Uncertainty Reduction in Reservoir Characterisation through Inverse Modelling of Dynamic Data: an Evolutionary Computation Approach

Mohammad Sayyafzadeh, B.Sc., M.Sc.

A thesis submitted for the degree of Doctor of Philosophy (Ph.D.)

Australian School of Petroleum

Faculty of Engineering, Computer and Mathematical Sciences

The University of Adelaide

July 2013

Contents

List of figures

List of tables

Summary

Precise reservoir characterisation is the basis for reliable flow performance predictions and unequivocal decision making concerning field development. History matching is an indispensable phase of reservoir characterisation in which the flow performance history is integrated into the initially constructed reservoir model to reduce uncertainties. It is a computationally intensive nonlinear inverse problem and typically suffers from ill-posedness. Developing an efficient automatic history matching framework is the core goal of almost all studies on this subject.

To overcome some of the existing challenges in history matching, this thesis introduces new techniques which are mostly based on evolutionary computation concepts. In order to examine the techniques, in the beginning, the foundations of an automatic history matching framework are developed in which a reservoir simulator (ECLIPSE) is coupled with a programming language (MATLAB). Then, the introduced methods along with a number of conventional methods are installed on the framework, and they are compared with each other using different case studies.

Thus far, numerous optimisation algorithms have been studied for history matching problems to conduct the calibration step accurately and efficiently. In this thesis, the application of a recent-developed algorithm, artificial bee colony (ABC), is assessed, for the first time. It is compared with three conventional optimisers, Levenberg-Marquette, Genetic Algorithm, and Simulated Annealing, using a synthetic reservoir model. The comparison indicates that ABC can deliver better results and is not concerned with the landscape shape of problem. The most likely reason of its success is having a suitable balance between exploration and exploitation search capability. Of course, similar to all stochastic optimisers, its main drawbacks are computational expenses and being inefficient in high-dimensional problems.

Fitness approximation (proxy-modelling) approaches are common methods for reducing computational costs. All of the applied fitness approximation methods in history-matching problems use a similar approach called uncontrolled fitness approximation. It has been corroborated that the uncontrolled fitness approximation approach may mislead the optimisation direction. To prevent this issue, a new fitness approximation is developed in that a model management (evolution-control) technique is included. The results of the controlled (proposed) approach are compared with the results of conventional one using a case study (PUNQ-S3 model). It is shown that the computation can be reduced up to 75% by the proposed method. The proxy-modelling methods should be applied when the problem is not high-dimensional.

None of the current formats of the applied stochastic optimisers is capable of dealing with high-dimensional problems efficiently, and they should be applied in conjunction with a reparameterisation technique which causes modelling errors. On the other hand, gradient-based optimisers may be trapped into a local minimum, due to the nonlinearity of the problem. In this thesis, an inventive stochastic algorithm is developed for high-dimensional problems based on wavelet image-fusion and evolutionary algorithm concepts. The developed algorithm is compared with six algorithms (genetic algorithm with a pilot point reparameterisation, BFGS with a zonation reparameterisation, BFGS with a spectral decomposition reparameterisation, artificial bee colony, genetic algorithm and BFGS in full-parameterisation) using two different case studies. It is interesting that the best results are obtained by the introduced method.

Besides, it is well-known that achieving high-quality history matched models using any of the methods depends on the reliability of objective function formulation. The most widespread approach of formulation is Bayesian framework. Because of complexities in quantifying measurement, modelling and prior model reliability, the weighting factors in the objective function may have uncertainties. The influence of these uncertainties on the

outcome of history matching is studied in this thesis, and an approach is developed based on Pareto optimisation (multi-objective genetic algorithm) to deal with this issue. The approach is compared with a conventional (random selection) one. The results confirm that a high amount of computation can be saved by the Pareto approach.

In last part of this thesis, a new analytical simulator is developed using the transfer function approach. The developed method does not need the expensive history matching, and it can be used for occasions that a quick forecasting is sought and/or history matching of grid-based reservoir simulation is impractical. In the developed method, it is assumed a reservoir consists of a combination of TFs, and then the order and arrangement of TFs are chosen based on the physical conditions of the reservoir ascertained by examining several cases. The results reveal a good agreement with those obtained from the grid-based simulators.

An additional piece of work is done in this thesis in which the optimal infill drilling plane is estimated for a coal seam gas reservoir (semi-synthetic model constructed based on the Tiffany unit in the San Juan basin) by the use of the developed framework in which the objective function and the decision variables are set to be the net present value, and the location of infill wells, respectively.

Thesis declaration

I certify that this work contains no material which has been accepted for the award of any other degree or diploma in any university or other tertiary institution in my name and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. In addition, I certify that no part of this work will, in the future, be used in a submission in my name, for any other degree or diploma in any university or other tertiary institution without the prior approval of the University of Adelaide and where applicable, any partner institution responsible for the joint-award of this degree.

I give consent to this copy of my thesis, when deposited in the University Library, being made available for loan and photocopying, subject to the provisions of the Copyright Act 1968.

I also give permission for the digital version of my thesis to be made available on the web, via the University's digital research repository, the Library catalogue and also through web search engines, unless permission has been granted by the University to restrict access for a period of time.

Mohammad Sayyafzadeh 3 July 2013

Acknowledgement

My first and sincere appreciation goes to my supervisor ***Dr. Manouchehr Haghighi*** for being a great advisor and excellent supervisor. His constant support, encouraging character, and worthwhile suggestions added more value to my academic experience. Without his guidance and persistent help this dissertation would not have been possible.

I would also like to express my special thanks to ***Dr. Jonathon Carter***. It has been an honour to have him as a co-author in my publications. I appreciate all his contributions.

My grateful thanks also go to my co-supervisor ***Prof. Pavel Bedrikovetski*** for his motivating discussions and kindness.

I am grateful to ***Dr. Pourafshary*** and ***Prof. Rashidi,*** and my fellow friends, ***Elaheh Arjomand, Keivan Bolouri, Alireza Salmachi and Azadeh Mamghaderi***. Their academic support and friendly help are greatly appreciated.

Last but not least, I would like to thank whole my family for all their love and support.

List of publications

Journals:

- SAYYAFZADEH, M., POURAFSHARY, P., HAGHIGHI, M. & RASHIDI, F. 2011. Application of transfer functions to model water injection in hydrocarbon reservoir. Journal of Petroleum Science and Engineering, 78, 139-148.
- SAYYAFZADEH, M. & HAGHIGHI, M. 2013. An assessment of different model-management techniques in history matching problems for reservoir modelling, Australian Petroleum Production and Exploration Association Journal, 53, 391-406.
- SAYYAFZADEH, M., HAGHIGHI, M., BOLOURI, K. & ARJOMAND, E. 2012. Reservoir characterisation using artificial bee colony optimisation, Australian Petroleum Production and Exploration Association Journal, 52, 115-128.
- SALMACHI, A., SAYYAFZADEH, M. & HAGHIGHI, M. 2013. Infill well placement optimization in coal bed methane reservoirs using genetic algorithm. Fuel Journal, 111, 248–258.
- SALMACHI, A., SAYYAFZADEH, M. & HAGHIGHI, M. 2013. Optimisation and economical evaluation of infill drilling in CSG reservoirs using a multi-objective genetic algorithm. Australian Petroleum Production and Exploration Association Journal, 53, 381-390.
- SAYYAFZADEH, M., MAMGHADERI, A., POURAFSHARI, P. & HAGHIGHI, M. 2013. A fast simulator for hydrocarbon reservoirs during gas injection. Journal of Petroleum Science and Technology, (accepted).

Conferences:

- SAYYAFZADEH, M., HAGHIGHI, M. & CARTER, J. 2012. Regularization in history matching using multi-objective genetic algorithm and Bayesian framework. SPE EUROPEC 2012 Copenhagen, Denmark: Society of Petroleum Engineers.
- SAYYAFZADEH, M. & HAGHIGHI, M. 2013. High-resolution reservoir modeling using image fusion technique in history matching problems. SPE EUROPEC 2013 London, UK: Society of Petroleum Engineers.
- SAYYAFZADEH, M., MAMGHADERI, A., POURAFSHARI, P. & HAGHIGHI, M. 2011. A new method to forecast reservoir performance during immiscible and miscible gas-flooding via transfer functions approach. SPE Asia Pacific Oil and Gas Conference and Exhibition. Jakarta, Indonesia: Society of Petroleum Engineers.

به نام خداوند جان و خرد

Dedicated to my mother, *Mahin Ghebti*, *who is* a constant source of support and encouragement, and to the memory of my father, *Abdolmajid Sayyafzadeh* who passed away just before I embarked on this journey.

تقدیم به

مادرم بلند تکیه گاهم، که دامان پر مهرش یگانه پناهم است و روح پاک پدرم که حاصل دستان خسته اش رمز موفقیتم شد.

Chapter 1

1. Introduction

The fate of huge investments in the oil and gas industry rests on decision making. Reservoir decision making, for instance field development planning, can be improved by having access to accurate forecasts for the future performance of reservoirs. A reservoir simulator is usually employed for forecasting and providing foresight in the consequences of the decisions. Although, the literature is an indication of the maturity of the reservoir simulation subject (Fanchi, 2005, Aziz and Settari, 2002, Farmer, 2005, Sun, 1994), the forecasts using the reservoir simulators are not always reliable. The uncertainties in forecasting are mostly on account of the low-quality input data of reservoir simulators. Among input data, usually the geological model (static model) has a considerable uncertainty (Berta et al., 1994). In the process of constructing a geomodel, limited data usually are available to be manipulated. Therefore, some of the geomodel parameters are estimated based geostatistical (Journel and Huijbregts, 1978, Caers, 2005) and/or geophysical interpretations. The uncertainties in the static model may lead to incorrect investment decisions (figure 1.1). Therefore, it is crucial to reduce these uncertainties.

