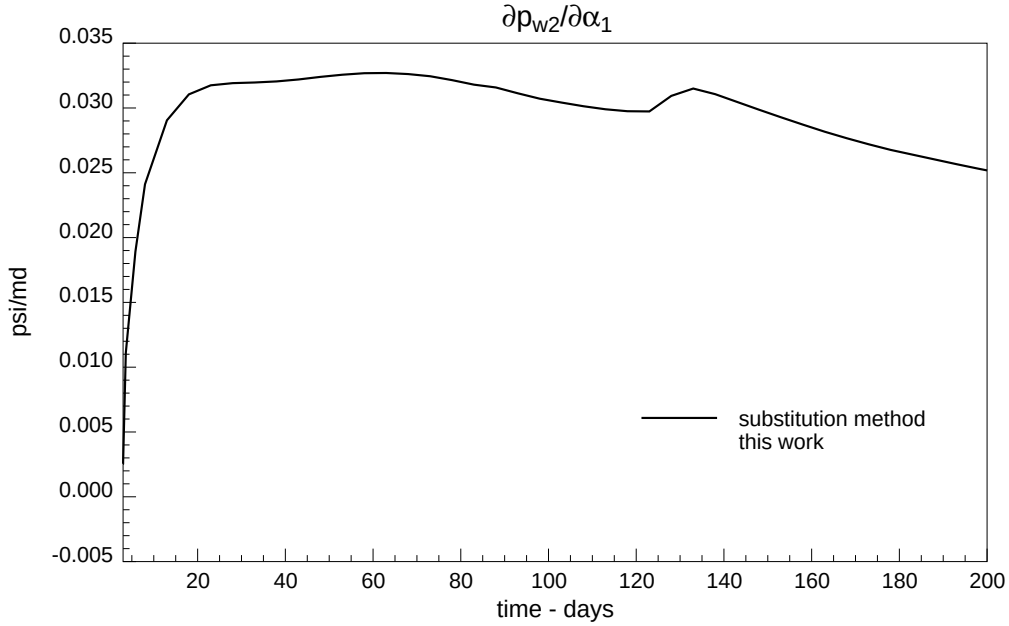



Figure 3.38: Sensitivity of long term pressure in well #2 with respect to $\alpha_1$ (Permeability inside the channel).

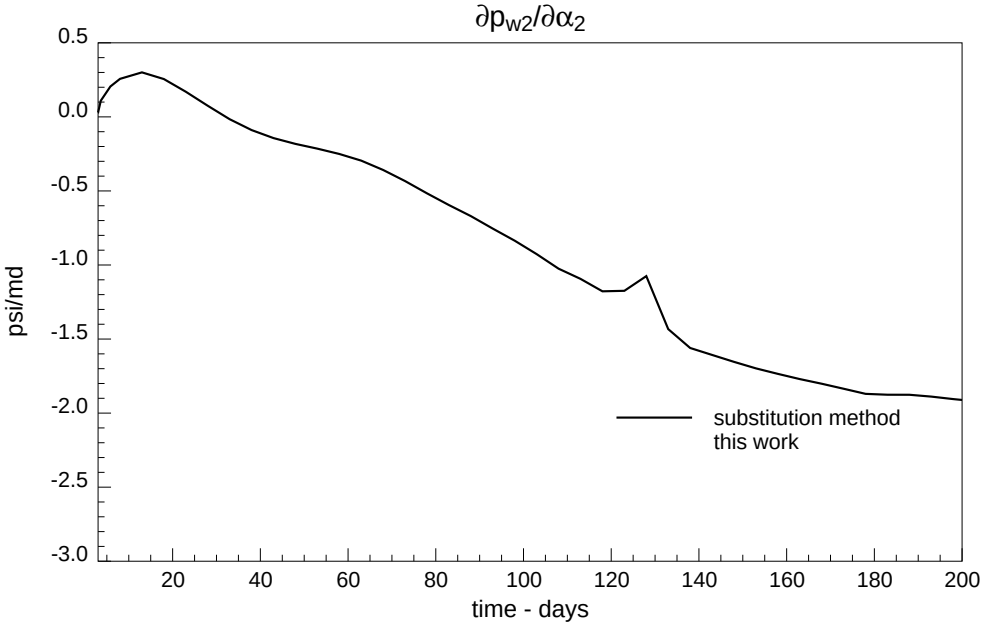



Figure 3.39: Sensitivity of long term pressure in well #2 with respect to $\alpha_2$ (Permeability outside the channel).

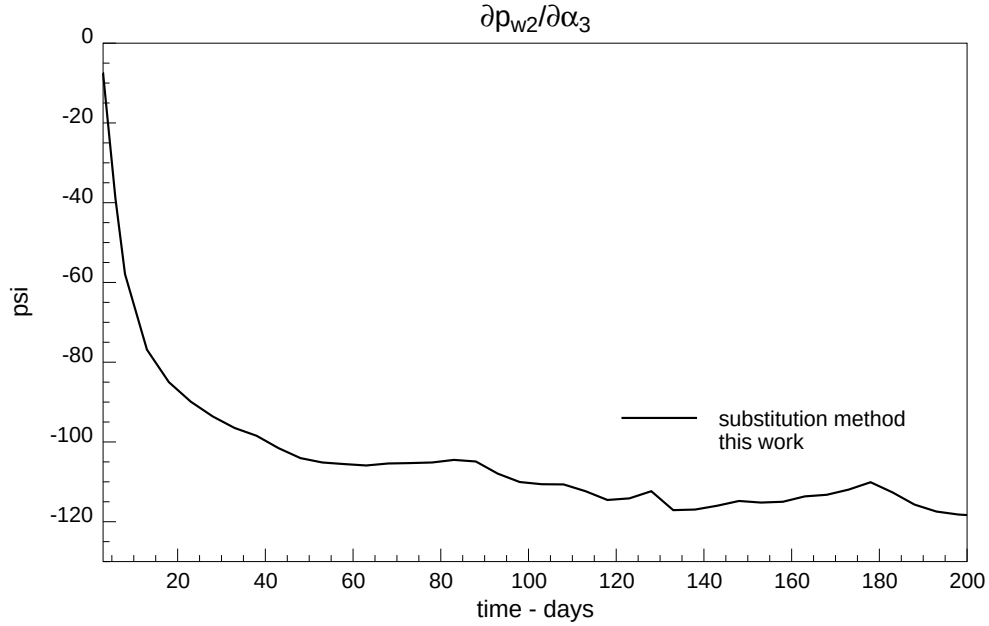



Figure 3.40: Sensitivity of long term pressure in well #2 with respect to $\alpha_3$ ("Amplitude" of the channel).

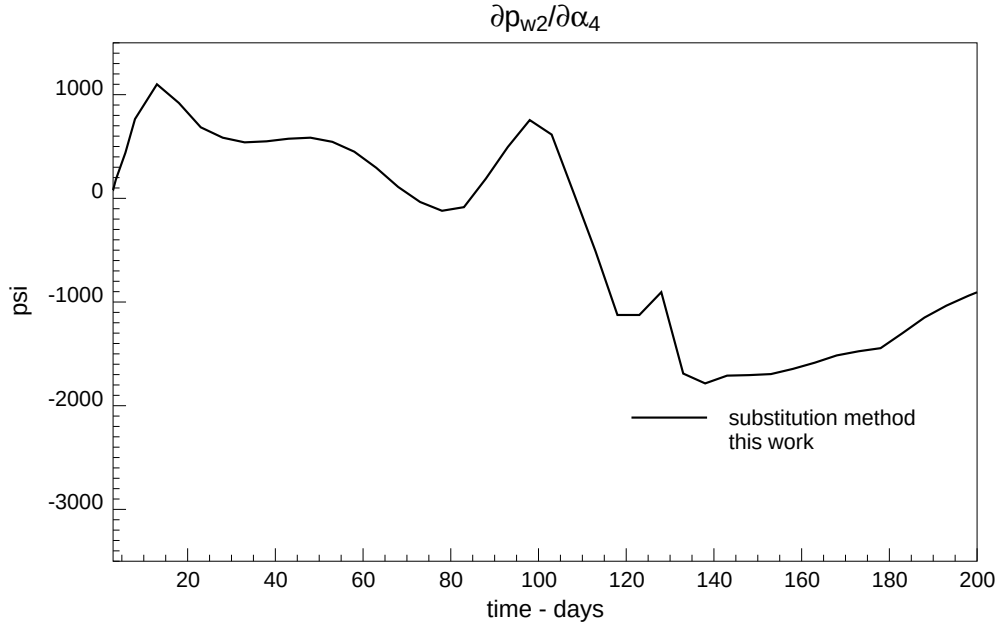



Figure 3.41: Sensitivity of long term pressure in well #2 with respect to $\alpha_4$ ("Sinuosity" of the channel).

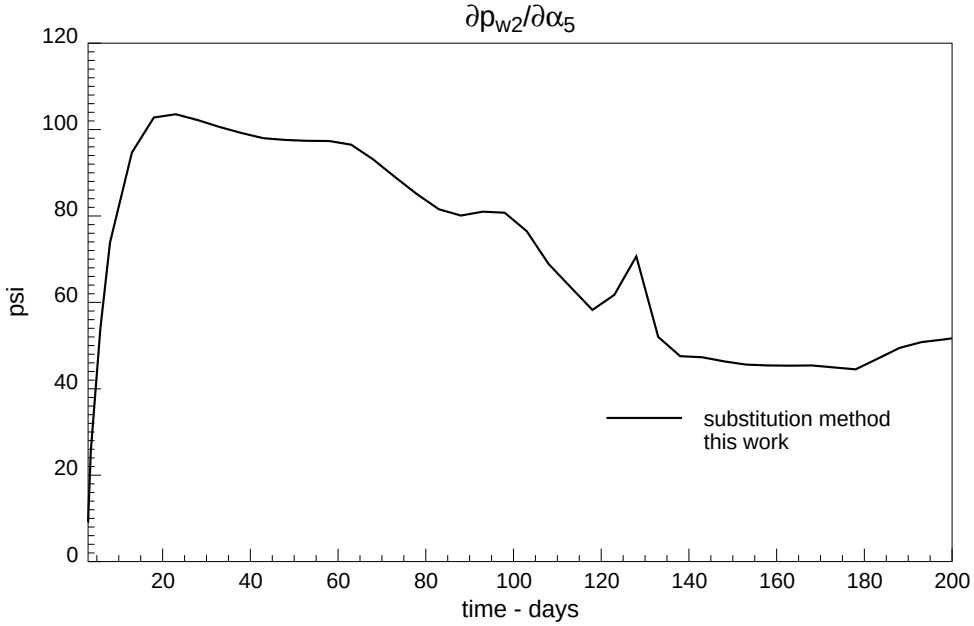



Figure 3.42: Sensitivity of long term pressure in well #2 with respect to $\alpha_5$ ("Width" of the channel).

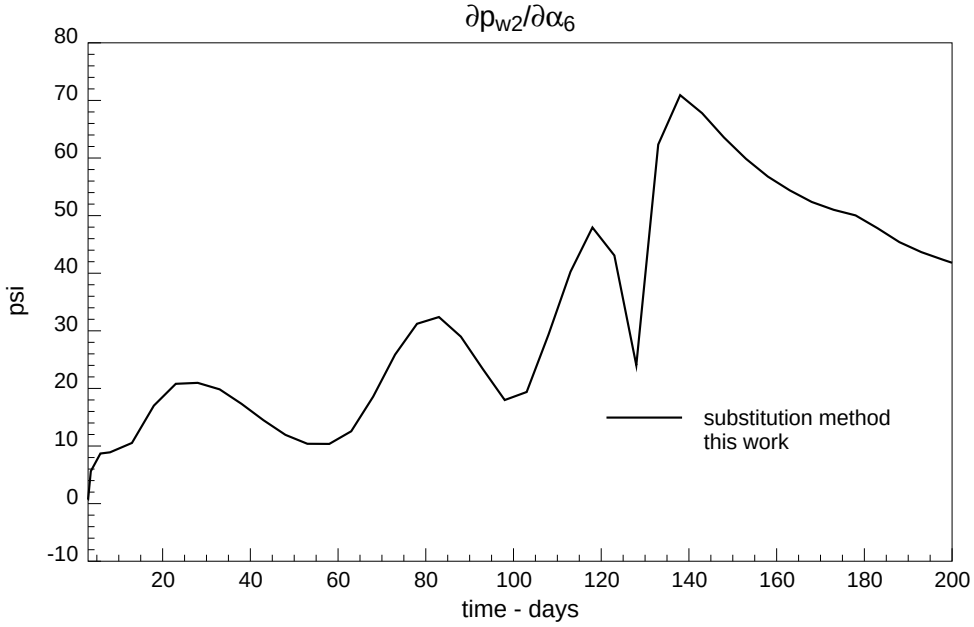



Figure 3.43: Sensitivity of long term pressure in well #2 with respect to $\alpha_6$ ("Translation x–direction" of the channel).

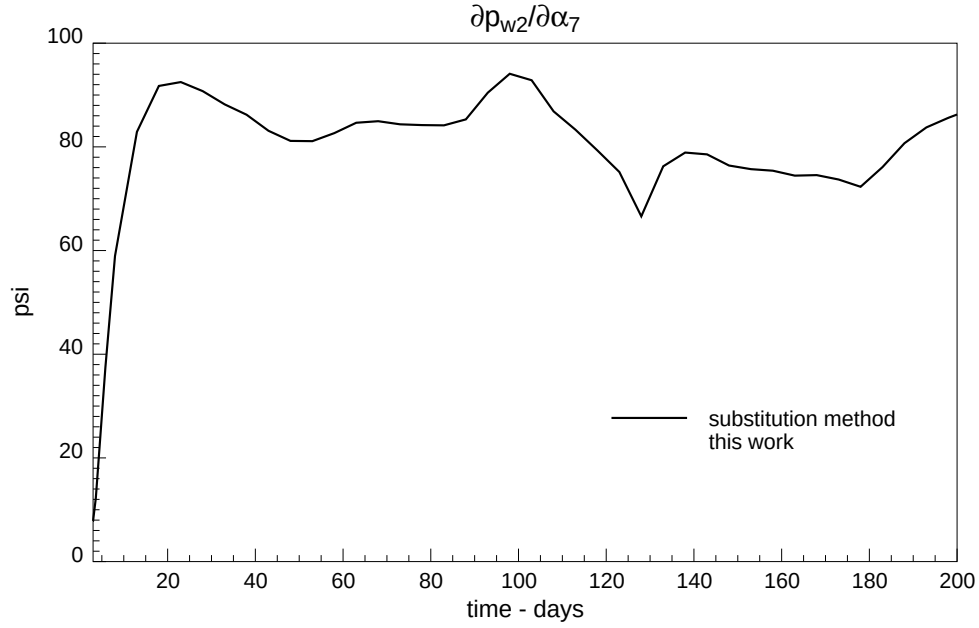



Figure 3.44: Sensitivity of long term pressure in well #2 with respect to $\alpha_7$ ("Translation y–direction" of the channel).

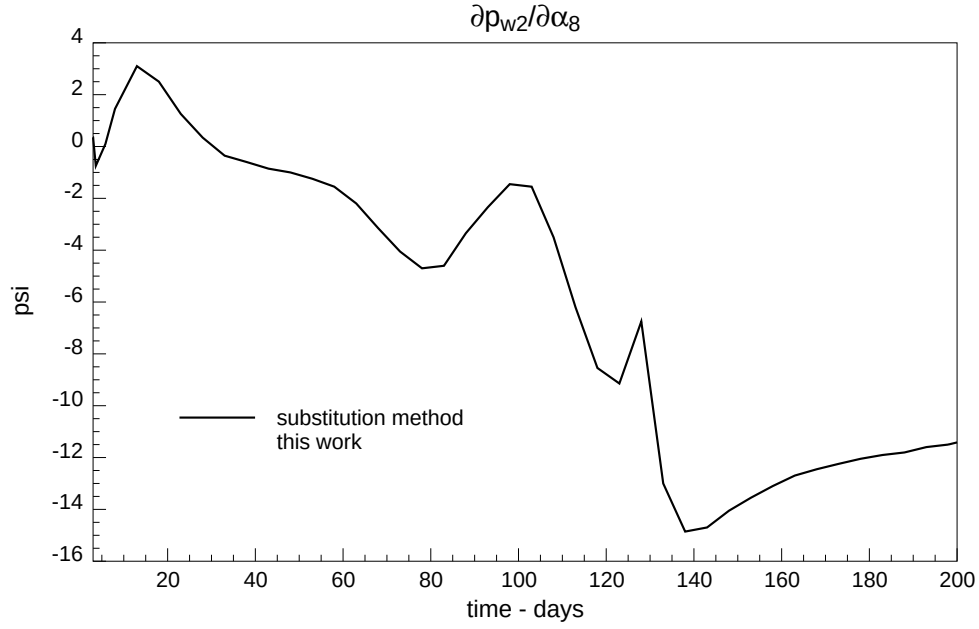



Figure 3.45: Sensitivity of long term pressure in well #2 with respect to $\alpha_8$ ("Rotation" of the channel).

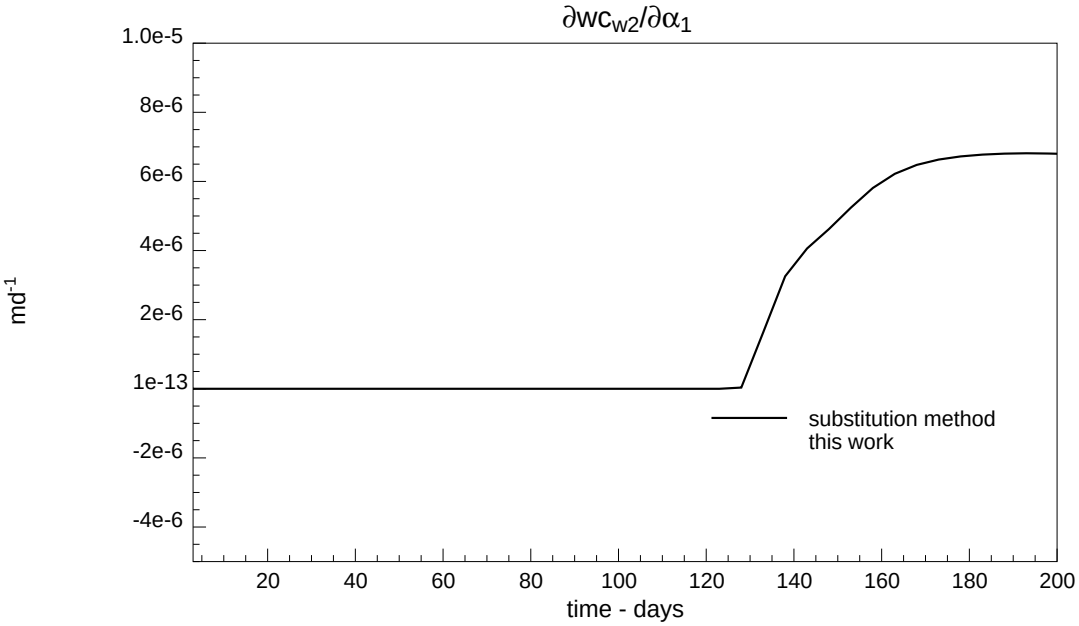



Figure 3.46: Sensitivity of the water cut in well #2 with respect to $\alpha_1$ (Permeability inside the channel).

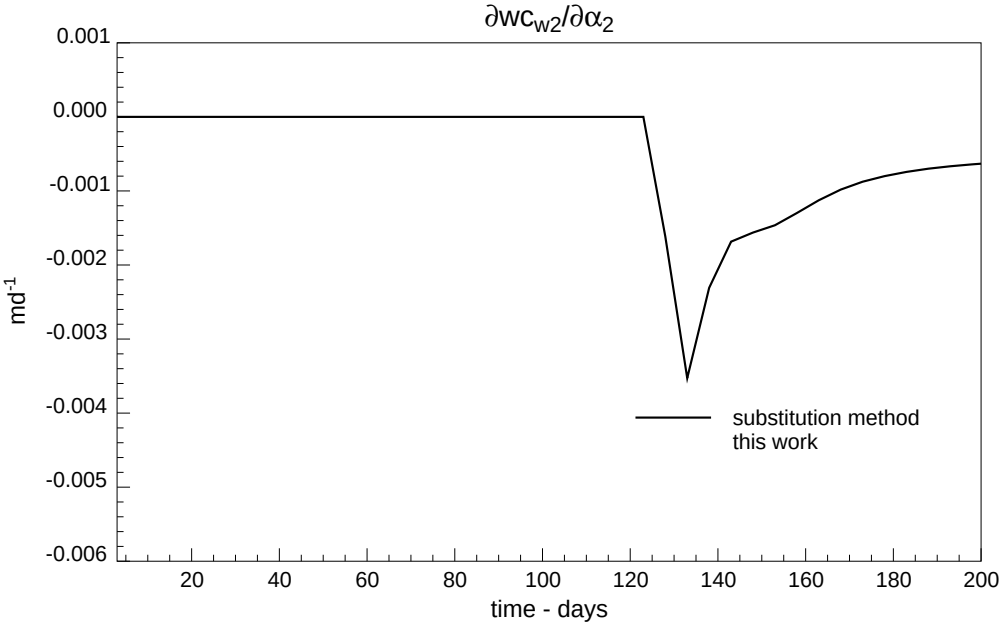



Figure 3.47: Sensitivity of the water cut in well #2 with respect to $\alpha_2$ (Permeability outside the channel).

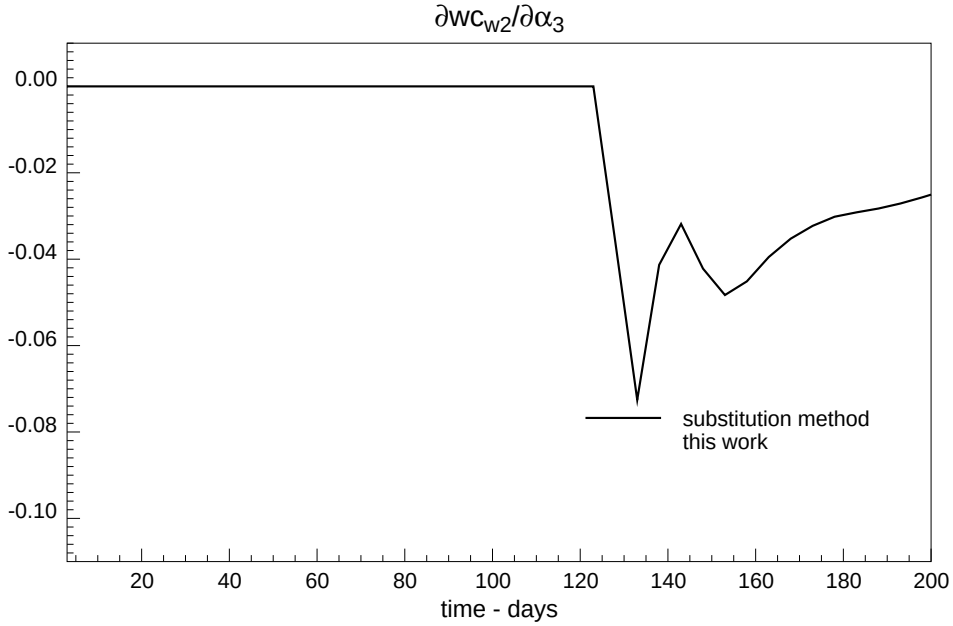



Figure 3.48: Sensitivity of the water cut in well #2 with respect to $\alpha_3$ ("Amplitude" of the channel).

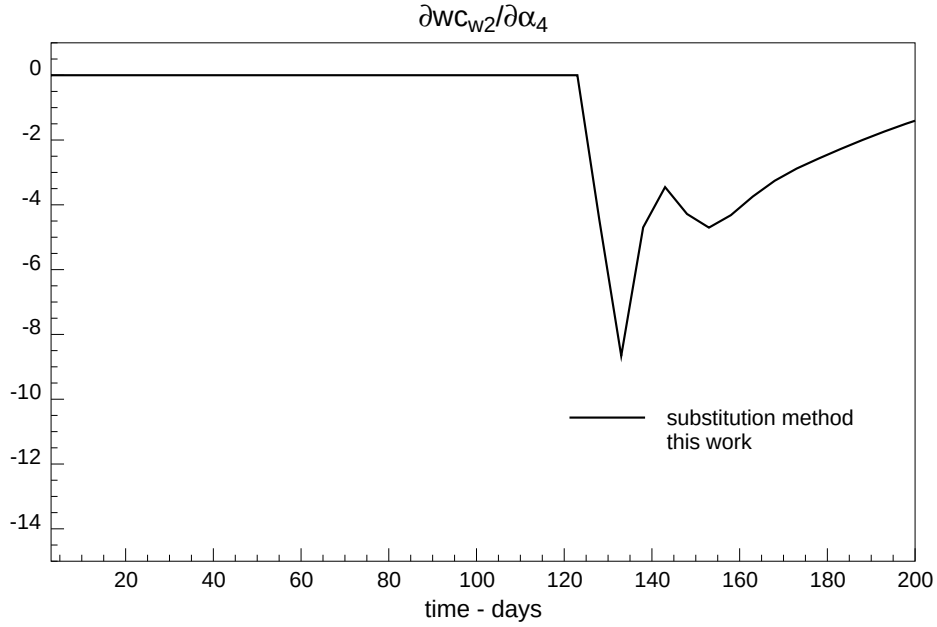



Figure 3.49: Sensitivity of the water cut in well #2 with respect to $\alpha_4$ ("Sinuosity" of the channel).

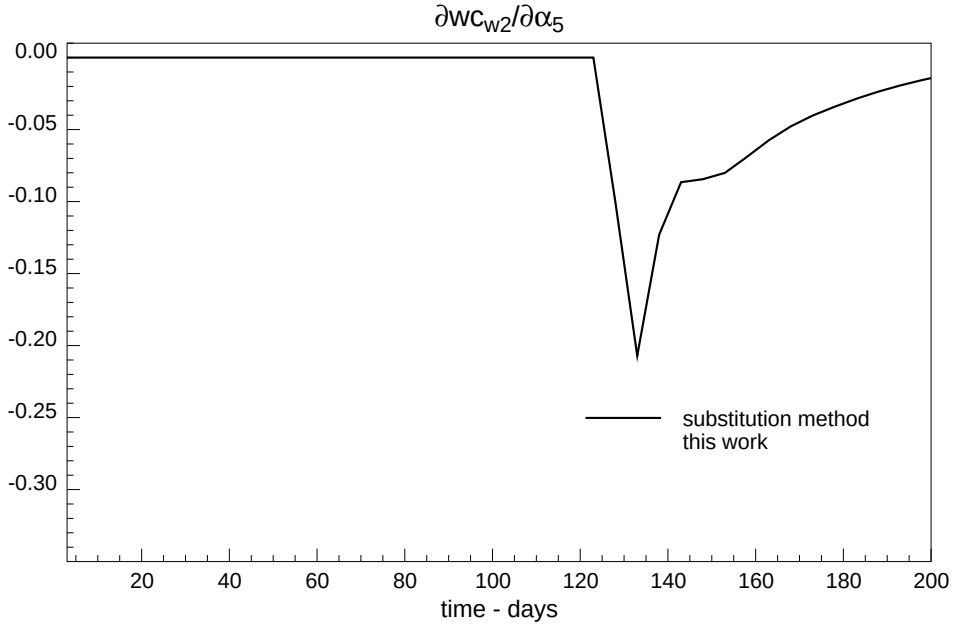



Figure 3.50: Sensitivity of the water cut in well #2 with respect to $\alpha_5$ ("Width" of the channel).

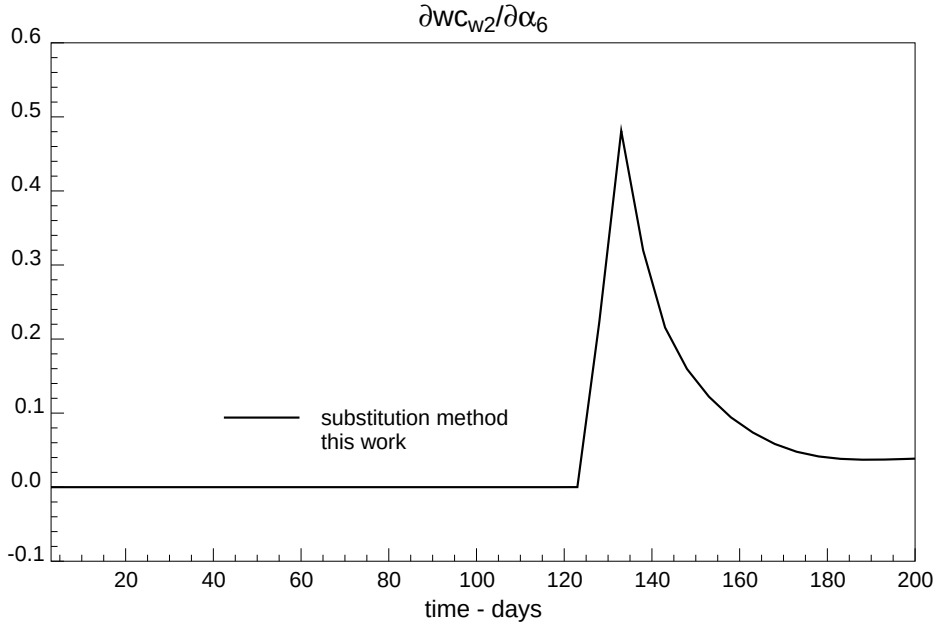



Figure 3.51: Sensitivity of the water cut in well #2 with respect to $\alpha_6$ ("Translation x–direction" of the channel).

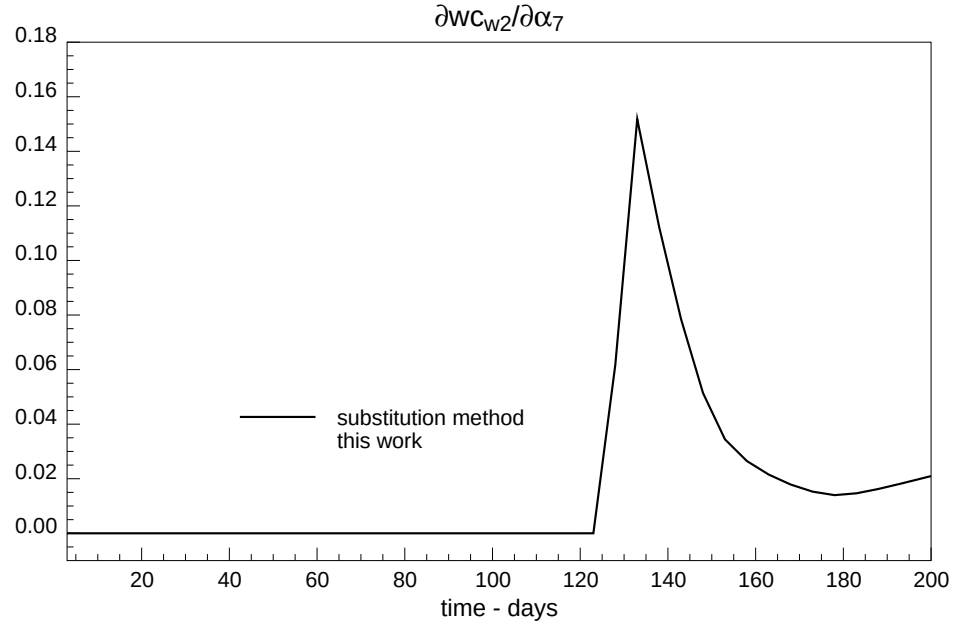



Figure 3.52: Sensitivity of the water cut in well #2 with respect to $\alpha_7$ ("Translation y–direction" of the channel).

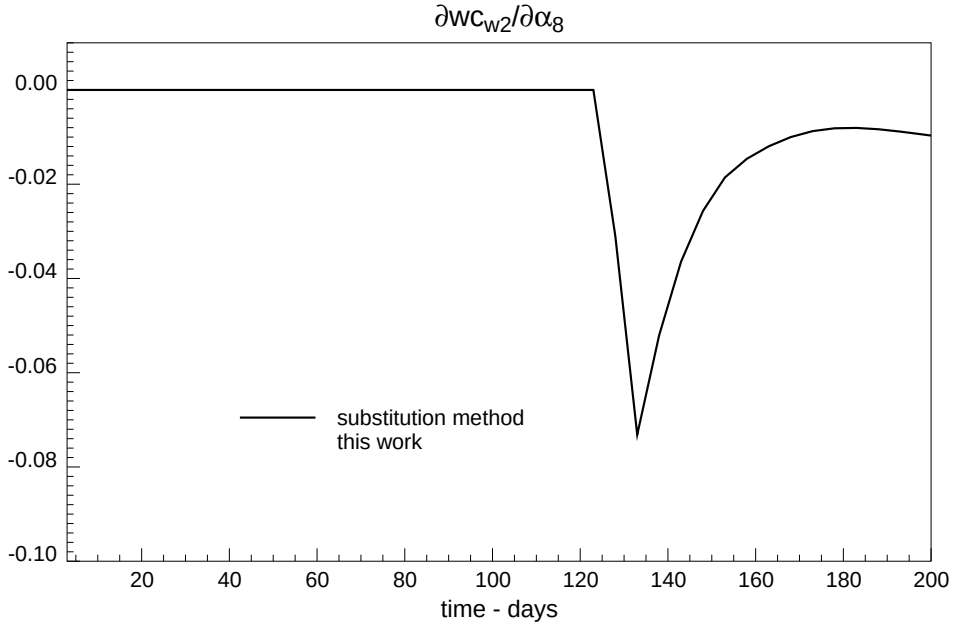



Figure 3.53: Sensitivity of the water cut in well #2 with respect to $\alpha_8$ ("Rotation" of the channel).

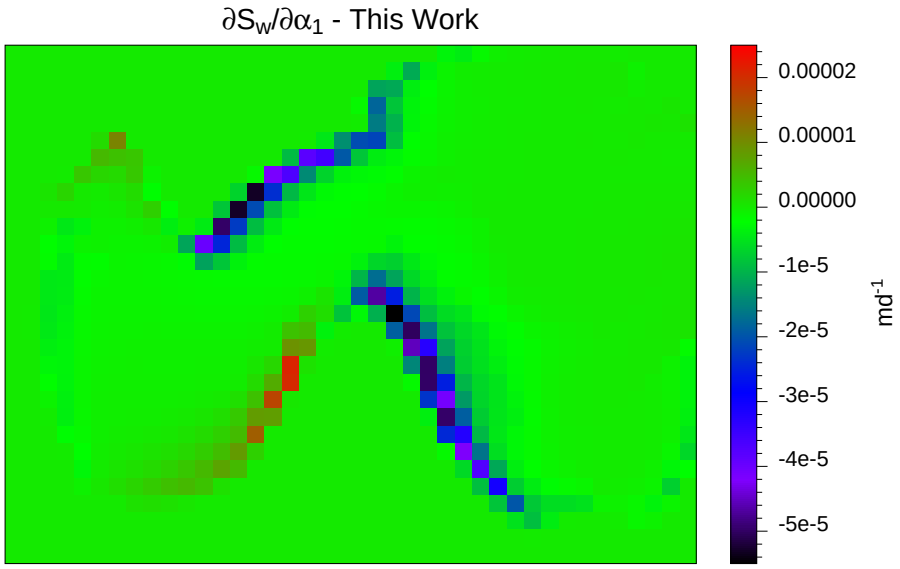



Figure 3.54: Sensitivity of water saturation distribution with respect to $\alpha_1$ (Permeability inside the channel), as calculated by this work.

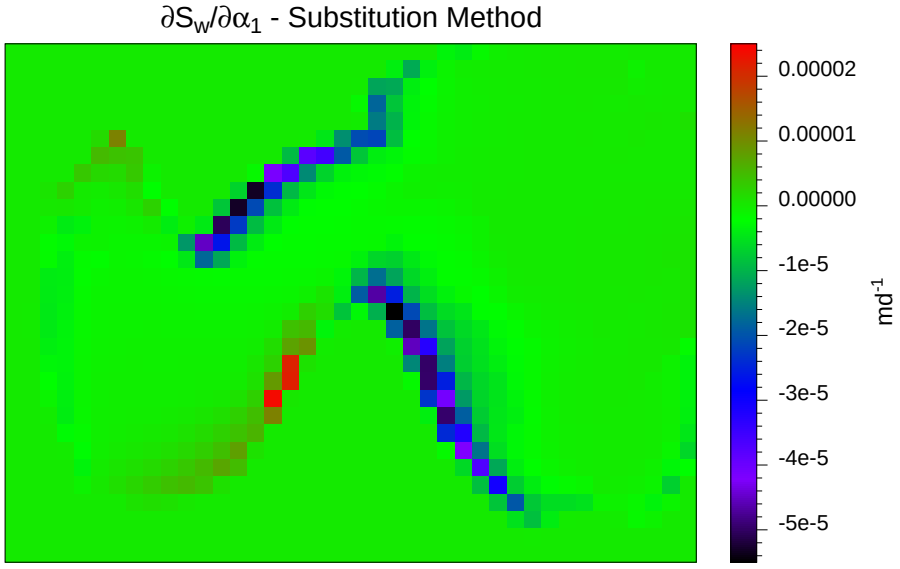



Figure 3.55: Sensitivity of water saturation distribution with respect to $\alpha_1$ (Permeability inside the channel), as calculated by substitution method.

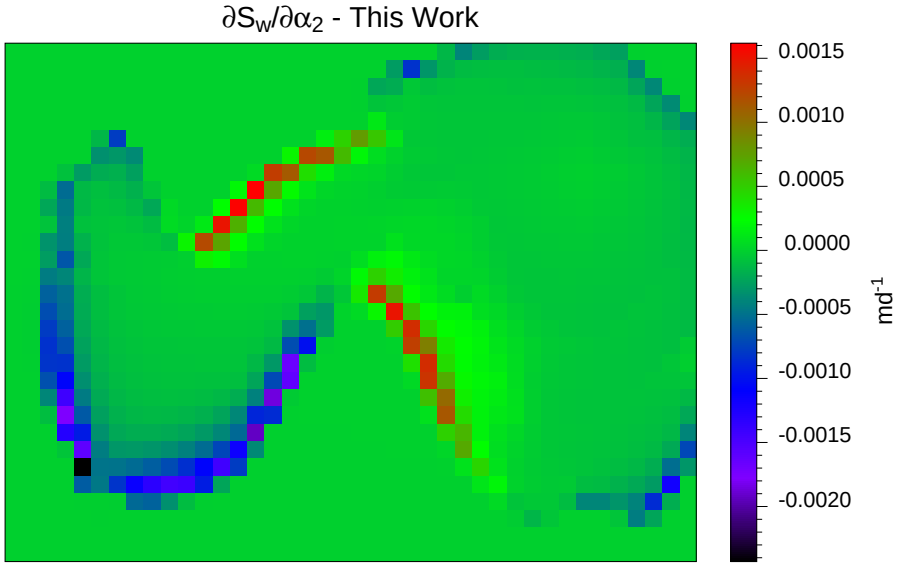



Figure 3.56: Sensitivity of water saturation distribution with respect to $\alpha_2$ (Permeability outside the channel), as calculated by this work.

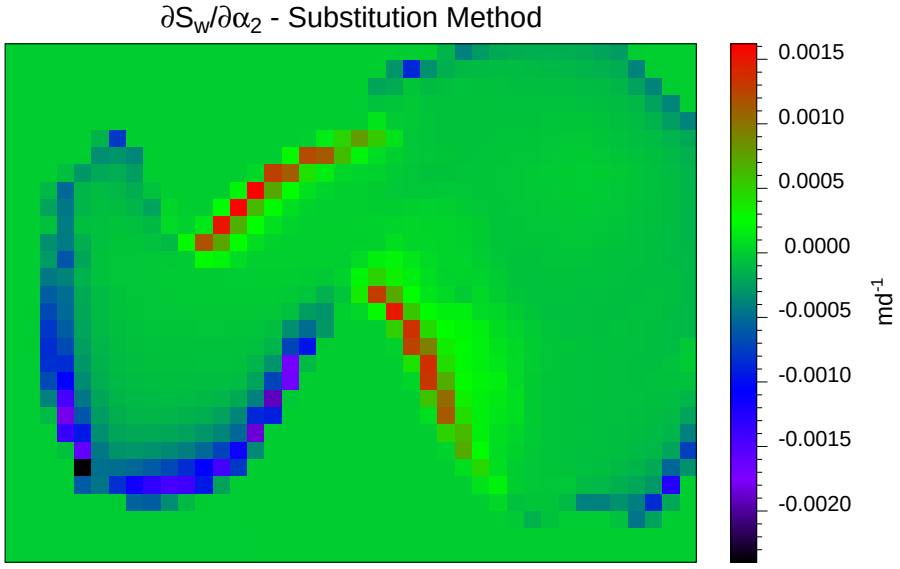



Figure 3.57: Sensitivity of water saturation distribution with respect to $\alpha_2$ (Permeability outside the channel), as calculated by substitution method.

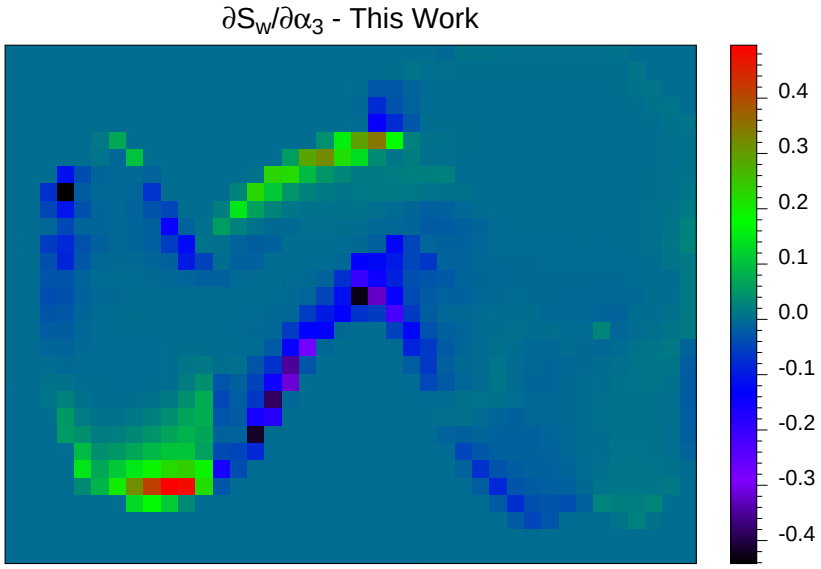



Figure 3.58: Sensitivity of water saturation distribution with respect to $\alpha_3$ ("Amplitude" of channel), as calculated by this work.