To reduce the uncertainties, supplementary data which are mostly dynamic data (well performance history and/or time-lapse seismic) may be used to update the initial model. Time-lapse seismic (4D seismic) data are not usually available, thus the normal practice is the implementation of well flow performance history. The process of tuning input

parameters based on actual data (history) is called history matching (figure 1.2). History matching is an inverse problem and performs contrary to reservoir simulators (the forward problem), i.e., in history matching problems, the flow performance data (observed) are the input and the outputs are the parameters which caused the observed data.

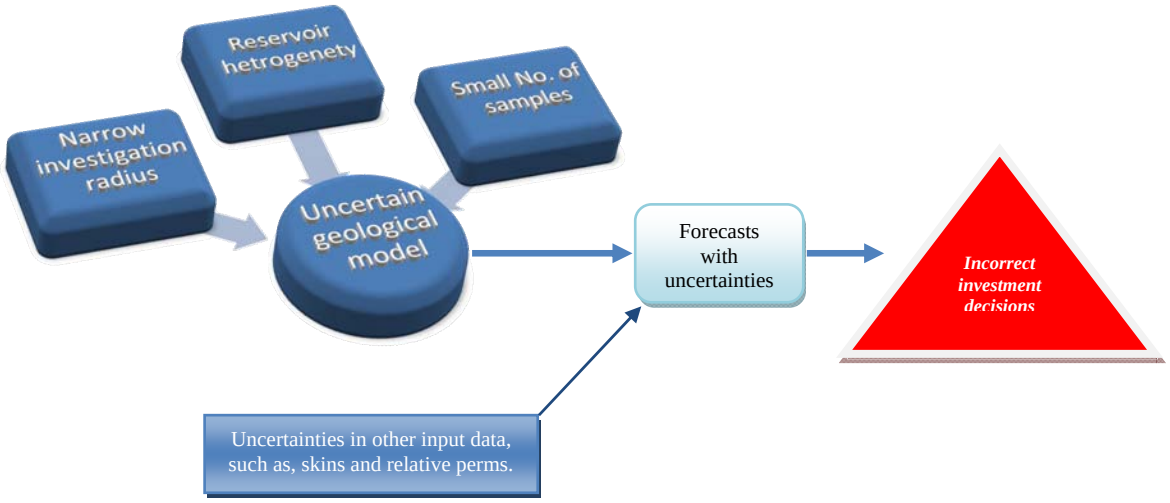


Figure 1.1 The effect of reservoir model uncertainties on investment decisions

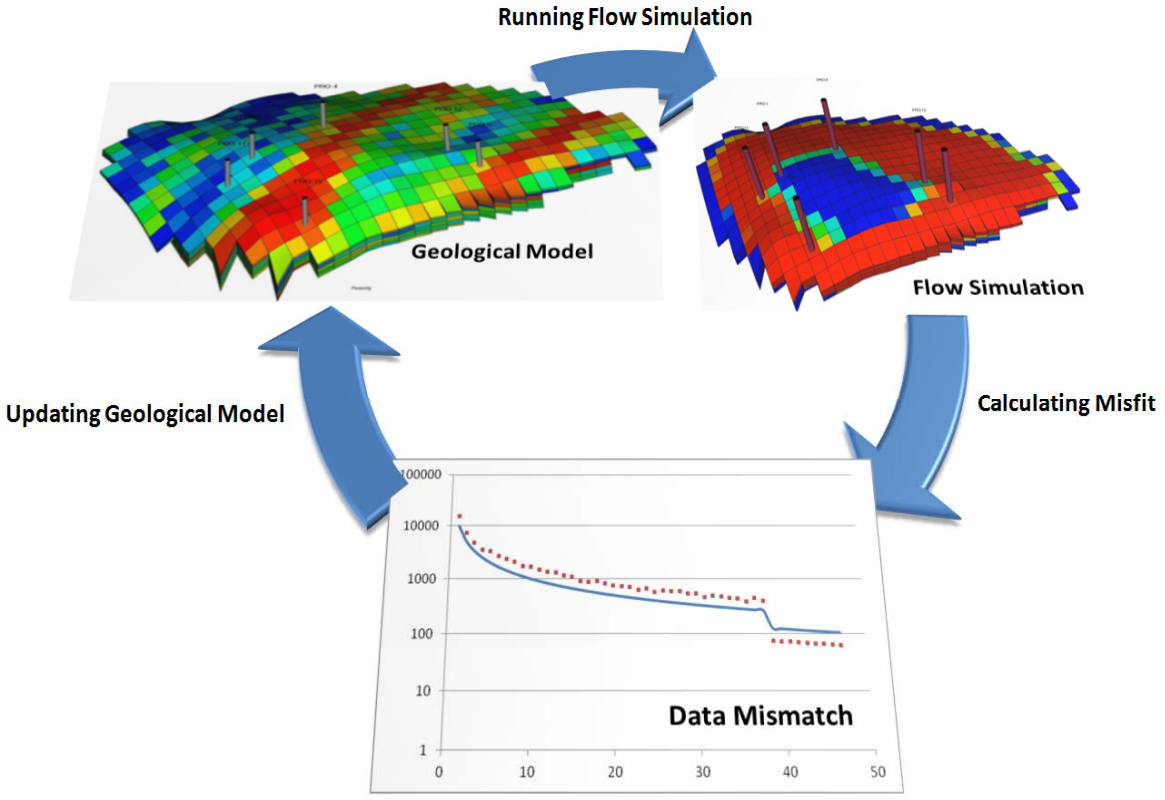


Figure 1.2 History matching schematically

1.1. Statement of the problem

History matching is a challenging task, due to the following issues:

i- The nonlinearity of the problem (which may cause nonconvexity)

ii- The large number of variables (which may cause underdetermination)

iii- Excessive computation time

iv- Measurement noise in observed data

v- Modelling errors

These issues create a complex inverse problem which usually suffers from ill-posedness.

1.2. Brief background and research gaps

To solve any nonlinear inverse problem (carry out history matching), the following three main steps are usually taken: 1- uncertain parameters (decision variables) are defined, this step is known as parameterisation, 2- an objective function is formulated; the objective function should indicates the quality of history matched models quantitatively and also include all specifications of the desired history matched model, and 3- calibration in which the variables are being adjusted until predefined criteria are met (figure 1.3).

Figure 1.3 The steps of history matching

Calibrations can be carried out manually or automatically. Manual history matching needs expertises, and only a few global parameters, such as average pore volume, aquifer and relative permeability parameters, are tuned (Oliver and Chen, 2010). Where the number of variables is more than a few, the large degree of freedom restricts the manual history matching application; in these problems, an automatic (usually known assisted) history matching approach should be utilised. In the automatic approach, the calibration is assisted by a computer software framework which typically makes use of a numerical optimisation rule. The main scope of this thesis is the automatic history matching.

The calibration in automatic history matching problems is an arduous task, since it is required to find the best history matched model in a system with a large degree of freedom while the function is nonlinear and computationally intensive. To carry out the calibration, one of the following approaches may be taken.

i- ***Direct calibration:*** the variables are directly tuned according to an optimisation rule (figure 1.4).

ii- ***Indirect calibration:*** the variables are indirectly tuned according to an optimisation through a reparameterisation operator (figure 1.5).

In the problems with a reasonable (not large) number of variables, the direct calibration is applied. In this approach, the main concern is the performance of the optimisation algorithm. In the problems in which the number of variables is large (which is often the case), usually the indirect calibration is applied. In the indirect calibration approach, first, the number of variables (the dimension of search space) is reduced by a reparameterisation method, and then the reparameterised model is calibrated using an optimisation algorithm. After completing the calibration, the obtained reparameterised model will be transformed back to the main space. Variables which are functions of spatial coordinates, such as porosity and permeability distributions are candidates of being reparameterised (Oliver and

Chen, 2010). In this approach, there are two major concerns: a- the efficiency of reparameterisation method, and b- the performance of optimisation algorithm.

In the direct or indirect approach, the optimisation algorithm typically deals with a search space which is not very high-dimensional (normally second order of magnitude), according to the fact that even if the number of variables is large, the number of variables and accordingly the dimension of search space is reduced using a reparameterisation technique prior to carrying out the optimisation. Although, the dimension of search space is not very large, the performance of the optimisation is one of in common concerns in both approaches (direct or indirect). The concern is owing to the nonlinearity and high computational expenses of the function. Hence, the augmentation of optimisation is one of the focuses of majority of the published studies (Romero and Carter, 2001, Mohamed et al., 2011, Ouenes et al., 1993, Yang and Watson, 1988, Zhang et al., 2003b).

Figure 1.4 Calibration through an optimisation rule (direct calibration)

In order to select the suitable optimiser for a problem, the specifications of the related objective function are required. These specifications are not usually measurable. Hence, the selection of optimiser is controversial. To prevent the examination of several optimisation algorithms for each problem, an algorithm is sought which is able to deliver tolerable approximation for the best history matched model in every problem with reasonable computational costs.