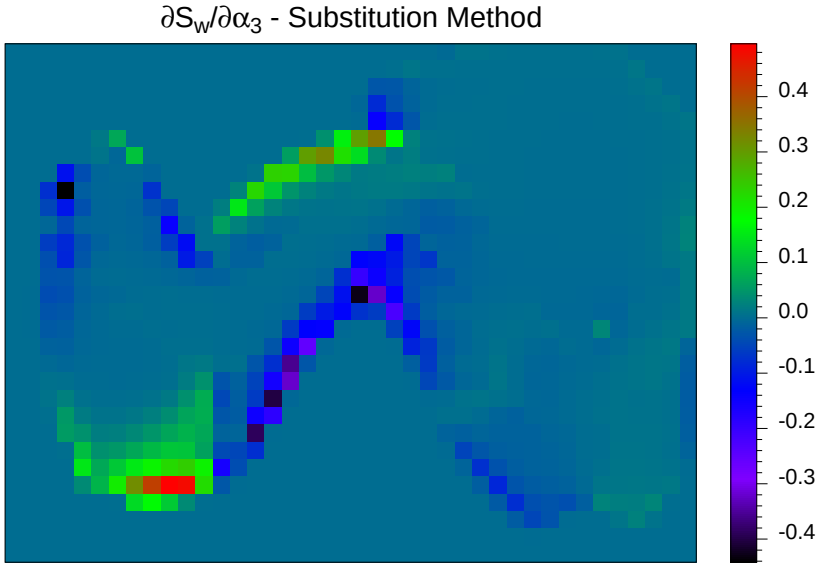



Figure 3.59: Sensitivity of water saturation distribution with respect to $\alpha_3$ ("Amplitude" of the channel), as calculated by substitution method.

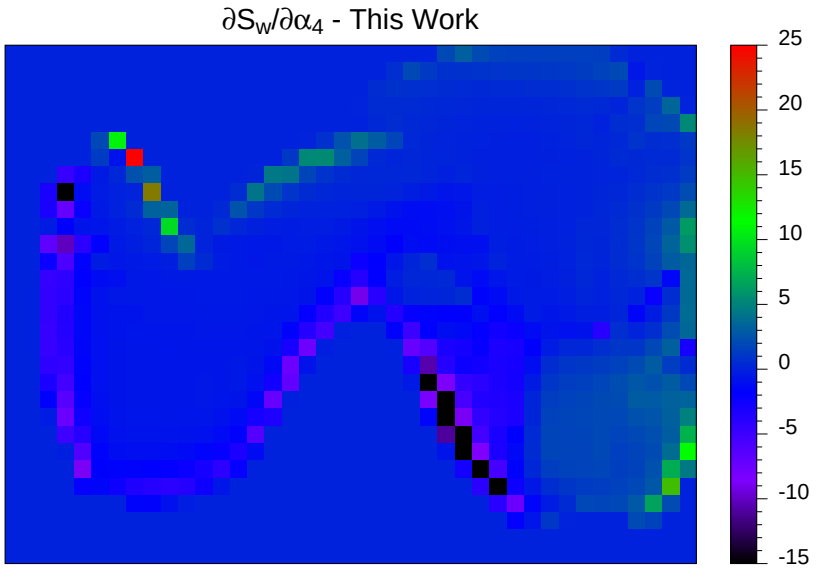



Figure 3.60: Sensitivity of water saturation distribution with respect to $\alpha_4$ ("Sinuosity" of channel), as calculated by this work.

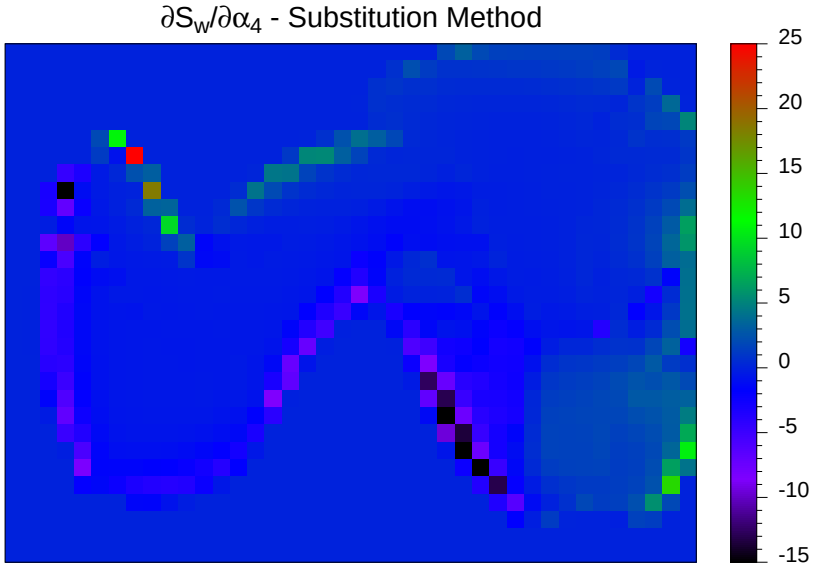



Figure 3.61: Sensitivity of water saturation distribution with respect to $\alpha_4$ ("Sinuosity" of the channel), as calculated by substitution method.

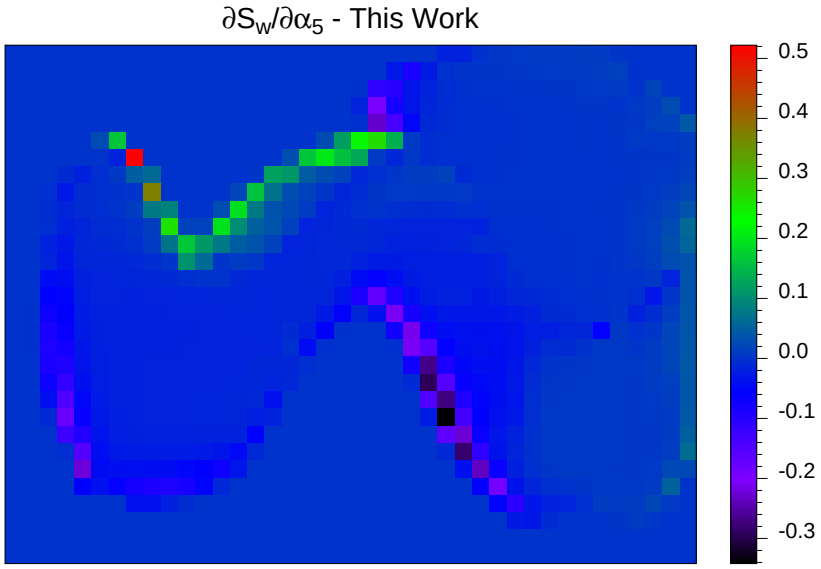



Figure 3.62: Sensitivity of water saturation distribution with respect to $\alpha_5$ ("Width" of channel), as calculated by this work.

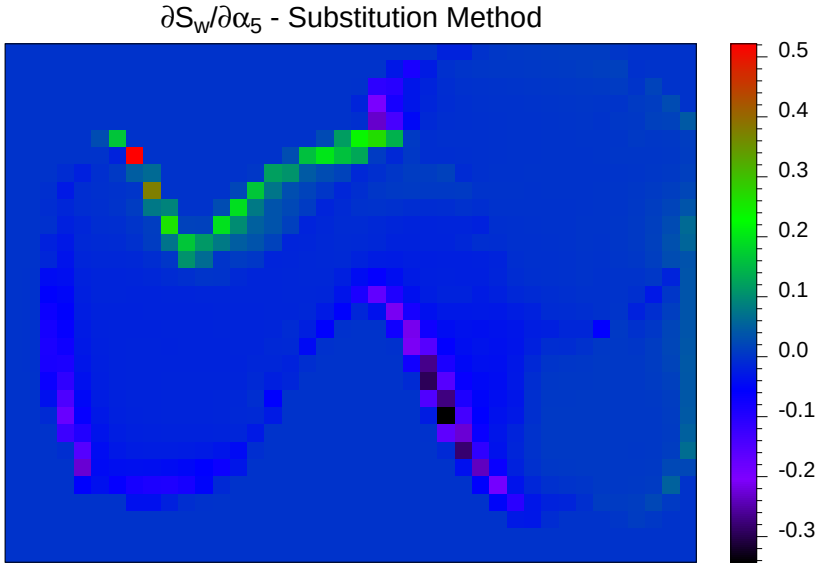



Figure 3.63: Sensitivity of water saturation distribution with respect to $\alpha_5$ ("Width" of the channel), as calculated by substitution method.

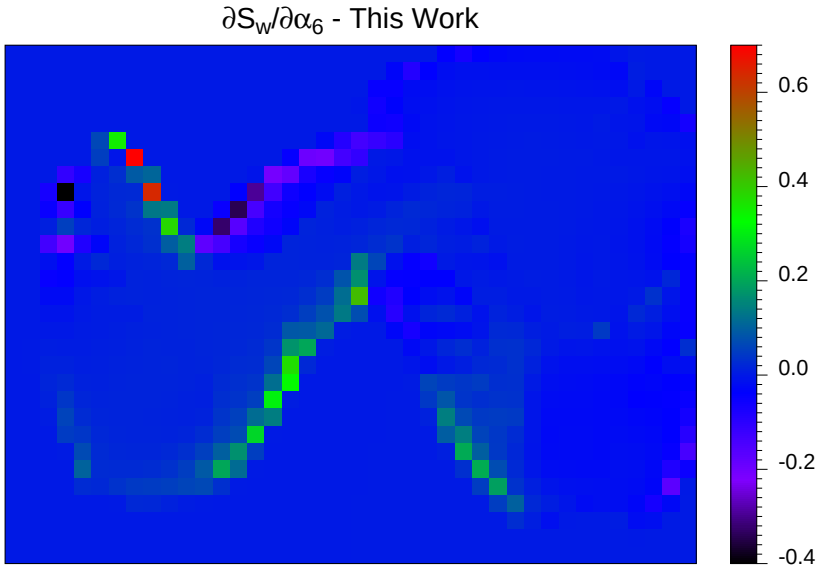



Figure 3.64: Sensitivity of water saturation distribution with respect to $\alpha_6$ ("Translation x–direction" of channel), as calculated by this work.

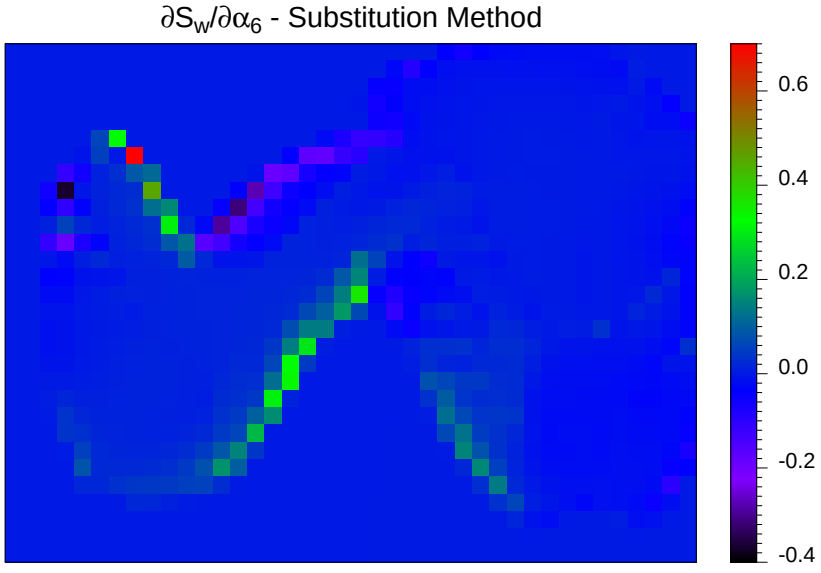



Figure 3.65: Sensitivity of water saturation distribution with respect to $\alpha_6$ ("Translation x–direction" of channel), as calculated by substitution method.

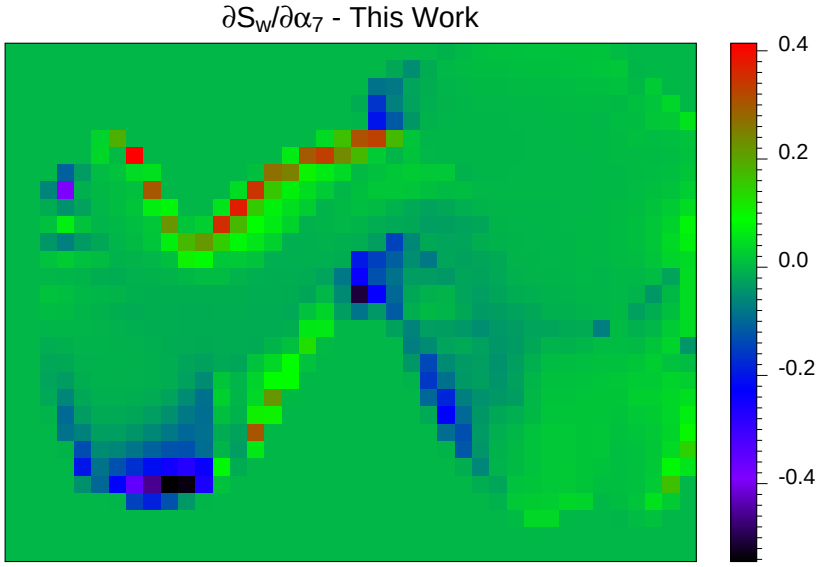



Figure 3.66: Sensitivity of water saturation distribution with respect to $\alpha_7$ ("Translation y–direction" of channel), as calculated by this work.

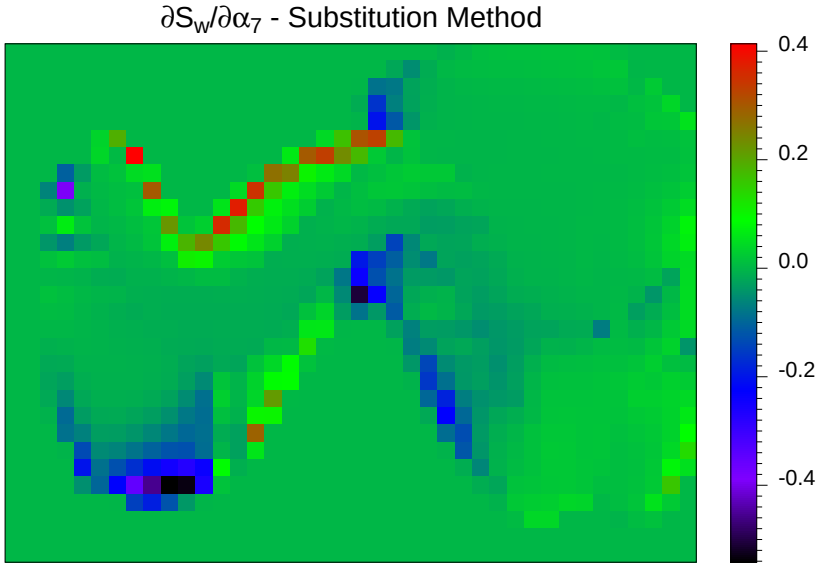



Figure 3.67: Sensitivity of water saturation distribution with respect to $\alpha_7$ ("Translation y–direction" of channel), as calculated by substitution method.

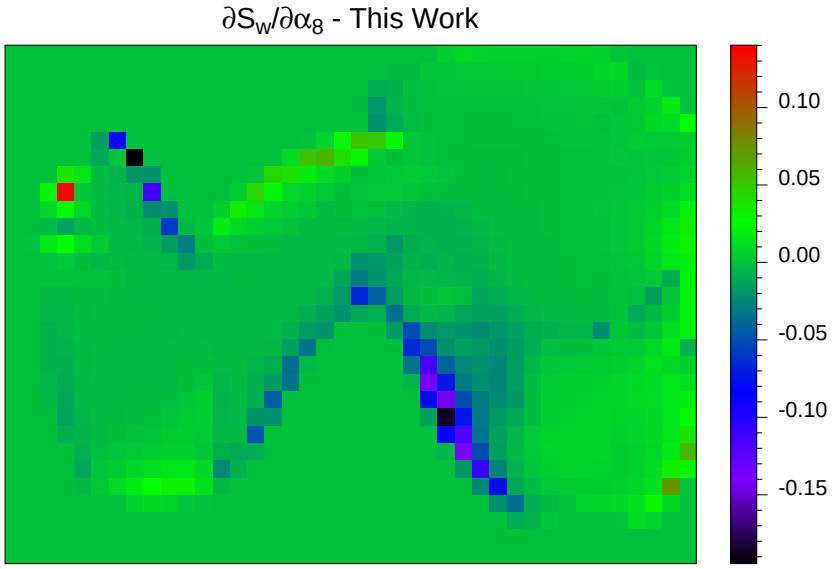



Figure 3.68: Sensitivity of water saturation distribution with respect to $\alpha_8$ ("Rotation" of channel), as calculated by this work.

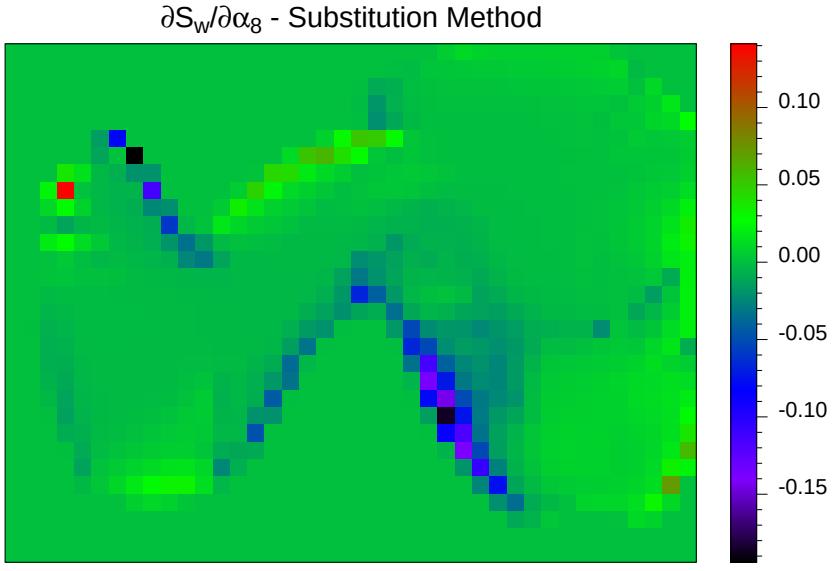



Figure 3.69: Sensitivity of water saturation distribution with respect to $\alpha_8$ ("Rotation" of channel), as calculated by substitution method.

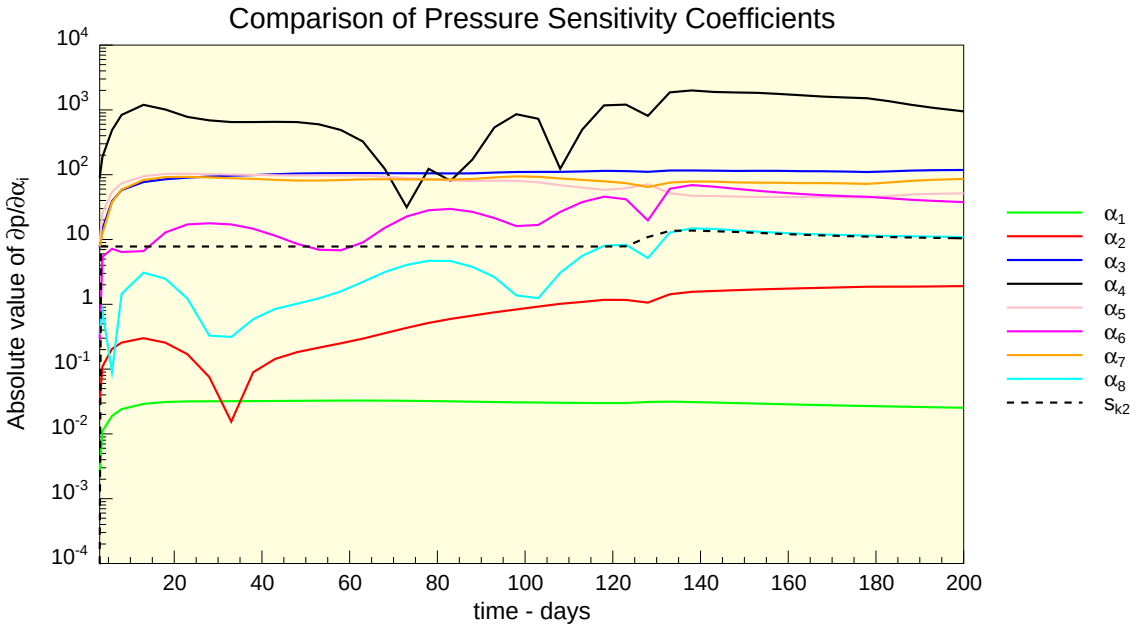



Figure 3.70: Comparison of magnitude of pressure sensitivity coefficients with respect to the parameters that define the object "channel".

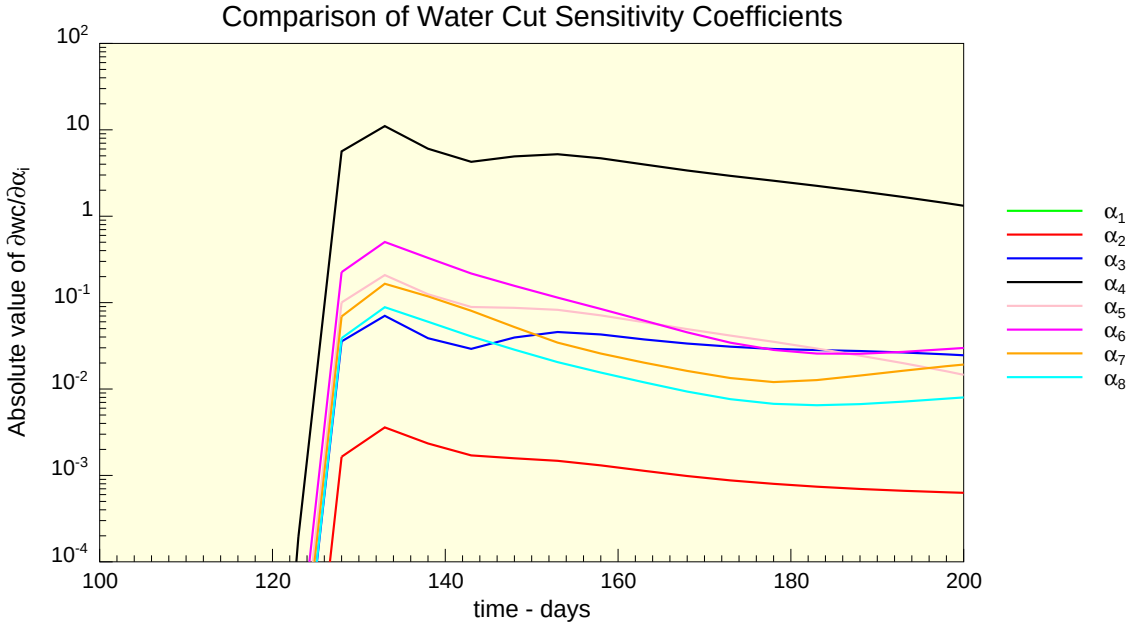



Figure 3.71: Comparison of magnitude of water cut sensitivity coefficients with respect to the parameters that define the object "channel".

# Chapter 4

# Resolution and Variance of Parameters

This chapter analyzes the meaning of the parameters estimated from the reservoir inverse problem. The analysis is performed using the mathematical approach presented in Section 2.6.2.

The questions we wish to answer are:

1. What data are necessary to determine a given reservoir parameter?
2. How much does each type of data contribute to reducing the uncertainty in the parameter estimation?
3. What is the significance of each parameter value estimated from the nonlinear inverse problem?

A basic concept to approach with these questions is the use of the resolution and covariance matrices of the parameter estimates assuming a linear model in the surroundings of a given solution $\vec{\alpha^*}$. $\vec{\alpha^*}$ is a parameter vector that provides a reasonable match of the observations and may be obtained by any method for nonlinear parameter estimation. The approach developed here can be used for the design of an optimal strategy for data collection.

In order to have an understanding of how the data is related to the resolution of parameters and their corresponding variance, a set of numerical experiments was constructed as described in the following sections.

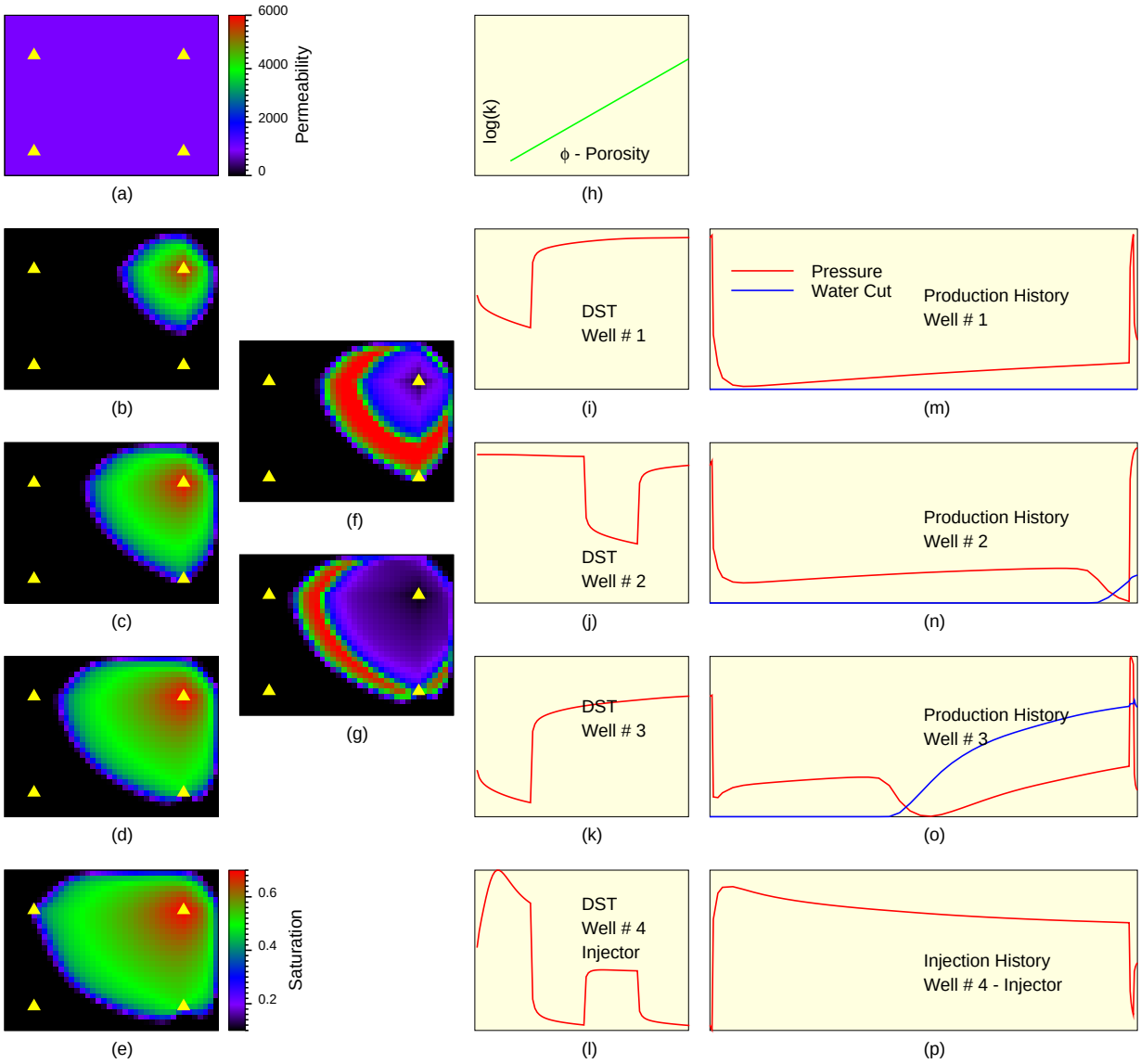
## 4.1 Analysis of Variance and Resolution for a Homogeneous Reservoir and Different Data.

A two–dimensional, black–oil reservoir was modeled as shown in Fig. 4.1(a). In the first instance, to avoid the effects of the heterogeneity in understanding the problem from the conceptual point of view, the permeability and porosity were set constant throughout the reservoir. Also, the geometry (dimensions) and rates were set in such a way to allow us to obtain water breakthrough in a short period of time. In the model, Well #4 is injecting water, the others are producing at constant total rate (at bottom hole conditions). Fig. 4.1(b)–(e) shows the water saturation distribution in the reservoir as a function of time, that is $t_b < t_c < t_d < t_e$. The water saturations maps are shown here only to illustrate what is going on in the reservoir and to help understand the results of the analysis, these maps were not available as data for reservoir parameter estimation. Fig. 4.1(f) through (p) show the information that can be obtained from measurements in the field and thus that can be used to perform reservoir parameter estimation. This information includes:

- 4–D seismic data. Fig. 4.1(f) and (g) show the changes of the water saturation distribution between (b)–(c) and (c)–(d) respectively, we assumed that this information would be available from the interpretation of three 3–D seismic surveys, that is two 4–D seismic interpretations.
- A known relationship between permeability and porosity. Fig. 4.1(h) shows that such relationship is of the type $\log k \sim \phi$, this relationship can be obtained from the analysis of cores.

- Well test data. Fig. 4.1(i)–(l) shows pressure versus time for drill stem tests (DST) at each well.

- Long term pressure and water cut. Fig. 4.1(m)–(p) shows pressure and water cut versus time for each well. At late time there was a shut–in pressure measurement at each well. The shut–in pressures were obtained in the simulated field by shutting the well for a 24 – 48 hour period and collecting a few bottom–hole pressure measurements.




(a): Reservoir geometry. (b)–(e): Water saturation distribution as a function of time. (f): Change of water saturation between (b) and (c). (g): Change of water saturation between (c) and (d). (h): Permeability–Porosity relationship. (i)–(l): Pressure vs. time for DST's. (m)–(p): Long term pressure and water cut vs. time.

Figure 4.1: Data for analysis of resolution and variance.

In order to calculate the variance of the estimates the following covariance model $\mathbf{C}_{data}$ was assumed for the data:

$$\mathbf{C}_{data} = \begin{pmatrix} \mathbf{C}_{DST} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{pressure} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{C}_{watercut} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{C}_{dsat} \end{pmatrix} \tag{4.1}$$

Where $\mathbf{C}_{DST}$ is the covariance matrix of the pressure measurements taken during the DST's, $\mathbf{C}_{pressure}$ is the covariance matrix of the long term pressure measurements (permanent gauges), $\mathbf{C}_{watercut}$ is the covariance matrix of the water cut measurements, and $\mathbf{C}_{dsat}$ is the covariance matrix of the change of water saturation in the reservoir as they are interpreted from 4–D seismic surveys. These covariance matrices are assumed to have the following form:

$$\mathbf{C}_{DST} = \sigma^2_{DST}\,\mathbf{I} \tag{4.2}$$

$$\mathbf{C}_{pressure} = \sigma^2_{pressure}\,\mathbf{I} \tag{4.3}$$

$$\mathbf{C}_{watercut} = \sigma^2_{watercut}\,\mathbf{I} \tag{4.4}$$

$$\mathbf{C}_{dsat} = \sigma^2_{dsat}\,\mathbf{I} \tag{4.5}$$

Where $\mathbf{I}$ is the square identity matrix of the appropriate dimension for each data set.

The structure of $\mathbf{C}_{data}$ as presented in Equations 4.1, 4.2, 4.3, 4.4, 4.2, and 4.5 implies that we have assumed that the variabilibility of each data point is independent of the others.

For this numerical experiment the following values of $\sigma$ were assumed:

$$\sigma_{DST} = 0.1 \text{ psi}$$

$$\sigma_{pressure} = 0.1 \text{ psi}$$

$$\sigma_{watercut} = 0.01$$
$$\sigma_{dsat} = 0.05$$

The water cut and change of saturation have no units. They are expressed as fractions, that is a value between 0–1.

The data set is composed of 3155 observations or data points. The breakdown of these data is as follows:

| *Data Set Type* | *Number of Observations* |
|---|---|
| DST | 384 |
| Long term pressure | 212 |
| Water Cut | 159 |
| $\Delta$ Saturation #1 | 1200 |
| $\Delta$ Saturation #2 | 1200 |
| *TOTAL* | 3155 |

The number of $\Delta$ Saturation points for each 4–D survey is equal to the number of cells in the simulation grid, which in this case was 1200 (40 x 30).

The *static object* type of modeling was used to parameterize this numerical experiment. 100 static objects were evenly distributed over the simulation grid. Each object was made of (4 x 3) adjacent cells, the parameter defining each object was permeability, that is the 12 cells that make up the object were assigned the same permeability. The porosity was assigned by using a fixed relationship to the permeability. Figures 4.2 and 4.3 show the simulation grid and the location of objects in the reservoir. This type of parameterization was chosen because it allows a physical link between the parameters and the geometry in the reservoir, and thus facilitates the understanding of the meaning of the resolution matrix as will be shown later in this chapter. Fig. 4.3 also shows the location of the object #56 which will be used later for analysis.

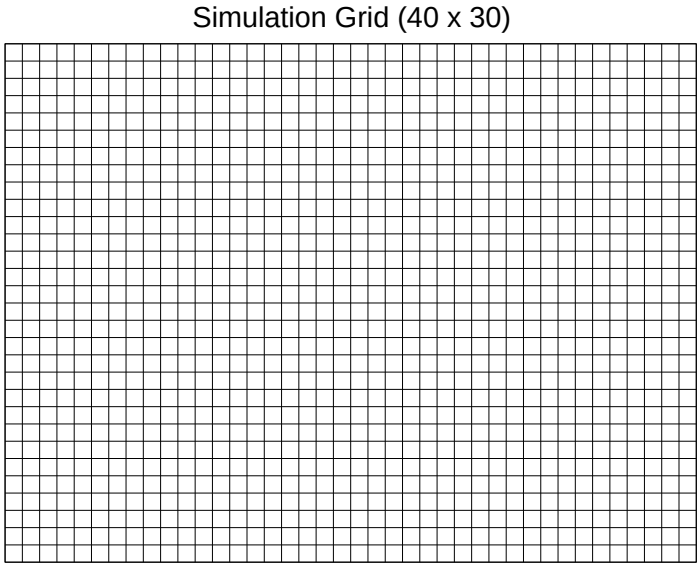



Figure 4.2: Simulation grid for resolution analysis (1200 blocks).

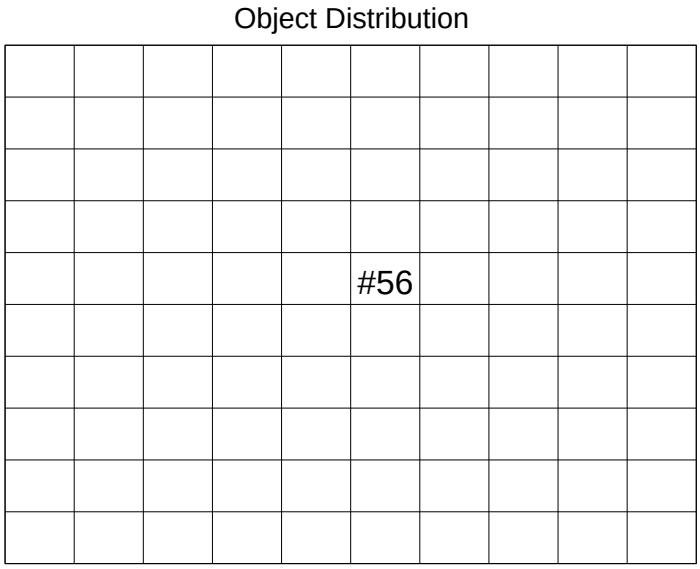



Figure 4.3: Parameterization of the reservoir for resolution analysis (100 static objects).

The total set of parameters for this problem includes 100 permeabilities and four skin factors. In the analysis we considered only the 100 permeabilities as the parameters to estimate. As mentioned earlier, in this way it is possible to relate the parameters being estimated to the reservoir geometry.

The dimensions of the matrices of interest are in this specific case:

| *Matrices* | *Symbol* | *Dimensions* | | | *Type* |
|---|---|---|---|---|---|
| Data Weight | $\mathbf{W}$ | 3155 | x | 3155 | diagonal |
| Sensitivity | $\mathbf{G}$ | 3155 | x | 100 | full |
| SVD | $\mathbf{U}$ | 3155 | x | 3155 | full |
| SVD | $\mathbf{V}$ | 100 | x | 100 | full |
| SVD | $\mathbf{S}$ | 3155 | x | 100 | diagonal |
| SVD | $\mathbf{U}_p$ | 3155 | x | p | full ($p \leq 100$) |
| SVD | $\mathbf{V}_p$ | 100 | x | p | full |
| SVD | $\mathbf{S}_p$ | p | x | p | diagonal |
| SVD | $\mathbf{H}$ | 100 | x | 100 | full |
| SVD | $\mathbf{H}^{-g}$ | 100 | x | 100 | full |
| Information Density | $\mathbf{S}_{inf}$ | 3155 | x | 3155 | full |
| Resolution | $\mathbf{R}$ | 100 | x | 100 | full |
| Covariance Data | $\mathbf{C}_{data}$ | 3155 | x | 3155 | diagonal |
| Covariance Parameter Estimated | $\mathbf{C}_{\alpha}$ | 100 | x | 100 | full |

Where $p$ is the number of nonzero *singular values* in $\mathbf{S}$. The matrices corresponding to SVD (*Singular Value Decomposition*) are defined for this case as follows:

$$\mathbf{W}^{\frac{1}{2}}\,\mathbf{G} = \mathbf{U}\,\mathbf{S}\,\mathbf{V}^T = \mathbf{U}_p\,\mathbf{S}_p\,\mathbf{V}_p^T \tag{4.6}$$

$$\mathbf{H} = \mathbf{G}^T\mathbf{W}\,\mathbf{G} = \mathbf{V}\,\mathbf{S}^2\,\mathbf{V}^T = \mathbf{V}_p\,\mathbf{S}_p^2\,\mathbf{V}_p^T \tag{4.7}$$

$$\mathbf{H}^{-g} = \mathbf{V}_p\,\mathbf{S}_p^{-2}\,\mathbf{V}_p^T \tag{4.8}$$

$$\mathbf{S}_{inf} = \mathbf{U}_p\,\mathbf{U}_p^T \tag{4.9}$$

$$\mathbf{R} = \mathbf{H}^{-g}\,\mathbf{H} = \mathbf{V}_p\,\mathbf{V}_p^T \tag{4.10}$$

$$\mathbf{C}_\alpha \;=\; \mathbf{H}^{-g}\left(\mathbf{G}^T\,\mathbf{W}^{\frac{1}{2}}\,\mathbf{C}_{data}\,\mathbf{W}^{\frac{1}{2}}\,\mathbf{G}\right)\,\mathbf{H}^{-g} \tag{4.11}$$

Obviously the diagonal matrices do not need to be formed explicitly since the diagonal elements are the only elements of interest. Despite the relatively small number of parameters compared to the size of the simulation grid, the matrices $\mathbf{U}$ and $\mathbf{S}_{inf}$ are of large dimension. Since the interest is to explore the relationship between the data (observations) and the resolution and variance of the parameter estimates, then it will not be necessary to compute $\mathbf{U}$ and $\mathbf{S}_{inf}$.

The data weight matrix $\mathbf{W}$ was defined as:

$$\mathbf{W} = \mathbf{C}_{data}^{-1} \tag{4.12}$$

That is, the data were scaled as the inverse of their standard deviations, and thus the covariance of the weighted data became the identity matrix.

The method to compute the covariance matrix of the parameter estimates was outlined in Section 2.6.3. With this approach it is not necessary to compute the matrix $\mathbf{U}$. The following sections describe the results of applying such calculations. The computer code used to calculate the singular value decomposition was from *Numerical Recipes in Fortran* [51].

### 4.1.1 Singular Values

The matrix $\mathbf{G}$ was computed by using the method to compute the sensitivity coefficients described in Chapter 3. The nonzero singular values of $\mathbf{G}$ are shown in a linear scale in Fig. 4.4 and in logarithmic scale in Fig. 4.5. It is observed that the singular values are all larger than zero, but that they become very small.

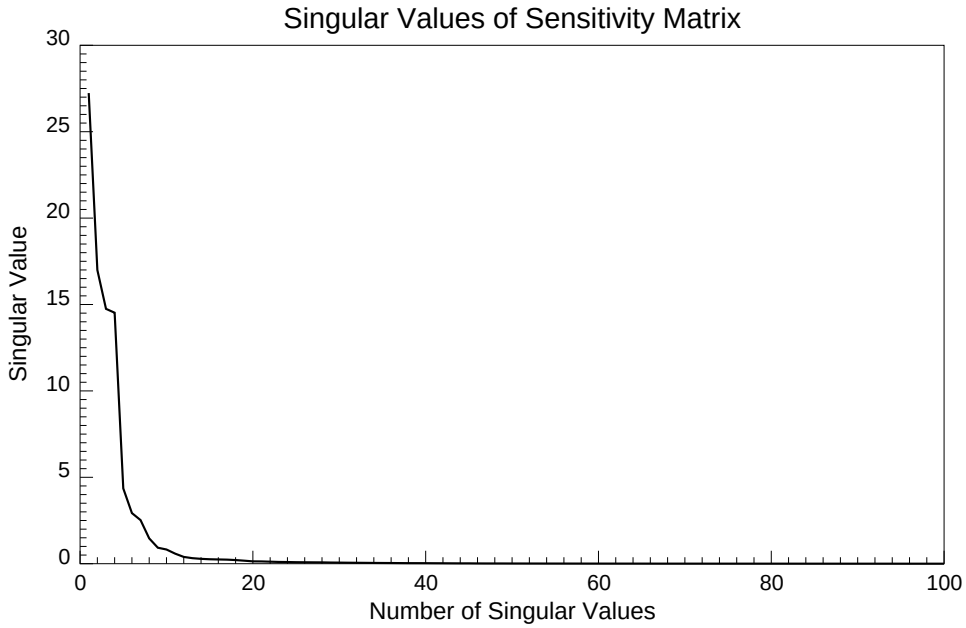



Figure 4.4: Singular Values of **G** - linear scale.

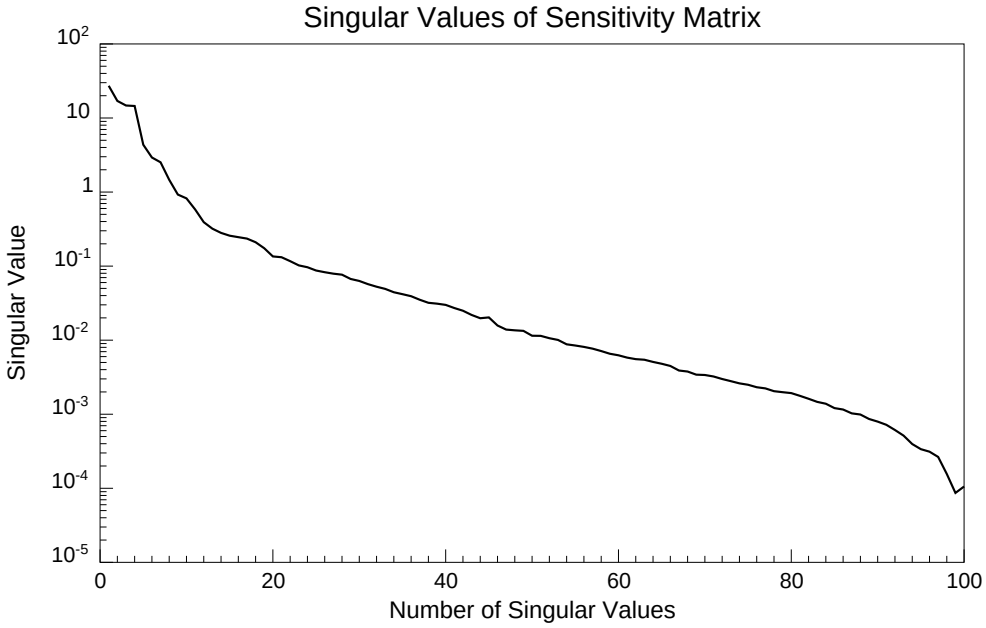



Figure 4.5: Singular Values of **G** - logarithmic scale.

### 4.1.2 Resolution and Variances

The first analysis was performed using the full spectrum of the singular values, that is $p = 100$, then another analysis was performed using $p = 50$.

#### Resolution and Variances for $p = 100$

In this case all the singular values were used ($p = 100$). By construction $\mathbf{R} = \mathbf{I}$, since $p = npar$. Fig. 4.6 shows the covariance matrix, since this matrix has 10,000 elements it is shown as a color map in which the color scale gives an indication of the value of each element of the matrix. A number of interest is the inverse of the square root of the variance of the parameters (diagonal elements of the covariance matrix) and because in this particular case it is possible to relate the parameters directly to the permeability in the reservoir, this variance is shown as a color map in Fig. 4.7. The darker zones of the map indicate which parameters can be resolved more accurately (smaller variance). The locations of the wells have been indicated to facilitate understanding. The following observations and qualitative interpretations can be made from this plot:

1. The permeabilities in the areas surrounding the wells are very well determined. This comes from the fact that the pressure and water cut information are gathered at the well locations.

2. The second best area is the area located between wells #4 (injector) and #3, this is because the water cut information is mainly from well #3.

3. The next best determined area is the area behind the water front. This is consistent with the observations made in the analysis of the sensitivity coefficients corresponding to the change of water saturation information.

4. The next best determined area is the surrounding of well #1, only pressure information is available for this area.

5. Finally, the worst area is the one located at the middle of the West and South boundaries, this arises from the weak information from these areas, far from the

wells and ahead of the water front.

### Resolution and Variances for $p = 50$

In this case we used the 50 largest singular values $p = 50$. In this case $p \leq npar$, and consequently the resolution matrix is no longer the unity matrix. Fig. 4.8 shows the resolution matrix as a color map, the colors are proportional to the value of each element in this matrix. It is observed that, as a whole, the resolution matrix seems not to be a bad approximation to the unity matrix. The covariance matrix is shown in Fig. 4.9. The map of $\sigma^{-1}$ is shown in Fig. 4.10. An analysis of these plots shows:

1. When $p < npar$ there is a trade–off between the resolution and the variance of the parameter estimates. The covariance matrix in Fig. 4.9 shows values much smaller than in the case $p = 100$ (Fig. 4.6), but the resolution matrix is no longer the unity matrix as may be wished.

2. The interpretation of the map of $\sigma^{-1}$ for $p = 50$ is not as straightforward as in the case of $p = 100$ since the resolution matrix is no longer the unity matrix. This raises one of the questions stated at the beginning of this chapter, namely what is the meaning of the parameter being calculated. This is an important issue and is discussed as a separate topic next.

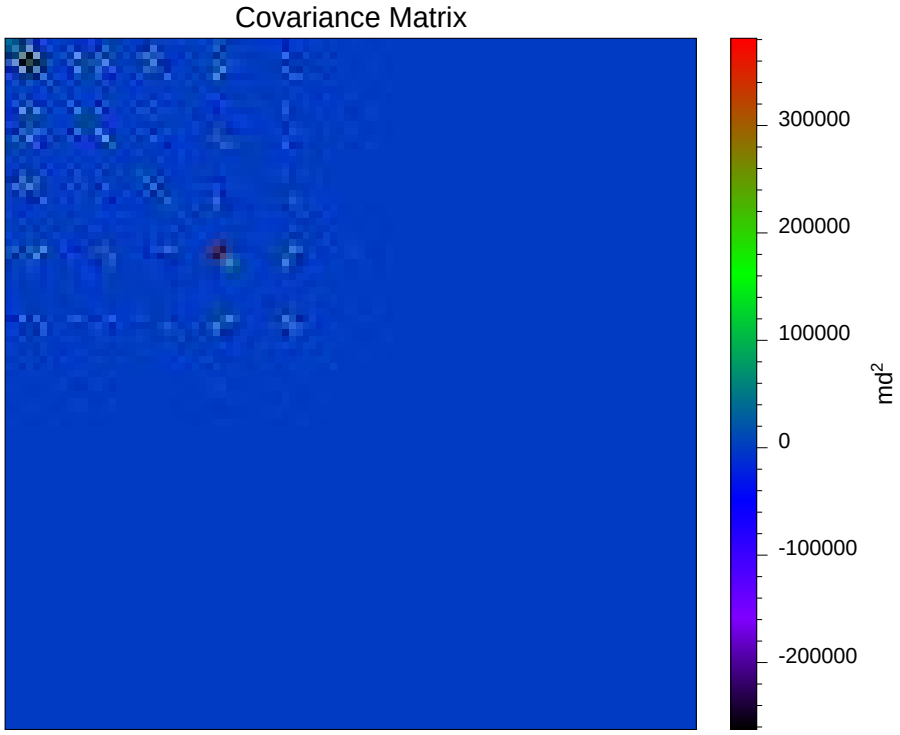



Figure 4.6: Covariance matrix of the parameter estimates $\mathbf{C}_\alpha$, $p = 100$.

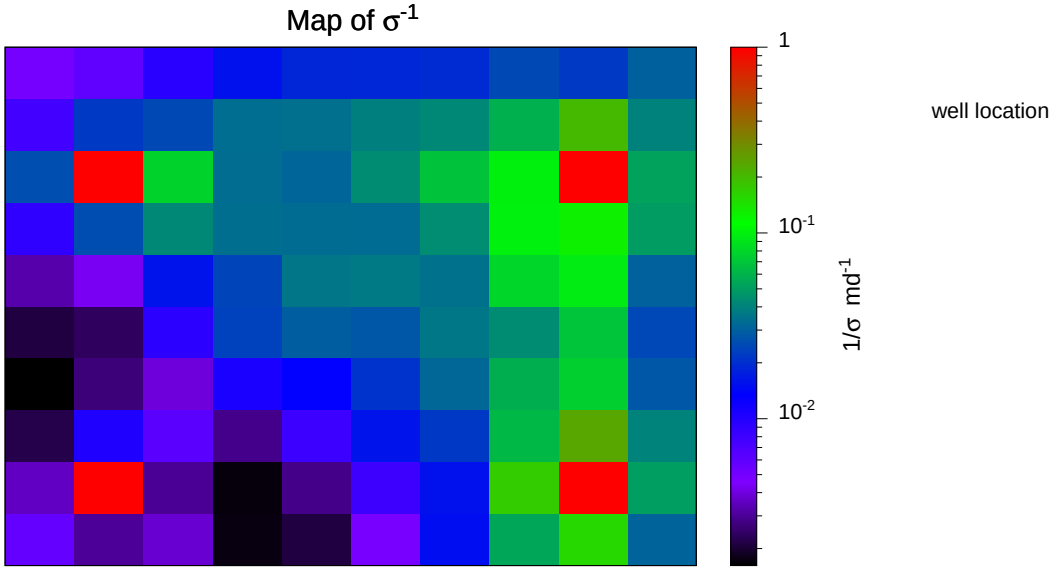



Figure 4.7: Variance of parameter estimates $\sigma^{-1}$, $p = 100$.

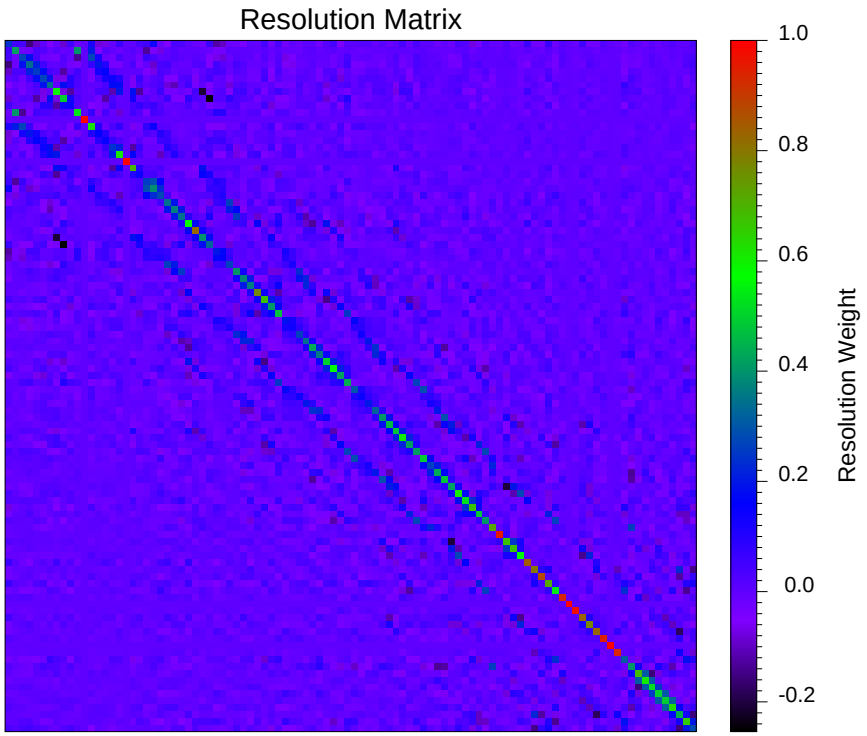



Figure 4.8: Resolution matrix of the parameter estimates $\mathbf{R}$, $p = 50$.

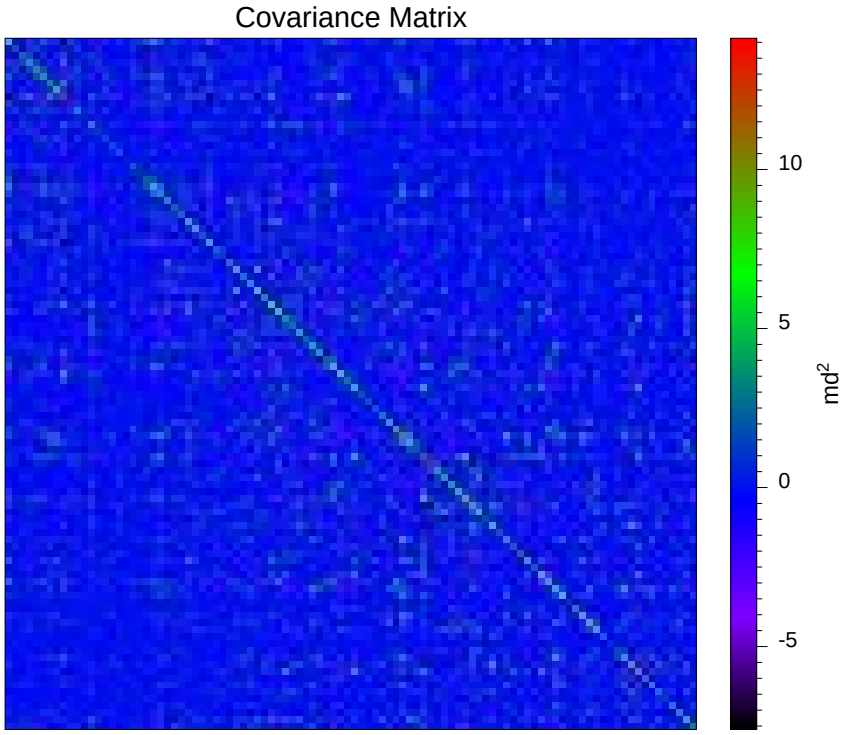



Figure 4.9: Covariance matrix of the parameter estimates $\mathbf{C}_\alpha$, $p = 50$.

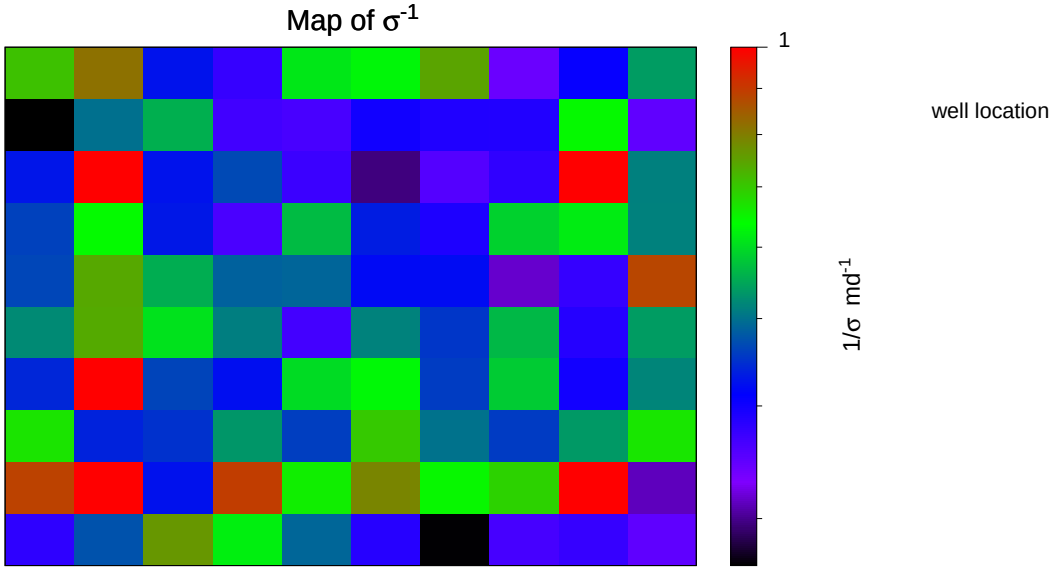



Figure 4.10: Variance of parameter estimates $\sigma^{-1}$, $p = 50$.

### 4.1.3 Trade–off Between Resolution and Variance

The variance and resolution of the parameter estimates are directly related to the number of active singular values. This was discussed as a general topic in Chapter 2. Here this subject will be discussed in more detail.

When all the singular values are larger than zero, then it is possible to have a perfect resolution of the parameters, that is $\mathbf{R} = \mathbf{I}$. In general for large problems there is not a clear cut–off between the nonzero and zero singular values, and as is shown in Fig. 4.5 the singular values approach zero very smoothly, this results in very large variances of the parameter estimates. One way to reduce the variance is by choosing only the largest singular values, this results in a lower variance as may be wished but at the cost of losing resolution. The effect of choosing only the largest singular values is the same as to say that the smallest eigenvalues are set to zero. The more singular values are zero the more undetermined is the problem. In the case in which all singular values are larger than zero, the problem is fully determined, by setting some singular values artificially to zero we force the problem to be undetermined but without specifying which are the parameters being set undetermined. Despite the problem of becoming undetermined we still calculate the full set of parameters, but the calculated values are not associated with individual values, but represent a linear combination of the parameters, and the weights of such linear combinations are given by the resolution matrix. This results in smaller variance of the linear combination of parameters. Now the calculated parameters can be seen as an average of the actual parameters. In our case the parameters can be mapped to the reservoir, so such averaging can be visualized. If the averaging is constrained to the neighborhood of the physical location of the parameter then such averaging may be accepted. All this analysis has to be done keeping in mind that the variances and resolutions we are calculating are not the variance and resolution of the actual nonlinear problem but of a linear approximation to it, anyway these are useful indications of the meaning of the calculated parameters and the relative magnitude of the uncertainty attached to them.

The trade–off between resolution and variance may be seen by analyzing the effect

of the $p$ (active number of singular values), on the calculation of a predetermined parameter. For this particular numerical experiment we will describe the effect on the parameter #56, which represents the permeability of object #56, the location of which is shown in Fig. 4.3. The first part of the analysis proceeds by constructing the trade–off curve for this parameter, this is following the concept presented by Jackson [43]. Fig. 4.11 shows the curves for the spread of the resolution and the square root of the variance for parameter #56.

The spread of the resolution matrix is a measure of its approximation to the identity matrix. The *Dirichlet spread function* is defined as:

$$\text{spread}(\mathbf{R}) = \| \mathbf{R} - \mathbf{I} \|_2^2 = \sum_{i=1}^{npar} \sum_{j=1}^{npar} (R_{i,j} - I_{i,j})^2 \tag{4.13}$$

and $\text{spread}(\mathbf{R}) = 0$ when $\mathbf{R} = \mathbf{I}$.

Another spread function to evaluate the approximation of $\mathbf{R}$ to $\mathbf{I}$ is the *Backus–Gilbert spread function* [15] which is defined as:

$$\text{spread}(\mathbf{R}) = \sum_{i=1}^{npar} \sum_{j=1}^{npar} w(i,j)\ (R_{i,j} - I_{i,j})^2 \tag{4.14}$$

$w(i,j)$ is a weight that allows us to weight each element in the row according to a certain order, that is when the parameters have a spatial meaning then the weights penalize the elements that are far from the spatial location corresponding to the parameter in the diagonal position of the matrix. Usually the weight corresponding to the diagonal element is set to zero, that is $w(i,i) = 0$. In our case the weight can be assigned to be equal to the square of the distance between parameters, that is:

$$w(i,j) = (x_i - x_j)^2 + (y_i - y_j)^2 \tag{4.15}$$

where $(x, y)$ are the spatial coordinates associated with each parameter.

Figures 4.11 and 4.12 show the plots of the variance and spread corresponding to parameter #56 as a function of $p$. The spread corresponding to parameter #56 was calculated as the spread of the row #56 of the resolution matrix. The spread was calculated using the Dirichlet and Backus–Gilbert methods. The Backus–Gilbert

method gives a mistakenly good spread for small $p$, this is due to the fact that the weight is set equal to zero on the diagonal element, and thus the method does not take into account that the diagonal is much smaller than unity. The Backus–Gilbert method provides a good measure for $p > 60$. Both Figures 4.11 and 4.12 show a clear trade–off between the resolution and the variance.

A second analysis was performed to interpret the meaning of the weights given by the resolution matrix. Fig. 4.13 shows a series of plots prepared as a function of $p$. The left column of plots shows the resolution matrices (as color maps) for $p = 100, 80, 50$, and 20. The location of the diagonal element corresponding to the parameter #56 is marked with a circle. The column of plots in the center shows the values of the elements in row #56 of the resolution matrix. The right column of plots shows how the elements of row #56 are spatially distributed.

For $p = 100$ the resolution matrix is equal to the identity matrix and has perfect resolution. By reducing $p$, the resolution is impaired and also the spread. For example for $p = 20$ the resolution is very poor but the variance is very low (from the trade–off plots).

From this analysis we see that a parameter estimate for this problem can be viewed not as an absolute value but as a relative one. That is we can consider it as "the value" of the parameter with a high variance, or as "a weighted average" of all the parameters with a low variance associated with such an average.

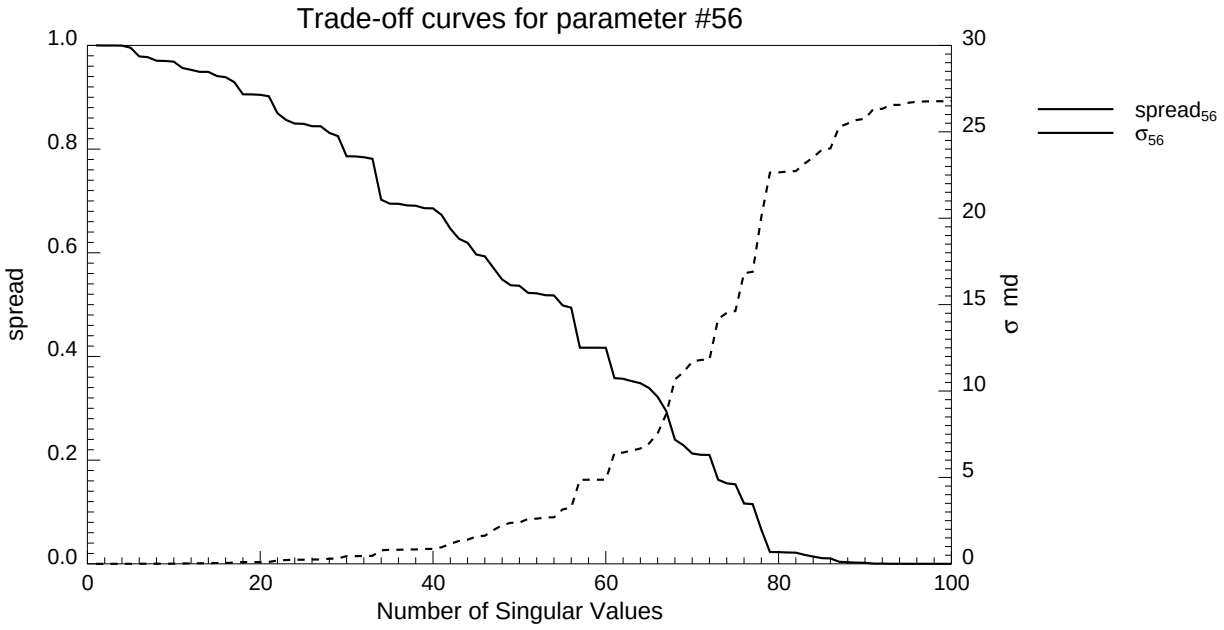



Figure 4.11: Trade–off curves for parameter #56. Dirichlet spread function.

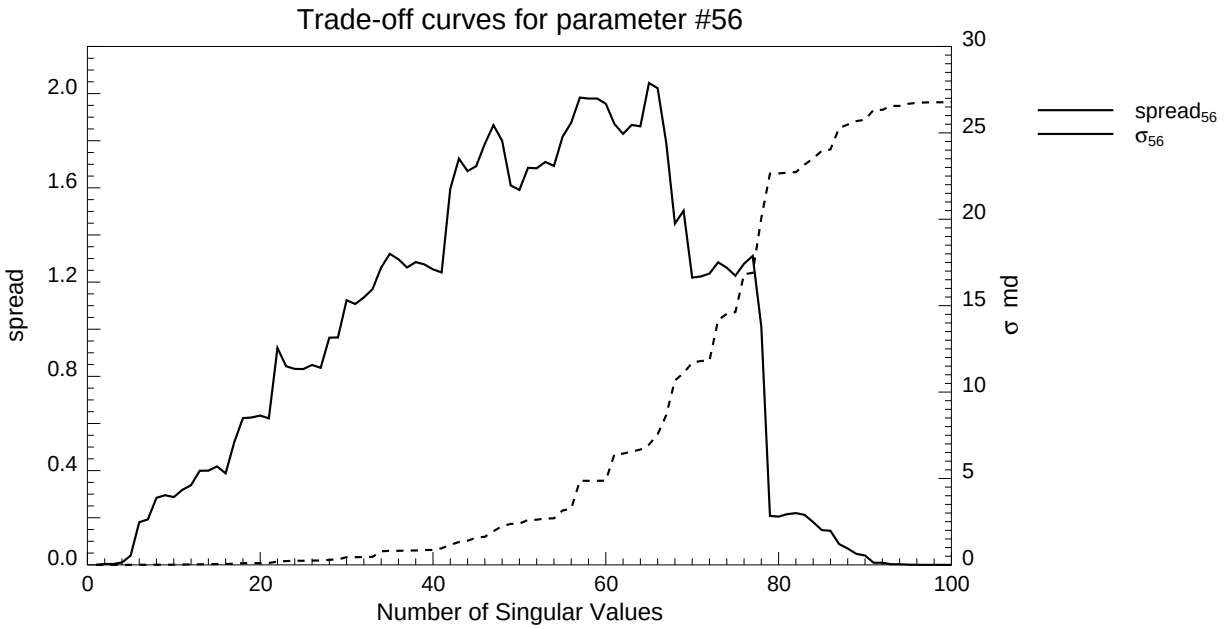



Figure 4.12: Trade–off curves for parameter #56. Backus–Gilbert spread function.

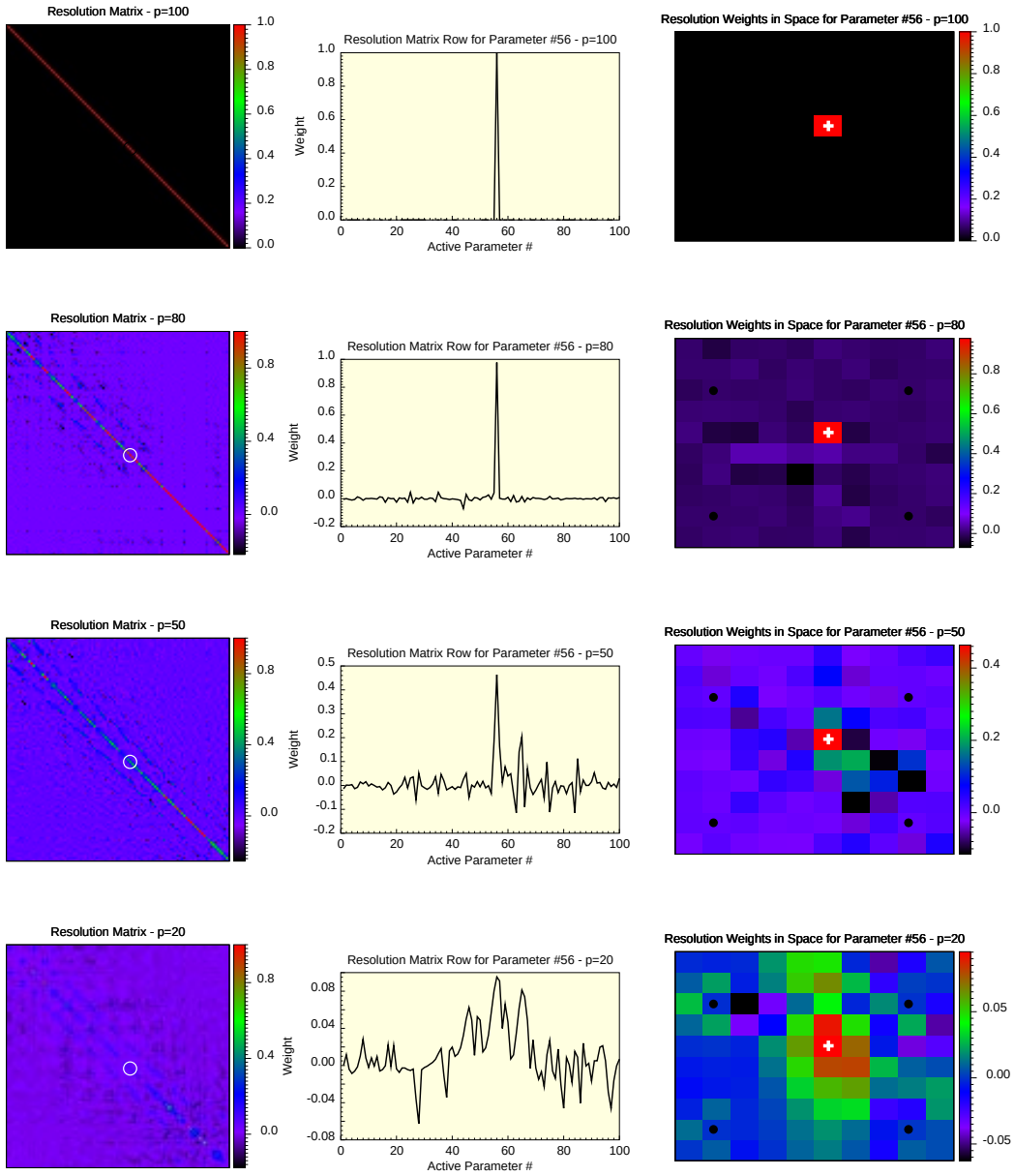



Figure 4.13: Resolution of parameter #56 as a function of number of $p$ – active singular values.

### 4.1.4 Resolution Power of the Data

One of the questions posed at the beginning of this chapter was how much each data contributes to reduce the uncertainty in the parameter estimation. This issue was explored by following the same approach shown for the study of resolution and variance, that is assuming a linear model in the surroundings of the optimal point. The approach was to calculate the variance for each parameter and for each data set both individually and in combination. The results, for the homogeneous reservoir and data as depicted in Fig. 4.1, are summarized in Fig. 4.14, which maps the normalized variance $\frac{\sigma_{\alpha_i}}{\alpha_i}$ in the reservoir. Also the locations of the wells have been indicated with small triangles. The following nine combinations of data were analyzed:

1. DST pressure only.
2. Long term pressure only.
3. Pressure only (DST pressure and long term pressure).
4. Water cut only.
5. Change of saturation only (two 4–D seismic interpretations).
6. Pressure and water cut.
7. Pressure and change of saturation.
8. Water cut and change of saturation.
9. All (pressure and water cut and change of saturation).

The variance was computed by taking the full spectrum of singular values, that is $p = 100$. The color scale for $\frac{\sigma}{k_{calc}}$ was set to be logarithmic with a range $[20–10^{-1}]$. A value of 20 means that $2\,\sigma = 0.10\,k_{calc}$, and a value of 1 means $\sigma = k_{calc}$. Thus any value above 10 can be considered as a reasonable level of uncertainty. The following remarks and observations can be drawn from Fig. 4.14.

1. Combinations of data always reduce the uncertainty. That is, adding more uncertain information to a data set does not downgrade the solution, on the contrary, it always produces a reduction in the level of uncertainty.

2. The addition of 4–D seismic information produced a remarkable improvement despite this information having a relative high uncertainty ($\sigma_{dsat} = 0.05$). This is because the 4–D seismic data provides information that is spatially distributed.

3. The water cut information seems to help little in this specific problem. This is because most of this information is concentrated at one well, #3.

4. The long term pressure seems to provide strong information, the analysis performed in Chapter 3 showed that the highest sensitivity in the long term pressure information was concentrated at the time when the water front reaches the wells (Fig. 3.7). This is confirmed again in the color map, here the best area determined by the long term pressure is the area between the injector well and well #3.

5. The DST information alone will not help much in the inverse problem in the way it was set, that is the DST data alone cannot be used to compute 100 spatially distributed parameters. DST's resolve simple models only. On the other hand the combination of the apparently poor information from DST with the long term pressure data (permanent gauges) produced a remarkable improvement in the quality of the data.

6. The DST plot shows how this data should be used in reservoir inverse problems. If no other data are available then the number of parameters should be small enough to ensure a low variance. Later in the life of the reservoir, when other data are available, such as production history and 4–D seismic, then the DST information can be used again but without the need to constrain the reservoir to a small number of parameters.

7. The appropriate number of parameters, or level of parametrization (size of the model) can be adjusted with the amount of data, that is when data are scarce the model should be kept small. When the data are abundant then certain areas of the reservoir can be parameterized with a more complex model.

Fig. 4.15 shows how the uncertainty, as measured by $\sigma$, at the center of the reservoir (parameter #56), is reduced by different combinations of all the available data.

From these analyses it was observed that the pressure data will play an important role in the reservoir parameter problem. Therefore a more detailed analysis was performed in the case of the pressure data. The analysis was performed by breaking down the pressure data into major components, which are: DST, shut–in, and data from permanent pressure gauges (referred to here as *long term pressure only*). Thus the following combinations of pressure data were investigated:

1. DST pressure only.
2. Shut–in pressure only.
3. Long term pressure only.
4. DST and shut–in pressure.
5. DST and long term pressure.
6. Shut–in and long term pressure.
7. All (DST and shut–in and long term pressure).

The results of this analysis are shown in Fig. 4.16. This time the color scale for $\frac{k}{\sigma_k}$ was adjusted to be in the range $[20 - 10^{-3}]$ in order to observe in more detail how each type of pressure data has an effect on the relative uncertainty. The higher resolution associated with the regions between the injector well #4 and the wells #3 and #2 is directly connected to the water front reaching these two wells within the time covered by the data. The apparently poor shut–in pressure information enhances

the quality of the data. The implication to the "real world" is that if permanent gauges are installed, then any sporadic or unpredicted shut–off of a well will produce a short transient, and indirectly will enhance the quality of the rest of the data. Another analysis was performed by analyzing how the spectrum of the singular values changed with different combinations of pressure data. The variance is related to the singular values through Equations 2.118 and 2.117, thus the most important point to look at is how the smallest singular values vary with each data set. Fig. 4.17 shows the results of this analysis. Interestingly we observe that the the different combinations have an effect on the smallest singular values while the largest ones do not change to a great extent.

Resolution of the data is also directly associated with the cost. The following table provides an indication of the order of magnitude of cost and the resolution power.

| *Data* | *Cost ($)* | *Resolution* |
|---|---|---|
| water cut | $10^3$ | low |
| DST | $10^4$ | low |
| long term pressure | $10^5$ | medium |
| change of saturation | $10^6$ | medium–low |
| all combined | $10^6$ | high |

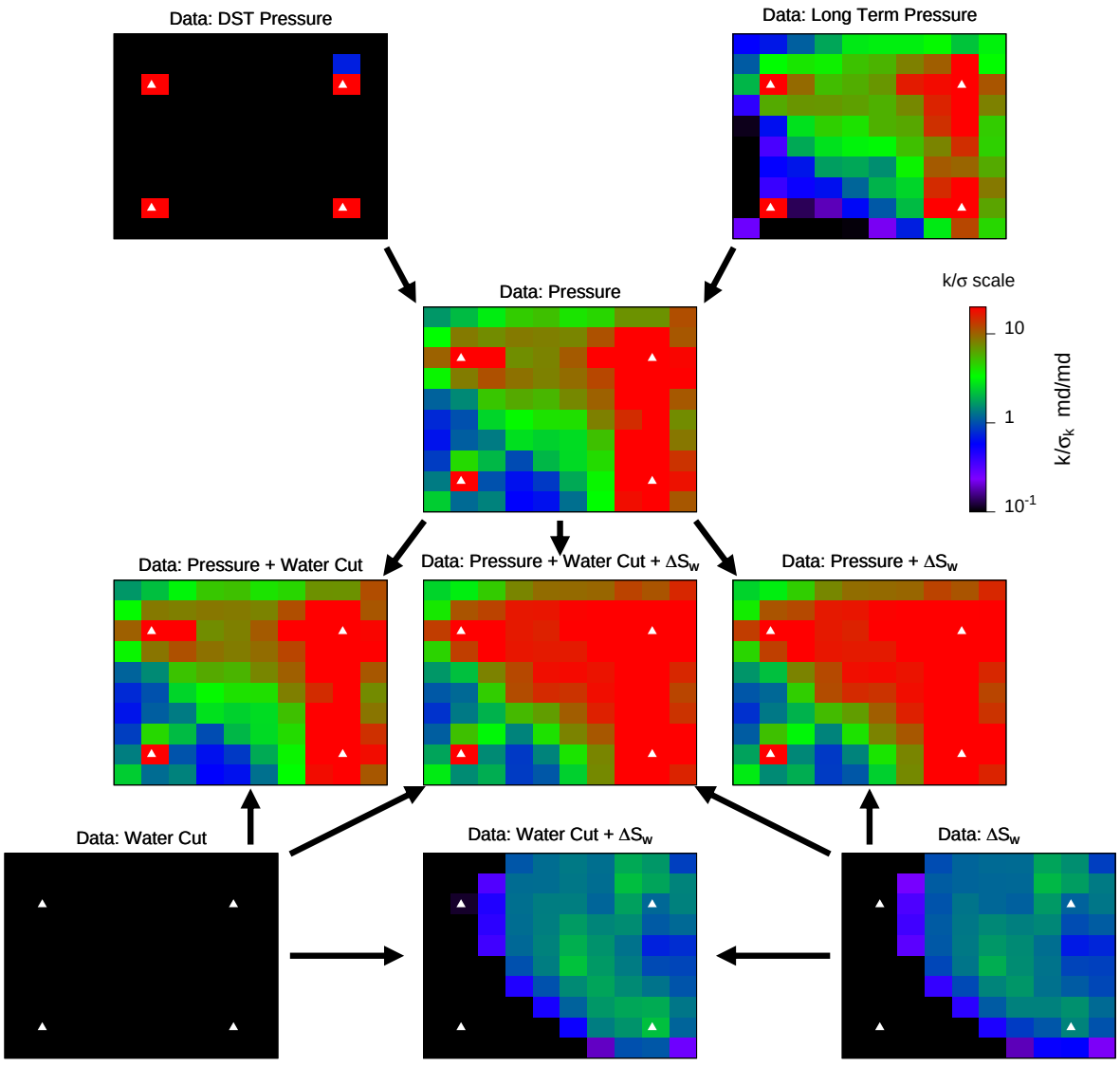



Figure 4.14: $\frac{\alpha_i}{\sigma_{\alpha_i}}$ as a function of data, $p = 100$.

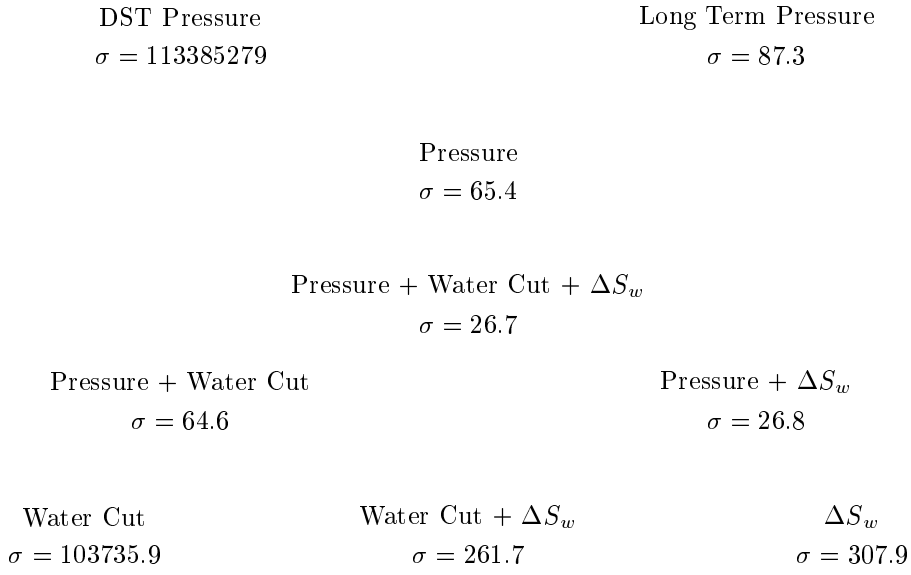



Figure 4.15: $\sigma$ for parameter #56 as function of data.

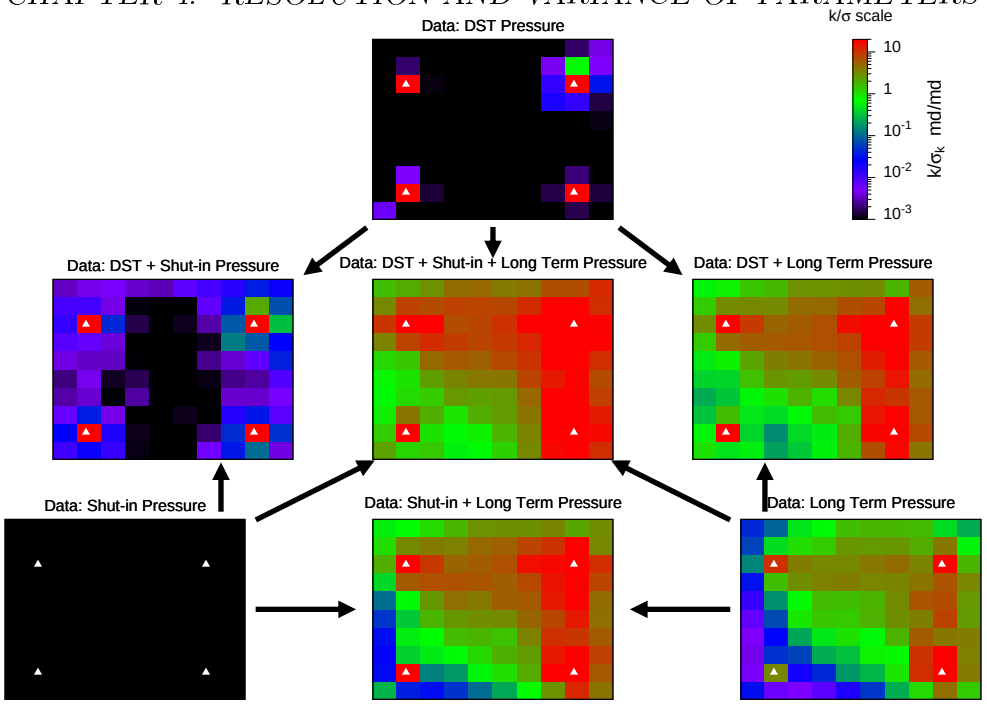



Figure 4.16: Analysis of pressure data – comparison of $\frac{k}{\sigma_k}$ for different combinations. $p = 100$.