Figure 1.5 Calibration through a reparameterisation and an optimisation (indirect calibration)

Thus far, many different optimisation algorithms, from gradient-based (classical) optimisers (Yang and Watson, 1988, Zhang et al., 2005a, Wu et al., 1999) to non-gradient (stochastic) optimisers (Romero and Carter, 2001, Ouenes and Bhagavan, 1994), have been introduced and studied, but still a unique algorithm is not being taken toward it. In computer and mathematical sciences, several new black-box optimisers have been developed based on the evolutionary computation concepts. The application of some of these algorithms has not been investigated in history matching problems. *By assessing these algorithms in history matching problems, a suitable optimiser may be found which fulfils the requirements.*

Almost all of the recent-developed optimisers make use of a stochastic search. In general, stochastic optimisers are not computationally efficient, and this drawback usually limits the application of these algorithms in history matching problems (Oliver and Chen, 2010) and causes to disregard their advantages in contrast with gradient-based algorithms. The computation issue of history matching, which is highlighted particularly when a stochastic optimiser is used, caused to launch another research branch in which the goal is computational cost reduction. In a number of studies, proxy-modelling (fitness

approximation) methods have been proposed and developed. In these methods, the original fitness function which is time-consuming to be evaluated is substituted with an undemanding function known as the approximation function (proxy). Instead of approximating the global optimum point of objective function directly, the global optimum point of the approximation function (which is trained by a set of samples from the original function) is sought. This substitution reduces the expenses dramatically (Zubarev, 2009). As the approximation function is trained by a limited number of samples, there is the chance of misdirection of history matching procedure to wrong optimum points by applying these methods (Zubarev, 2009). So far, several techniques have been introduced to diminish the proxy-modelling errors (Ramgulam et al., 2007, Silva et al., 2008), the presented methods have error in estimating the global optimum (Zubarev, 2009). *Therefore, the improvement of proxy-modelling is an additional research area which may be achieved by designing or implementing the new techniques.*

In the indirect approach, despite the importance of the optimisation challenges, the reparameterisation operator plays the other important role. An inappropriate reparameterisation technique may lead to wrong history matched model(s), even if the optimisation algorithm performs perfectly. In the published studies related to the improvement of reparameterisation, a method is sought which is capable of describing full-parameterised models by significantly fewer number of variables with an enough precision (reasonable error). So far, many different techniques have been introduced (Jafarpour et al., 2010, Jafarpour and McLaughlin, 2009, Reynolds et al., 1996, Leo et al., 1986); each technique has own advantages and drawbacks and a comprehensive technique has not been developed.

Another way of removing the reparameterisation error entirely is making use of the direct approach, instead of an indirect calibration (Oliver and Chen, 2010). By a direct approach, although reparameterisation error does not exist, the optimiser needs to deal with very

high-dimensional search space. The only optimisation algorithms which have been examined as a direct calibrator in large problems are gradient-based ones (Zhang et al., 2005a). The nonlinearity and having multiple minima (which is likely (Mantica et al., 2002)) may concern the performance of the gradient-based algorithms. On the other hand, stochastic optimisers with their present forms are unlikely to be efficient in high-dimensional search space. Up to now, none of the stochastic algorithms has been used in full-parameterised history matching problems. *Thus, developing an optimiser which is able to deal with large number of variables can be one more research area in history matching.*

Obtaining high-quality history matched models using any of the history matching algorithms is subjected not only to the performance of optimisation and maybe reparameterisation algorithm, but also it depends on the reliability of objective function formulation. The most widespread approach of objective function formulation in history matching problems is Bayesian framework. The modelling error and measurement noise are required to be quantified and incorporated into the objective function formulation as the weighting factors. These elements are not accurately quantifiable, therefore they may have uncertainty. The influence of uncertainty of these elements on the outcomes of history matching has not been studied extensively and also a proper approach has not been developed for dealing with this issue. *The investigation of effects of uncertainties in objective function formulation on the performance of calibration is another research area which needs further study*.

With all these advances in the history matching subject, still this procedure is time-consuming. In some occasions, a quick overview about the performance of the reservoir is adequate, or history matching is not attainable. In these occasions, it is better to make use of a fast simulator. However, some methods like Decline Curve Analysis (DCA) (Arps, 1944, Baker et al., 2003) and the Capacitance Resistance Model (CRM) (Sayarpour et al., 2009a, Sayarpour et al., 2010) supply quick forecasts and also require the minimum data to

simulate reservoirs, their outputs are not very reliable in many cases, particularly secondary recovery and gas-flooding, respectively. On the other hand, grid-based simulation (Aziz and Settari, 2002, Fanchi, 2001) and streamline simulation (Rust and Caudle, 1972) are accurate methods, but they are time-consuming (especially grid-based simulators), and also large quantities of data and expensive history matching are required. In these occasions, a fast simulator which has reasonable accuracy is sought. *Therefore, researchers look for a method which needs minimum amount of data and has reasonable accuracy.*

1.3. Objectives

In this study, the main goal is introducing new methods to augment the automatic history matching process, and accordingly increase the quality of future performance predictions. Each of the proposed methods addresses a specific issue in history matching problems and belongs to the aforementioned research gaps. The study begins by *evaluating the application of one of the most recent optimisation algorithms in history matching problems* (first objective). In the next step (second objective), it is sought to *improve proxy-modelling by including model management techniques to reduce computational costs reliably, particularly for stochastic optimisers*. The subsequent step (third objective) is *to remove reparameterisation error by developing a specific stochastic direct calibration method which can cope with large number of variables*. Another (fourth) objective of this thesis is to *comprehend the effects of uncertainties in objective function formulation on the history matching results, and developing a method to prevail over this issue*. The last objective of this thesis is *to introduce a reliable fast simulator for occasions that quick forecasts are looked for*.

1.4. Outline of the methods and contributions

Five different methods are introduced in this thesis; each of these methods corresponds to the aforementioned objectives. The outlines of these methods are briefly explained in the following subsections, and they are described in details, in the chapters. To study the proposed methods and provide comparison with conventional methods, an automatic history matching framework is required. Thus, a software framework is developed in the start. This framework is coded in MATLAB programming language (MathWork, 2011b) to make use of the great capabilities of the MATLAB toolboxes. ECLIPSE-E100 (Schlumberger, 2010) (numerical reservoir simulator) is coupled with MATLAB. Also, in order to use geostatistical estimation, SGeMS software (Remy et al., 2009) is used. SGeMS was already coupled to MATLAB by Thomas Mejer Hansen (http://mgstat.sourceforge.net/), and the codes with some adjustments are utilised in our framework. By the developed package, it is possible to call any of the following software, ECLIPSE, SGeMS and MATLAB toolboxes (figure 1.6). Furthermore, it is possible to read the output files of them, and write or modify their input files. The proposed methods and a number of conventional methods of history matching are coded in this framework to deliver assessments.

It should be mentioned that the framework not only can be used in history matching problems, but also it may be used in production optimisation and economical analysis problems. The developed codes are implemented for a well placement optimisation problem in a coal bed methane reservoir (San Juan Basin). Infill drilling optimisation in CBM reservoirs is challenging, as several terms, including, permeability, water saturation, pressure, water treatment costs and gas price are influential. Thus, it needs an automatic framework to enable the evaluation of many different scenarios. The automatic framework and a customised genetic algorithm are used for this purpose. The corresponding results of

the accomplishment can be found in ***two published journal papers*** which are attached to the appendix, “Salmachi, Alireza, Mohammad Sayyafzadeh, and Manouchehr Haghighi "*Infill well placement optimization in coal bed methane reservoirs using genetic algorithm*" ***Fuel (2013)***”, and II- “Salmachi, Alireza, Mohammad Sayyafzadeh, and Manouchehr Haghighi "*Optimisation and economical evaluation of infill drilling in CSG reservoirs using a multi-objective genetic algorithm*" ***APPEA 2013***”.

Figure 1.6 The developed framework between MATLAB and ECLIPSE

1.4.1. Artificial bee colony algorithm

In the first part of this thesis, the major focus is placed on the optimisation (calibration) step. The artificial bee colony (ABC) (Karaboga, 2005) is incorporated into the framework, and its application in history matching is studied, for the first time. ABC is one of the most recent swarm intelligence algorithms and has an evolutionary computation approach. It was inspired by studying the swarm behaviour of honey bees. Its applications were examined in different fields, such as digital IIR filters (Nurhan, 2009), heat-transfer coefficient estimations (Zielonka et al., 2011), and several numerical functions (Karaboga

and Akay, 2009). The outcomes of this algorithm verify that ABC can deliver good-results in different problems, and it is not very sensitive to the objective function specifications (the shape of landscape).

The advantages and disadvantages of ABC in history matching problems are explored by evaluating its performance on a synthetic model, and its outcomes are compared with three regular history matching algorithms. In this part, the algorithms are used along with a pilot point reparameterisation method, since stochastic algorithms cannot be efficient in very high-dimensional search spaces (the inefficiency of stochastic algorithms in very high-dimensional search spaces is investigated in chapter 6). This method along with its results and discussions are presented in chapter 4. In the chapter, also, the effects of reparameterisation on the shape of landscape and the performance of the optimisation algorithms are cursorily studied. A peer-reviewed paper was published on this topic: "SAYYAFZADEH, M., HAGHIGHI, M., BOLOURI, K., & ARJOMAND, E. 2012. *Reservoir characterization using artificial bee colony*, in ***APPEA Journal,*** 2012; 52:115-128".