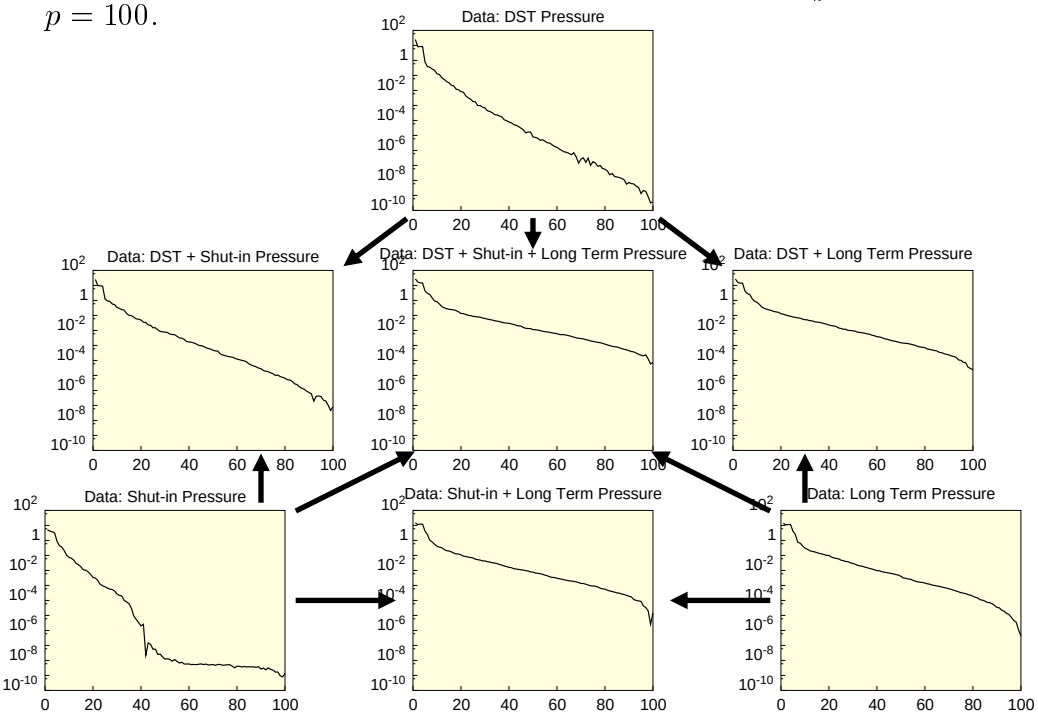
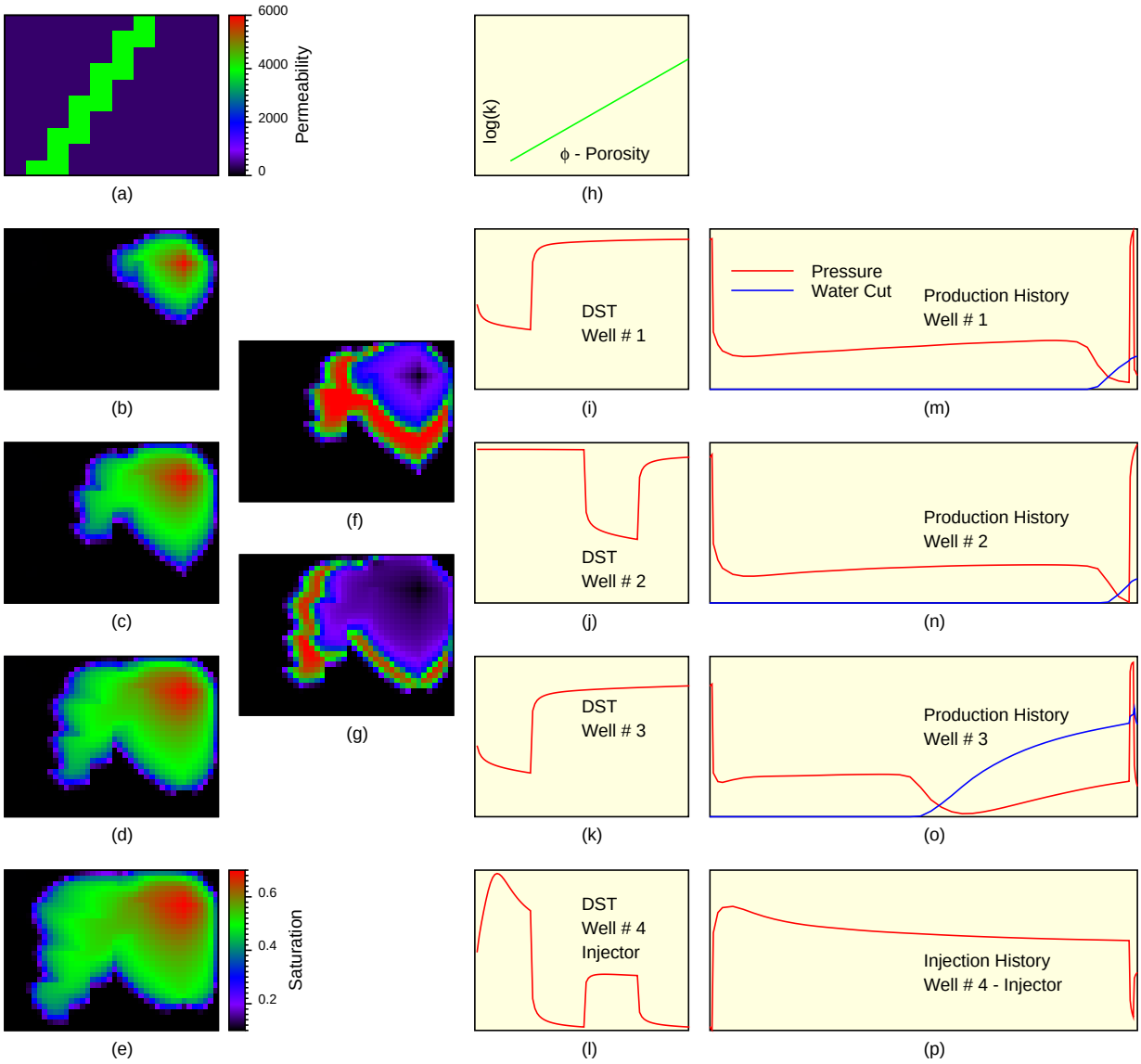



Figure 4.17: Analysis of pressure data – comparison of singular values vs. $p$ for different combinations.

## 4.2 Analysis of Variance and Resolution for Heterogeneous Reservoir and Different Data.

This section shows an analysis of the effect of the heterogeneity in the resolution power of the data. The reservoir model and data for the study are shown in Fig. 4.18. The reservoir model includes a streak of high permeability crossing the reservoir. The permeability in the streak is 8000 md, the permeability in the rest of the reservoir is 500 md. The reservoir was discretized and parameterized as in the previous case, as shown earlier in Figures 4.2 and 4.3 respectively.

The results of this analysis are summarized in Fig. 4.19, and the conclusions that can be drawn from there are basically the same as were obtained for the homogeneous case, except that in this case the better determined area is the high permeability zone and the area covered by the water front. Also the water cut seems to have more weight than in the homogeneous case.




(a): Reservoir geometry. (b)–(e): Water saturation distribution as a function of time. (f): Change of water saturation between (b) and (c). (g): Change of water saturation between (c) and (d). (h): Permeability–Porosity relationship. (i)–(l): Pressure vs. time for DST's. (m)–(p): Long term pressure and water cut vs. time.

Figure 4.18: Heterogeneous case. Data for analysis of resolution and variance.

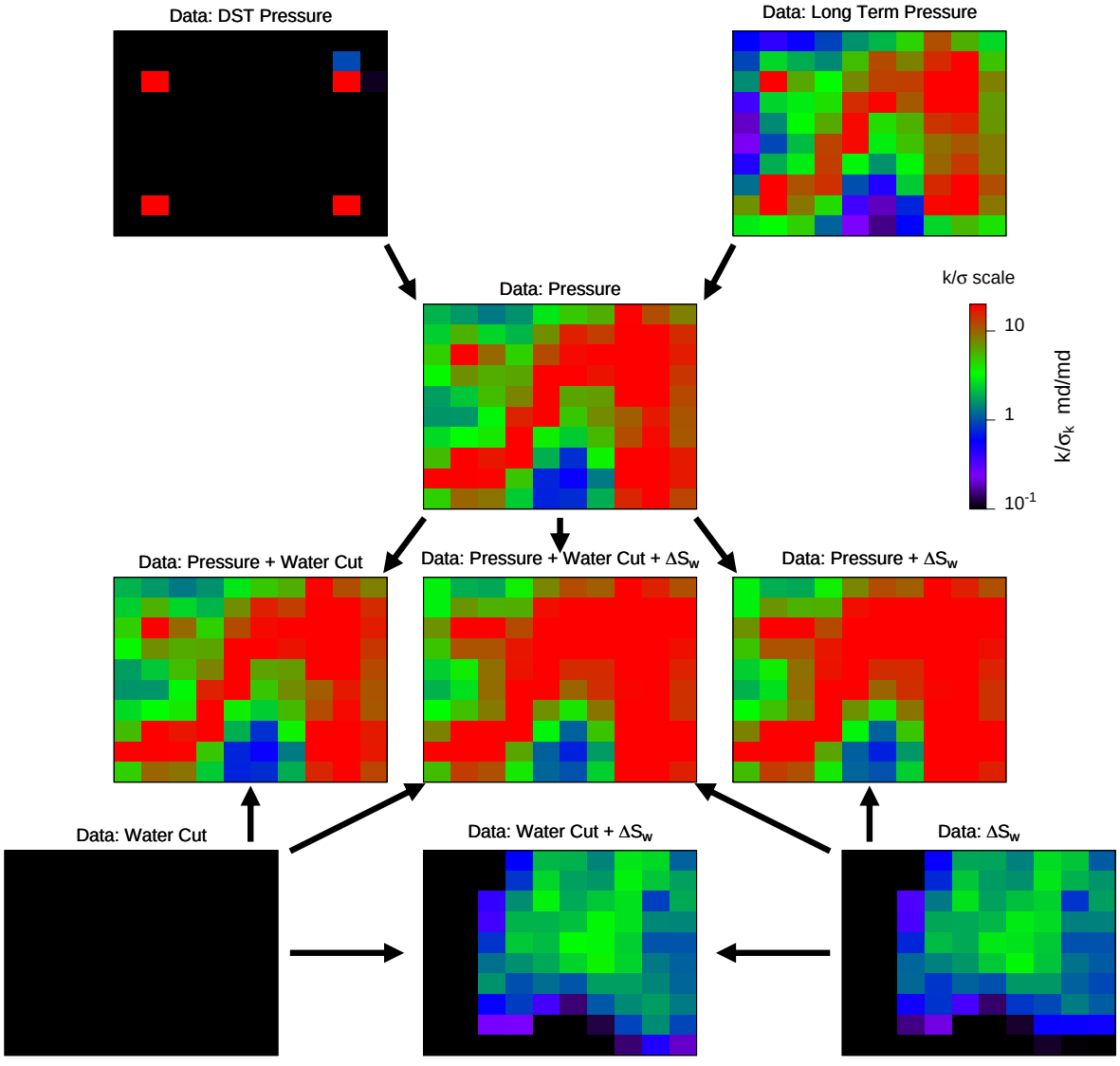



Figure 4.19: Heterogeneous case. $\frac{\alpha_i}{\sigma_{\alpha_i}}$ as a function of data, $p = 100$.

## 4.3 Analysis of the Results from a Reservoir Inverse Problem

The previous sections presented a systematic procedure to analyze the issue of how each data type contributes to the reduction of the uncertainty attached to each parameter. To perform such analysis we assumed that we knew the true values of the parameters. Here we will use the variance approach to analyze the results of an inverse problem, that is after the data have been matched to a calculated set of parameters. Then a new question arises, namely how good are such calculated parameters. This issue was investigated by using the variance analysis.

This case used the same heterogeneous reservoir and data as in Section 4.2 as shown earlier in Fig. 4.18. The first guess for the inverse problem was a homogeneous distribution of permeability. A solution to the inverse problem is a set of parameters that results in a reasonable match of the data. Fig. 4.20 shows the true and calculated permeability and their corrresponding variance maps. Both true and calculated sets of parameters provide a good match to the data, the "true case" provides the exact match because the data was generated with this distribution of permeability. At first glance both the true and the calculated permeability fields seem to be different, but when both are evaluated from the $\frac{k}{\sigma_k}$ point of view, they are seen to have essentially the same uncertainty. Both maps reveal that the permeability at the middle of the South and West boundary will be ill–determined, consequently any value there cannot be taken seriously, since any arbitrary but reasonable value assigned to those regions would provide an acceptable match to the data. That is the same as saying that the available data are not sufficient to resolve the parameters in these regions of the reservoir.

Both variance maps are similar from the qualitative point of view, this implies that it is not necessary to know the *true* reservoir values to perform the variance analysis, a reservoir description that provides a good match of the field observations is enough for the analysis.

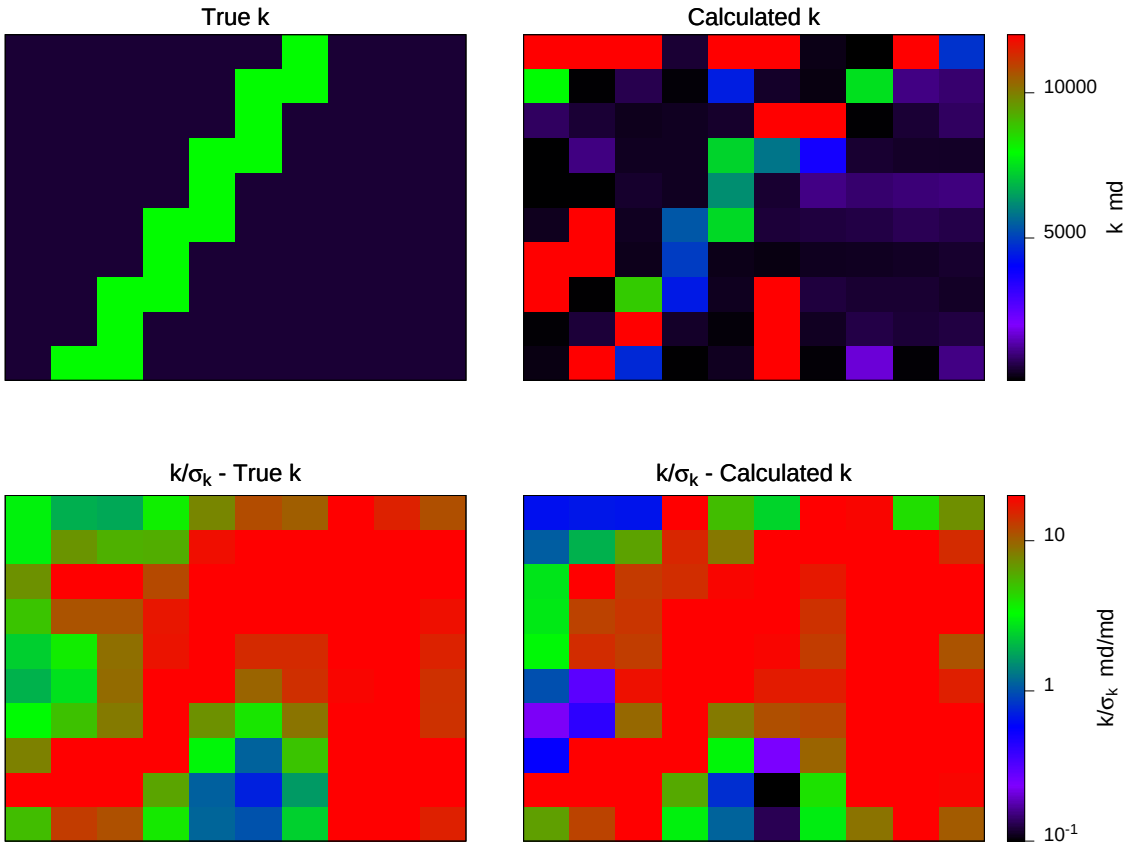



Figure 4.20: Comparison of $\frac{k}{\sigma_k}$ for true case and calculated case, $p = 100$.

## 4.4 Application of the Variance Analysis in Reservoir Monitoring

The variance analysis shown in the previous sections can be used to organize and optimize the data collection policy in a reservoir. In general it is possible to obtain a first approximation to the reservoir model before production and/or dynamic data are collected. Then it is possible to use this *approximated* reservoir to perform the variance analysis and to prepare plots similar to the plots shown in Figures 4.14 and 4.16 before any data is actually collected in the field. Then the best strategy for data collection can be devised considering that each piece of information will be used later in conjunction with the others. The analysis can also be used to find the best level of parameterization at a given time considering the data available. The level of parameterization will change as more field observations are made available. Both processes: reservoir parameter estimation and future data collection policy, have to be updated continually since every time new field data arrives the reservoir model is updated via the inverse problem solution, then the variance analysis is updated considering the new description of the reservoir, next the data collection policy is updated and the cycle starts over again.

Since the variance analysis uses as input a covariance model for the data which is generally assumed to be diagonal and which is related to the accuracy or reliability of the field measurements, then this method can be used to investigate how the accuracy of the instruments used to collect the information has an effect on the calculated permeability and porosity.

# Chapter 5

# Applications of the Method

In this chapter the application of the procedure for reservoir parameter inversion is described in several cases, the purpose is to demonstrate the viability of the approach developed in this work at the field level and to explore the difficulties and areas that require further research.

Except for the case in Section 5.8, the cases shown in this chapter are *synthetic*, that is all the parameters, including the permeability and porosity distribution in the reservoir, are perfectly known, this will be referred to in this work as the *true* reservoir. The purpose in using synthetic data was to investigate problems in which the correct solution was known exactly so that the accuracy of the estimation process could be determined.

The *field data* were generated by the mathematical model (reservoir numerical simulator), thus the data are without noise unless it is specifically indicated.

## 5.1 Channel Case

This case is about the inversion of the permeability and porosity distribution in a reservoir that was generated with a *channel* model. Fig. 5.1 illustrates the problem:

1. Fig. 5.1(a) shows a two–dimensional reservoir map, the permeability and porosity are the unknowns to be determined. There are four wells, (marked with

triangles), one of them is injecting water and the others are producing, the production rates and injection rates are also known. Fig. 5.1(b)–(e) show the water distribution in the reservoir as a function of time, this is *not known* to us, but it is shown here *only* to visualize the problem.

2. Fig. 5.1(f)–(p) shows graphically what are the *field observations* that are available to us. These observations are:

   (a) 4–D seismic interpretation. Fig. 5.1(f) and (g) show two *change of water saturation* ($\Delta S_w$) maps. These maps are assumed to have been made available after three consecutive 3–D seismic surveys. The seismic surveys were performed at the time when the reservoir was in states shown in Fig. 5.1(b), (c) and (d). The geophysical interpretation cannot provide us with the value of the water saturation at each time, but it can provide the change of saturation in the reservoir between two surveys.

   (b) Permeability – porosity correlation. Fig. 5.1(h) shows such a correlation. This correlation may be obtained from measurements in cores from the wells. The maximum and minimum values possibles for the permeability and porosity are also specified, these values may come from *a priori* geological information.

   (c) DST pressures. A drill stem test (DST) has been performed at each well and the information is available. The information is a relative large number of pressure measurements over a small period of time. Each DST consists of a single flow followed by a build–up. The DST's are performed at early time in the productive life of the reservoir, thus only a single phase (oil) was produced.

   (d) Pressure data from permanent gauges. The solid lines in Fig. 5.1(m)–(p) show the plot of the bottom hole pressure at each well as a function of time. At the late time the permanent gauges recorded a simulated shut–in pressure at each well. These events can be planned or are due to operational reasons. The shut–in is of shorter duration than the DST, and fewer measurements are made.

(e) Water cut. The dotted lines in Fig. 5.1(m)–(p) show the history of the water cut measured at each well. This assumes that production of each well is being measured individually and continuously.

(f) Fluid and rock properties. Except for the permeability and porosity, rock and fluid properties such as viscosity, density, relative permeability and compressibility are known.

(g) The skin factors at each well were considered to be known.

The first attempt to match the data was by using the *dynamic object model*. The object model was a channel. The purpose was to test if it was possible to recover the *true* channel. The objective function to minimize was the *weighted least squares*, the weights were chosen in such a way that each data set had approximately the same weighted magnitude, that is the pressure component of the objective function was approximately the same as the components corresponding to the water cut and change of saturation data sets.

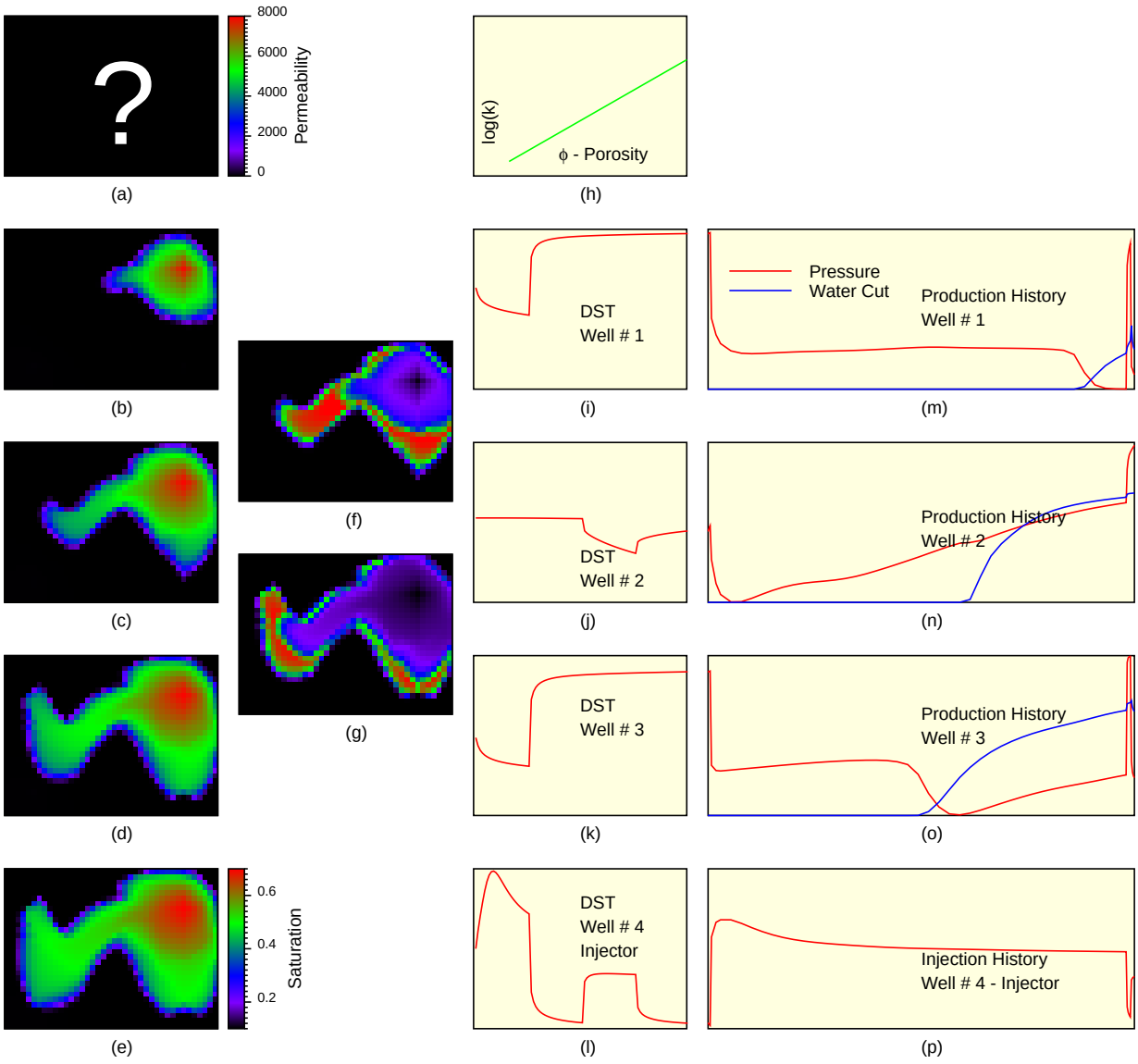
Since it is easy to visualize the channel object, it is possible to follow the evolution of the parameters at each iteration of the parameter estimation algorithm. Fig. 5.2 shows a selected set of pictures of the intermediate states of the channel and the respective value of the objective function. The colors in the map are proportional to the permeability. The first guess corresponds to an arbitrary channel that is intercepted by well #2. We may know this information from another source such as well logs, or cores. After 41 iterations the algorithm was able to recover the *true* reservoir. Except for the fact that the channel was intercepted by well #2, the first guess was substantially different from the *true* channel both in shape, location and permeability. Figures 5.4 and 5.5 show the evolution of the objective function. It is seen that the algorithm spent most of the time between iterations 5 through 34 without any appreciable progress in reducing the value of the objective function. The reason for this small improvement can be seen in Fig. 5.2 in the pictures corresponding to the iterations 11 through 25, in which it is seen that the channel is being moved very slowly by keeping well #2 in the edge of the channel. This is a very difficult task because of the shape of the edge. There was no mathematical constraint to keep the

well inside the channel, this constraint arose naturally since if the channel were modified in such a way that well #2 were to fall outside the channel, then the pressure in this well would be greatly affected, the objective function would increase and then such a change would be rejected. By iteration 19 it was observed that the progress was slow, the match of the pressure was in general good and the match of the change of saturation was very bad. At this point the weighting scale of the data was changed, most of the weight was given to the change in saturation information, the pressure and water cut were given a second order importance. As a result of this modification the *driver* for change is in the change of saturation and some increase in the partial error is allowed in the pressure and water cut. The progress of the algorithm was slow but steady. The key step arose at iteration #28 when the well moved to the other side of the channel, after which the algorithm converged very quickly to the *true* channel.

In this case the reasons for slow progress were understood by visual inspection, from the mathematical point of view this is due to a stationary point in the objective function (local minimum or saddle point). In the case of the *pixel* approach this problem is not easily detected because of the large number of parameters. The way to detect such a problem is by checking the progress of the objective function and the mismatch of each type of information, and the way to fix the problem is by changing the relative weights of the data.

As an alternative to the *dynamic object* modeling the same problem was solved using the *static object* approach. The parameterization was similar to that shown in Fig. 4.3. There are 100 parameters related to permeability, that is each parameter represents the permeability of a group of 12 simulation cells. This approach is very close to the *pixel* approach. There were two main purposes to this exercise, first to evaluate the method and second to explore what happens when the parameterization is much larger than the heterogeneity. Due to the size of the group of cells represented by each parameter it is impossible to reproduce the *true* channel with this large discretization. Thus it is known in advance that in this case we cannot recover

the *true* reservoir. The first guess, or starting point for the parameter estimation algorithm was a homogeneous reservoir, which means a complete ignorance of the channel. After approximately 500 iterations most of the data were matched reasonably closely. Fig. 5.6 shows the calculated permeability and the match of the data. It is observed that despite the coarseness of the parameterization the pressure and water cut are matched almost exactly. The case of the change of saturation is different, it is not possible to match this data perfectly, but the match can be considered to be acceptable. This case illustrates what probably would happen in field cases where the information is more detailed than the coarse discretization for parameter estimation. Fig. 5.7(a) and (b) show enlarged maps of the *true* and *calculated* reservoirs respectively. Fig. 5.7(c) shows a superposition of the true channel with the large pixel description. Fig. 5.7(d) shows a map of $\frac{\sigma_k}{k}$, this map provides an indication of the uncertainty associated with to each parameter in the large pixel model, the darker areas of the map indicate higher uncertainty, thus it is observed that the higher uncertainty corresponds to areas not swept by the water front (compare Fig. 5.7(d) and Fig. 5.1). The variance of the parameters was computed following the procedure shown in Chapter 2.




(a): Reservoir geometry. (b)–(e): Water saturation distribution as a function of time. (f): Change of water saturation between (b) and (c). (g): Change of water saturation between (c) and (d). (h): Permeability–Porosity relationship. (i)–(l): Pressure vs. time for DST's. (m)–(p): Long term pressure and water cut vs. time.

Figure 5.1: Data for inverse problem: channel model.

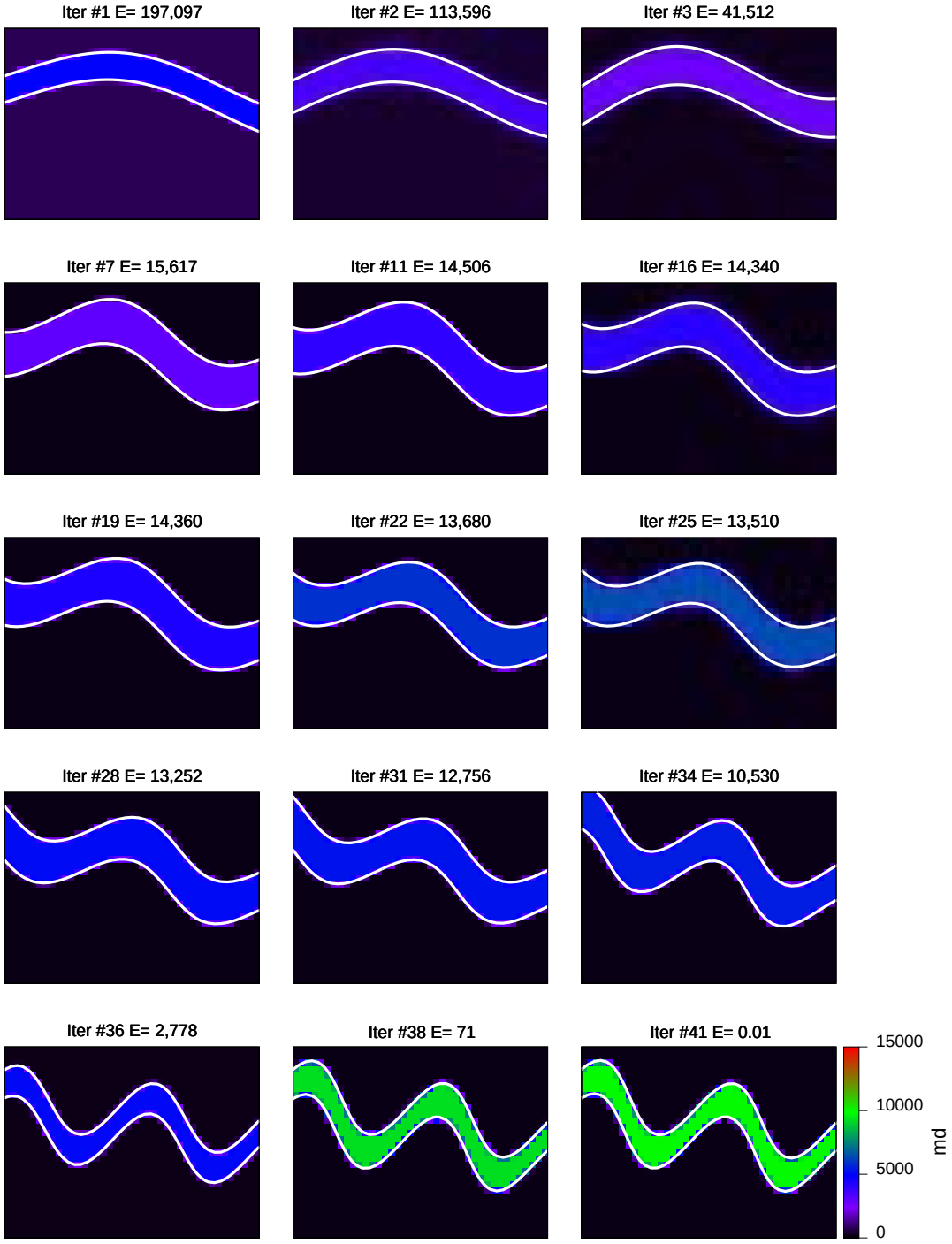



Figure 5.2: Matching history – channel model.

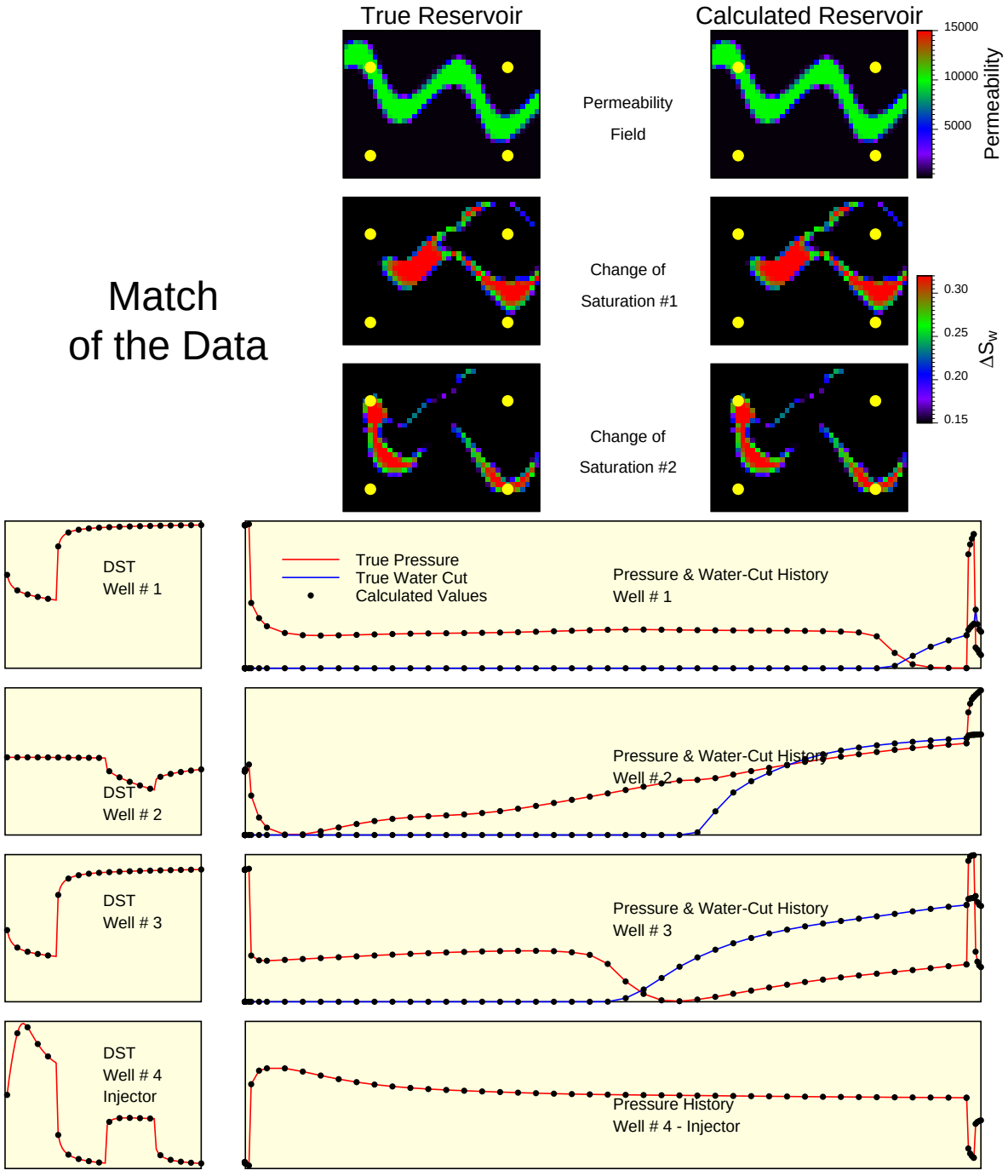



Figure 5.3: Match of the data: channel model.

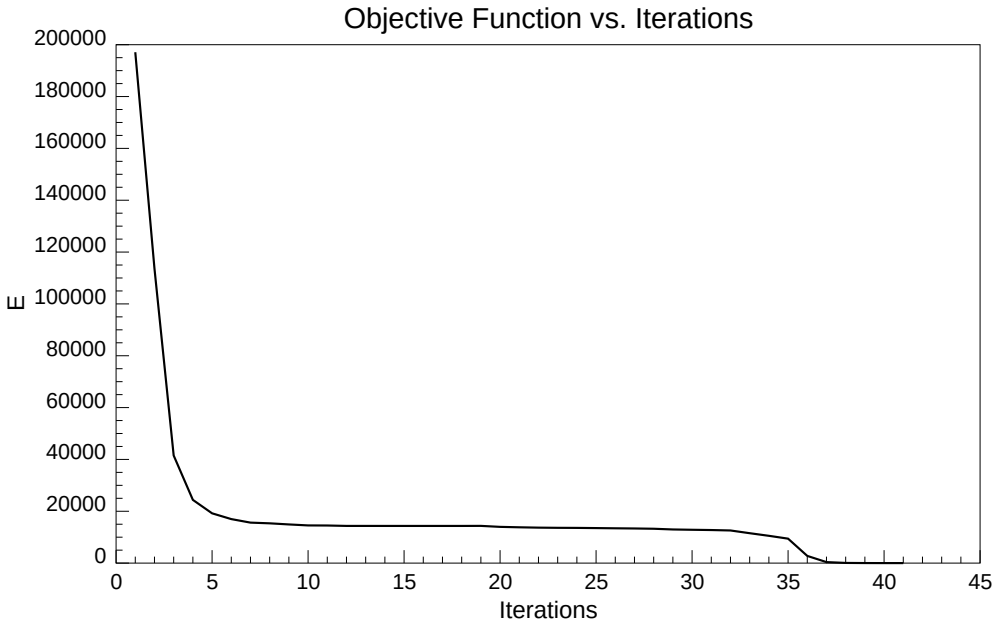



Figure 5.4: Objective function as function of number of iterations – linear scale.

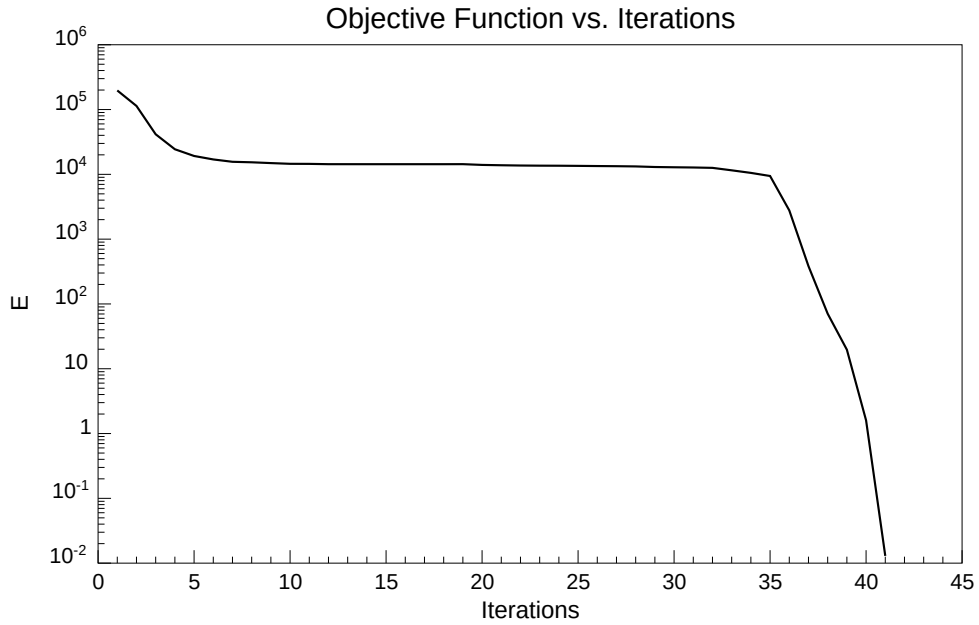



Figure 5.5: Objective function as function of number of iterations – logarithmic scale.

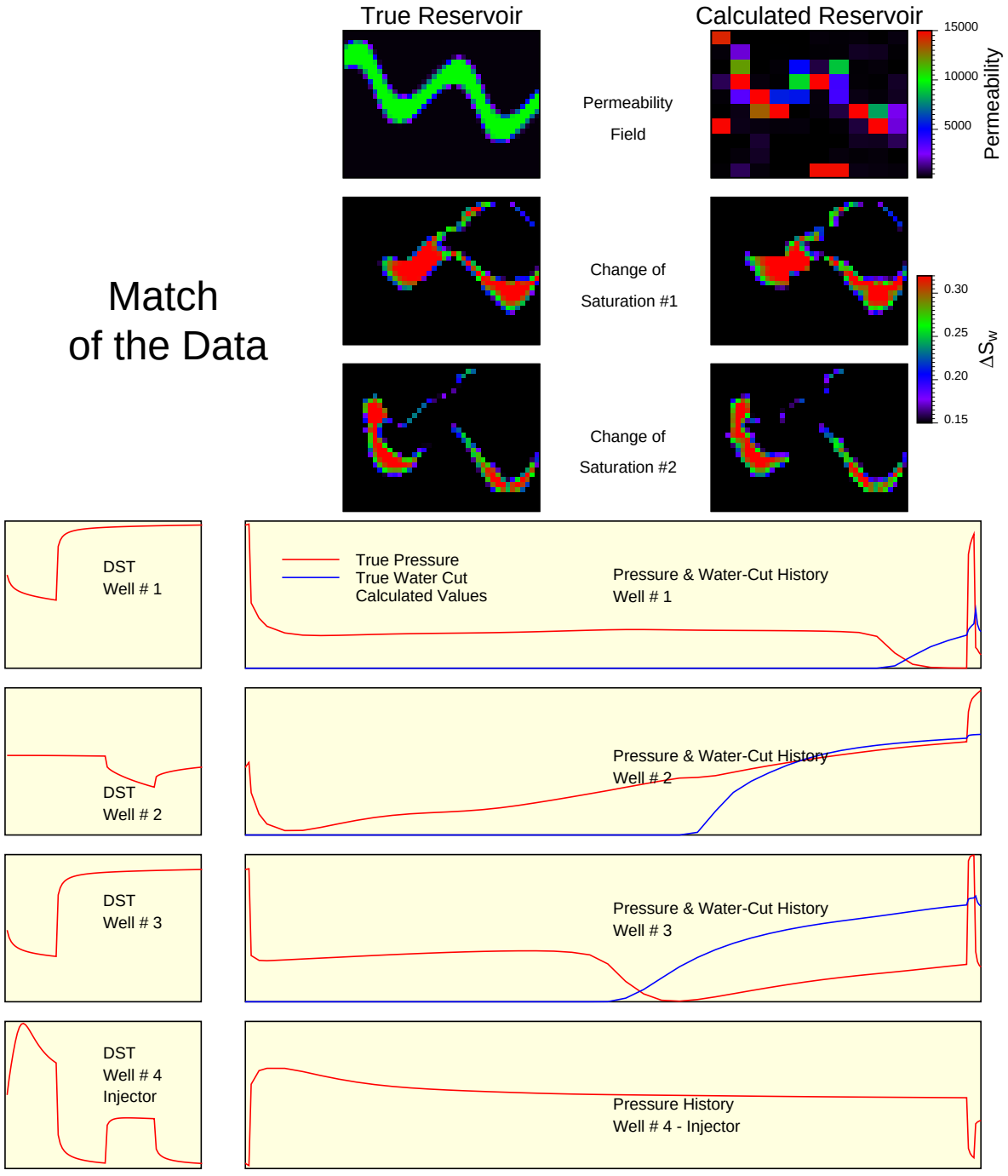
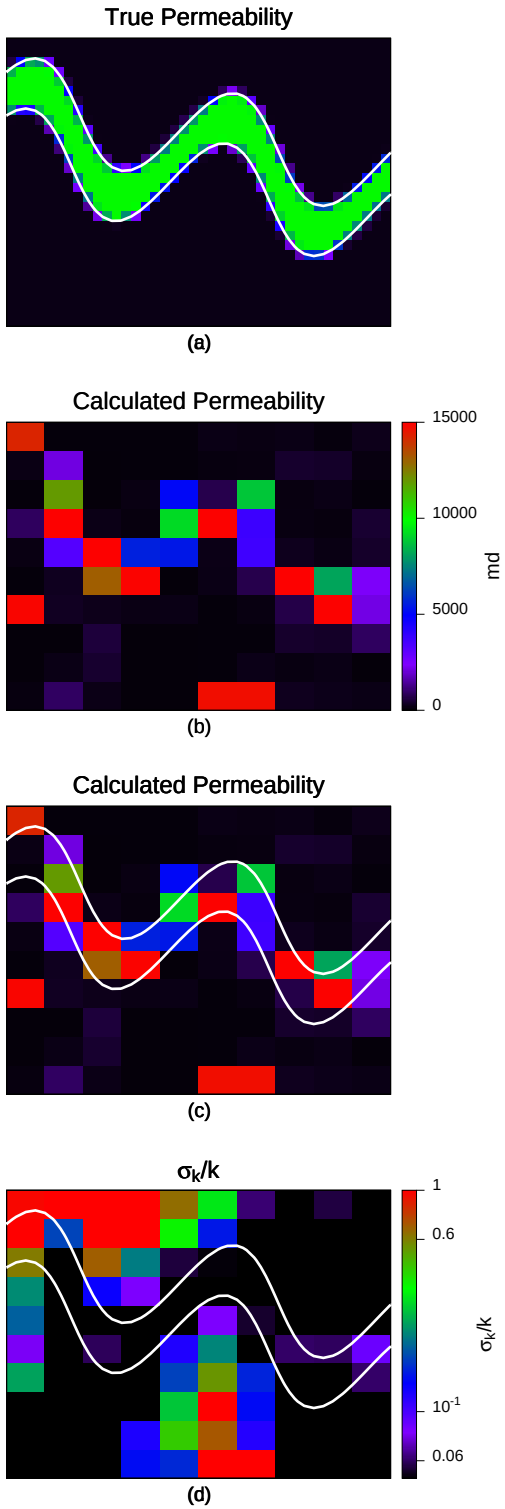



Figure 5.6: Match of the data: channel reservoir matched with large *pixel* model.