1.4.2. Proxy-modelling with evolution-control

The second part of this thesis addresses the computational challenges (costs). In this part, the applications of evolution-control (model management) techniques are introduced to enhance the quality of proxy-modelling. The uncontrolled approach (conventional approach in Petroleum discipline) has the potential of misleading the optimiser to wrong optimum points (Jin, 2005). To reduce the probability of misdirection, the original function should be applied along with the approximation function during the optimisation process. In order to make use of the original function efficiency, it is required to apply a model management (evolution-control) technique. There are three categories of model management: individual-based, population-based and adaptive. By employing each of

these techniques, a controlled fitness approximation is assembled which benefits from online learning. In this study, an adaptive evolution-control technique is customised based on heuristic fuzzy rules. These three approaches are compared with the uncontrolled (conventional) approach, and also they are compared with each other using PUNQ-S3 reservoir model. A genetic algorithm with a customised crossover, an artificial neural network and a Latin Hypercube Sampling strategy are used as optimiser, proxy model and experimental design respectively. The methodology, results and discussions are presented in chapter 5. Also, a peer-reviewed paper was published on this topic: "SAYYAFZADEH, M. & HAGHIGHI, M. 2013. *Assessment of different model management techniques in history matching problems for reservoir modelling*, in ***APPEA Journal***, 2013; 53"

1.4.3. Image-fusion

In this part, an inventive algorithm based on a multi-focus wavelet transform image fusion technique is proposed to deal with very high-dimensional search spaces using a stochastic search. Image fusion is a concept for combining a number of images into a single image to afford a more informative image. By making use of this technique and evolutionary computation concepts, an optimisation algorithm is designed in which different models intelligently and stochastically are merged based on their corresponding fitness values to produce a fitting model. It is a repetitive approach until stopping criteria are met. In order to evaluate the proposed algorithm, the outcomes of history matching using this algorithm on a synthetic reservoir model and also PUNQ-S3 reservoir model are compared with the results of a number of regular approaches (direct and indirect calibration methods). This method along with its results and discussion are presented in chapter 6. Also, a SPE conference paper was published on this topic: "SAYYAFZADEH, M. & HAGHIGHI, M. 2013. *High-resolution reservoir modelling using image-fusion in history matching problems*, in SPE EUROPEC conference London, 2013"

1.4.4. Pareto optimisation

The main emphasis of this part of the thesis is on the role of objective function formulation on the outcomes of history matching. To reduce risks in decision making for problems which have uncertain weighting factors, there is a conventional approach (Tarantola, 2005). This approach has not been used in history matching problems, before. It is based on random selection, thus it needs the execution of optimisation many times. Therefore, a computationally efficient alternative approach is introduced, in this study.

A Pareto optimisation is proposed for these problems in which likelihood and prior functions are considered as two separated objective functions. In the proposed approach, a multi-objective genetic algorithm is implemented to carry out the Pareto optimisation, and also specific post-optimisation trade-off rules are designed based on Taylor approximation series to enhance the quality of results. By a single execution of optimisation by this approach, a set of solutions is achieved that can be used for uncertainty analyses. A linear numerical example and PUNQ-S3 model are used to investigate the effect of weighting factors. The proposed and the conventional approach are compared with each other using these two examples, in terms of the required computation time and the quality of achieved solutions. This method along with its results and discussions are presented in chapter 7. Also, a SPE conference paper was published on this topic, "SAYYAFZADEH, M., HAGHIGHI, M. & CARTER, N. 2012, *Regularization in history matching using multi-objective genetic algorithm and Bayesian framework*, in SPE EUROPEC conference Copenhagen, 2012"

1.4.5. Fast simulator based on transfer functions

In this part, a new simulator is introduced for forecasting the performance of oil reservoirs during water and gas (miscible and immiscible) injection based on Transfer Function (TF).

In this method, it is assumed a reservoir consists of a combination of TFs. The order and arrangement of TFs are chosen based on the physical conditions of the reservoir which are ascertained by examining several cases. The method is combined to a fractional flow model to predict oil production rates along with total production rates. For water-flooding problems, the chosen fractional flow is the model introduced by Gentil, 2005, and for gas-flooding, a specific fractional flow is developed. Injection and production rates act as input and output signals to these TFs, respectively. By analysing input and output signals, the matching parameters of TFs are calculated. Subsequently, it is possible to predict reservoir performance. The results are compared with those obtained from the common grid-based simulator in different reservoir models.

This method along with its results and discussions are presented in chapter 8. A peer-reviewed journal and a SPE conference paper were published on this topic: "SAYYAFZADEH, M., POURAFSHARY, P., HAGHIGHI, M. & RASHIDI, F. 2011, *Application of transfer functions to model water injection in hydrocarbon reservoir*, ***Journal of Petroleum Science and Engineering***, 78, 139-148" and "SAYYAFZADEH, M., MAMGHADERI, A., POURAFSHARI, P. & HAGHIGHI, M. 2011. *A New Method to Forecast Reservoir Performance during Immiscible and Miscible Gas-Flooding via Transfer Functions Approach*, SPE APOGE Jakarta" (and accepted for being published by the ***Journal of Petroleum Science and Technology***).

This thesis continues by a mathematical preliminaries chapter in which the relevant mathematics is explained. The third chapter presents the past to present advances in history matching. The explanation of the abovementioned techniques along with their results and discussions form the chapters 4 to 8. The conclusive remarks and future works can be found in the last chapter. This thesis also has an appendix in which the related MATLAB codes and the results of infill drilling optimisation are provided.

Chapter 2

2. Mathematical preliminaries

This chapter presents the relevant theoretical and mathematical preliminaries of history matching. It begins with an introduction to the inverse problem theory and history matching, and then, the different steps of history matching are described.

2.1. Inverse problem theory

Forward problems (simulation) allow us to forecast the state of a performance by solving the corresponding mathematical equations, while in inverse problems, parameters that characterise a model (system) are estimated by inferring measurements (observations) (Tarantola, 2005). The solution of inverse problems delivers an estimation concerning the parameters which are not directly measurable, for example, the mass of sun, the mass distribution of earth and the subsurface properties of Earth. The inverse problem theory is being used in different fields, including seismic interpretations (Bouvier et al., 1989), image recovery (Stark, 1987), the estimations of heat transfer coefficients (Zielonka et al., 2011), machine learning (robotic) (Mjolsness and DeCoste, 2001), reservoir modelling and so on. Equation 1 demonstrates a forward problem.

$$d_{cal} = g(m) \tag{1}$$

m is the input model, g is the phenomena modelling (forward) operator and d is the state of performance of the model (calculation data). m is a column vector in a N_m dimensional

Hilbert space denoted by M (equation 2) and d is a column vector in a N_{obs} dimensional Hilbert space denoted by D (equation 3). N_m and N_{obs} are equal to the number of variables which characterise the model and the number of measurement data, respectively.

$$m = \begin{bmatrix} m_1 \\ \vdots \\ m_{N_m} \end{bmatrix} \tag{2}$$

$$d = \begin{bmatrix} d_1 \\ \vdots \\ d_{N_{obs}} \end{bmatrix} \tag{3}$$

g relates explicitly models from the M space to calculation data from the D space (equation 4). g can be either linear or nonlinear. If g is a linear operator, it will be denoted by G and will be a $N_{obs} \times N_m$ matrix. Usually g cannot perfectly predict the observations, and it almost always is accompanied by modelling errors. Modelling errors might be due to numerical solutions and/or imperfect phenomena modelling, and usually they are unknown. On the other hand, measurement noise is always included in observed data. Thus, in order to relate models to observations, equation 1 should be changed to equation 5 in which there are two other terms on right hand side, modelling error and measurement noise (error).

$$g: M \rightarrow D \tag{4}$$

$$d_{obs} = g(m) + \epsilon_{modelling} + \epsilon_{measurement} \tag{5}$$

In inverse problems, it is sought to estimate m from d_{obs} (equation 6). The forward operator is usually nonlinear and noninvertible in real problems. These issues along with unknown modelling and measurement errors cause complications in the estimation of m. To solve these inverse problems, usually the following three steps are being taken: parameterisation, objective function formulation and optimisation (Tarantola, 1987). In the parameterisation, variables which describe the model are determined. In the second step, an objective function is formulated in which the criteria of distinguishing between models (solutions) are delineated and quantified; equation 7 is one of the most famous objective function

definitions in inverse problems known as least squares (L^2). The last step is optimisation in which the global optimum of the defined objective function is sough. It is an iterative procedure in which the model is being updated according to a selected optimisation rule until predefined stopping criteria are met (Landa, 1997). These steps are described in details in the following sections. Figure 2.1 and 2.2 show the forward and inverse problem schematically.

$$m = g^{-1}(d_{obs} - \epsilon_{modelling} - \epsilon_{measurement}) \quad (6)$$

$$\varphi(m) = \left(d_{obs} - g(m)\right)^T \left(d_{obs} - g(m)\right) \quad (7)$$