(a) – True channel reservoir. (b) – Calculated reservoir with large *pixel* model. (c) – Comparison of true and calculated reservoirs. (d) – $\frac{\sigma_k}{k}$ for calculated reservoir.

Figure 5.7: Comparison and variance of channel reservoir matched with large *pixel* model.

## 5.2 Ellipse Model

In this case the true reservoir was represented by using an ellipse model. The construction of this model was described in Chapter 3 and the reservoir was assumed in single phase only, thus there is no water injection. The information for the inverse problem consisted of one DST and one shut–in pressure at each of the three wells in the reservoir. The ellipse model is relatively simple because the curvature of the boundary is always of the same sign (that was not the case in the channel model). The algorithm converged in a small number of iterations. Fig. 5.8 shows the evolution of the parameters when the *dynamic object* approach was used. In this case the algorithm converged in 14 iterations to the *true* reservoir. In the sequence of plots it is observed how the algorithm developed in this work has the capability to make the object to "float" in the reservoir, that is to change shape, to apply rigid translations–rotations and to change permeabilities.

Fig. 5.9 shows another interesting aspect of this case. Fig. 5.9(b) shows an *exact* match to the *true* reservoir. It may be presumed that in the case of *object* modeling the inverse problem is overdetermined because of the reduced number of parameters in the parameterization scheme and of the large amount of data, and thus there would be a unique solution that would match the data. Fig. 5.9(c) shows that such a presumption would be wrong, since the ellipse shown there results in an almost exact match of the data, the permeabilities inside and outside the ellipse were matched exactly but not the geometry. As an observation, the areas of the calculated and true ellipses are the same, and the distances from the ellipse boundaries to the wells are approximately the same. The DST information seems to be able to resolve the properties between the wells (equal distances) and the shut–in pressure is able to resolve the volume of the reservoir (equal areas). Fig. 5.9(d) shows an attempt to match the data with a very fine *pixel* model (1200 parameters), the data were matched exactly but it was not possible to recover the *true* reservoir (dotted line), this revealed the power of the dynamic object modeling as a way of honoring large scale information. Fig. 5.9(d) also confirms the previous statement about the DST's being able to resolve parameters between wells but not in other regions.

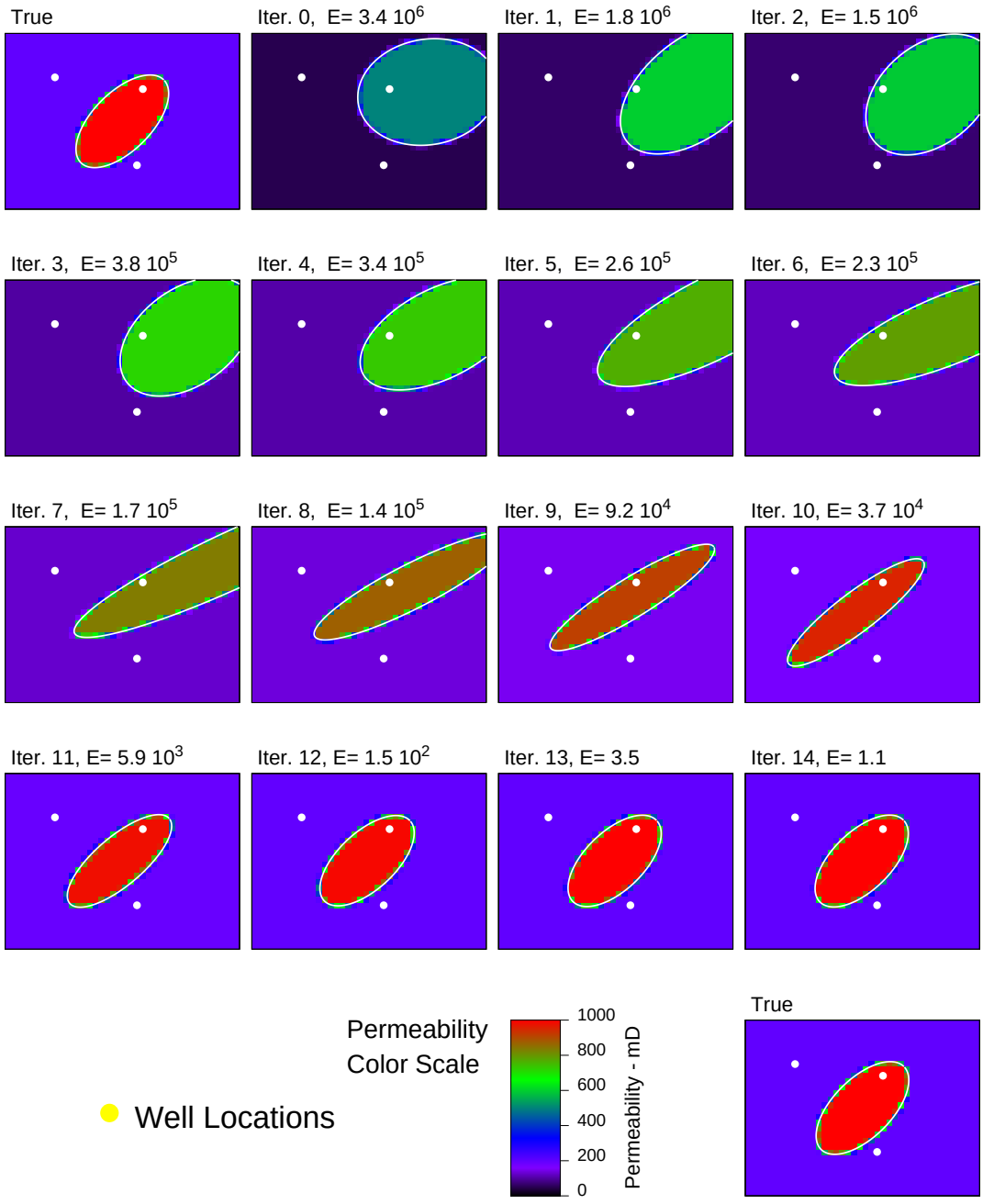
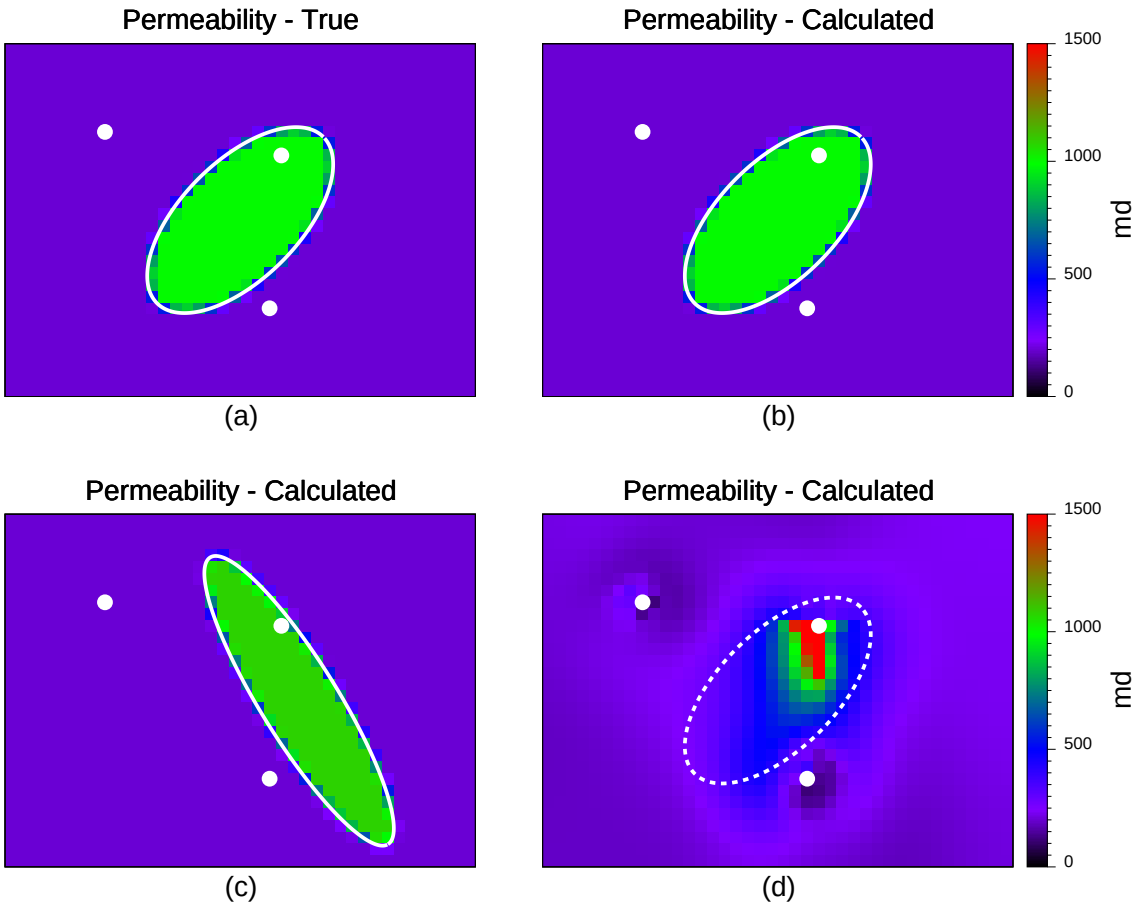



Figure 5.8: Matching history – ellipse model.



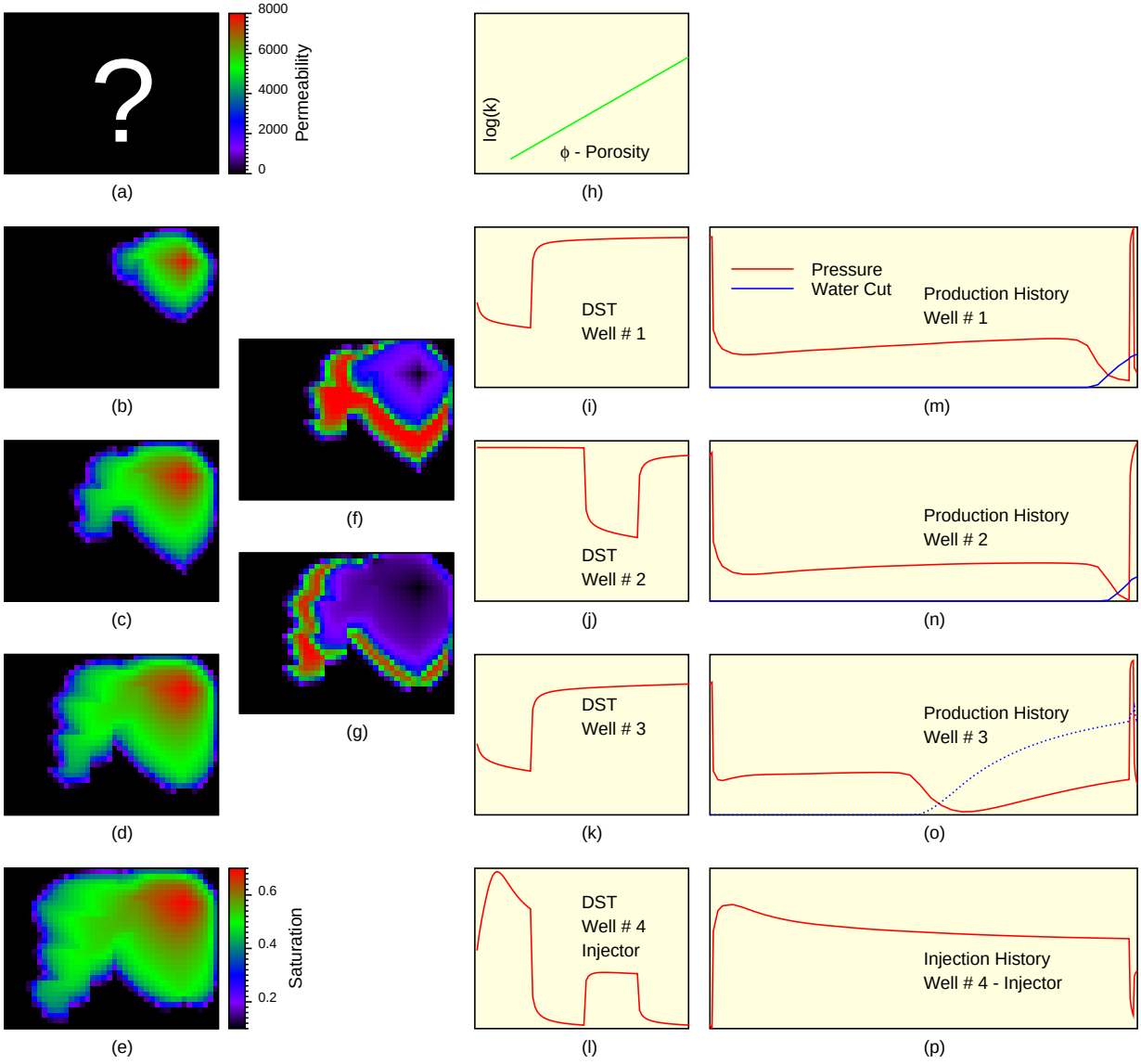

(a) – True permeability distribution. (b) – Calculated permeability distribution with object modeling, exact match. (c) – Calculated permeability distribution with object modeling, second match. (d) – Calculated permeability distribution with pixel modeling.

Figure 5.9: Match of the data: ellipse model, comparison of results.

## 5.3 Permeability and Porosity as Independent Variables Case

In this case the porosity and permeability were considered to be independent variables, and not related to each other as in all the previous cases. Fig 5.10 shows the *field observations* used as input to the parameter estimation algorithm. Fig. 5.11 shows the results of this numerical experiment. (c) and (d) show the *true* permeability and porosity fields that were used to generate the data. (a) and (b) show the calculated permeability and porosity when the permeability–porosity correlation information shown in Fig. 5.10(h) was used; most of the true reservoir values were recovered. Figures (e) and (f) show the calculated permeability and porosity when they are considered as independent variables; the true permeability field was recovered fairly well but not the porosity field. Also it is observed that in some areas the calculated values show high permeability and low porosity and vice versa, this is not consistent with the general reservoir configuration where high permeability is associated with high porosity. This distorted field of porosity has three causes. First, from the mathematical point of view permeability and porosity were considered to be independent variables whereas they are physically correlated. Second, the dimensions of the parameter estimation problem has been doubled which makes the problem more undetermined and higher uncertainty is associated with the parameter estimates. Third, probably the most important cause is that the sensitivity coefficients with respect to the permeability and porosity corresponding to the same area of the reservoir have opposite sign, this was illustrated earlier in Figures 3.7, 3.8 and Figures 3.9, 3.10, thus the optimization algorithm will change the permeability and porosity in opposite directions, and this results in areas with high permeability and low porosity or vice versa. This means that in order to avoid a disparity between the fields of permeability and porosity it will always advisable to have a constraint to impose a correlation between the permeability and porosity variables. Such a constraint can be either *hard* or *soft*; the hard constraint is when permeability and porosity are related through a fixed relationship, such as $\phi_i = \phi_i(k_i)$; the soft constraint is when we constrain the permeability and porosity to be within a certain area

of the permeability–porosity plane, this type of constraint can be enforced by using the *penalty* or *barrier* method presented in Section 2.9




(a): Reservoir geometry. (b)–(e): Water saturation distribution as a function of time. (f): Change of water saturation between (b) and (c). (g): Change of water saturation between (c) and (d). (h): Permeability–Porosity relationship. (i)–(l): Pressure vs. time for DST's. (m)–(p): Long term pressure and water cut vs. time.

Figure 5.10: Data – large pixel model.

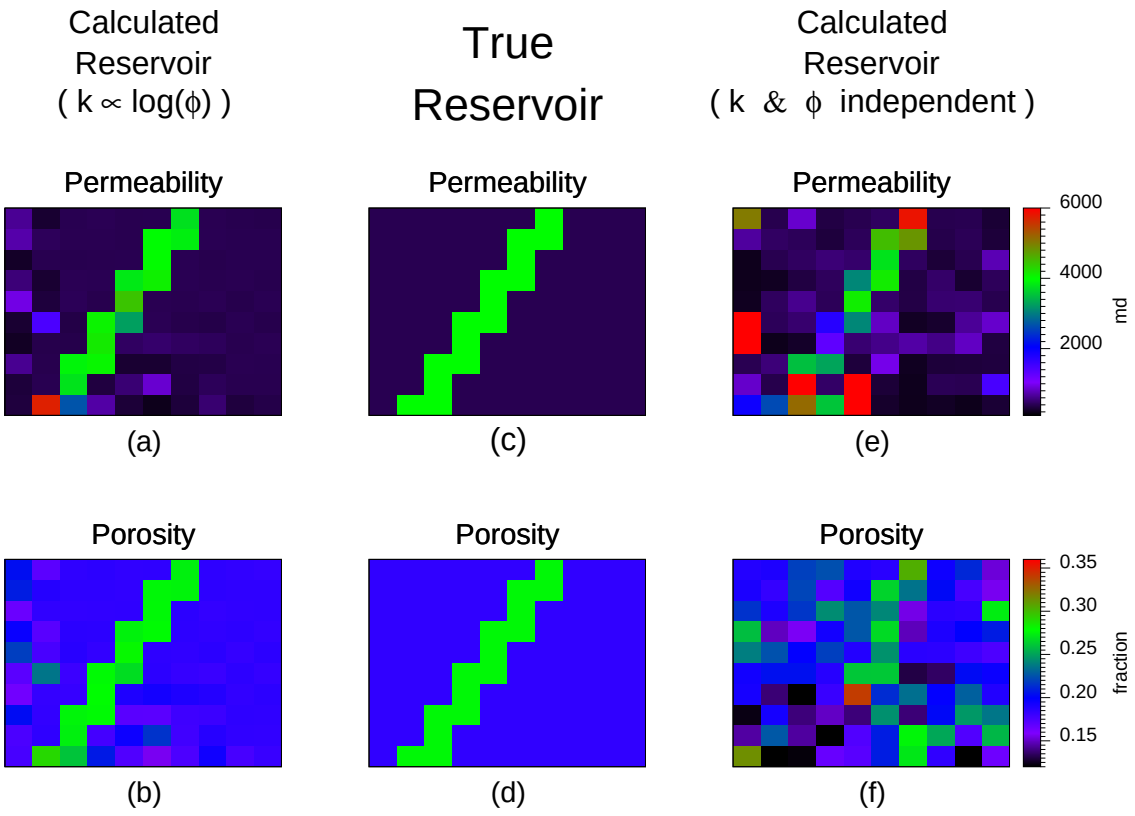



Figure 5.11: Change of saturation match – large pixel model.

## 5.4 Black and White 4–D Information Case

The previous sections presented cases showing the performance of the algorithm under different scenarios. These cases revealed the importance of using the change of water saturation as information. This information was assumed to be *exact*, that is the numerical simulator was loaded with the *true* reservoir model and the calculated change of water saturation was used as input for the parameter estimation problem. This may be regarded as an unrealistic approach. With the current state of 4–D seismic technology it is not possible to calculate the changes of saturation accurately, but it is possible to generate *black and white* maps of change of water saturation in the reservoir. By this we refer to maps that provide binary information, that is *black* means that a change in $S_w$ has been interpreted (but we do not know the magnitude), and *white* otherwise (no change in $S_w$ has been interpreted). This section presents the results of the performance of the algorithm when the 4–D seismic information is provided in the *black and white* format. Fig. 5.12(f)–(p) shows the data that is provided as input to the inverse problem. The main difference from the cases shown in the previous sections is in the quality of the information in Fig. 5.12(f) and (g), the information there is in the discrete *black and white* format.

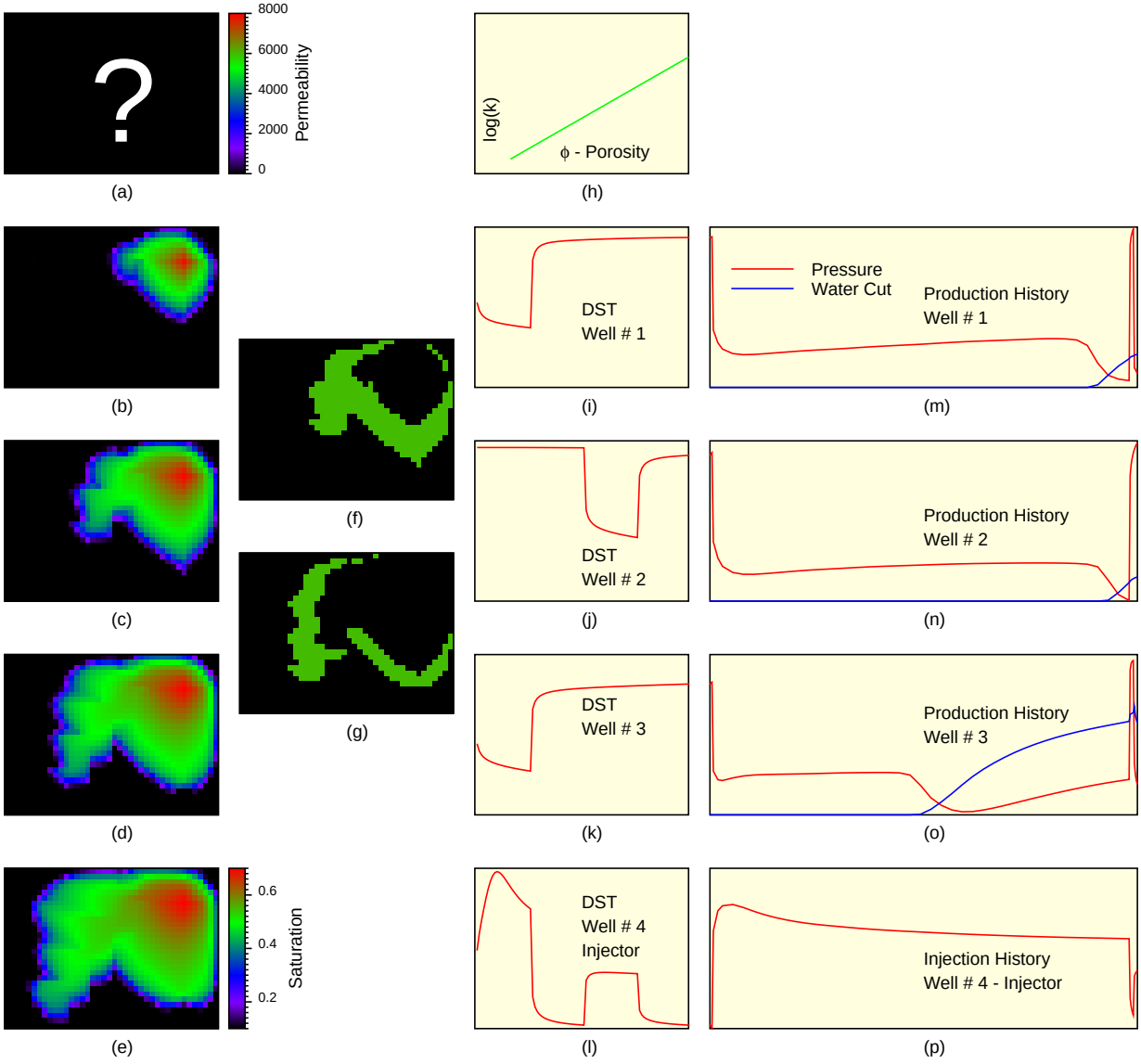
The procedure to create the *black and white* 4–D seismic data is shown in Fig. 5.13. The exact $\Delta S_w$ computed by the numerical simulator as shown in (a) and (c) were filtered to generate (b) and (d) respectively. The filter was defined to assign values as follows:

$$(\Delta S_w)_{filtered} \equiv \begin{cases} 0.00 \equiv \textit{white} & \text{if } (\Delta S_w)_{true} \leq 0.15 \\ 0.30 \equiv \textit{black} & \text{otherwise} \end{cases} \tag{5.1}$$

Fig. 5.14 shows the calculated permeability and the match of the data. It is observed that pressure and water cut were matched almost exactly, and the shapes of the *black and white* $\Delta S_w$ maps were also matched.

Fig. 5.15 shows a comparison of the calculated reservoirs: (a) is the *true* reservoir, (b) is the calculated reservoir when the *exact* $\Delta S_w$ (4–D) information is used, and (c) is the calculated reservoir when the *black and white* $\Delta S_w$ (4–D) information is

used as input. From (b) and (c) we observe that the *black and white* information does not impair the quality of the match to a great extent and thus we may conclude that 4–D information in this coarse format can be used successfully for reservoir parameter estimation.




(a): Reservoir geometry. (b)–(e): Water saturation distribution as a function of time. (f): Change of water saturation between (b) and (c). (g): Change of water saturation between (c) and (d). (h): Permeability–Porosity relationship. (i)–(l): Pressure vs. time for DST's. (m)–(p): Long term pressure and water cut vs. time.

Figure 5.12: Data for inverse problem: Filtered 4–D seismic interpretation case.

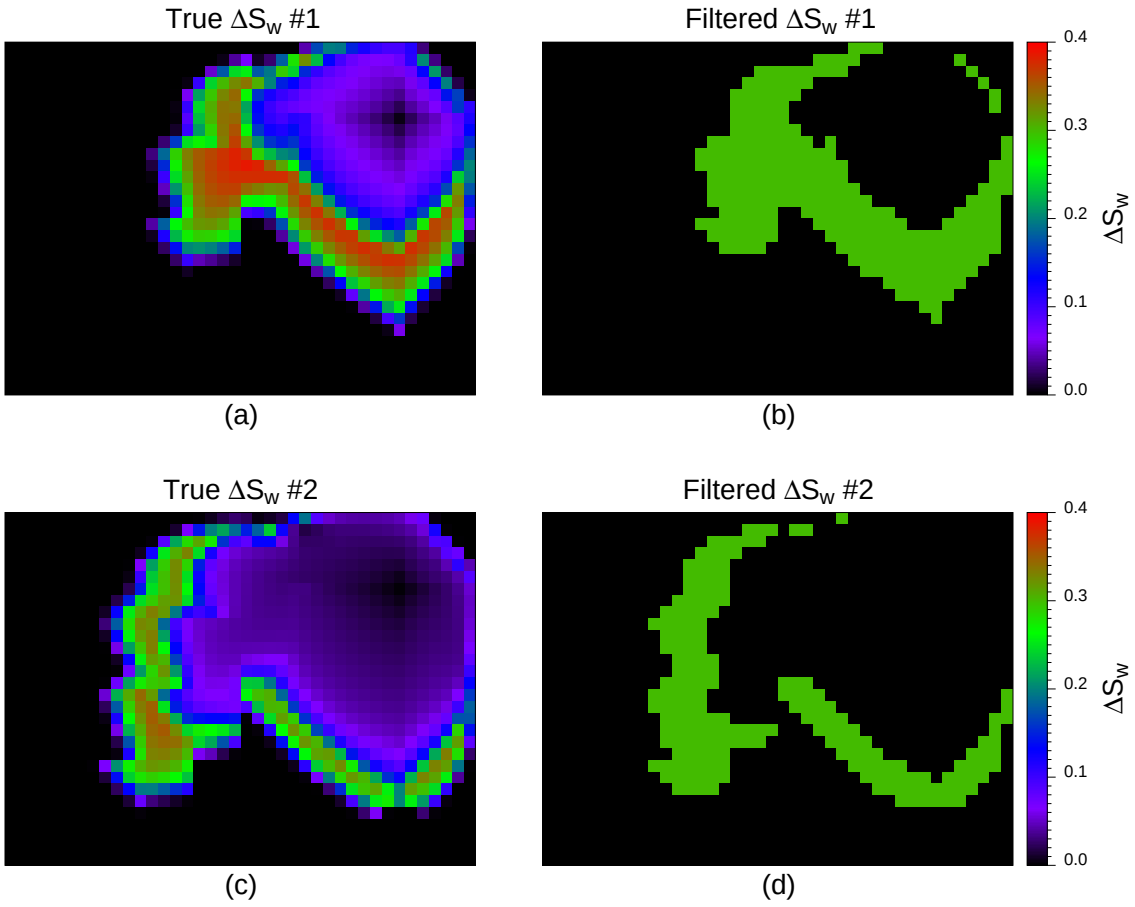



Figure 5.13: Filtering of the $\Delta S_w$ to obtain the 4–D seismic interpretation data for inverse problem.

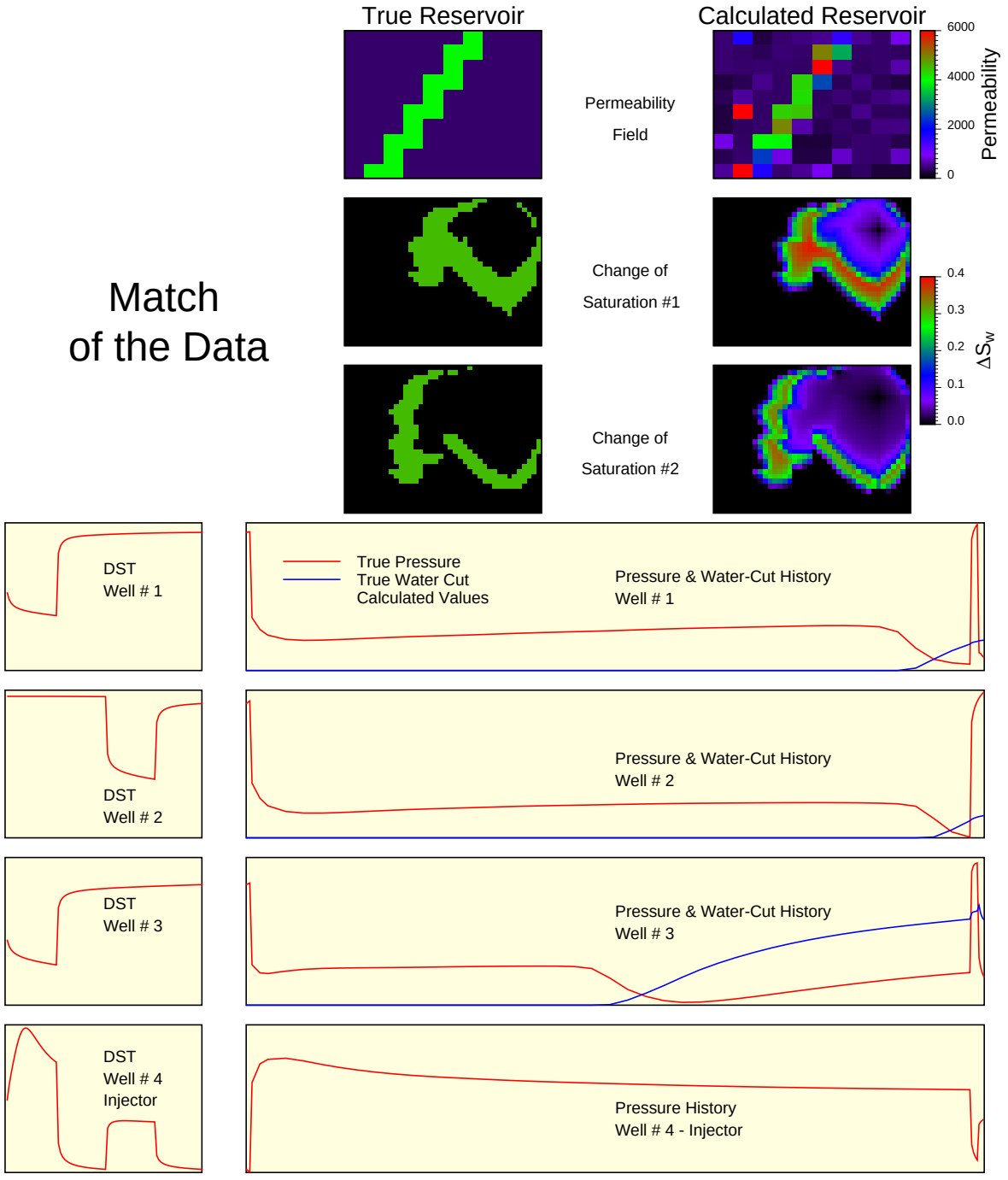
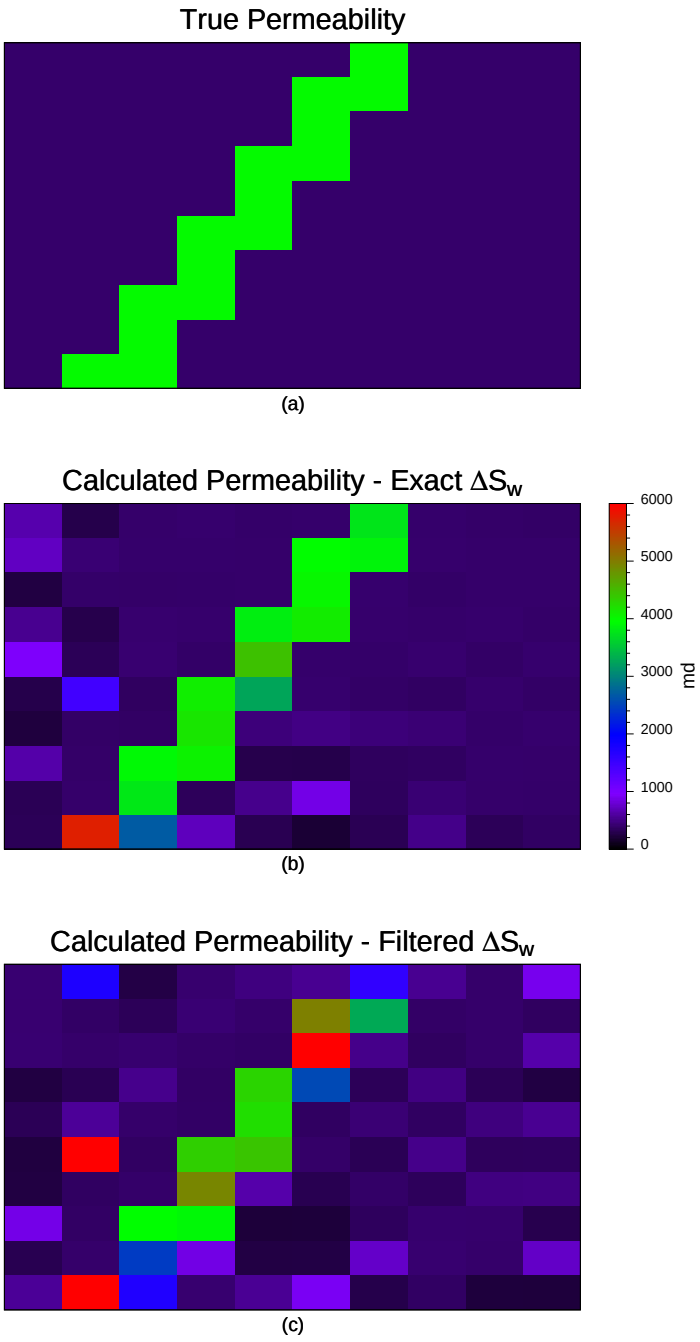



Figure 5.14: Match of the data: Filtered 4–D seismic interpretation case.




(a) – True reservoir. (b) – calculated reservoir when 4–D seismic data is exact. (c) – Calculated reservoir when the 4–D seismic data is provided in a *black and white* format.

Figure 5.15: Comparison of calculated permeability: Filtered 4–D seismic interpretation case.

## 5.5 Kriging Case

This section presents a special case of *static object* modeling that allows us to introduce some *a priori* information, such as the spatial correlation of the parameters. This provides a way to include geostatistical information. This type of information is usually available in the form of a variogram or covariance model for the reservoir, and may be obtained from outcrops or from another reservoir. Oliver et al. [12], [17], [18], [20], [19], [23], [21], [22], and [16] applied a theory developed in the geophysical field [14], and [15] that allows us to include the *a priori* covariance of the parameter information, this is accomplished by using an *objective function* in the form shown in Equation 2.6. There are two concerns in the use of this approach. First, Equation 2.6 is obtained by assuming that the unknown parameters are random variables with a *Gaussian* probability distribution, which means that the approach is applicable only when the permeability is considered as a realization of a random variable with a *Gaussian* probability distribution. Second, the application of this method becomes numerically expensive when the number of simulation cells is large, especially in a *pixel* approach.

In this work the *a priori* information was included following a procedure similar to that used by Fasanino, Molinard, de Marsily, and Pelce [9]. The concept is described here.

If there were a relatively large of permeability and porosity sample points, then it would be possible to estimate the permeability in the rest of the reservoir by using a linear estimation technique, such as kriging, that takes into consideration a predefined spatial correlation of the parameters given in the form of a covariance matrix. In our case we do not have the sample points, therefore we make the sampled permeability values the unknown parameters in the inverse problem. The covariance matrix used by the kriging estimator allows us to introduce the *a priori* information about the spatial correlation.

In this work the linear estimator method was *ordinary kriging*. The *ordinary kriging* interpolation makes an estimation of the parameters at the unsampled locations

as a linear combinations of the data at the sampled locations, that is

$$k_j = \sum_{i=1}^{nsamp} \lambda_{i,j}\, k_i \tag{5.2}$$

where $k_j$ is the permeability in the unsampled location, $k_i$ is the permeability at the $i^{th}$ sampled location, $nsamp$ is the number of sampled locations, and $\lambda_{i,j}$ are the weights to apply to the sampled permeability. The weights are obtained from a process that minimizes the variance of the error of the unsampled parameter constrained to $\sum_{i=1}^{nsamp} \lambda_{i,j} = 1\ \forall j$. The sampled values of $k_i$ are the parameters to be estimated by the nonlinear parameter estimation algorithm. The sampled locations are referred to in this work as *pilot points*. This procedure was implemented as follows:

1. Define the simulation grid (Fig. 5.16(b)).
2. Define the location of *npar* pilot points (Fig. 5.16(c)).
3. Define a variogram model for the permeability in the reservoir (Fig. 5.16(d)).
4. Compute the *ordinary kriging* weights $\lambda_{i,j}$, the $i$'s are the pilot point locations and the $j$'s are the simulation cells.
5. Define *npar static objects* whose identification parameter is $k_i$ .
6. Define the relationship between the parameters and the permeability at each simulation block as:

$$k_j = \sum_{i=1}^{npar} \lambda_{i,j}\, k_i \tag{5.3}$$

7. Launch the nonlinear parameter estimation algorithm.

To compute the *ordinary kriging* weights this work used the program *kb2d* from *GSLIB* [52].

During the nonlinear parameter estimation algorithm the kriging weights remain unchanged. Because the kriging weights can take negative values it is necessary to implement constraints in the parameter estimation process, in this case the constraints

are not on the parameters themselves but on the linear combinations of the parameters, that is there are twice the number of simulation cells constraints, namely:

$$\sum_{i=1}^{npar} \lambda_{i,j}\, k_i \quad \geq \quad k_{min} \quad \forall j \tag{5.4}$$

$$\sum_{i=1}^{npar} \lambda_{i,j}\, k_i \quad \leq \quad k_{max} \quad \forall j \tag{5.5}$$

The procedures to handle these types of constraint were presented in Section 2.9.

The main advantage of the method used here is that it reduces the dimensions of the parameter estimation problem to the number of pilot points, which is much smaller than the number of simulation cells.

In this particular case it is possible to compute the covariance for the permeability distribution in the reservoir. As was presented in Section 2.6.2, we can always compute an approximation to the covariance matrix for the parameter estimates, which in this case are the permeabilities at the pilot points. Because the permeability in the discretized reservoir, that is in each cell of the simulation grid, can be computed as a linear combination of the parameters, then we can write:

$$\vec{k} \quad = \quad \mathbf{A}\,\vec{\alpha} \tag{5.6}$$

where $\vec{k} \in R^{nblocks}$ is the vector of the permeability at each cell of the simulation grid; $\vec{\alpha}\, R^{npar}$ is the vector of parameter estimates, that is the permeability to be estimated at each pilot point; $\mathbf{A}$ is the matrix $(nblock \times npar)$ that provides the linear transformation, and $A_{i,j} = \lambda_{i,j}$ are the *ordinary kriging* weights. Then the $(nblock \times nblock)$ covariance matrix of the permeability in the reservoir, $\mathbf{C}\{\vec{k}\}$ can be computed with Equation 2.86,

$$\mathbf{C}\{\vec{k}\} \quad = \quad \mathbf{A}\,\mathbf{C}\{\vec{\alpha}\}\,\mathbf{A}^T \tag{5.7}$$

where $\mathbf{C}\{\vec{\alpha}\}$ is the $(npar \times npar)$ covariance matrix of the permeability at the pilot points.

The variance, or diagonal elements of $\mathbf{C}\{\vec{k}\}$ can be computed directly as:

$$\sigma^2_{k_i} = \left(\mathbf{C}\{\vec{k}\}\right)_{i,i} = \sum_{j=1}^{npar} \sum_{k=1}^{npar} \lambda_{i,j}\, \lambda_{i,k}\, \left(\mathbf{C}\{\vec{\alpha}\}\right)_{j,k} \tag{5.8}$$

and by taking in account the symmetry in $\mathbf{C}\{\vec{\alpha}\}$ as:

$$\sigma^2_{k_i} = \sum_{j=1}^{npar} \lambda^2_{i,j}\, \left(\mathbf{C}\{\vec{\alpha}\}\right)_{j,j} + 2 \sum_{j=1}^{npar-1} \sum_{k=j+1}^{npar} \lambda_{i,j}\, \lambda_{i,k}\, \left(\mathbf{C}\{\vec{\alpha}\}\right)_{j,k} \tag{5.9}$$

The calculation of the linear approximation to $\mathbf{C}\{\vec{\alpha}\}$ is performed using Equation 2.115.