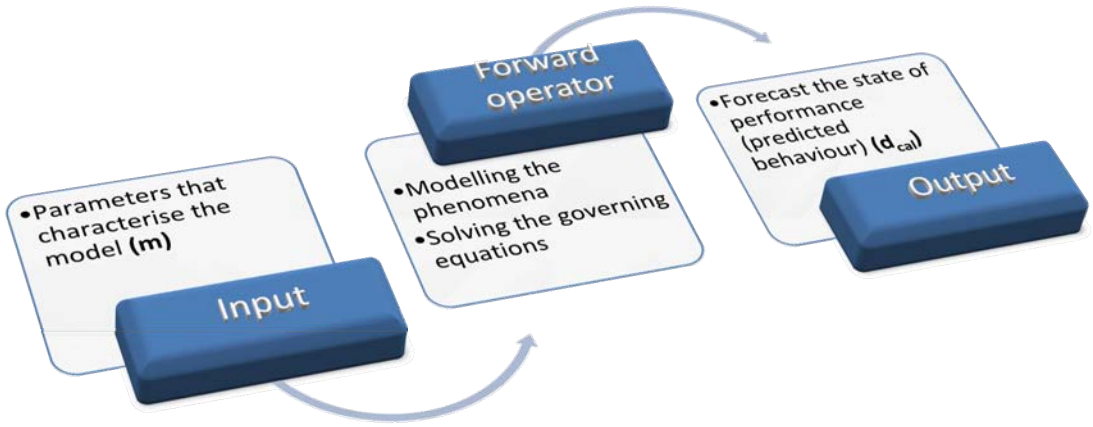


Figure 2.1 Forward problem

2.2. History matching in reservoir simulation

History matching is a nonlinear inverse problem (Ballester and Carter, 2007). In this inverse problem, the parameters of initially constructed models are updated based on the true behaviour of reservoir performance. History matching problems are often underdetermined, i.e., the number of variables is more than the number of independent observations. Furthermore, the forward operator is very time-consuming. In the forward problem, a system of differential equations formed by combining continuity equation, Darcy's equation, an equation of state, capillary pressure and relative permeability

relationships is solved (Landa, 1997, Aziz and Settari, 1979). Equation 8 is a sample equation which is solved in reservoir simulations for each phase and each gridblock (Fanchi, 2005). B_o, μ_o, k_{ro}, Φ_o, q_o and S_o are formation volume factor, viscosity, relative permeability, potential energy, flow rate and saturation of oil phase respectively. The aforementioned issues create a nonlinear ill-posed expensive inverse problem. Ill-posed problems have more than one solution with equally good fitting to the observations (Sun, 1994).

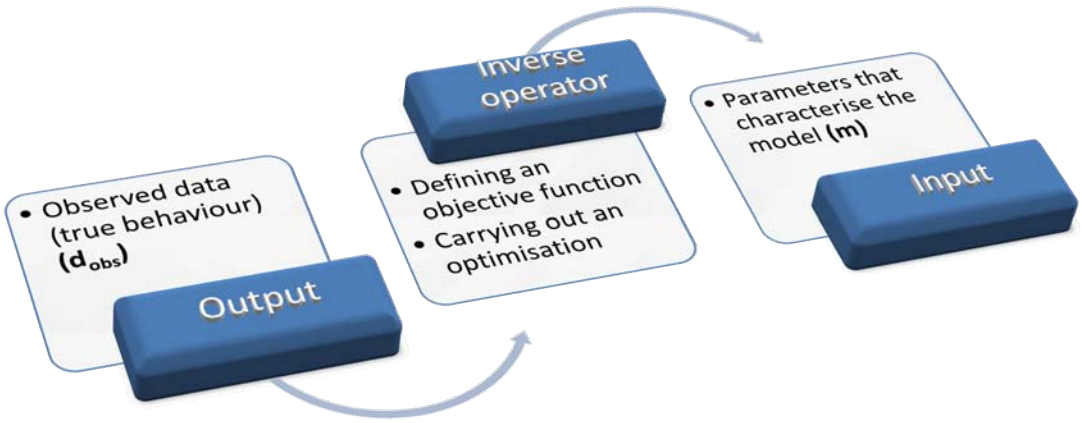


Figure 2.2 Inverse problem

$$\int \left[\nabla . \frac{K k_{ro}}{\mu_o B_o} \nabla \Phi_o - \frac{q_o}{\rho_o}\right] dV = \int \frac{\partial}{\partial t}\left(\frac{\phi S_o}{B_o}\right) dV \qquad (8)$$

History matching similar to any other inverse problems consists of the three steps which are described in the following sections.

2.3. Parameterisation

In the first step, it is required to define the uncertain parameters which can be any of the reservoir simulation input data. The reservoir simulation input data include fault transmissibility, well skins, fluid properties in each region (capillary pressure curves, relative permeability curves and fluid composition), aquifer parameters (size, pressure,

connections and permeability), water-oil contacts, gas-oil contacts, initial pressures and saturations, and geological model parameters (porosity, permeabilities, net to gross ratio in every gridblocks, and fracture's parameters) (Oliver and Chen, 2010). Therefore, *m* usually consists of only the uncertain parameters. The dimension of M is equal to the number of elements of *m*, and it is denoted by N_m.

2.3.1. Reparameterisation

The first step is not completed only by a parameterisation, due to the fact that usually there are a huge number of uncertain parameters which are predominantly the properties' distribution across the reservoir such as porosity and permeabilities in every gridblock. Solving a nonlinear inverse problem with a very large degree of freedom (or in other word, finding the best model in a very high-dimensional space) is challenging. In order to perform the calibration (optimisation) more efficient and quicker, making the search space smaller (reducing the number of dimension of M) is advantageous. A reparameterisation method may be utilised to reduce the number of variables. Variables which are functions of spatial coordinates, such as porosity and permeability are suitable candidates to be reparameterised (Oliver and Chen, 2010), since usually there are some correlations between these parameters, and hence they may be represented by a lower number of variables.

In order to reparameterise a model, an operator is required which is called reparameterisation operator and denoted by *f*. *f* maps M to R (equation 9). R is the corresponding space of the reparameterised model (*r*). The dimension of corresponding Hilbert space of *r* denoted by N_r is usually much smaller than N_m. Now, instead of calibrating the elements of *m* directly in a very high-dimensional space, it is possible to carry out history matching indirectly in a smaller space by calibrating the elements of *r*. In this approach, modifications in *m* are made by changes in *r* through h. *f* normally is not

invertible, therefore r is related to m using another function denoted by h. In equation 9, ϵ_R (reparameterisation modelling error) is usually an unknown function. It is very important to remove or reduce the reparameterisation error with choosing an appropriate operator (h) and the optimal number of variables, otherwise the approximation in equation 9 will be incorrect, and accordingly the achieved models via it will not be reliable models.

$$f: M \to R\,; \quad r = f(m) \;\; and \;\; h: R \to M\,; \quad m = h(r) + \epsilon_R \quad \to \quad m \approx h(r) \tag{9}$$

So far, several reparameterisation methods have been introduced; the advantages of disadvantages of the methods are reviewed in the literature review chapter. In the following subsections, the three of the methods are explained as examples. In order to demonstrate the methods, a synthetic porosity distribution (m_{Ref}) (figure 2.3) is reparameterised by the methods. The full-parameterised model consists of 1225 variables. Then, the best reparameterised models are found via an optimisation algorithm based on equation 10. In the end, the achieved models are compared with the reference model both quantitatively based on equation 10 and also qualitatively by presenting the figures.

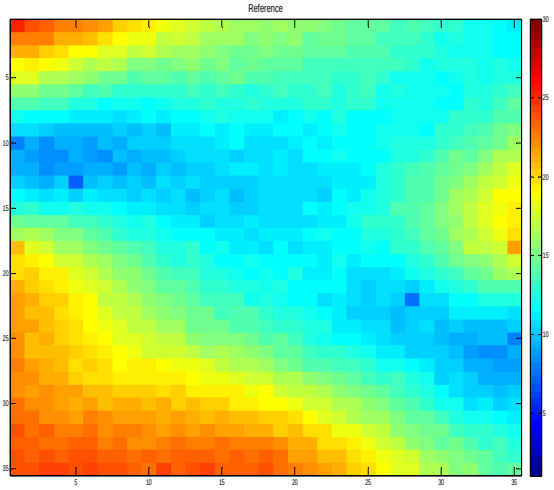


Figure 2.3 Porosity distribution of the reference model for the example

$$m_{Rep.} = h(r); \quad Misfit_{Rep.}(r) = \sqrt{\frac{1}{N_m}\sum_{i=1}^{N_m}\left(m_{Rep_i} - m_{Ref_i}\right)^2} \tag{10}$$

2.3.1.1. Zonation

In this method, the reservoir is divided into a number of coarse zones, and it is assumed properties are homogenous in each zone. The number of variables can be significantly reduced via this method. The procedure is as follows: the number of zones and the corresponding sizes are defined, and then the property in each zone will be sought by history matching. Figure 2.4 shows the best reparameterised model which can be achieved via a zonation method for the model. The figure 2.4 (right) shows the reference model. The left figure reveals the same reservoir after the zonation reparameterisation. In this zonation method, the reservoir model is divided into 25 same size zones with 7×7 gridblocks. Thus, the number of variables is significantly reduced from 1225 to 25. As it can be seen, it delivers the main features of the model reasonably (low porosity and high porosity sections are described acceptably), but it has a low resolution. The misfit value based on equation 10 for the reparameterised model using this approach is 1.62.