Another procedure closely related to the kriging–pilot points procedure presented in this section is the one referred to as "the self–calibrated method" by Wen, Gomez–Hernandez, Capilla, and Sahuquillo [53]. This method was developed in the groundwater modeling field, but can be adapted for use in the multiphase problem of petroleum reservoir modeling. The essence of the method is, given a first guess of the reservoir or *realization* obtained with a geostatistical tool, to calculate a perturbation to the field such that its response matches the dynamic data from the real reservoir. The perturbation is calculated as a kriging estimate from perturbations at certain predefined control or *pilot* points. The method assumes that the covariance or variogram of the perturbation field is known. The perturbations of the permeability field ($\Delta k$) at the pilot points are calculated with a nonlinear parameter estimation algorithm. The method as described by Wen et al. is used as a way to refine a set of stochastic realizations generated with geostatistical tools.

The approach developed here can be used not only to calculate the perturbation at the pilot points but also to compute the variance as a measure of uncertainty. The method of introducing a kriged perturbation can be considered as a generalization of the kriging method developed in the first part of this section. That is, when the *first guess or realization* is a zero permeability field everywhere, the "self–calibrated method" becomes the kriging method.

The permeability field in the "self–calibrated method" is calculated as:

$$\vec{k} = \vec{k_0} + \mathbf{A}\,\vec{\alpha} \tag{5.10}$$

where $\vec{k_0}$ is the vector of the permeability field generated with any stochastic method. The term $\mathbf{A}\,\vec{\alpha}$ represents the perturbation to the permeability field, $\vec{\alpha}$ are the unknown perturbations at the pilot points and $\mathbf{A}$ is the matrix with the kriging weights as before.

The parameter estimation has to be performed taking into account the following linear constraints:

$$k_{0_j} + \sum_{i=1}^{npar} \lambda_{i,j}\,\alpha_i \geq k_{min} \quad j = 1, 2, \ldots, nblock \tag{5.11}$$

$$k_{0_j} + \sum_{i=1}^{npar} \lambda_{i,j}\,\alpha_i \leq k_{max} \quad j = 1, 2, \ldots, nblock \tag{5.12}$$

$$\alpha_i \geq -\Delta_{max}k \quad i = 1, 2, \ldots, npar \tag{5.13}$$

$$\alpha_i \leq \Delta_{max}k \quad i = 1, 2, \ldots, npar \tag{5.14}$$

where $\Delta_{max}k$ is the maximum perturbation allowed at the pilot points. In this case the parameters $\alpha_i$ can take negative values.

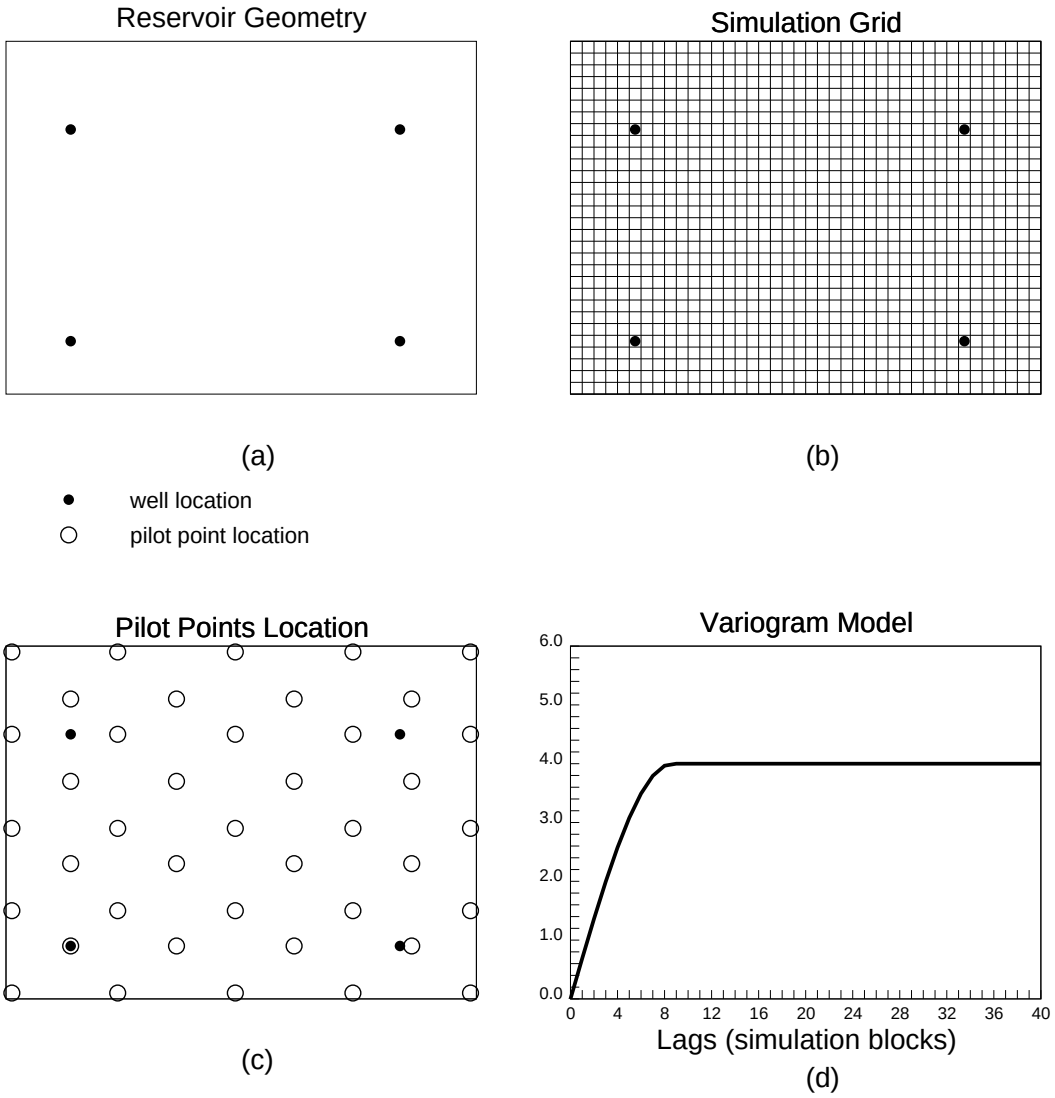
The key step in "the self–calibrated method" is in obtaining the *stochastic realization* that provides a relatively good match to the dynamic data. This realization may be cheap to find when the dynamic data is insufficient to resolve the reservoir, that is when the problem is highly undetermined, this may be the case when the only data are from a few DST's, but it is not clear how to find such a realization when there are a large number of measurements such as in the case of reservoir monitoring where the field observations (dynamic data) consist of sets of DST's, permanent gauge measurements, water cut history and 4–D seismic data. This point can be also inferred from the variance maps shown in Fig. 4.14.

### 5.5.1 Example of Kriging Model

The theory developed in the previous section was examined with a synthetic example. Fig. 5.17 shows the data for the inverse problem. Fig. 5.18 shows the evolution

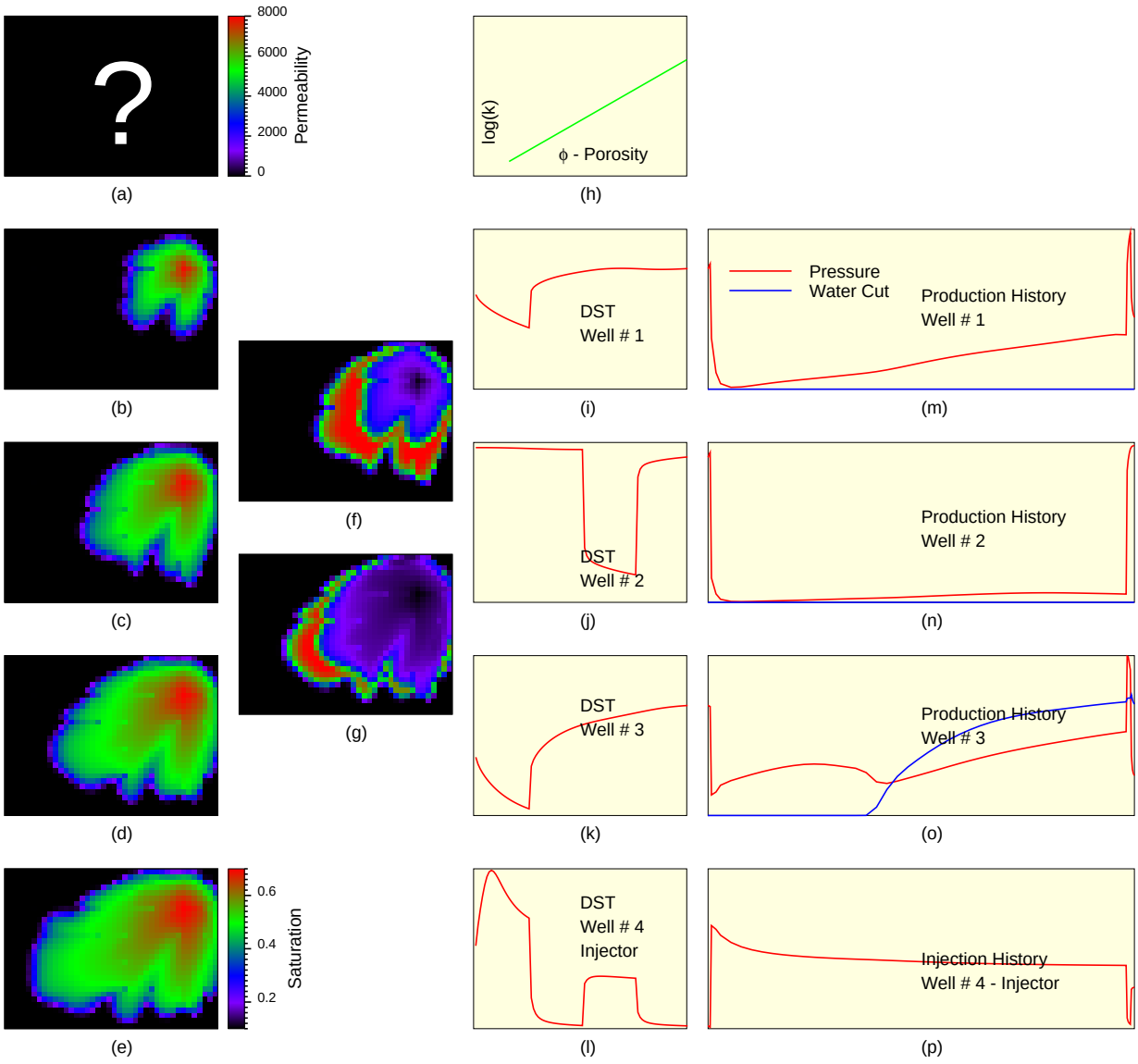
of the parameters. The first frame of Fig. 5.18 shows that the initial guess was a homogeneous reservoir, the method converged after 61 iterations of the Gauss–Newton algorithm. Fig. 5.19(a) shows the locations of the 48 pilot points that were used. The problem was to replicate the dynamic behavior of the reservoir (well test, production history and 4–D seismic data) when the permeability in the reservoir is computed using kriging estimates based on the (unknown) permeability values at the 48 pilot points. Fig. 5.19(b)–(c) shows that we were able to recover the true reservoir almost completely. This was possible because the parameter estimation procedure used the same location and number of pilot points, and the same variogram model as was used to generate the original data. The purpose was to test if the procedure could find the exact answer. Since in real reservoirs are unknown the appropriate locations or number of pilot points and the variogram model, then we can use this procedure to generate several realizations of the reservoir by using different pilot points and variogram models that are considered plausible for that particular reservoir.

This idea was tested by solving the same inverse problem with three different variogram models that differed from the *true* variogram in the azimuth angle $\alpha$. The results are shown in Fig. 5.20. In (b) is shown the calculated reservoir when we used the exact variogram model, $\alpha = 30^o$. In (c) and (d) are shown the results when the model variogram was rotated $90^o$ and $45^o$ from the true azimuth, the maps of the calculated reservoirs show that the calculated permeability is now reoriented taking into account the different azimuth angles but still matching the data. The matches are as good as in the case of the using the *true* model. Fig. 5.21 shows the match of the 4–D seismic data which is the most challenging because of the orientation effects introduced by the changes in the azimuth of the variograms; the maps shows that we were able to compute three very different reservoirs that have a very similar behavior from the dynamic point of view, and thus this confirms the capability of this method to compute different realizations of the same reservoir.




(a) – Reservoir geometry. (b) – simulation grid. (c) – Pilot points for kriging interpolation. (d) – Example of variogram model for kriging interpolation.

Figure 5.16: Pilot points for kriging interpolation.




(a): Reservoir geometry. (b)–(e): Water saturation distribution as a function of time. (f): Change of water saturation between (b) and (c). (g): Change of water saturation between (c) and (d). (h): Permeability–Porosity relationship. (i)–(l): Pressure vs. time for DST's. (m)–(p): Long term pressure and water cut vs. time.

Figure 5.17: Data for kriging model example.

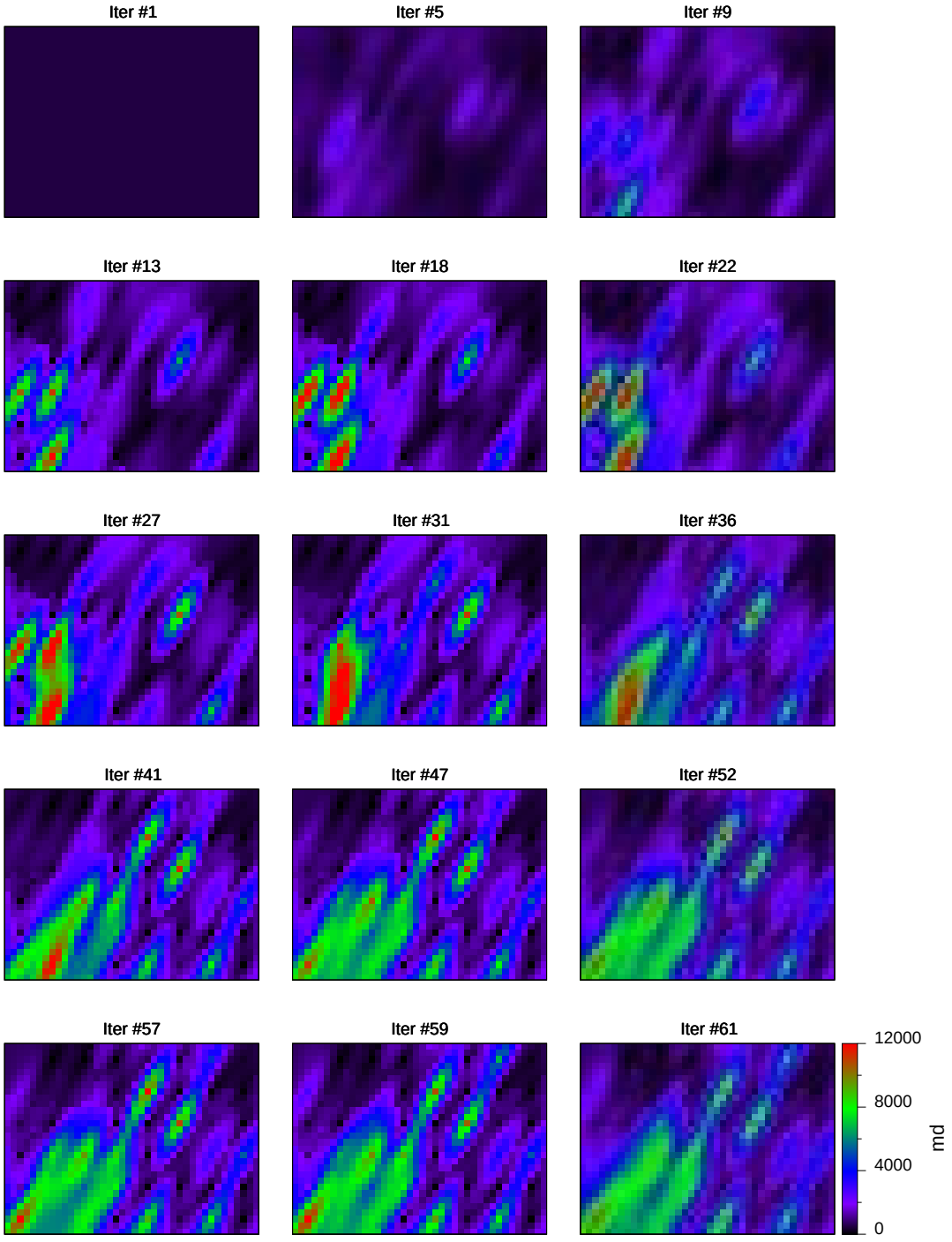
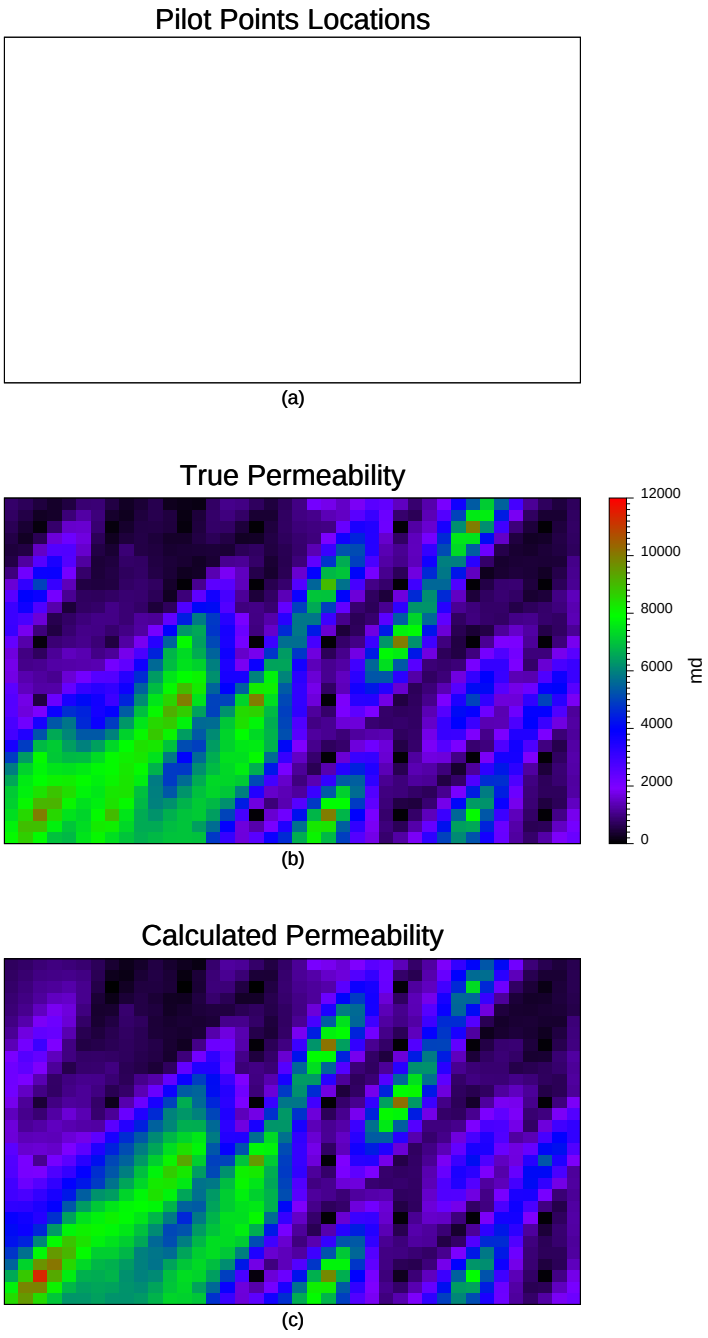
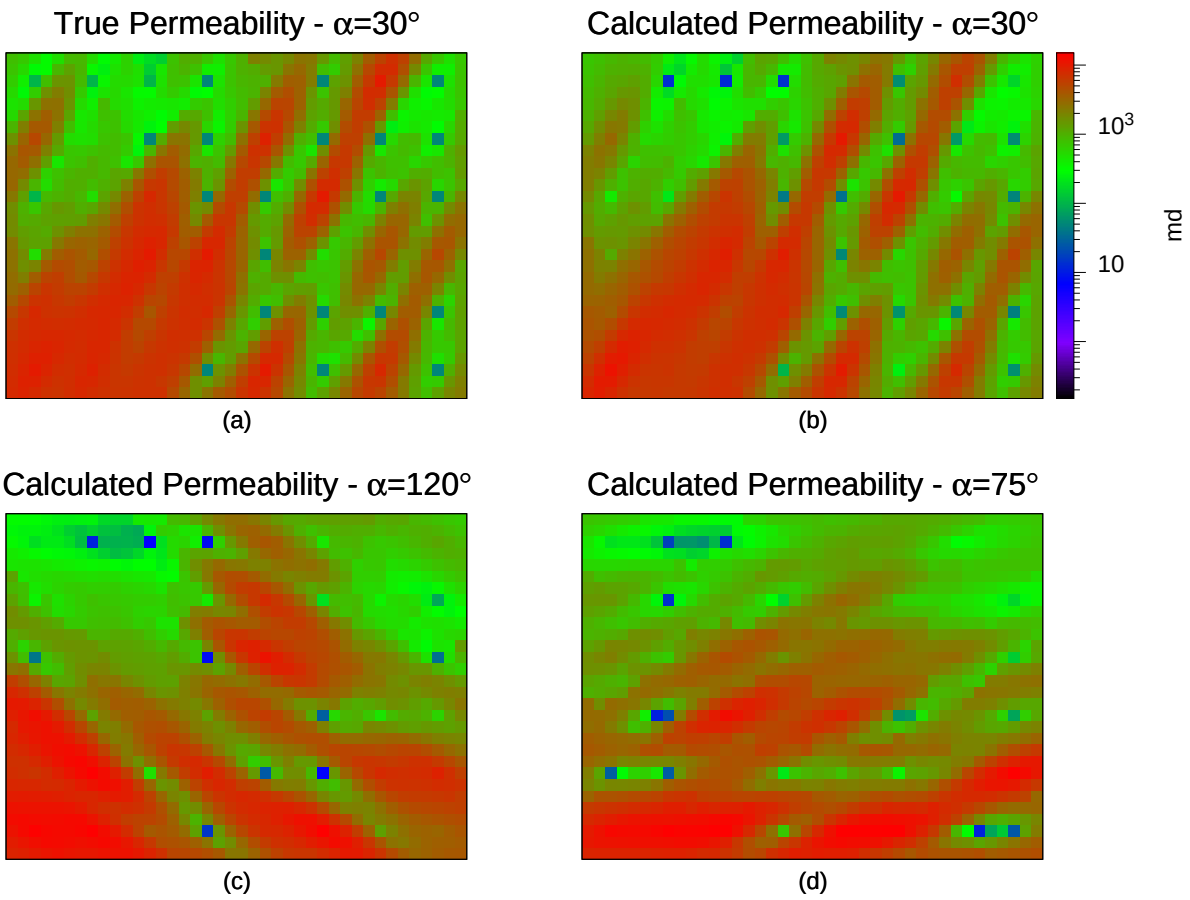



Figure 5.18: Matching history – kriging model.




(a) – Pilot points location. (b) – True reservoir. (c) – Calculated reservoir.

Figure 5.19: Match of the true reservoir – kriging model.




(a) – True reservoir. (b) – Calculated reservoir with the right variogram model. (c) – Calculated reservoir with a variogram rotated 90 deg. (d) – Calculated reservoir with a variogram rotated 45 deg.

Figure 5.20: Comparison of calculated reservoirs using different variogram models – kriging model.

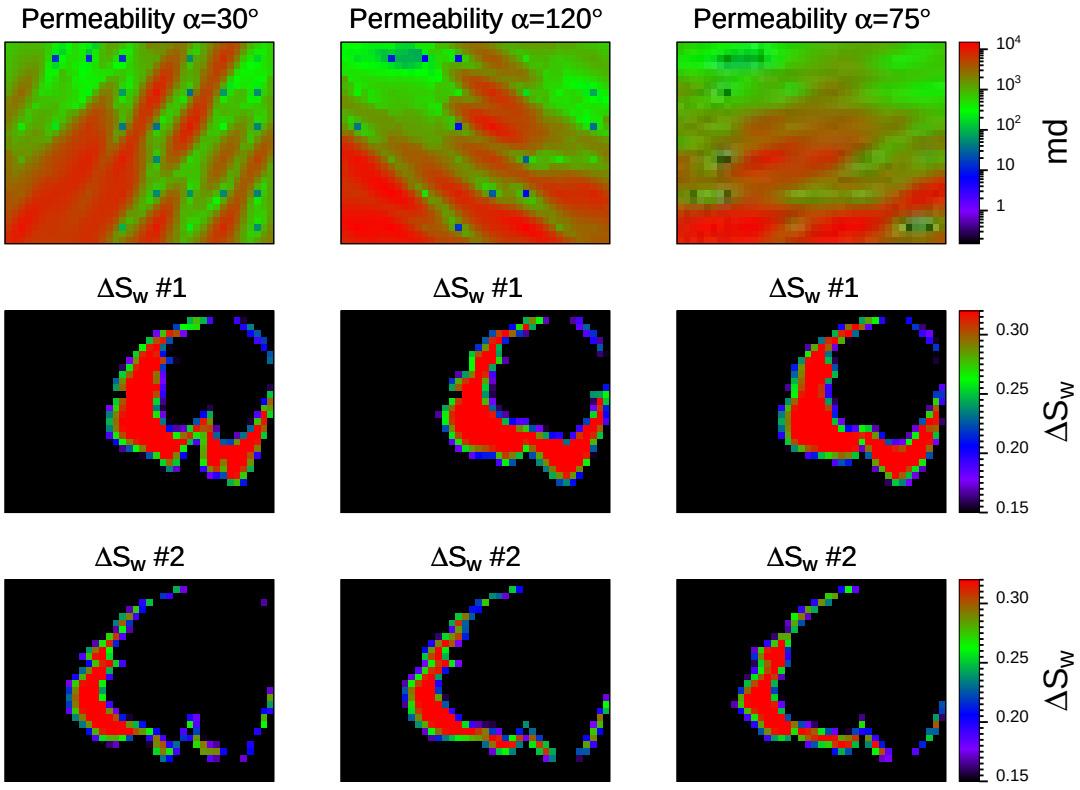



Figure 5.21: Comparison of calculated reservoirs using different variogram models, match of the 4–D data – kriging model.

## 5.6 Fault Model

This section illustrates the use of *dynamic object* modeling to locate the position of a fault in a reservoir.

As a first approximation a fault can be modeled as a rectangle. Fig. 5.22 shows how the reservoir and the object rectangle can be parameterized:

$\alpha_1$ = permeability in the reservoir.
$\alpha_2$ = permeability inside the rectangle.
$\alpha_3$ = length of the rectangle.
$\alpha_4$ = width of the rectangle.
$\alpha_5$ = x–coordinate of the center of the rectangle.
$\alpha_6$ = y–coordinate of the center of the rectangle.
$\alpha_7$ = rotation of the axis of the rectangle.
$\alpha_{8-11}$ = skin factor at each well.

In order to simulate the fault $\alpha_2$ and $\alpha_4$ were set to be constant, that is they were not active parameters in the inverse problem. The permeability inside the fault ($\alpha_2$) was assigned a very low value ($10^{-5}$ md); the width, ($\alpha_4$) was set to 2.5 simulation grid cell units.

With this approach the fault can *float* in the reservoir, all the parameters except the width and the inside permeability could take any value within a specified range. Thus the problem is to locate the position of the fault by using well test data, production history and 4–D seismic information. In the model, the fault was allowed to be partially outside the domain of the reservoir. Fig. 5.23 summarizes the data for this problem.

From a visual inspection of the data it is almost trivial to see the approximate location of the fault, but in this exercise it was decided to start with a first guess that results in a substantially different saturation distribution with the only purpose to investigate the difficulties from the mathematical point of view. Fig. 5.24 shows the evolution of the estimates for the location of the fault and the water saturation at the end of the simulation; the water saturation is shown only to facilitate the

understanding of the pictures, the fault is *moved* by matching the well test data, production history and 4–D seismic information (not the water saturation). The first frame corresponds to the first guess, in the successive frames the fault, first moves to the East, gets longer and turns to become almost horizontal. The movement to the right stops after the water is completely shut-off, then the fault moves to the North and rotates to finally reach a position very close to the *true* location. It is seen that the movement to the North and the rotation are relatively small at each iteration. The reason is that the fault edge is sliding over the East boundary, and this is a very unstable process. This is illustrated in the *linear search* frame of Fig. 5.24 which shows a case when the algorithm tried to apply too large a rotation to the fault and this resulted in water flow around the end, this produced a large increase in the objective function and hence the rotation was rejected. This part of the path in the parameter space is along a sharp valley, so a movement in the wrong direction produces a large increase in the objective function.

This example illustrated that it is possible to handle the problem of reservoir compartmentalization as a reservoir inverse problem. The fault model shown in this case it is probably too simple. In a real case would be necessary to have the capability of dynamic grid refinement around the fault in order to make the width of the fault much thinner.

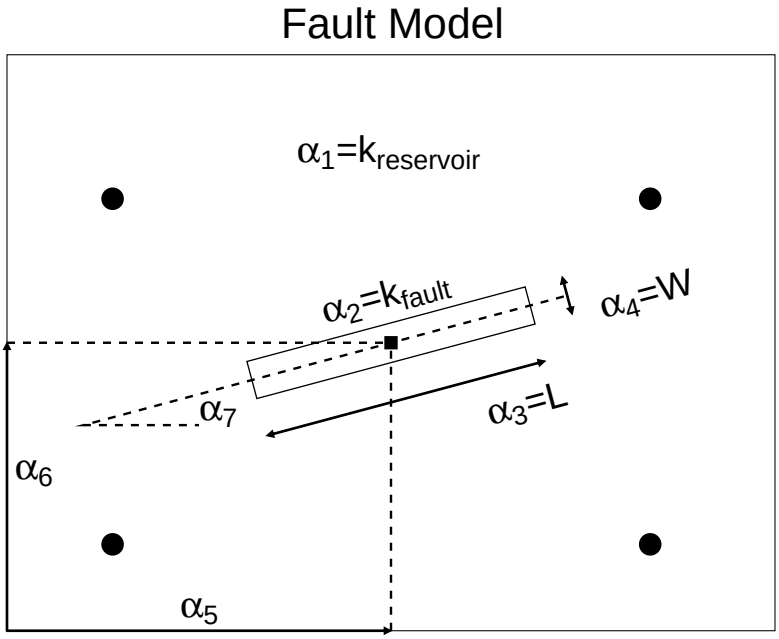
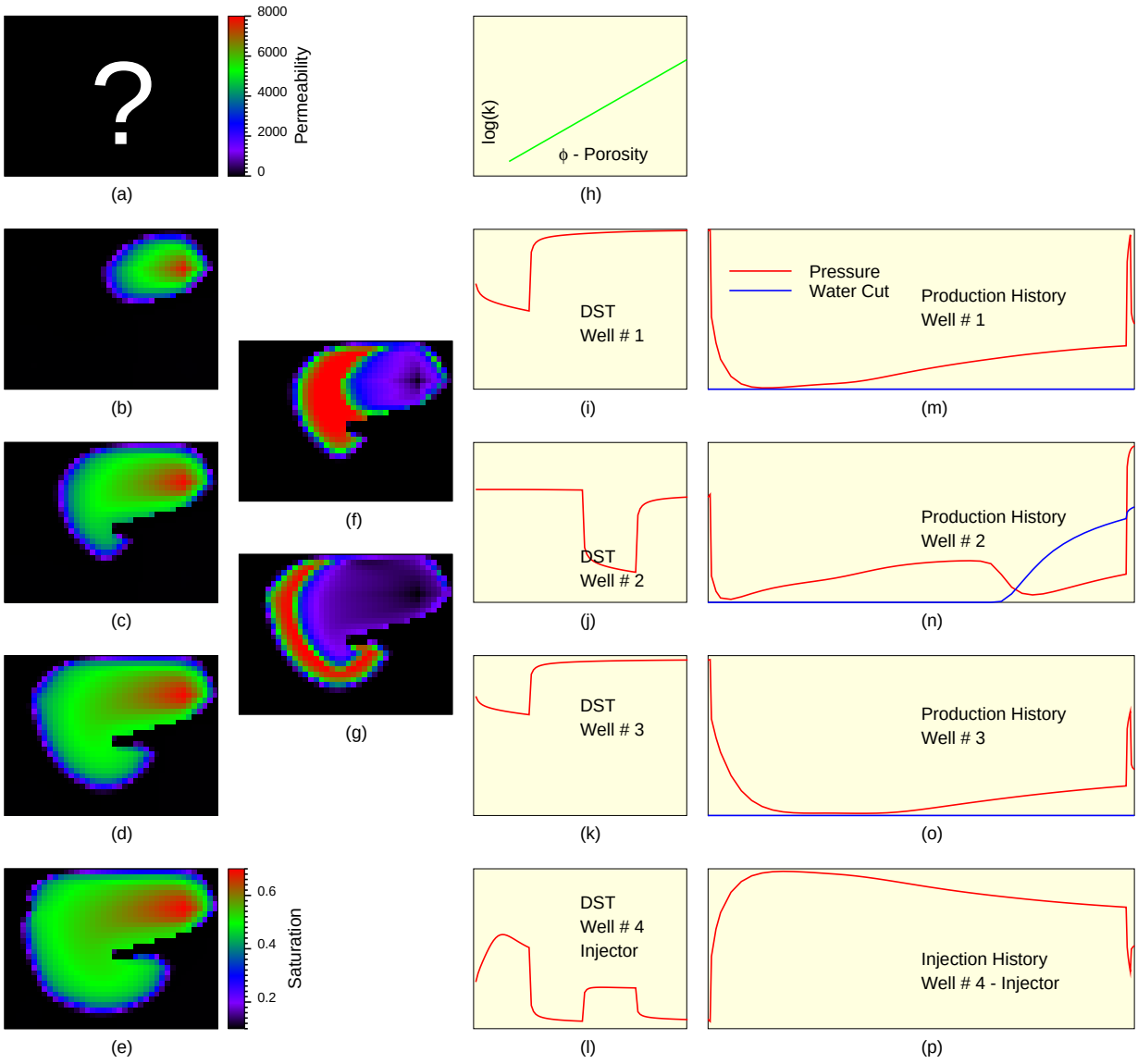



Figure 5.22: Fault model.




(a): Reservoir geometry. (b)–(e): Water saturation distribution as a function of time. (f): Change of water saturation between (b) and (c). (g): Change of water saturation between (c) and (d). (h): Permeability–Porosity relationship. (i)–(l): Pressure vs. time for DST's. (m)–(p): Long term pressure and water cut vs. time.

Figure 5.23: Data for fault model.

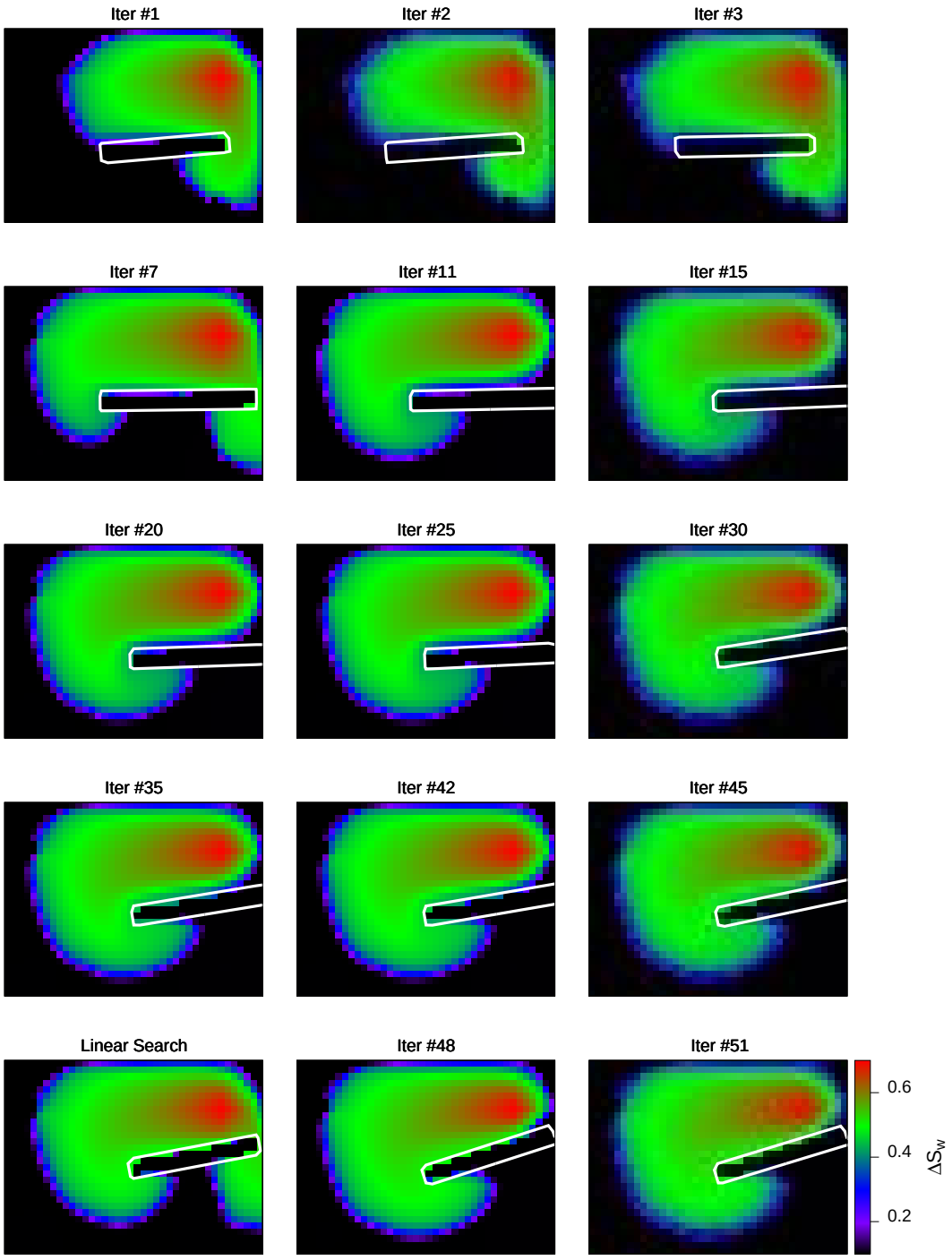



Figure 5.24: Matching history – fault model.

## 5.7 3–D Seismic Data Case

This section illustrates the use of interpreted 3–D seismic data along with dynamic data for reservoir parameter estimation.

Fig. 5.25 shows a 16 color map that represents an hypothetical interpretation of a 3–D seismic survey. This figure is an adaptation of a true interpreted 3–D seismic survey shown as *Plate 15 (b)* in Ref. [54]. We will use this map as data for the case shown in this section.

We can associate each of the 16 colors in Fig. 5.25 with a permeability/porosity number but we do not know the value. The problem now becomes in finding the value that we should assign to each color of the interpreted 3–D seismic data that results in a match of the dynamic and static data from the reservoir.

In order to generate the synthetic data we assigned each color a permeability value that was considered reasonable for the channel environment depicted in Ref. [54]. The porosity was calculated using the permeability–porosity relationship used in the previous examples to simulate measurements in cores. The *true* distributions of permeability and porosity that were used to generate the dynamic data are shown in Figures 5.26 and 5.27. In this case there were 10 wells in the reservoir and their location were as shown in Ref. [54]. The wells producing oil were indicated with a triangle, the wells injecting water were indicated with a solid dot.

In order to replicate the case of a new reservoir, where the only information is the 3–D seismic and pressure measurements, the data used in the inverse problem consisted of field observations from one DST and from permanent gauges at each well over a 200 days period. Also the geometry of the reservoir was adjusted to avoid the water front reaching any of the oil producing wells. The simulation grid consisted of a 30 x 30 mesh.

Fig. 5.28 shows the evolution of the calculated permeability distribution as a function of the number of iterations. In this case the first iteration, which corresponds to an initial guess, consisted of a homogeneous reservoir; this was a very unrealistic guess since we can start always with a guess that at least takes into account some *a priori* information such as the relative magnitude and ordering of the parameters,

but in this experiment we were interested in investigating if it was possible to recover the ordering of parameters. In this case there were 16 parameters to estimate, one for each color from the 3–D seismic interpretation. The procedure was able to recover the true reservoir permeability distribution in 32 iterations. The final match, that is the one corresponding to iteration #32 is shown in Fig. 5.29(b).

One observation about this experiment is that even though the dynamic information was relatively scarce (pressure data only) the number of wells (10) was of the same order of magnitude as the number of parameters (16) thus we can expect not to require a great effort to converge to the true reservoir. This can be explained from the mathematical point of view: the sensitivity coefficients corresponding to the color regions intercepted by the wells are much larger than the others, and because the of ratio colors to wells is small then we may expect that most of the colors will be well determined. Thus a second experiment was designed to investigate this issue.

The second experiment consisted of the same true reservoir and dynamic data, in this case we also had to recover the permeability corresponding to each of the 16 colors, but we also added another 10 parameters that represented the permeability in each well block. In other words the number of parameters was increased from 16 to 26. Fig. 5.29(c) shows the calculated reservoir after 62 iterations. This time only the major features of the reservoir were recovered. The permeabilities in the well blocks were recovered exactly. Also in this case some of the estimated parameters were in the wrong order, these were the ones located farthest from the wells. This second experiment reveals the enormous value associated with the connectivity information available from the 3–D seismic interpretation. The procedure was much less effective when we ignored the knowledge that the wellbore permeabilities had counterparts in other parts of the reservoir.

As a conclusion the procedure developed in this work seems to be useful for resolving reservoir inverse problems including 3–D seismic information.