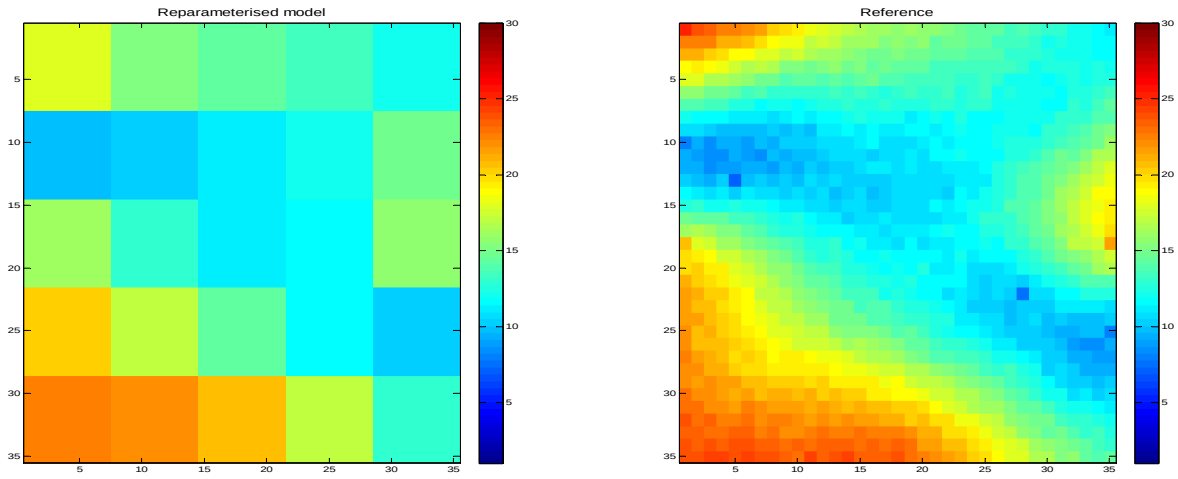


Figure 2.4 Reparameterisation using the zonation method

2.3.1.2. Pilot point reparameterisation

In this method, only a number of gridblocks are tuned by the optimisation, and the rest of the gridblocks are estimated by geostatistical interpolations. The gridblocks which are calibrated are called pilot points and act as pseudo-wells. The procedure of this method is

usually as follows: 1- The number of pilot points is defined and fixed. 2- The location of them is defined and fixed. 3- Using an optimisation algorithm, the best values for the selected pilot points are estimated. In this process, to evaluate each scenario, the properties in other gridblocks should be estimated using a geostatistical correlation. Thus, in the *h* function, *m* is estimated from *r* by a geostatistical computation in which usually the well data (the properties in gridblocks which wells are located at) are used along with the pilot points' data. In this study, SGeMS software (Remy et al., 2009) is used for generating geostatistical correlations.

In this paragraph, it is briefly explained how estimations in geostatistics are made by presenting the mathematical preliminaries for one of the most common geostatistical estimators, the Ordinary Kriging. Kriging is a method which estimates the value in unsampled locations based on the correlation of the sampled data. In this method, the property in each gridblock (x^*) is estimated based on the *n* measured samples with a specific weighting factor (ω). The summation of these weighting factors is equal to one (equation 11). Each weighting factor is calculated by solving a linear system shown in equation 12 in which the variogram (covariance) information is required. The variogram (γ) is usually interpreted by analysing the sample data (well logs or core samples) and providing a correlation for the variogram; the most famous three variogram models are: Spherical, Gaussian and Exponential. After calculating the weighting factors, the property in x^* can be estimated through equation 13. There are different methods, except ordinary kriging, such as simple kriging, co-kriging and several stochastic methods (sequential Gaussian simulation, sequential indicator simulation). In the following studies, the concepts of geostatistics can be found in more details (Journel and Huijbregts, 1978, Deutsch, 2002, Kelkar and Perez, 2002, Caers, 2005).

$$\sum_{i=1}^{n} \omega_i = 1 \qquad (11)$$

$$\begin{bmatrix} \omega_1 \\ \vdots \\ \omega_n \\ \mu \end{bmatrix} = \begin{bmatrix} \gamma(x_1,x_1) & \cdots & \gamma(x_1,x_n) & 1 \\ \vdots & \ddots & \vdots & \vdots \\ \gamma(x_n,x_1) & \cdots & \gamma(x_n,x_n) & 1 \\ 1 & \cdots & 1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} \gamma(x_1,x^*) \\ \vdots \\ \gamma(x_1,x^*) \\ 1 \end{bmatrix} \quad (12)$$

$$\hat{Z}(x^*) = \begin{bmatrix} \omega_1 \\ \vdots \\ \omega_n \end{bmatrix}^{-1} \begin{bmatrix} Z(x_1) \\ \vdots \\ Z(x_n) \end{bmatrix} \quad (13)$$

Using the pilot point method, the example is reparameterised by considering 6 points with unfixed locations and all the main elements of variogram unknown (nugget, contribution, range, anisotropy in x-direction and y-direction) as variables. Thus, the total number of variables is 23; each pilot point is represented by three parameters, x-location y-location and value and the variogram is represented by 5 parameters. The best reparameterised model found by an optimisation algorithm is shown in figure 2.5. As it can be seen, the reparameterised model, which is achieved by an almost similar number of variables as the zonation, has a high-resolution and also it carries the main characteristic of the model. The misfit for this model is 0.79 which is lower than the previous method.

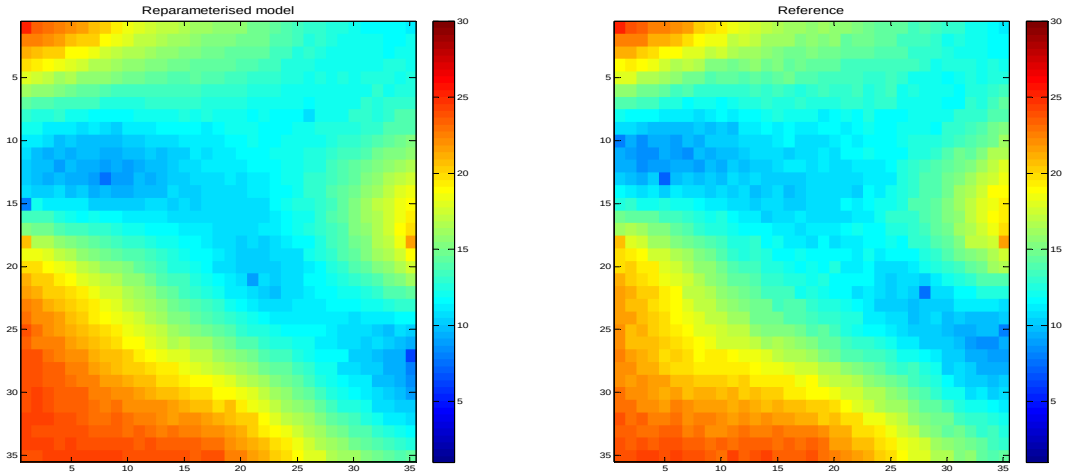


Figure 2.5 Reparameterisation using the pilot point method

2.3.1.3. Spectral decomposition

The spectral decomposition of prior covariance matrix is another method of reparameterisation. In this method, models are reparameterised based on eigenvalues and eigenvectors of prior covariance matrix (C_m). The procedure is as follows: 1- A diagonal

square matrix denoted by D should be produced in which the diagonal elements are equal to square root of diagonal elements of the prior covariance matrix ($\sqrt{c_{ii}}$). 2- A correlation matrix should be calculated via equation 14. 3- The eigensystem of $\tilde{C}_m$ should be computed; then, the eigenvalues should be ordered from largest to smallest ($\lambda_m \leq ... \leq \lambda_1$), assume u_j is the corresponding eigenvector for λ_j. 4- The number of parameter reduction should be defined (p); usually equation 15 is used to defined p in which θ is the fraction (0-1) of the total spectrum included in the first p eigenvalues. 5- U_p matrix should be constructed which is $N_m \times p$; the columns of this matrix are the first *p* eigenvectors, 6- *m* is varied by changes in the elements of a column vector (*r*) which is p-dimensional through equation 16. Thus, *h* in this reparameterisation technique is as equation 16 in which D and U_p are calculated based on a prior covariance matrix decomposition.

$$\tilde{C}_m = D^{-1} C_m D^{-1} \tag{14}$$

$$\sum_{i=1}^{p} \lambda_i \geq \theta \sum_{i=1}^{N_m} \lambda_i \tag{15}$$

$$m = D U_p r \tag{16}$$

The example is reparameterised using this method. It is assumed that p is equal to 25. A prior covariance matrix is constructed using geostatistical correlations. It should be mentioned that the prior covariance matrix is generated numerically based on only nine gridblock data and therefore, it has uncertainty. The best model is shown in figure 2.6. As it can be seen, it does not properly represent the reference model and the misfit value is equal to 5.1. In order to deliver a proper model, it is required to increase the number of variables of p and also modify the prior covariance matrix by gathering more information. To calculate the eigenvalues and eigenvectors, MATLAB functions are used.

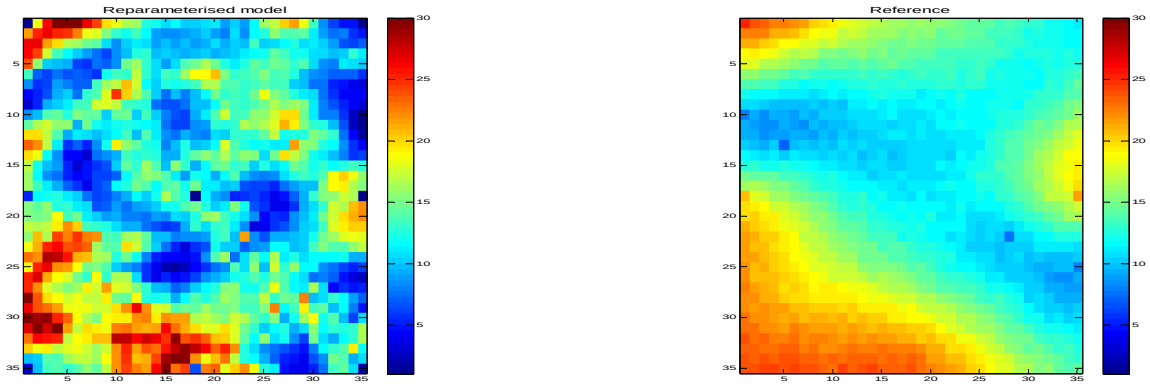


Figure 2.6 Reparameterisation using the spectral decomposition method

2.4. Objective function formulation

After defining the variables in the first step (parameterisation), it is required to identify decisive factors which are sought. These factors should express the quality of history matched models and also distinguish between the models. With regards of criteria (factors), an objective function (fitness function) is formulated to demonstrate the goodness of models quantitatively. In history matching problems, the objective function is typically formulated in two different manners: deterministic and probabilistic. In the deterministic approach, usually the objective function is similar to the least squares problems (L^2), and the minimum of squared data misfit is sought. In the probabilistic approach, normally, the objective function is formulated using a Bayesian interpretation, and the posterior probability function is being sought (Aanonsen et al., 2009, Shah et al., 1978, Oliver, 1994, Tarantola, 2005). These two approaches are explained in the following sections.

2.4.1. Deterministic formulation

The most common deterministic formulation is least squares of data misfits. In this method, the objective function is defined as the sum of the squares of misfits between observations and calculations provided by model (residuals) in every equation. In the linear problems, calculation data (d_{cal}) are related to model by a forward operator which is a $N_{obs} \times N_m$ matrix (equation 17, 18). The solution of the least square problems is the

minimum of summation of the residuals (equation 19, 20). In order to find the minimum of equation 20, its derivative for each element of *m* should be equal to zero (equation 21).

$$G = \begin{bmatrix} X_{11} & \cdots & X_{1N_m} \\ \vdots & \ddots & \vdots \\ X_{N_{obs}1} & \cdots & X_{N_{obs}N_m} \end{bmatrix} \tag{17}$$

$$\begin{bmatrix} d_{cal_1} \\ \vdots \\ d_{cal_{N_{obs}}} \end{bmatrix} = \begin{bmatrix} X_{11} & \cdots & X_{1N_m} \\ \vdots & \ddots & \vdots \\ X_{N_{obs}1} & \cdots & X_{N_{obs}N_m} \end{bmatrix} \begin{bmatrix} m_1 \\ \vdots \\ m_{N_m} \end{bmatrix} \tag{18}$$

$$r_i = d_{obs_i} - \sum_{j=1}^{N_m} (X_{ij} m_j) \tag{19}$$

$$S = \sum_{i=1}^{N_{obs}} \left(d_{obs_i} - (Gm)_i\right)^2 \tag{20}$$

$$\forall\, (j = 1, \dots, N_m), \qquad \frac{\partial S}{\partial m_j} = 0 \tag{21}$$

The derivation of equation 20 is as equation 22.

$$\frac{\partial S}{\partial m_j} = 2 \sum_{i=1}^{N_{obs}} r_i \frac{\partial r_i}{\partial m_j} \tag{22}$$

The derivation of the residual is as equation 23.

$$\frac{\partial r_i}{\partial m_j} = -X_{ij} \tag{23}$$

If equation 23 and 19 are incorporated into equation 22, equation 22 will be equation 24:

$$\frac{\partial S}{\partial m_j} = -2 \sum_{i=1}^{N_{obs}} \left(\left(d_{obs_i} - \sum_{j=1}^{N_m} (X_{ij} m_j) \right) X_{ij} \right) = 0 \tag{24}$$

The solution (m^*) of equation 24 is as equation 25 which can be rewritten as equation 26.

$$\sum_{i=1}^{N_{obs}} (d_{obs_i} X_{ij}) = \sum_{i=1}^{N_{obs}} \left(\sum_{j=1}^{N_m} X_{ij}\, X_{ij} {m^*}_j \right) \tag{25}$$

$$m^* = (G^T G)^{-1} G^T d_{obs} \quad (26)$$

If a sum of weighted squares of misfit is sought to be minimised, the solution will be as equation 27. The weighting factors are usually estimated based on measurement noise in observations. Measurement noise is usually assumed Gaussian with a zero mean. The weighting factors are assumed as standard deviations (σ) of the measurement noise.

$$m^* = (G^T W G)^{-1} G^T W d_{obs} \quad (27)$$

If there is any measurement noise in observations, the final residual (r(m^*)) will not be zero and will be a positive value. If the weighting factors in the objective function are equal to the standard deviation of the measurement noises, usually the residual at the m^* will be around N_{obs} ($r_i(m^*) = \frac{d_{obs_i} - \sum_{j=1}^{N_m}(X_{ij} m^*{}_j)}{w_i}$, $\sum_{j=1}^{N_m}(X_{ij} m^*{}_j) = d_{cal_i}, d_{obs_i} - d_{cal_i} \approx \sigma_i, r_i(m^*) \approx \frac{\sigma_i}{\sigma_i}$, $\mathrm{R}(m^*) = \sum_{i=1}^{N_{obs}} r_i(m^*)$, $thus\ \mathrm{R}(m^*) \approx N_{obs}$).

Linear problems are quadratic and also have closed-form solutions, but the nonlinear least squares problems do not have closed-form solutions, and they should be found by iterative approaches. History matching is a nonlinear least squares problem. In these problems, the derivative of g is not only a function of coefficients, but also it is a function of variables (m). To solve nonlinear least squares problems, an optimisation algorithm is required to find the minimum of equation 28. The optimisation methods are reviewed in a separate section later.

$$S(m) = \sum_{i=1}^{N_{obs}} \left(d_{obs_i} - \left(g(m) \right)_i \right)^2 \quad (28)$$

In this formulation, a regularisation term will be added, if the problem is underdetermined or/and ill-posed (Tikhonov and Arsenin, 1977). Equation 29 shows an objective function which has weighting factors and a Tikhonov regularisation term. The regularisation acts as a penalty term, and it can convert ill-posed problems to well-posed problems. λ is the

regularisation factor and can be any value from zero to infinity. W and W` are weighting factors for individual elements of data misfit and the prior model, respectively. More details about the least squares methods can be found in the following books (Wonnacott and Wonnacott, 1981, Stewart, 1973).

$$S(m) = \sum_{i=1}^{N_{obs}} \left(\frac{d_{obs_i} - \left(g(m)\right)_i}{W_i} \right)^2 + \lambda \sum_{i=1}^{N_m} \left(\frac{m_i - m_{prior_i}}{W'_i} \right)^2 \quad (29)$$

2.4.2. Probabilistic formulation

In this approach, the objective function is formulated based on probability distribution functions, and it is sought to find the probability distribution function of the model while is conditioned to (gives) observations ($p(m|d_{obs})$). It is widely used for parameters estimations (Sivia and Skilling, 2006). Finding a model using this approach by any number of experiments is practical. The degree of confident regarding the model is increased, if more independent experiments become available. The following example taken from Sivia and Skilling book is a good mean to describe this approach. There is a coin which should be investigated to answer the following questions a- is the coin bias, and b- if it is bias, what is the factor. To answer these questions, the best way is flipping the coins and gathering information. The following figure shows the corresponding graphs based on the number of experiments (figure 2.7). As it can be seen, via an increase of the number of experiments from zero to 64, a better probability distribution is generated, and accordingly a better decision can be made.

The probability distribution of *m* while is conditioned to observations is usually not directly measurable, especially in nonlinear inverse problems. In these problems, based on Bayes' theorem, it will be possible to define the function based on three other probability distribution functions as equation 30. It demonstrates how a subjective degree of belief logically varies based on observations (evidence). It is used to update the degree of belief

regarding a hypothesis (p(m)) in line with additional observations. Bayes' theorem is the basis of Bayesian statistics. In the following section, the Bayesian framework is described.

$$p(m|d_{obs}) = \frac{p(d_{obs}|m) \times p(m)}{p(d_{obs})} \tag{30}$$

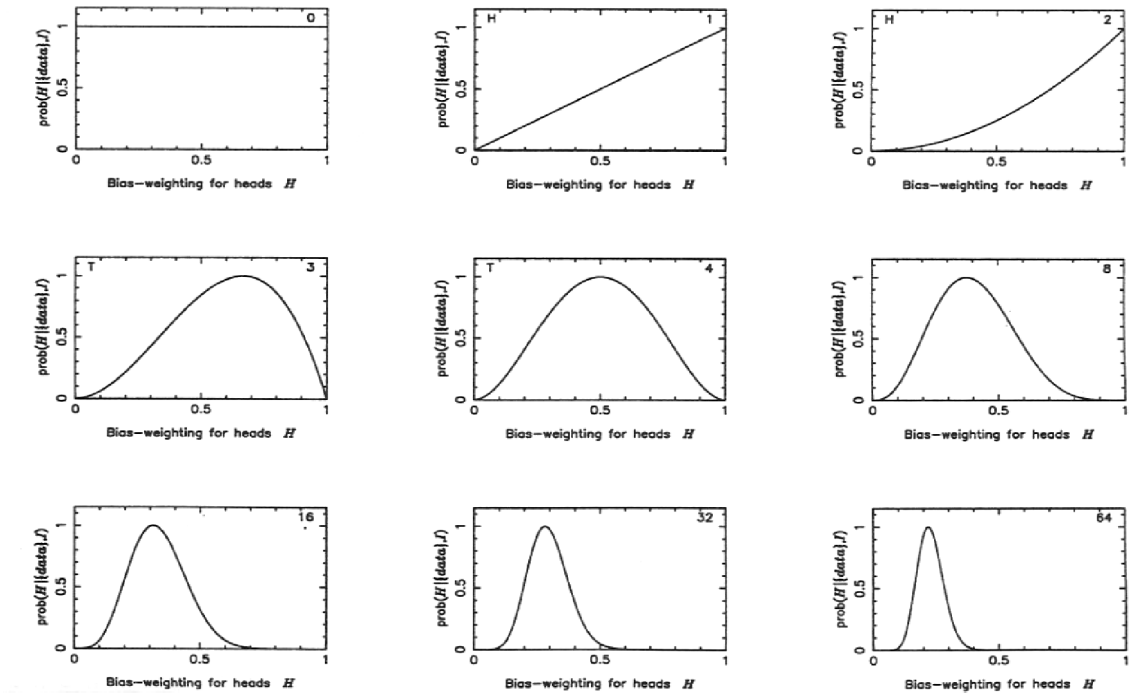


Figure 2.7 The degree of biasness of a coin based on observations (Sivia and Skilling, 2006)