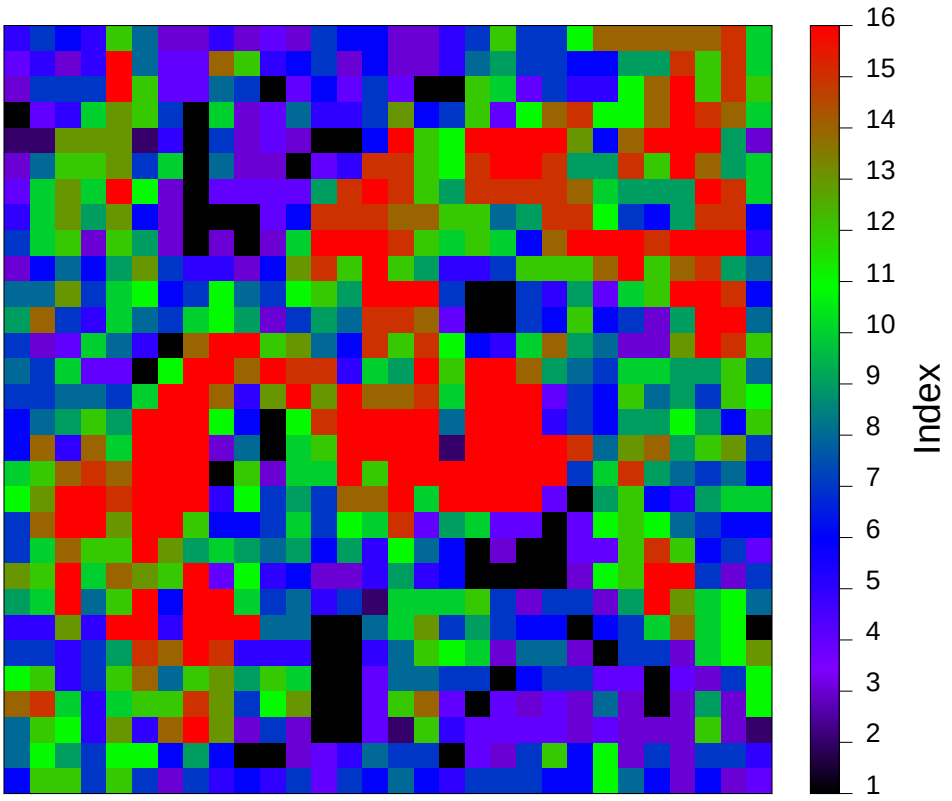



Figure 5.25: Interpreted 3–D seismic data presented as a 16 colors map.

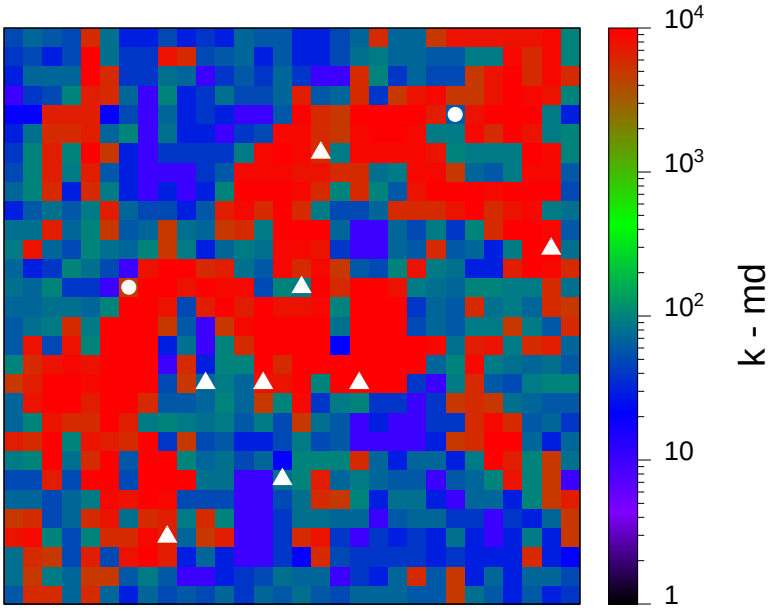



Figure 5.26: True permeability – 3–D seismic data case.

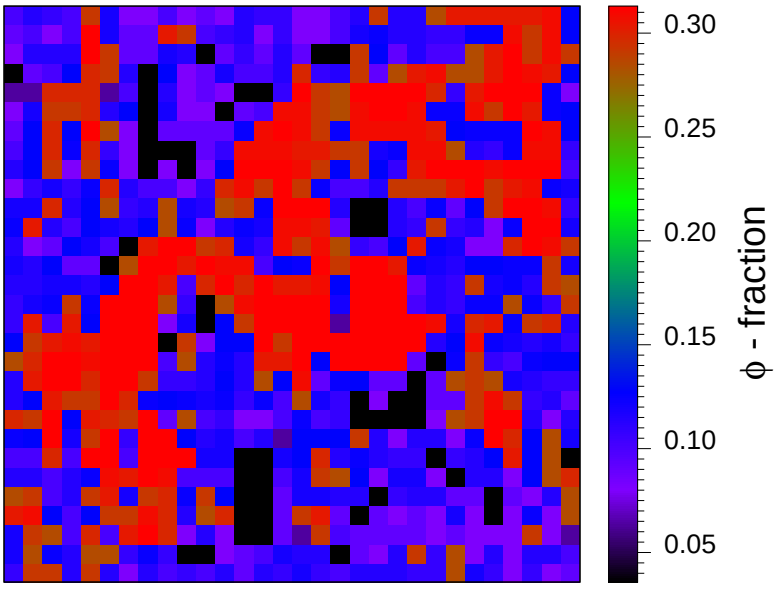



Figure 5.27: True porosity – 3–D seismic data case.

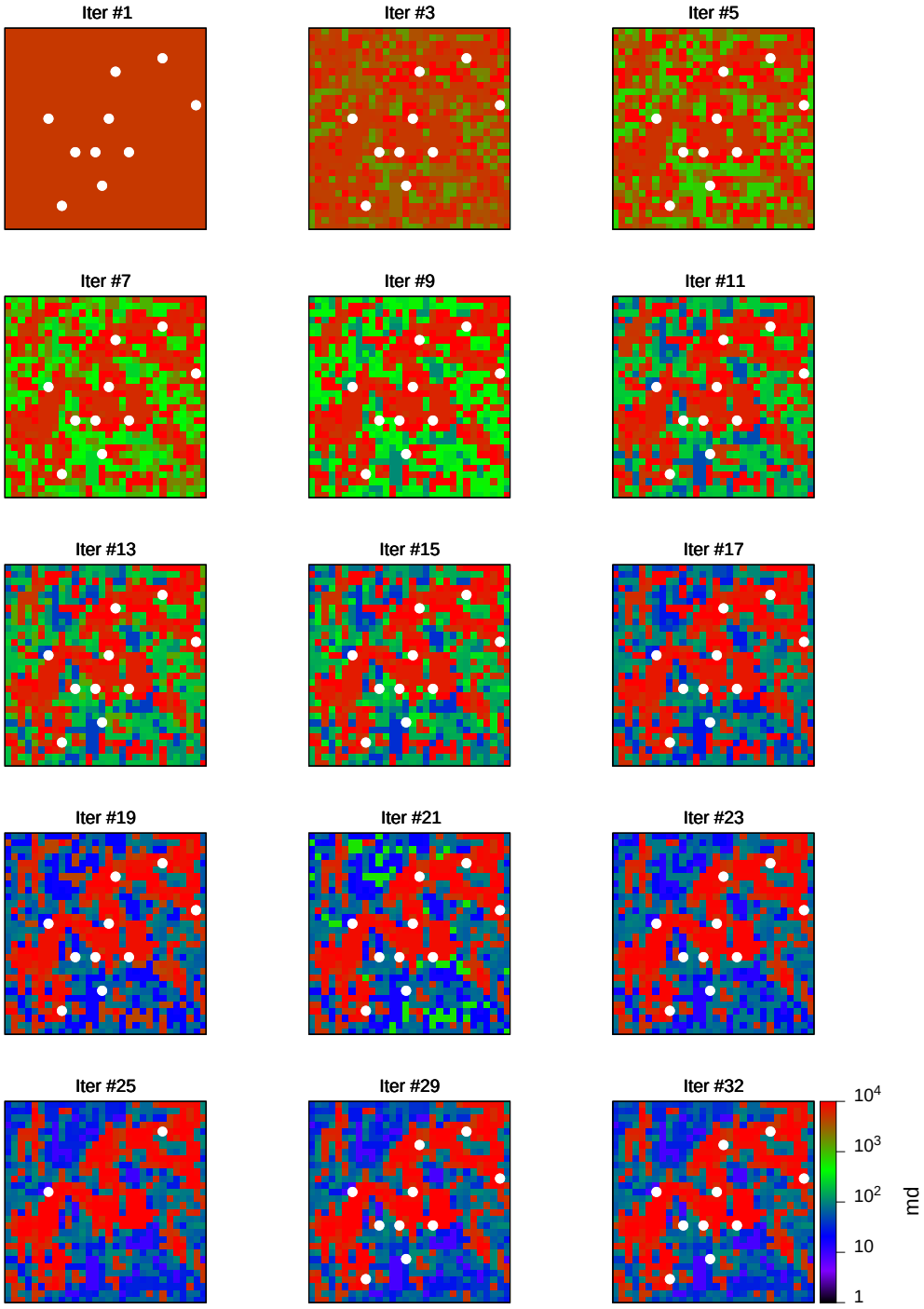
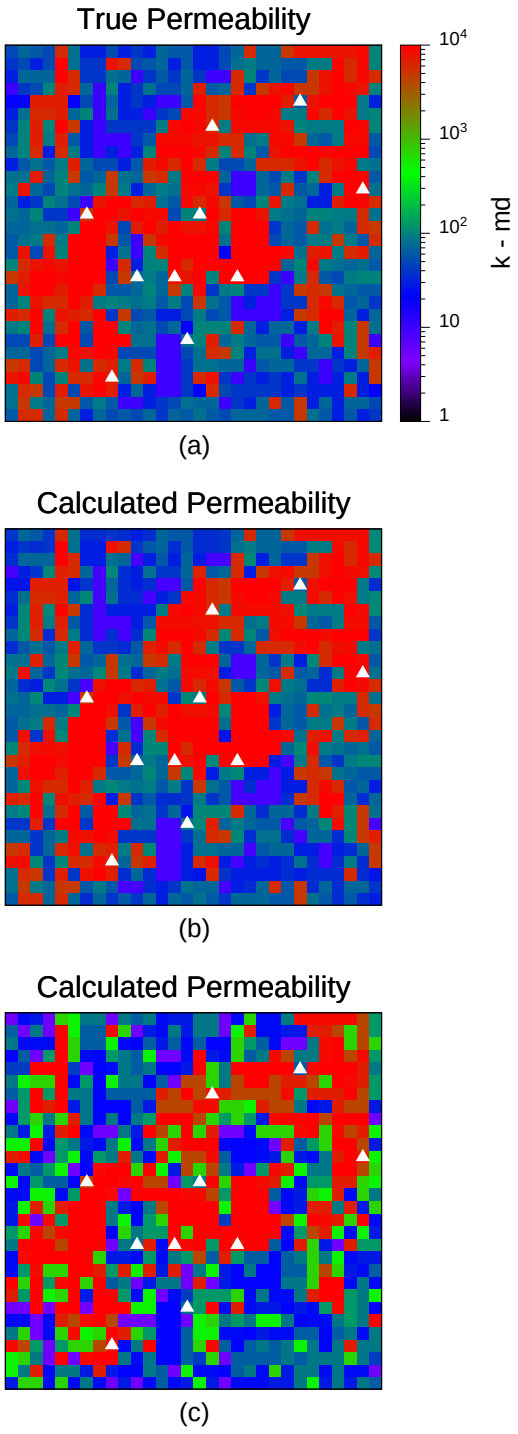



Figure 5.28: Matching history – 3–D seismic data case.




(a) – True reservoir. (b) – Calculated reservoir considering 16 parameters. (c) – Calculated reservoir considering 26 parameters.

Figure 5.29: Match of the true reservoir – 3–D seismic data case.

## 5.8 Field Case

In the previous sections the method developed in this work for reservoir parameter estimation was tested with synthetic cases, that is the *field observations* were generated by the numerical model of the reservoir, thus the data was controlled and the *true* reservoir was known. This approach allowed us to develop and test the method in a *perfect* environment.

The method developed in this work was also applied to an actual case, using data from an actual field. The purpose of this exercise was to explore the difficulties of handling actual data. The results were presented by Landa, Kamal, Jenkings, and Horne [29] in 1996.

The field is located in offshore Indonesia, and has been under commercial production since 1994. The reservoir is composed of sand and shales deposited in a river/tide dominated delta. Most of the production is single phase dry gas from a single layer. Fig. 5.30 shows a map of the reservoir. The locations of the wells under production and the location of the faults are marked on the map as they were interpreted by the geologist of the company that operates this field. The reservoir boundary at the West is determined by the water–gas contact between the gas pool and an aquifer. The method developed in this work was applied to investigate a small area of the field, which was referred to as the *area of study* as shown in Fig. 5.31. There were three wells under production in the area of study, and they are referred to as wells #1, #2, and #3. The boundaries of the area of study were set as follows:

- North. Fault #1.
- East. An arbitrary line drawn between well #1 and the nearest producing well to the East.
- South. Fault #2.
- West. An arbitrary line drawn to the West of well #3.

The purpose of this exercise was to estimate a distribution of permeability within the *area of study* consistent with the data available at the time.

The information available for this study is summarized in Table 5.1.

Table 5.1: Field case – summary of information available for area of study. (From Ref. [29]).

| *Well* | *E-Logs* | *Cores* | *DST* | *Bottom Hole Shut–in* | *Production History* |
|---|---|---|---|---|---|
| #1 | yes | yes | 1 | 3 | yes |
| #2 | yes | | 1 | 2 | yes |
| #3 | yes | | 1 | 2 | yes |

There was no geostatistical information available.

### 5.8.1 Analysis of the Data

Fig. 5.32 shows the production history. The information corresponded to nearly 500 days of production. The figure also indicates the major events at each well, that is the DST's and shut–in surveys. The shut–in information consisted of the *3 day extrapolated pressure*, that is the well was closed at the surface for a 12–48 hour period while the bottom hole pressure was recorded, then the pressure recordings were analyzed in the field and a single value was reported as the *3–day extrapolated pressure* along with the cumulative production for the well. Fig. 5.33 shows the DST information as it was made available from the company's records. The DST's were performed during the completion of the wells and before any commercial production. The shut–in pressure surveys were performed approximately every six months at each well.

The first step in the analysis was to transform the data to a uniform system of units and to a time reference. The time zero was set at the start of the first DST in well #1.

The next step was to perform an analysis of the data by using traditional reservoir engineering approaches. The results of these analysis *were not used* as input to the parameter estimation procedure. The purpose of this traditional type of analysis was to gain understanding of the reservoir and also to interpret the results of the application of the method used in this work.

The DST information was analyzed with conventional tools for pressure transient analysis. Fig. 5.33 shows the *diagnostic plots* also known as *log–log plots* for the final build–up of the DST's. A qualitative interpretation of these plots follows:

- Well #1. The derivative curve shows a slope of $\frac{1}{2}$ which can be explained as linear flow, as in a channel. The downward trend at the end can be attributed to either the effects of a constant pressure boundary or to increasing permeability as we move away from the well.
- Well #2. The approximately constant derivative can be attributed to a fairly homogeneous permeability field.
- Well #3. The downward slope in the derivative can be attributed to an increase of permeability as we move away from the well.

The shut–in pressure was used to construct a plot of $(p/z)$ vs. cumulative production, the plot is shown in Fig. 5.34. The plot suggested that there was a mechanism that provides pressure support to the wells. This could be attributed to the aquifer acting at the western boundary of the reservoir.

### 5.8.2 Mathematical Model

The next step was to construct a mathematical model of the problem. A two dimensional, single phase simulation grid was constructed to compute the pressure in the wells. The reservoir was discretized within the area of study with a 10 x 32 grid. The eastern side was set to be a *no flow* boundary because of the wells producing to the East of well #1. It was not clear if fault #1 could be considered to be a *no flow*

boundary, therefore the simulation grid was extended to the North but with a much coarser block size than in the area of study. The southern boundary was modeled as *no flow* because fault #2 was considered by the geology interpretation to be a sealing fault. In order to account for the aquifer, which as noted in the interpretation of the $p/z$ plot seemed to have an effect on the area the study, the simulation grid was extended to the West, to the water–gas contact. The extended grid was modeled with a much coarser block size than in the area of study, and the aquifer was modeled as a constant pressure boundary. The effect of the constant pressure boundary provided a key element in the inversion problem since this allowed resolution of the permeability in the western part of the area of study.

The fluid properties were estimated using the correlations normally applied in well test analysis.

### 5.8.3 Parameterization

Two approaches were applied to parameterize the inverse problem: *pixel* and *object* modeling.

- *Pixel* modeling. The purpose was to compute the permeability and porosity of each block of the simulation grid. In order to include the information from cores, the porosity was related to the permeability with a fixed correlation of the type:

$$\log(k) = m\,\phi + a \tag{5.15}$$

  This correlation was obtained from the cross plot of the measured permeability and porosity. This plot is shown in Fig. 5.35. By making the porosity a function of the permeability we also reduced the number of unknowns by a factor of two. The skin factors at each well were set as parameters to be estimated. Thus the inverse problem consisted of estimating 330 permeabilities and three skin factors.

- *Object* modeling. A channel model was chosen to parameterize the reservoir. The development of this model was presented in Section 3.3.3 and shown in Fig. 3.24. The porosity and permeability were assumed to be linked by a fixed relationship, of the type shown in Equation 5.15. Thus the parameters to estimate were two permeabilities (inside and outside the channel), six parameters that define shape and location of the channel, and three skin factors. That resulted in a total of 11 parameters for estimation.

### 5.8.4 Objective Function

The objective function was defined in the form of *weighted least squares*, that is:

$$E = \left(\vec{d}^{obs} - \vec{d}^{cal}\right)^T \mathbf{W} \left(\vec{d}^{obs} - \vec{d}^{cal}\right) \tag{5.16}$$

**W** is defined in the next section.

### 5.8.5 Filtering the Data

The first task was to adjust all the measurements to a common depth, or *datum* that was set at -6000 ft.

The next step was to remove the data that was considered noisy. Two new problems were encountered that were not experienced in the synthetic cases analyzed in the previous sections: first, a large number of pressure measurements and second, wellbore effects in the pressure.

The information from well testing consisted of a large number of measurements that came from electronic pressure gauges. In traditional well test analysis it is possible to process a relative large amount of data because the mathematical model is simple and can be solve with analytical formulas. This is not the case in this work because the model is much more complex and is solved numerically. Trying to calculate a large number of pressure data points would require a prohibitively large number of time steps in the simulator. Thus it was necessary to choose the data

points that were regarded as the most representative. The second problem was the wellbore storage effect at early time after each change of rate in the DST's, this is not a problem in conventional well test analysis because the mathematical model includes the modeling of the wellbore effects and the wellbore storage coefficient is considered a parameter to be estimated. This is not the case in this work, firstly the interest was in investigating permeability far from the well and not in the wellbore effects, thus the mathematical model was constructed without the capacity of handling such wellbore effects. Therefore it was necessary to remove the data that included the wellbore storage effects. Leaving in such data would have resulted in distorted estimates of the permeability field around the wells.

For each flow period and build–up in each DST we constructed the diagnostic (or log–log) plot, such as the ones shown in the left column in Fig. 5.33. From these plots it was possible first to identify and remove the noisy data, and second to remove the data with wellbore effects. The wellbore effect was considered to end at $1\frac{1}{2}$ cycles from the end of the unit slope in the derivative. Finally, from the remaining data, the most representative data points were picked from the derivative curve if they were considered to contain important information. For example from the diagnostic plot corresponding to the build–up in well #1, we picked data points more or less equally spaced (in the logarithmic scale) lying in the $\frac{1}{2}$ slope section and in the downward section, this is because it was considered that the derivative curves provide more information about the major heterogeneity in the reservoir than the pressure curve. From the plots it was observed that the data corresponding to the flow in the DST in well #3 were the nosiest data, and this was consistent with the field report that acknowledged operational problems in controlling the flow rates.

Also it was necessary to make sure that the numerical reservoir simulator was able to reproduce pressure transients. This consideration comes from the fact that the numerical simulator computed the well pressure by using the well known Peaceman equation [49], which was developed for steady state or pseudosteady state only. Peaceman [55], and [56] recognized this fact and provided new guidelines to compute

the pressure in wells. In this work we compared the transient pressure calculated by the numerical simulator with a known exact solution (analytical model) for a simple homogeneous reservoir case. As a result of this analysis it was necessary to make small adjustments in the equivalent radius $r_0$ in Equation 3.45 at early time.

Another problem was how to combine data of different quality and quantity, that is there were a large number of observations during the DST's but a smaller number of data during the build–ups. The build–up data were considered to be more accurate than the data points from the flowing periods because of the effects of small fluctuations in the flow and inaccuracies in the reporting of the flow rates. Finally the scarcest data were the shut–in pressures, a single number for an important event as was revealed in the variance analysis in Chapter 4. The concern was that the procedure for inversion would match the largest but most inaccurate data and would ignore the shut–in data. This issue was treated in this work by applying different weights to the data, as indicated in Table 5.2

Table 5.2: Field case – weighting of the data. (From Ref. [29]).

| *Pressure from* | *Weight* | *Number of Data Points* | *Total Weight* | *% Weight* |
|---|---|---|---|---|
| DST Flow | 1.0 | 124 | 124 | 22% |
| DST Build–up | 3.0 | 79 | 237 | 42% |
| Shut–in | 30.0 | 7 | 210 | 36% |
| TOTAL | | 210 | 571 | 100% |

### 5.8.6 Results

The final results of the inverse problem are shown in Figures 5.36 and 5.37 for the *pixel* model approach and in Fig. 5.38 for the *object* model approach.

Fig. 5.39 show the match of the DST pressures for the *pixel* model. It is observed that in general the match is of good quality. As an example of problem with the data, it may be observed that it is not possible to obtain a good match in the last flow period in the DST at well #1, the data shows increasing pressure with time and this is not possible in a mathematical model that assumes *constant* flow rate and skin.

Fig. 5.40 shows the match of the build–up in the DST in well #1 when it is plotted in a diagnostic plot. It is observed that the calculated pressure follows the trend shown in the derivative curve. This plot also shows the effect of the filtering, the true data is plotted showing the complete data set and the calculated data shows only the points that were picked during the filtering process.

Fig. 5.41 shows the match of the shut–in pressure presented as a $p/z$ vs. cumulative production plot. The match shows that it was possible to capture the general trend, and that the matches to the last shut–in pressures were the most difficult.

Next follows an analysis of the estimates of the permeability distributions as they were calculated by the procedure for inversion derived in this work.

Fig. 5.36 shows the results of the application of the *pixel* approach. The distribution of permeability around the well #1 resembles a channel, this is consistent with the qualitative interpretation from the diagnostic plot of the DST in well #1. The distribution of permeability in the western part of the reservoir explains the pressure support observed at late time in the $p/z$ vs. cumulative plot. From the distribution of permeability on the northern boundary may be interpreted that fault #1 is not sealing, that is there is communication with the northern part of the reservoir. Since no geostatistical information about the spatial correlation of the permeability and porosity was used in this study, it may be that we obtained a distribution of permeability that matches the data but that is completely uncorrelated in space, which would mean that is an unrealistic distribution. To examine this issue we computed the directional variograms (N–S and E–W) of the calculated permeability, as shown in

Fig. 5.42. It is observed that there is a spatial correlation in the E–W direction, this was consistent with the geological information. In this case the information about the aquifer was a key to resolving the inverse problem.

The match to the data obtained with the channel model was reasonable but of much lower quality than the match obtained with the *pixel* approach. This was expected since the parameter space in the object modeling is much smaller than in the in the case of pixel modeling. In this case the object modeling was too simple to capture the small scale heterogeneity. A visual comparison between Figures 5.37 and 5.38 helps to understand the different scale of the reservoir information that it is recovered with each approach.

### 5.8.7 Conclusions from the Field Case Study

Some conclusions and suggestions for future development can be drawn from the study of this field case:

- Recognition of the presence of constant pressure boundaries represents very important information since these boundaries influence the resolution of parameters in large areas of the reservoir.

- More frequent shut–in pressure surveys and the complete pressure records would have provided a more realistic description of the permeability field.

- It is necessary to develop more accurate numerical procedures to model the pressure transients since that is the information that will be matched. If the numerical simulator is unable to reproduce the transients then that will result in distorted estimates of the permeability field.

- A hybrid approach, that is a combination of *pixel* and *object* modeling may be the best approach. The regions close to the wells would be modeled with the pixel approach and with a finer grid (using grid refinement) and the large areas between the wells could be modeled with multiple objects.

- Short duration pressure information (DST and shut–in) is insufficient to provide a detailed description of the reservoir. Tracer and pressure interference tests would have provided better information about the heterogeneity field between the wells.
- This approach seems to be very helpful in determining the major features in the reservoir, such as the communications between compartments in the reservoir separated by faults.

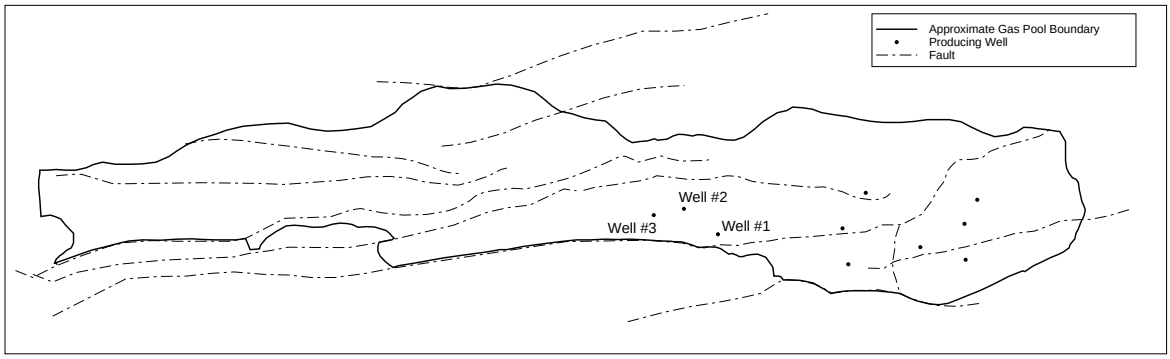



Figure 5.30: Field case – map of the field. (From Ref. [29]).

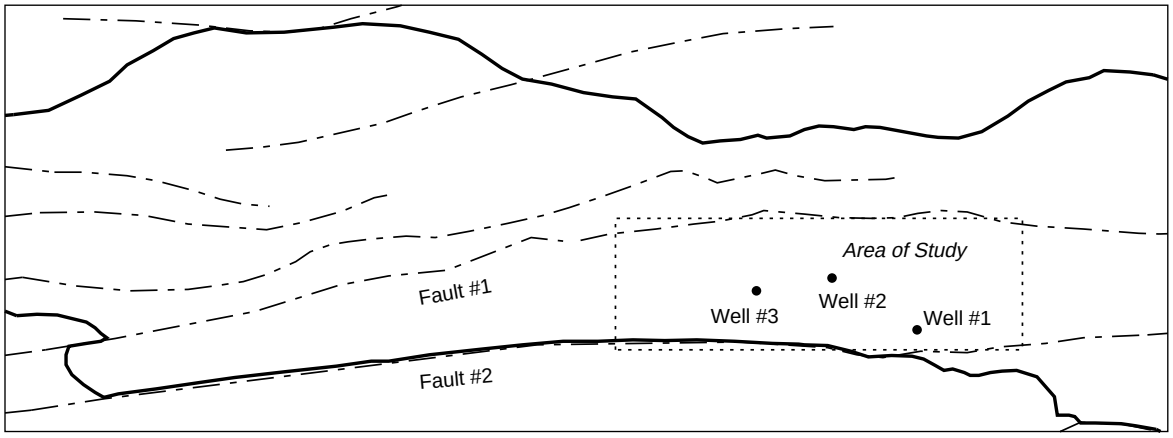



Figure 5.31: Field case – area of study. (From Ref. [29]).

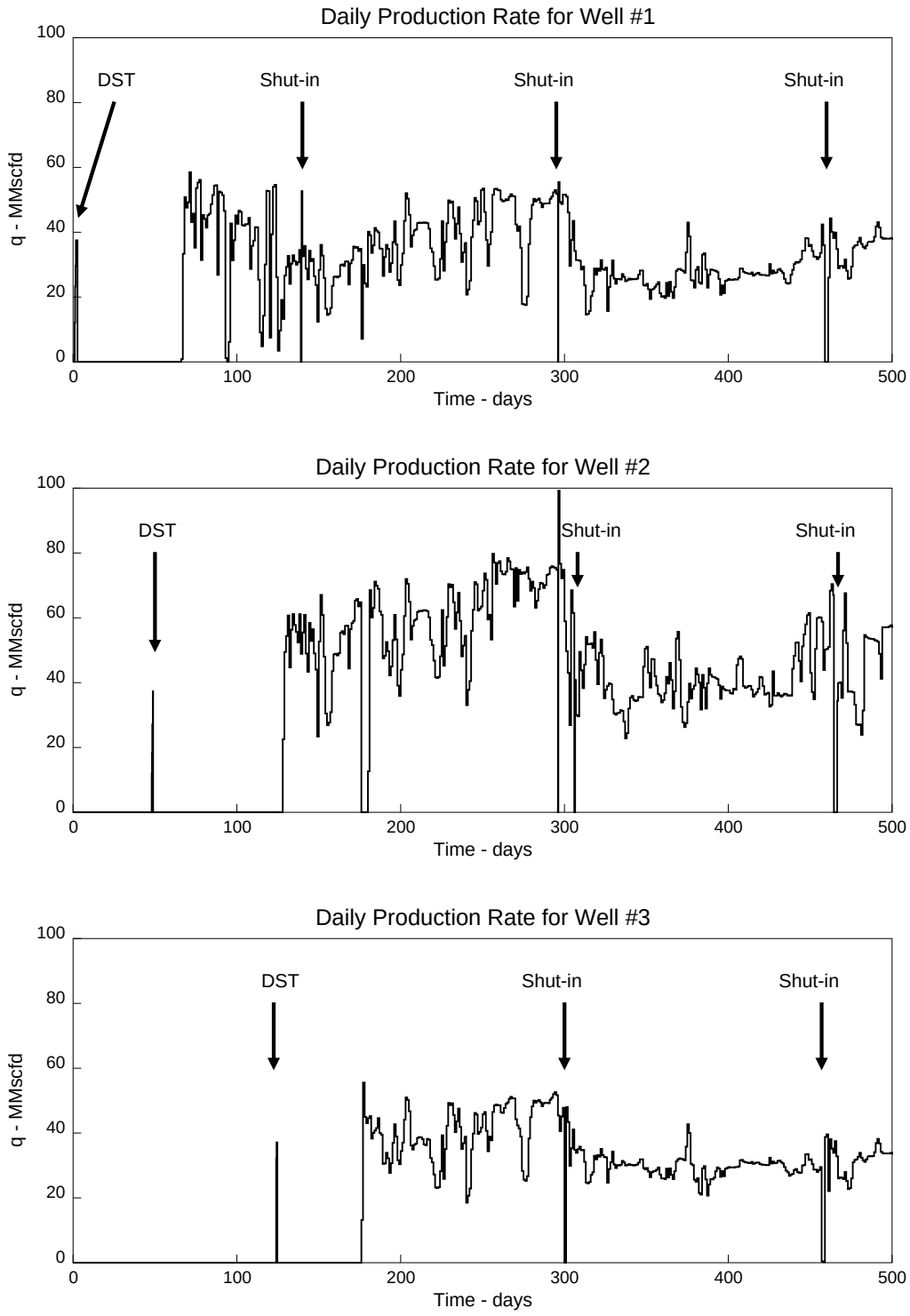



Figure 5.32: Field case – production history. (From Ref. [29]).

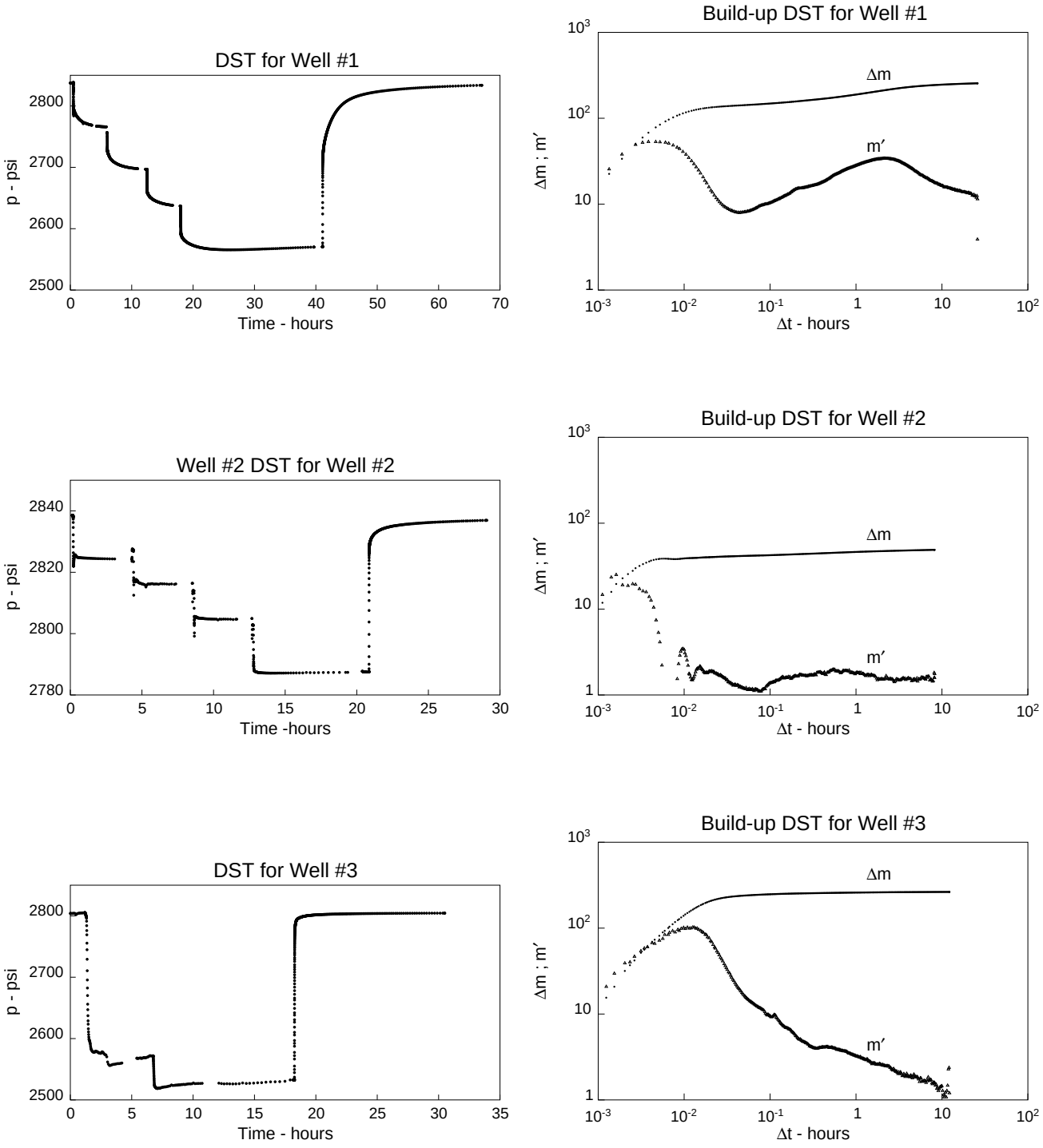



Figure 5.33: Field case – DST data and diagnostic plots for build–up. (From Ref. [29]).

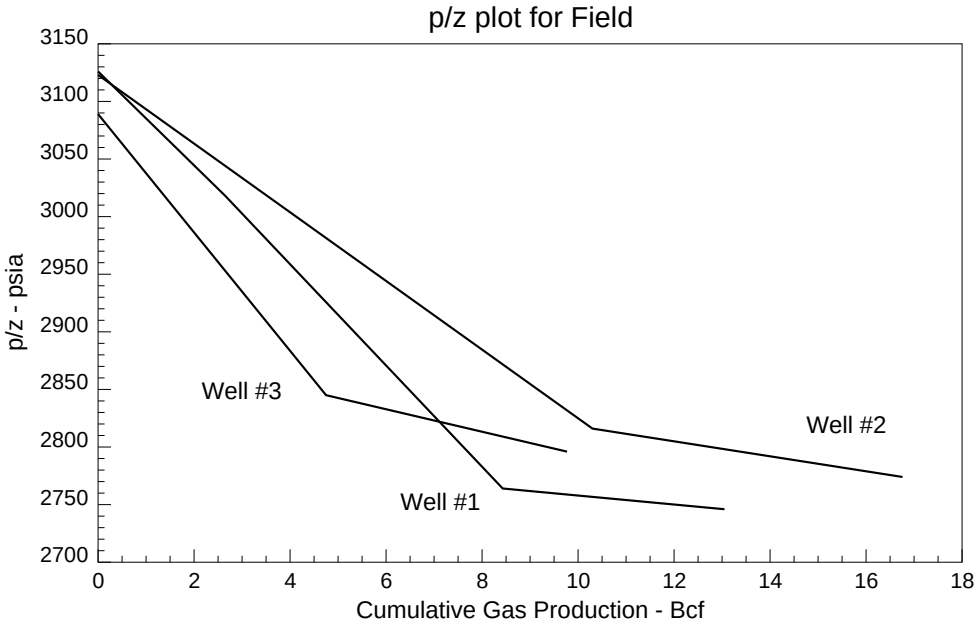



Figure 5.34: Field case – p/z vs. cumulative production plot. (From Ref. [29]).

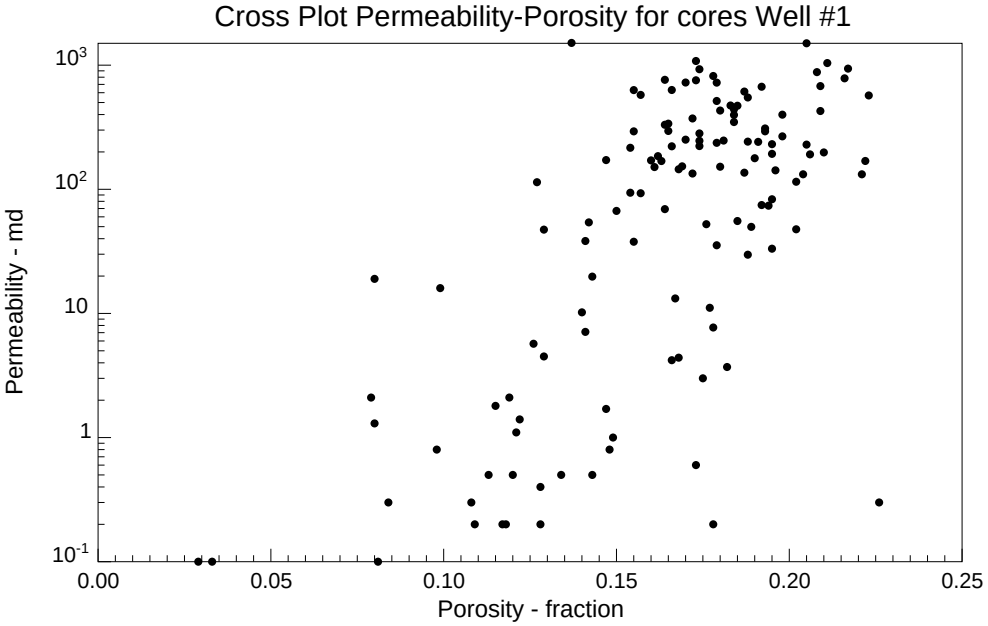



Figure 5.35: Field case – cross plot of permeability and porosity from cores taken in well #1. (From Ref. [29]).

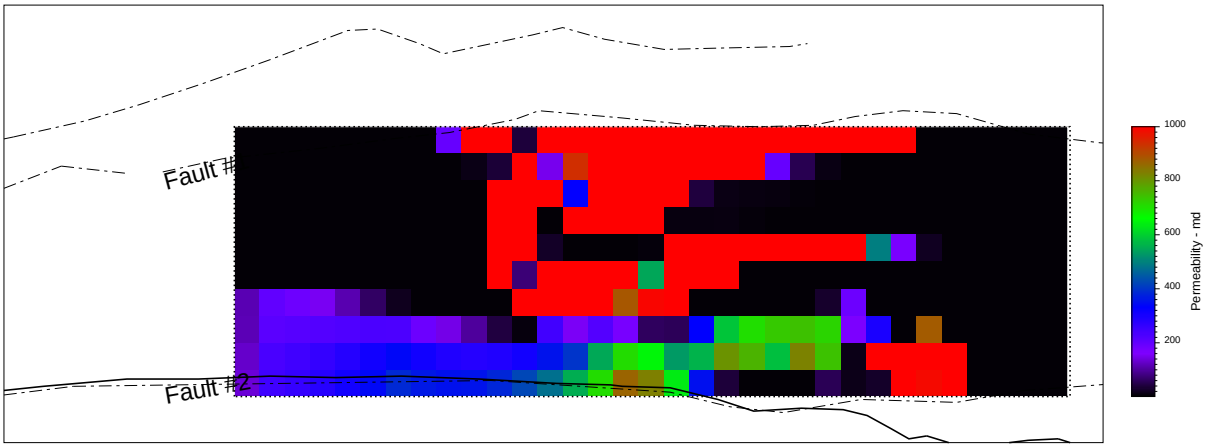



Figure 5.36: Field case – calculated permeability – pixel model. (From Ref. [29]).

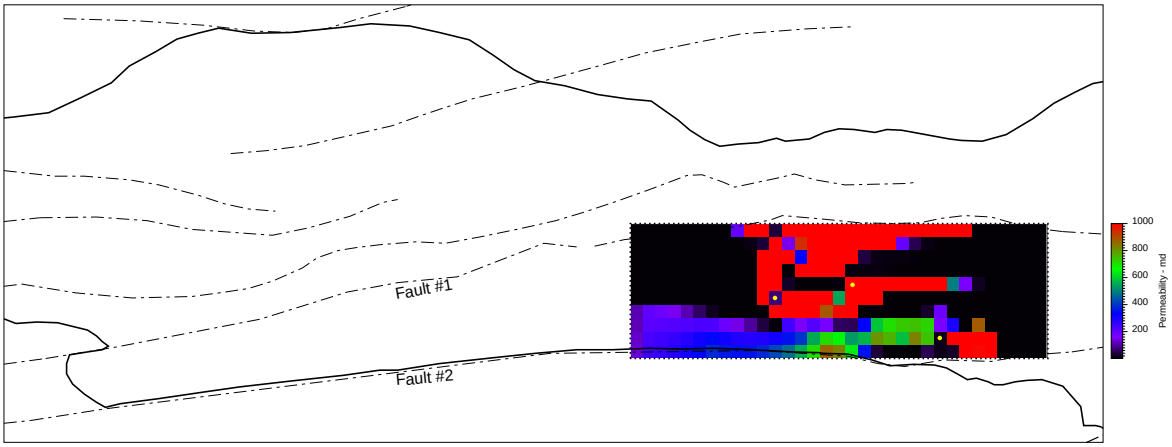



Figure 5.37: Field case – calculated permeability – pixel model. (From Ref. [29]).

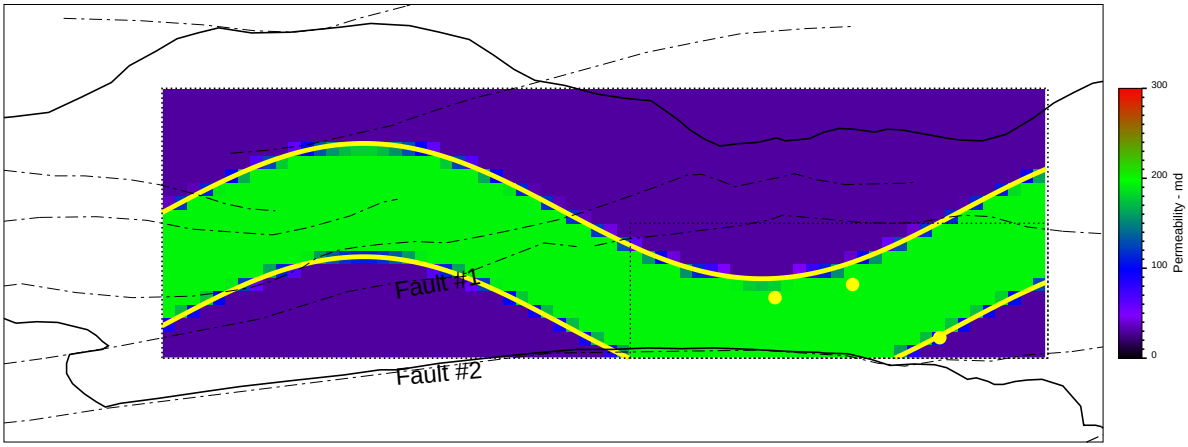



Figure 5.38: Field case – calculated permeability – object model. (From Ref. [29]).

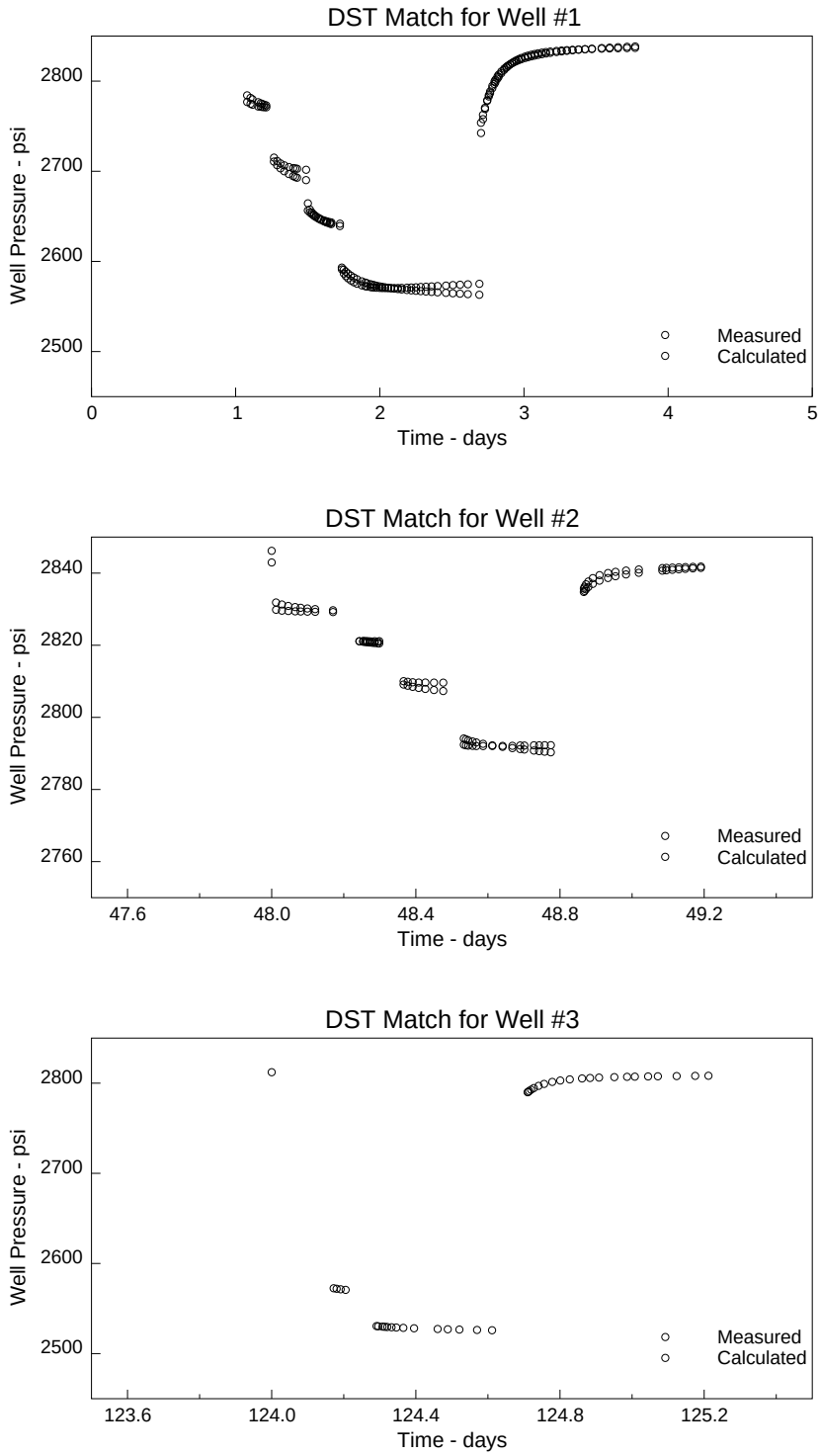



Figure 5.39: Field case – match of DST data. (From Ref. [29]).

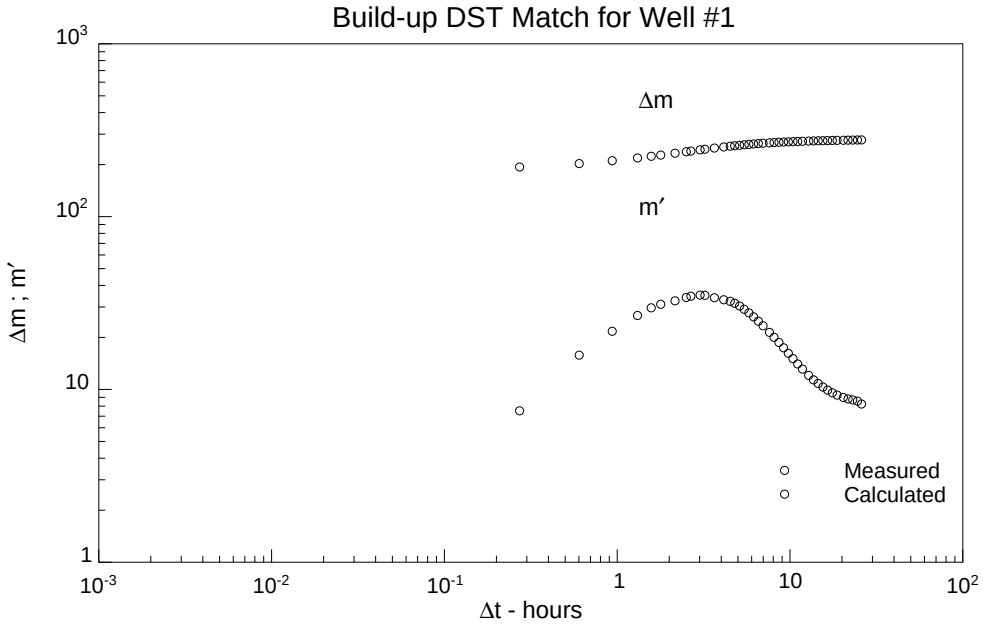



Figure 5.40: Field case – match of build–up in well #1 – diagnostic plot. (From Ref. [29]).

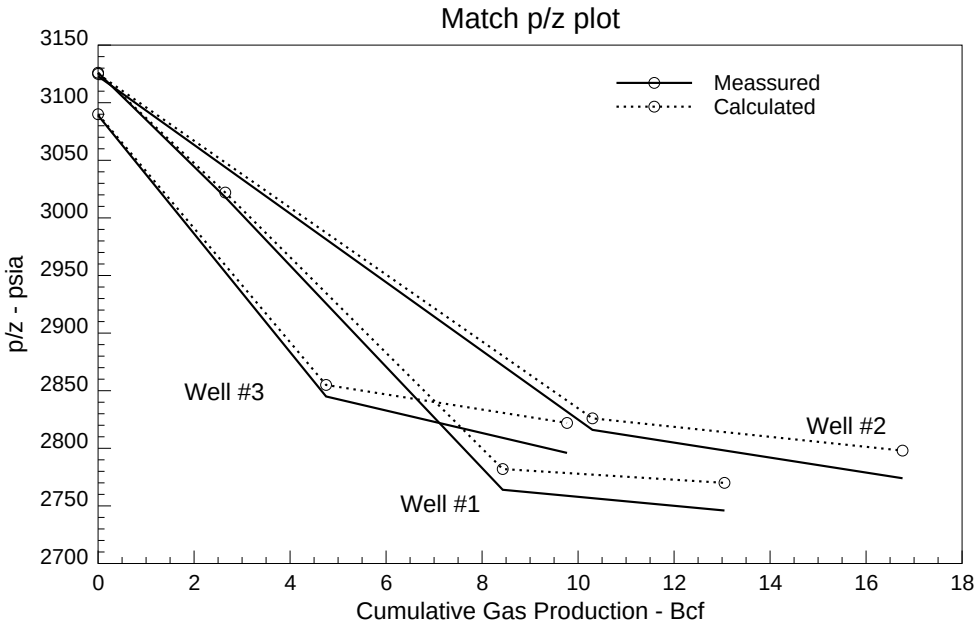



Figure 5.41: Field case – match of shut–in pressure – p/z vs. cumulative production plot. (From Ref. [29]).

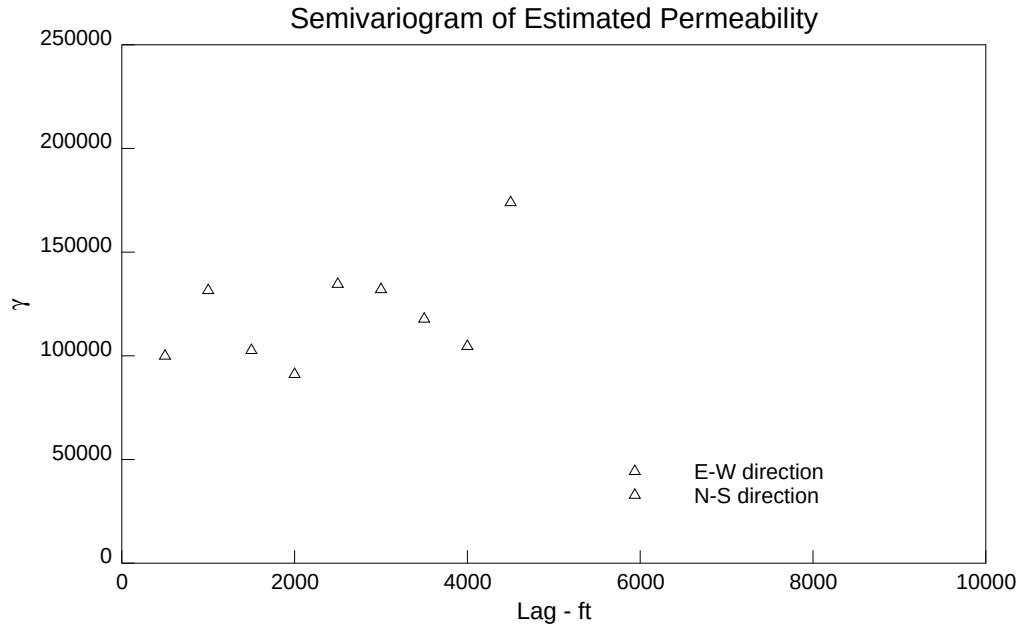



Figure 5.42: Field case – semivariogram of the estimated permeability with the *pixel* approach. (From Ref. [29]).

# Chapter 6

# Conclusions

The main purpose of this work was to develop procedures to estimate the distributions of permeability and porosity in the reservoir by using dynamic data. A second objective was to develop methods to assess the uncertainty associated to the calculated permeability and porosity and to determine the value of each data type from the point of view of the inverse problem.

The research resulted in the development of a procedure that allows us to integrate information from several sources to compute distributions of permeability and porosity. The information that can be integrated includes:

- Well test data
- Shut–in pressure
- Long term pressure (from permanent gauges)
- Production history (water cut, rates)
- Interpreted 4–D seismic maps
- Permeability–porosity correlations
- Variogram model
- Large scale geological information

This list seems to cover most of the information that can be realistically obtained from a program for reservoir monitoring with current technology, but it is not necessary to have all of them to apply the method described in this work.

In this work the *interpreted 4–D seismic* information was assumed to be in the form of maps of *change of saturation* in the reservoir. The procedure can be adapted to the case when the 4–D seismic interpretation is not related to a simple change of saturation but to a more complex relationship that could be a function of not only saturation but pressure, density, rock type, etc. The only limitation is that such a function should be known.

In this work the unknown parameters of interest were assumed to be the absolute permeability, the porosity and the skin at the wells. Thus it was implicitly assumed that all the other parameters such as relative permeability, fluid properties and thickness were known, if this is not the case the method could be adapted to include the other unknown parameters within the parameter estimation algorithm.

The components of the procedure developed during this work included:

- A numerical reservoir simulator with the capability of computing *sensitivity coefficients* as part of the solution process. The simulator was limited to two–dimensional reservoirs and water–oil systems but there is no constraint whatsoever from the theoretical point of view to expand the approach to three dimensions and to gas-oil-water systems.
- Mathematical procedures to construct *dynamic* reservoir objects.
- A parameter estimation algorithm based on the Gauss–Newton algorithm, with a Marquardt method for stabilization, penalty function for constraints and an optimized linear search.
- A method to undertake variance and resolution analysis based on the theory of *singular value decomposition.*

The examples of reservoir objects shown here were relatively simple, but the method is applicable to more complex cases.

The method developed in this work relies on the computation of the *sensitivity coefficients* of the field observations with respect to the parameters of the inverse problem. This computation might be considered an unnecessary burden since there are other methods to solve the inverse problem without the sensitivity coefficients, but we found such an effort worthwhile for the following reasons:

1. The sensitivity coefficients are necessary to perform the variance and resolution analysis, and this is a key to understanding the meaning of what it is being calculated and to plan *reservoir monitoring*. The sensitivity coefficients are also needed to find the most adequate parameterization of the reservoir for a given amount of information. The covariance matrix calculated in the variance analysis can be used for the generation of other realizations of the reservoir that also honor the data.

2. The method to compute the sensitivity coefficients, as developed in this work, is relatively simple to implement when it is possible to have access to the numerical reservoir simulator computer code. Thus the complexity of modeling the reservoir is left to the simulation part and not to the inverse problem as it would be the case for the optimal control approach.

3. The Gauss–Newton approach is very efficient as a parameter estimation algorithm, compared to direct search methods that do not require sensitivity coefficients.

4. The computation of the sensitivity can be performed with high efficiency, especially in the case of object or large pixel modeling. The efficiency can also be enhanced by parallel computing.

From the results of the numerical experiments performed during this research it is possible to conclude that the method can be utilized as a tool for reservoir

characterization. The method seems to be more useful when the dynamic data set consists of a large number of measurements that are difficult to honor with the existing geostatistical approaches; this may be the case in mature fields that have already gone through a secondary recovery process and in which data have been gathered over many years.

The more interesting conclusions obtained from this work are about the roles of well testing, 4–D seismic and integration.

Well testing has often been considered as an isolated tool to solve relatively simple reservoir models in the neighborhood of the wells and to evaluate if it is necessary to perform a well stimulation to improve the productivity. These are very important considerations from the engineering and economical point of view, but since the well tests are performed generally during the earlier part of the life of the reservoir then this information is not used later mainly because of the assumption that the area investigated is small and also because of the lack of a mathematical tool to be used in a more complex environment. The analysis performed here shows that the well test information can be used to resolve much more complex reservoir models later in the life of the reservoir, not as a standalone approach but as a piece of information in a larger data set. The addition of well west information enhances the resolution value of the other information, thus well tests designed early in the life of the reservoir must take into account that the data gathered will be useful later for reservoir characterization.

The value of 4–D seismic data is due to the fact that t provides information that is spatially distributed in the reservoir in contrast to the other data that are localized at the wells. Even so, it was found that the 4–D information does not assist much in reservoir characterization when it is considered alone. The value of this type of information appears when it is combined with other more traditional data such as production history. Also it was found that this kind of information does not need to be very accurate since data in *black and white* format can still be useful. In all the examples shown in this work it was assumed that two consecutive 3–D seismic surveys were performed after production had begun in the reservoir. If the first survey were to be performed before any production then it is very likely that the 4–D information will provide even more value to the reservoir inverse problem.

The value of data integration can be evaluated with the variance analysis. It was shown that addition of data always enhances the overall resolution of the parameters. The addition of data that seems to provides little information value when considered in isolation results in a large increase in the resolution of the other data. A dramatic example is in the case of the shut–in pressure data, which includes only a very small number of pressure measurements obtained when the wells are closed at the surface for a 24 – 48 hour period. This information when considered by itself cannot be used to solve even the simplest models, but its value is considerable when it is used in an integrated procedure such as the one developed in this work.

## 6.1 Areas that Need Further Research

One of the main assumptions in this work, and implicit in the theory of the inverse problems, is that the forward problem has already been solved, that is to say that the mathematical model can perform *accurate* predictions. This is critical because in an inverse problem the input is *actual* field observations and if the mathematical model (numerical reservoir simulator) cannot reproduce the main features of the data then this will result in distorted estimates of permeability and porosity. Thus more research has to be conducted in the areas of forward modeling, especially in the area of well modeling in multiphase regimes, and in the area of controlling numerical dispersion in finite difference models.

Another area for research is the extension of the method to three–dimensional reservoirs, including the modeling of objects and the addition of well–logs and bottom–hole flow rate measurements as part of the field observations for the inverse problem. In the three-dimensional reservoir the 4–D seismic information will have to be considered as an average in the vertical direction and thus it will be necessary to develop mathematical procedures for such averaging.

In this work the sensitivity coefficients were computed with high accuracy, probably this is not necessary, thus some further research may produce a method to obtain an approximation to the sensitivity coefficients with less computation.

The sensitivity coefficients also provide information that would be useful in the area of down–scaling and up–scaling for reservoir simulation, although this application was not explored in this work.

# Nomenclature

| | |
|---|---|
| $B$ | Formation Volume Factor |
| $\mathbf{C}\{\}$ | Covariance matrix |
| $d$ | data |
| $E$ | Objective function |
| $E\{\}$ | Expected value |
| DST | Drill Stem Test |
| $dsat$ | Change of saturation |
| $f$ | Residual in material balance equation |
| $\mathbf{G}$ | Sensitivity matrix |
| $\mathbf{H}$ | Hessian matrix |
| $h$ | Reservoir pay zone thickness |
| $\mathbf{J}$ | Jacobian matrix |
| $k$ | Permeability |
| $nblocks$ | Number of active blocks in the simulation grid |
| $ncons$ | Number of constraints |
| $nobs$ | Number of observations |
| $npar$ | Number of parameters |
| $nsamp$ | Number of samples |
| $q$ | Production (injection) rate at wells |
| $\mathbf{R}$ | Resolution matrix |
| $S$ | Saturation - $fraction$ |

| | |
|---|---|
| $\mathbf{S}_{inf}$ | Information density matrix |
| $\mathbf{S}$ | Singular Values matrix |
| $s_k$ | Skin factor |
| $t$ | Time |
| $\mathbf{U}$ | Singular Value Decomposition factor matrix |
| $\mathbf{V}$ | Singular Value Decomposition factor matrix |
| $\mathbf{W}$ | Weight matrix |
| $W_f$ | well factor |
| $w$ | weight to each data |

## Symbols

| | |
|---|---|
| $\alpha$ | Parameter |
| $\phi$ | Porosity - $fraction$ |
| $\mu$ | Viscosity |
| $\rho$ | Step size in linear search |
| $\sigma$ | Standard deviation |
| $\Lambda$ | Matrix of eigenvalues |
| $\|\|\ \|\|$ | 2–norm of a vector |

## Superscripts

| | |
|---|---|
| $cal$ | Calculated with Mathematical Model |
| $obs$ | Observed, true data |
| $\vec{}$ | vector |

## Subscripts

| | |
|---|---|
| $cal$ | Calculated with Mathematical Model |
| $obs$ | Observed, true data |
| $GN$ | Gauss–Newton |
| $o$ | oil |
| $p$ | number of nonzero singular values |
| $SVD$ | Singular Value Decomposition |
| $w$ | water |

# Bibliography

[1] Wahl, W. L., Mullins, L., Barham, R., and Bartlet, W.: "Matching the Performance of Saudi Arabian Oil Fields With an Electrical Model," paper SPE 414 presented at the 1962 SPE 37th Annual Fall Meeting, Los Angeles, CA, October, 7-10.

[2] Jacquard, P. and Jain, C.: "Permeability Distribution From Field Pressure Data," *Soc. Pet. Eng. Journal* (December 1965) 281–294.

[3] Carter, R. D., Pierce, A. C., Kemp, L., and Willians, D. L.: "Performance Matching With Constraints," *Soc. Pet. Eng. Journal* (April 1974) 187–196.

[4] Chen, W. H., Gavalas, G. R., Seinfeld, J. H., and Wasserman, M. L.: "A New Algorithm for Automatic History Matching," paper SPE 4545 presented at the 1973 SPE–AIME 48th Annual Fall Meeting, Las Vegas, NV, September, 30–October, 3.

[5] Chavent, G., Dupuy, M., and Lemonnier, P.: "History Matching by Use of Optimal Theory," paper SPE 4627 presented at the 1973 SPE–AIME 48th Annual Fall Meeting, Las Vegas, NV, September, 30–October, 3.

[6] Watson, A. T., Seinfeld, J. H., Gavalas, G. R., and Woo, P. T.: "History Matching in Two–Phase Petroleum Reservoirs," paper SPE 8250 presented at the 1979 SPE 54th Annual Technical Conference and Exhibition, Las Vegas, NV, September, 23-26.

[7] Yang, P. H. and Watson, A. T.: "Automatic History Matching With Variable–Metric Methods," paper SPE 16977 presented at the 1987 SPE 62nd Annual Technical Conference and Exhibition, Dallas, TX, September, 27-30.

[8] Gill, P. E., Murray, W., and Wright, M. H.: *Practical Optimization*, Academic Press, San Diego, CA (1981).

[9] Fasanino, G., Molinard, J., and de Marsily, G.: "Inverse Modeling in Gas Reservoirs," paper SPE 15592 presented at the 1986 SPE 61st Annual Technical Conference and Exhibition, New Orleans, LA, October, 5-8.

[10] Tang, Y. N. and Chen, Y. M.: "Application of GPST Algorithm to History Matching of Single–Phase Simulator Models," *Unsolicited paper SPE 13410* (1985).

[11] Tang, Y. N. and Chen, Y. M.: "Generalized Pulse–Spectrum Technique for Two–Dimensional and Two–Phase History Matching," *Applied Numerical Mathematics* (1989) **5**, 529–539.

[12] Oliver, D. S.: "Incorporation of Transient Pressure Data Into Reservoir Characterization," *In Situ* (1994) **18**, No. 3, 243–275.

[13] Tarantola, A. and Valette, B.: "Generalized Nonlinear Inverse Problems Solved Using The Least Squares Criterion," *Reviews of Geophysics and Space Physics* (May 1982) **20**, No. 2, 219–232.

[14] Tarantola, A.: *Inverse Problem Theory – Methods for Data Fitting and Model Parameter Estimation*, Elsevier Science Publishers, Amsterdam (1987).

[15] Menke, W.: *Geophysical Data Analysis: Discrete Inverse Theory*, Academic Press, Inc., San Diego, CA (1989).

[16] Chu, L., Reynolds, A. C., and Oliver, D. S.: "Computation of Sensitivity Coefficients With Application to the Integration of Static and Well–Test Pressure Data," Paper presented at the Eclipse International Forum, Milan, Italy (September, 6-9 1994).

[17] 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) **19**, No. 2, 179–223.

[18] Oliver, D. S.: "Multiple Realizations of the Permeability Field From Well Test Data," paper SPE 27970 presented at the 1994 University of Tulsa Centennial Petroleum Engineering Symposium, Tulsa, OK, August, 29-31.

[19] 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 30558 presented at the 1995 SPE Annual Technical Conference and Exhibition, Dallas, TX, October, 22-25.

[20] Chu, L., Reynolds, A. C., and Oliver, D. S.: "Reservoir Description From Static and Well–Test Data Using Efficient Gradient Methods," paper SPE 29999 presented at the 1995 SPE International Meeting on Petroleum Engineering, Beijing, P.R. China, November, 14-17.

[21] Oliver, D. S.: "A Comparison of the Value of Interference and Well–Test Data for Mapping Permeability and Porosity," *In Situ* (1996) **20**, No. 1, 41–59.

[22] Oliver, D. S., He, N., and Reynolds, A. C.: "Conditioning Permeability Fields to Pressure Data," Paper presented at the 5th European Conference on the Mathematics of Oil Recovery, Mining University, Leoben, Austria (September, 3-6 1996).

[23] He, N., Reynolds, A. C., and Oliver, D. S.: "Three–Dimensional Reservoir Description from Multiwell Pressure Data," paper SPE 36509 presented at the 1996 SPE Annual Technical Conference and Exhibition, Denver, CO, October, 6-9.

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

[25] Bissell, R., Sharma, Y., and Killough, J. E.: "History Matching Using the Method of Gradients: Two Case Studies," paper SPE 28590 presented at the 1994 SPE Annual Technical Conference and Exhibition, New Orleans, LA, September, 25-28.

[26] Tan, T. B. and Kalogerakis, N.: "A Fully Implicit, Three–Dimensional, Three–Phase Simulator with Automatic History–Matching Capability," paper SPE 21205 presented at the 1991 SPE 11th Symposium on Reservoir Simulation, Anaheim, CA, February, 17-20.

[27] Tan, T. B.: "A Computational Efficient Gauss-Newton Method for Automatic History Matching," paper SPE 29100 presented at the 1995 SPE Symposium on Reservoir Simulation, San Antonio, TX, February, 12-15.

[28] Bissell, R.: "History Matching A Reservoir Model by the Positioning of Geological Objects," Paper presented at the 5th European Conference on the Mathematics of Oil Recovery, Mining University, Leoben, Austria (September, 3-6 1996).

[29] Landa, J. L., Kamal, M. M., Jenkins, C. D., and Horne, R. N.: "Reservoir Characterization Constrained to Well Test Data: A Field Example," paper SPE 36511 presented at the 1996 SPE Annual Technical Conference and Exhibition, Denver, CO, October, 6-9.

[30] Ounes, A., Bréfort, B., Meunier, G., and Dupéré, S.: "A New Algorithm for Automatic History Matching: Application of Simulated Annealing Method (SAM) to Reservoir Inverse Modeling," *Unsolicitated paper SPE 26297* (1993).

[31] Sultan, A. J., Ounes, A., and Weiss, W. W.: "Automatic History Matching for an Integrated Reservoir Description and Improving Oil Recovery," paper SPE 27712 presented at the 1994 SPE Permian Basin Oil and Gas Recovery Conference, Midland, TX, March, 16-18.

[32] Ounes, A., Weiss, W., and Sultan, A. J.: "Parallel Reservoir Automatic History

Matching Using a Network of Workstations and PVM," paper SPE 29107 presented at the 1995 SPE 13th Symposium on Reservoir Simulation, San Antonio, TX, February, 12-15.

[33] Sen, M. K., Datta-Gupta, A., Stoffa, P. L., Lake, L. W., and Pope, G. A.: "Stochastic Reservoir Modeling Using Simulated Annealing and Genetic Algorithms," paper SPE 24754 presented at the 1992 SPE Annual Technical Conference and Exhibition, Washington, DC, October, 4-7.

[34] Schmidt, H., Stright, D. H., and Forcade, K. C.: "Multiwell Data Acquisition and Analysis for Permanent Bottomhole Pressure Gauge Installations," paper SPE 16511 presented at the 1987 SPE Petroleum Industry Applications of Microcomputers, Lake Conroe, TX, June, 23-26.

[35] Gallivan, J. D., Kilvington, L. J., and Shere, A. J.: "Experience With Permanent Bottomhole Pressure/Temperature Gauges in a North Sea Oil Field," *SPEPE* (November 1988) 637.

[36] Unneland, T. and Haugland, T.: "Permanent Downhole Gauges Used in Reservoir Management of Complex North Sea Oil Fields," paper SPE 26781 presented at the 1993 SPE Offshore Europe Conference, Aberdeen, UK, September, 7-10.

[37] Seymour, R. H. and Barr, F. J.: "Seabed–Seismic 4D Data–Collection Method for Reservoir Monitoring," *JPT* (January 1997) **49**, No. 1, 40–41.

[38] Tyler, K., Henriquez, A., Georgsen, F., Holden, I., and Tjelmeland, H.: "A Program for 3–D Modeling of Heterogeneities in a Fluvial Reservoir," presented at the 3rd European Conference on the Mathematics of Oil Recovery (1992).

[39] Wietzerbin, L. J. and Mallet, J.-L.: "Parameterization of Complex 3D Heterogeinities: A New CAD Approach," paper SPE 26423 presented at the 1993 SPE 68th Annual Technical Conference and Exhibition, Houston, TX, October, 3-6.

[40] Bratvold, R. B., Holden, L., Svanes, T., and Tyler, K.: "STORM: Integrated 3D Stochastic Reservoir Modeling Tool for Geologists and Reservoir Engineers,"

paper SPE 27563 presented at the 1994 SPE European Petroleum Computer Conference, Aberdeen, UK, March, 15-17.

[41] Petit, F. M., Biver, P. Y., Calatayud, P. M., Lesueur, J.-L., and Alabert, F. G.: "Early Quantification of Hydrocarbon in Place Through Geostatistical Object Modelling and Connectivity Computations," paper SPE 28416 presented at the 1994 SPE 69th Annual Technical Conference and Exhibition, New Orleans, LA, September, 25-28.

[42] Deutsch, C. V. and Wang, L.: "Hierachical Object–Based Geostatistical Modeling of Fluvial Reservoirs," paper SPE 36514 presented at the 1996 SPE Annual Technical Conference and Exhibition, Denver, CO, October, 6-9.

[43] Jackson, D.: "Interpretation of Inaccurate, Insufficient and Inconsistent Data," *Geophysical Journal of the Royal Astronomical Society* (1972) **28**, 97–109.

[44] Hillier, F. S. and Lieberman, G. J.: *Introduction to Operations Research – 5th Edition*, McGraw–Hill Publishing Company, New York, NY (1990).

[45] Murray, W.: "Class Notes OR343–Nonlinear Programing," Stanford University (Fall 1996).

[46] Bard, Y.: *Nonlinear Parameter Estimation*, Academic Press, New York, NY (1970).

[47] Ounes, A., Bhagavan, S., Bunge, P. H., and Travis, B. J.: "Application of Simulated Annealing and Other Global Optimization Methods to Reservoir Description: Myths and Realities," paper SPE 28415 presented at the 1994 SPE 69th Annual Technical Conference and Exhibition, New Orleans, September, 25-28.

[48] Aziz, K. and Settari, A.: *Petroleum Reservoir Simulation*, Elsevier Applied Science Publishers, New York (1979).

[49] Peaceman, D. W.: "Interpretation of Well–Block Pressures in Numerical Reservoir Simulation," paper SPE 6893 presented at the 1977 1977 SPE Annual Technical Conference and Exhibition, Denver, CO, October, 9-12.

[50] Eisenstat, S. C., Schultz, M. H., and Sherman, A. H.: "Yale Sparse Matrix Package, Technical Reports 112 and 114," Yale University Departament of Computer Science (1977).

[51] Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P.: *Numerical Recipes in FORTRAN –The Art of Scientific Computing– Second Edition*, Cambridge University Press, New York, NY (1992).

[52] Deutsch, C. and Journel, A.: *GSLIB Geostatistical Software Library and User's Guide*, Oxford University Press, New York (1992).

[53] Wen, X.-H., Gomez-Hernandez, J., Capilla, J. E., and Sahuquillo, A.: "Significance of Conditioning to Piezometric Head Data for Predictions of Mass Transport in Groundwater Modeling," *Mathematical Geology* (1996) **28**, No. 7, 951–968.

[54] Sheriff, R. E. and Geldart, L. P.: *Exploration Seismology*, Cambridge University Press, New York, NY (1995).

[55] Peaceman, D. W.: "Interpretation of Well–Block Pressures in Numerical Reservoir Simulation with Nonsquare Grid Blocks and Anisotropic Permeability," paper SPE 10528 presented at the 1982 SPE Symposium on Reservoir Simulation, New Orleans, LA, January 31 – February 3.

[56] Peaceman, D. W.: "A New Method for Representing Multiple Wells with Arbitrary Rates in Numerical Reservoir Simulation," *SPE Reservoir Engineering* (November 1995) 253–257.

# Appendix A

# Lists of Programs

## A.1 General Instructions

The set of programs that were developed during the course of this work are located in a directory, with a subdirectory `example1` which contains a set of example files.

The programs are written in Fortran-77 language and were run originally on DEC-Alpha workstations. The compiler switches used for compilation are set such that local variables within subroutines are not reinitialized between calls.

The procedures to compile and run the programs are as follows:

**To compile:** key in the `make` command to execute the compilation instructions in the `makefile` file; the created executable file is named `proj80.exe`.

**To run:** key in the `proj80.exe` and the program will present a deck from which it is possible to launch most of options that are possible to run.

## A.2 Files in `makefile`

The following is the list of the files that must be present at the time of compiling with the `make` command. A brief description of the task performed by the code is as follows.

| | |
|---|---|
| `makefile` | compiling instructions to create executable file `proj80.exe`. |
| `proj80.f` | driver, parameter estimation algorithm, read and write input and output files, computation of resolution and variance of estimated parameters. |
| `sim80.f` | finite difference reservoir simulator, computation of sensitivity coefficients. |
| `yalex.f` | driver for YALE direct sparse matrix solver, written by R.C. Wattenbarger at Stanford University in 1992. |
| `ndrv.f` | subroutine of YALE direct sparse matrix solver [50]. |
| `ordv.f` | subroutine of YALE direct sparse matrix solver [50]. |
| `modcholesky.f` | performs modified Cholesky factorization. |
| `solmodcholesky.f` | solves linear system of equations when matrix has bee factorized using the modified Cholesky method. |
| `cholesky.f` | standard Cholesky factorization. From Ref. [51]. |
| `objx.f` | handling of reservoir objects. |
| `svdcmpx.f` | performs singular value decomposition of a matrix. From Ref. [51]. |
| `dkdparaelipse.f` | computes the derivatives of the reservoir permeability with respect to the parameters of the inverse problem. |
| `proj.inc` | defines dimensions for arrays and vectors. |
| `sim1.inc2` | defines conversion factors. |

## A.3 Ancillary Programs

The following is the list of the ancillary programs that are needed to perform some tasks before or after the job. Some of them need to be compiled (Fortran–77).

| | |
|---|---|
| `gps` | generate color maps and plots in Postscript format, this program was written by R.C. Wattenbarger at Stanford University in 1992. |
| `kb2d` | ordinary kriging program from *GSLIB* [52]. |
| `kb2dread.f` | reads output from `kb2d` and creates files `obj.data` and `obj2.data`. |

## A.4 Data Files

Once the job has been launched with the `proj80.exe` command the program presents a menu to set the major flags for the program. For each option chosen from the menu, a list of the files that are needed is shown on the screen.

The following list presents the names of the data and generated files with a description of their contents.

| | |
|---|---|
| `data1.data` | data file for simulator, grid size, well location, rates. |
| `presatmaps.data` | specifies the time schedule to generate maps of pressure and saturation in the reservoir. |
| `wantresol.data` | flag to activate calculation of resolution, variance or change data weights. |
| `obj.data` | specifies the relationship between each object and the simulation cells. |
| `obj2.data` | specifies the relationship between each simulation cell and the objects. |
| `covardata.data` | specifies the covariance of each type of data. |
| `timex.data` | specifies the time scheme for the simulator. |
| `timex.datainput` | specifies the time scheme when the program is used to generate synthetic data. |
| `perpor.data` | permeability and porosity for simulation. |
| `skin.data` | skin factor at each well. |

| | |
|---|---|
| `tphis.data` | pressure data to match. |
| `twchis.data` | water cut data to match. |
| `dsatmap1.data` | change of saturation #1 data to match. |
| `dsatmap2.data` | change of saturation #2 data to match. |
| `par.data` | current value of the parameters for the inverse problem. |
| `parmax.data` | upper bounds for the parameters and permeability in the reservoir. |
| `parmin.data` | lower bounds for the parameters and permeability in the reservoir. |
| `penal.data` | penalty coefficients for constraints. |
| `deltaxmax.data` | specified maximum absolute change in the parameters at each iteration (only for dynamic object case). |
| `regrespar.data` | specifies which parameters are active for the inverse problem. |
| `method.data` | set flags for regression. |
| `weightpr.data` | assigns individual weight to each pressure measurement. |
| `weightwc.data` | assigns individual weight to each water cut measurement. |
| `dsatmaps.data` | specifies historic times for each change of saturation distribution and assigns a weight to each set. |
| `par.log` | keeps a record of the parameters for each iteration. |
| `iteration` | keeps a record of the value of the objective function for each iteration. |
| `pre.out` | well pressure output from simulator. |
| `watcut.out` | well water cut from simulator. |
| `lastpre.out` | well pressure output in same format than `tphis.data`. |
| `lastwac.out` | well water cut output in same format than tt twchis.data. |
| `pilot.data` | location of pilot points (for kriging option). |
| `krigdata.data` | input file for kriging program `kb2d`. |
| `kb2d.par` | parameter file for `kb2d`. |
| `*.gps` | script files for plotting utility `gps`. |

## A.5 Sample Session

The simplest way to create all the input files with the right format is to run a synthetic case. Under this option the programs creates all the files needed to run an actual case. Once the synthetic case has been run then the files can be edited in order to represent the actual data. Following is a sample session that illustrates the creation of the input files.

1. Before launching the job, create the following files, or copy them from the directory `example1`.

   (a) `data1.data`, data for simulation.

   (b) `perpor.data`, permeability and porosity at each simulation cell.

   (c) `skin.data`, skin factor at each well.

   (d) `wantresol.data`, make sure there is a "0" as input.

   (e) `obj.data`, link between each parameter and the simulation cells.

   (f) `timex.datainput`, cumulative time (days) for time steps in simulation.

   (g) `dsatmaps.data`, change of saturation information.

   (h) `presatmaps.data`, time for generation of pressure and saturation maps.

2. Key in `proj80.exe`, the following menu will appear:

```
**************************************************
O P T I O N S

[ 1 ] Run Program.

[ 2 ] Run Simulator ONLY and Stop.

[ 3 ] Run Simulator and Calculate Sensitivity
```

```
[ 4 ] Generate Input Files for History Match
                                        and Stop

**************************************************

ENTER OPTION #
```

3. Enter "4", the following screen will be show:

```
The following files need to exist:
----------------------------------

  data1.data      (input data for simulation)
  obj.data        (link parameter-simulation cell)
  timex.datainput (cumulative time for time steps)
  dsatmaps.data   (for delta-saturation maps)
  presatmaps.data (for pressure and saturation maps)
  perpor.data     (perm-porosity file)
  skin.data       (skin in all wells)

The program will create the following files:
--------------------------------------------
(All  previous input files will be deleted)
******************************************

  timex.data      (cumulative time for time steps)
  tphis.data      (time-pressure history)
  twchis.data     (time-water cut history)
  weightpr.data    (weight to pressure data)
  weightwc.data    (weight to water-cut data)
  parmax.data     (upper limit for each parameter)
```

```
  parmin.data     (lower limit for each parameter)
  penal.data        (penalty weight)
  regrespar.data  (active parameters for estimation)
  dsatmap1.data   (delta saturation map)
  dsatmap2.data   (delta saturation map)
  permap.data0    (true perm map to plot)
  pormap.data0    (true perm map to plot)

< Enter > to continue, < ^C > to abort

FORTRAN PAUSE
PAUSE prompt>
```

4. Key in <enter> to continue, the following screen will be shown:

```
Make sure that the times reported in
                              dsatmaps.data
                              presatmaps.data
                              data1.data
are consistent with the times
listed in  timex.datainput
-------------------------------------------------

Enter the number of objects
          (parameters less skin factor)
```

5. Enter "100", the following screen will be shown:

```
To prepare the input files for parameter estimation
```

```
need the following information

**************************************************
O P T I O N S

[ 1 ] No Kriging - Simple Object.

[ 2 ] Kriging in Permeability.

[ 3 ] Kriging in Delta_Permeability

[ 4 ] Dynamic Object
**************************************************

ENTER OPTION #
```

6. Key in "1", the following menu will show up:

```
ENTER maximum permeability allowed in Reservoir
```

7. Key in "12000.0", the following menu will show up:

```
ENTER minimum permeability allowed in Reservoir
```

8. Key in "50.0", the following menu will show up:

```
ENTER maximum value allowed to parameters
```

9. Key in "12000.0", the following menu will show up:

```
ENTER minimum value allowed to parameters
```

10. Key in "50.0", the following menu will show up:

```
ENTER constant value for par.data
```

11. Key in "500.0", the following screen will show up:

```
-------------------------------------------
Now will run simulation to create
pressure, watercut and delta-saturation data

You may need to edit some of the files that
are being created for you.
-------------------------------------------

==============================
krun  =           4
kcons =           1
ikrig =           0
kelip =          -1
==============================

< Enter  > to continue, < ^C > to abort
FORTRAN PAUSE
PAUSE prompt>
```

12. Press <enter> to continue, the screen will show the time steps in the simulation.

```
nsvdmax is         4000
completed reading data1.data    1.00000000000000        10.0000000000000
  2.00000000000000
nobj is          100
```

```
nobj is          100
npar is          104
ntph =         149
completed reading readhistory
completed readpar
created new perpor.data0
Active Parameters

J-Active Parameters

npar is          104
time=   2.080000000000000E-002  iterations=           4
time=   4.170000000000000E-002  iterations=           4
time=   6.250000000000000E-002  iterations=           4


time=    206.541700000000       iterations=           3
time=    207.041700000000       iterations=           3
time=    211.000000000000       iterations=           4
time=    221.000000000000       iterations=           5
Done with sim80.f
created new tphis.data
created new twchis.data
```

13. By this time all the input files have been created in the proper format. To run a real case it is necessary to edit the pressure, water cut and change of saturation files (`tphis.data`, `twchis.data` and `dsatmap1.data - dsatmap2.data` respectively). For further details read the file `readme` in the same directory than the set of programs.