2.4.3. Bayesian framework

Using a Bayesian interpretation, inverse problems can be formulated. In inverse problems, it is sought to find the maximum of $p(m|d_{obs})$ denoted by MAP. MAP is conditioned to the observations and also it is the most probable model. If the $p(m|d_{obs})$ function can be estimated, not only it will be possible to pick the MAP, but also a number of other models can be selected from the function which are all conditioned to the observations. These models have a lower probability of existence than MAP. They can provide a high-quality uncertainty analysis and quantification, and accordingly, the risks in decision making can be reduced, since other probable models are also available and can be investigated. Thus, the goal in this interpretation is finding the posterior probability distribution function.

In order to find the posterior probability function, the best way is to relate this function to some other terms which are measurable. According to Bayes' theorem, the posterior probability function is a function of three other probability functions. The $p(d_{obs})$ called marginal likelihood (model evidence) is not a function of *m* and is a fixed value. Therefore, this term is not used to estimate posterior probability functions ($p(m|d_{obs})$). The other two terms change posterior probability functions (equation 31). If a natural logarithm is taken from both sides of equation 31, it will be as equation 32. The first term in the right handside is called likelihood probability distribution function and the second term is called prior probability distribution function. Hence, the posterior degree of belief regarding a hypothesis is a function of the inherent degree of belief regarding the hypothesis (the prior) and the compatibility of the observations (evidences) with the hypothesis (the likelihood).

$$p(m|d_{obs}) \propto p(d_{obs}|m) \times p(m) \tag{31}$$

$$\ln\big(p(m|d_{obs})\big) \propto \big(\ln\big(p(d_{obs}|m)\big) + \ln\big(p(m)\big)\big) \tag{32}$$

With an assumption of Gaussian distribution of modelling error and measurement noise and zero means, the likelihood function can be defined as equation 33. In this equation, C_D is the covariance matrix of observation data which consists of two covariance matrixes (C_d and C_T), as equation 34. C_d should include the measurement noise of observation data and C_T should include the modelling errors. If observations are uncorrelated, the covariance matrix will be diagonal and the entries will be equal to the squared standard deviation of measurement noise, as equation 35.

$$\mathrm{p(d_{obs}|m)} = ((2\pi)^{\mathrm{N_{obs}}}\det(\mathrm{C_D}))^{-\frac{1}{2}}\exp\left(-\frac{1}{2}(\mathrm{g(m)}-\mathrm{d_{obs}})^{\mathrm{t}}\mathrm{C_D}^{-1}(\mathrm{g(m)}-\mathrm{d_{obs}})\right) \tag{33}$$

$$\mathrm{C_D} = \mathrm{C_d} + \mathrm{C_T} \tag{34}$$

$$\mathrm{C_d} = \begin{bmatrix} \sigma_1^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_{N_{obs}}^2 \end{bmatrix} \tag{35}$$

With an assumption of Gaussian distribution of model, prior probability distribution function will be as equation 36. In this equation, m_{prior} is the mean of distribution and usually is the initial guess regarding the model and C_m is the corresponding covariance matrix for the initial probability distribution function. C_m expresses the initial degree of confidence about the model. If m is a single variable, the C_m will be the standard deviation, but usually m is a multivariate distribution.

$$\mathrm{p(m)} = ((2\pi)^{\mathrm{N_m}}\det(\mathrm{C_m}))^{-\frac{1}{2}}\exp\left(-\frac{1}{2}(\mathrm{m}-\mathrm{m_{prior}})^{\mathrm{t}}\mathrm{C_m}^{-1}(\mathrm{m}-\mathrm{m_{prior}})\right) \quad (36)$$

By incorporating equation 33 and 36 in 32, equation 37 will be achieved. The coefficients of probability distributions are constant and they are not variables of m, thus equation 37 can be rewritten as equation 38 in which C_1 and C_2 are two constants.

$$\ln(p(m|d_{obs})) \propto \left(\ln\left(((2\pi)^{\mathrm{N_{obs}}}\det(\mathrm{C_D}))^{-\frac{1}{2}}\exp\left(-\frac{1}{2}(\mathrm{g(m)}-\mathrm{d_{obs}})^{\mathrm{t}}\mathrm{C_D}^{-1}(\mathrm{g(m)}-\mathrm{d_{obs}})\right)\right) + \ln\left(((2\pi)^{\mathrm{N_m}}\det(\mathrm{C_m}))^{-\frac{1}{2}}\exp\left(-\frac{1}{2}(\mathrm{m}-\mathrm{m_{prior}})^{\mathrm{t}}\mathrm{C_m}^{-1}(\mathrm{m}-\mathrm{m_{prior}})\right)\right)\right) \quad (37)$$

$$\ln(p(m|d_{obs})) \propto \left(C_1 + \left(-\frac{1}{2}(\mathrm{g(m)}-\mathrm{d_{obs}})^{\mathrm{t}}\mathrm{C_D}^{-1}(\mathrm{g(m)}-\mathrm{d_{obs}})\right) + C_2 + \left(-\frac{1}{2}(\mathrm{m}-\mathrm{m_{prior}})^{\mathrm{t}}\mathrm{C_m}^{-1}(\mathrm{m}-\mathrm{m_{prior}})\right)\right) \quad (38)$$

In order to find the maximum of $p(m|d_{obs})$, it is required to find the minimum of equation 39, i.e., the minimum of equation 39 is equal to the maximum of $p(m|d_{obs})$. This form is very similar to equation 29 (least squares with a Tikhonov regularisation term). In this equation covariance matrixes acts as weighting factors for the individual elements and the ratio of covariance matrixes play the role of a regularisation factor. The likelihood function demonstrates the fitting of observed data over calculation data. The prior knowledge function (regularisation term) exhibits the distance of calibrated model from an initial model. It should be pointed out that sometimes only the likelihood term is used for parameter identification. In these cases, it is sought to find the maximum of likelihood

which is called ML (maximum likelihood), and it will be similar to least squares method without a regularisation term.

$$S(m) = \left(\frac{1}{2}(\mathrm{g(m)} - \mathrm{d_{obs}})^{\mathrm{t}}\mathrm{C_D}^{-1}(\mathrm{g(m)} - \mathrm{d_{obs}})\right) + \left(\frac{1}{2}\left(\mathrm{m} - \mathrm{m_{prior}}\right)^{\mathrm{t}}\mathrm{C_m}^{-1}\left(\mathrm{m} - \mathrm{m_{prior}}\right)\right) \quad (39)$$

In linear inverse problem, the posterior probability function will be Gaussian with a mean equal to MAP and a covariance matrix denoted by $C_{m`}$ (any multivariate Gaussian distribution is characterised by two terms, mean and covariance matrix), due to the fact that the misfit function is quadratic (Tarantola, 2005). The mean of posterior probability function (MAP) will be as equation 40 and the covariance matrix will be as equation 41. The posterior covariance matrix reveals the current degree of belief regarding the variables.

$$MAP = m_{prior} + C_m G^t (G C_m G^t + C_D)^{-1}\left(d_{obs} - G m_{prior}\right) \quad (40)$$

$$C_{m'} = C_m - C_m G^t (G C_m G^t + C_D)^{-1} G C_m \quad (41)$$

An example is created to demonstrate the application of Bayesian framework in parameters identifications. It is sought to find a set of models given the observations. There is prior information regarding the model as equation 42. 10000 samples are selected from the corresponding probability distribution function and shown in figure 2.8. As it can be realised from the histogram of the samples, there is a wide ranges for x and y of m which are varied almost between -5 to 3 and -4 to 4 respectively. There are two observations which are related to *m* using a linear operator *G* (equation 43). The true value which generated the observation is given in equation 44. Gaussian noise with zero mean and standard deviation of 5 percents of corresponding element are added to observations, thus, the diagonal elements of observation covariance matrix is equal to the squared standard deviations. Now, we are looking to update the initial probability distribution based on the observations.

$$m_{prior} = \begin{bmatrix} -1 \\ 1 \end{bmatrix}, C_m = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (42)$$

$$G = \begin{bmatrix} 0.004 & 0.769 \\ 0.774 & 0.947 \end{bmatrix} \quad (43)$$

$$m_{actual} = \begin{bmatrix} -2 \\ 1.5 \end{bmatrix} \quad (44)$$

Based on equation 40 and 41, MAP (m_∞) and $C_{m`}$ of posterior probability function will be as equation 45.

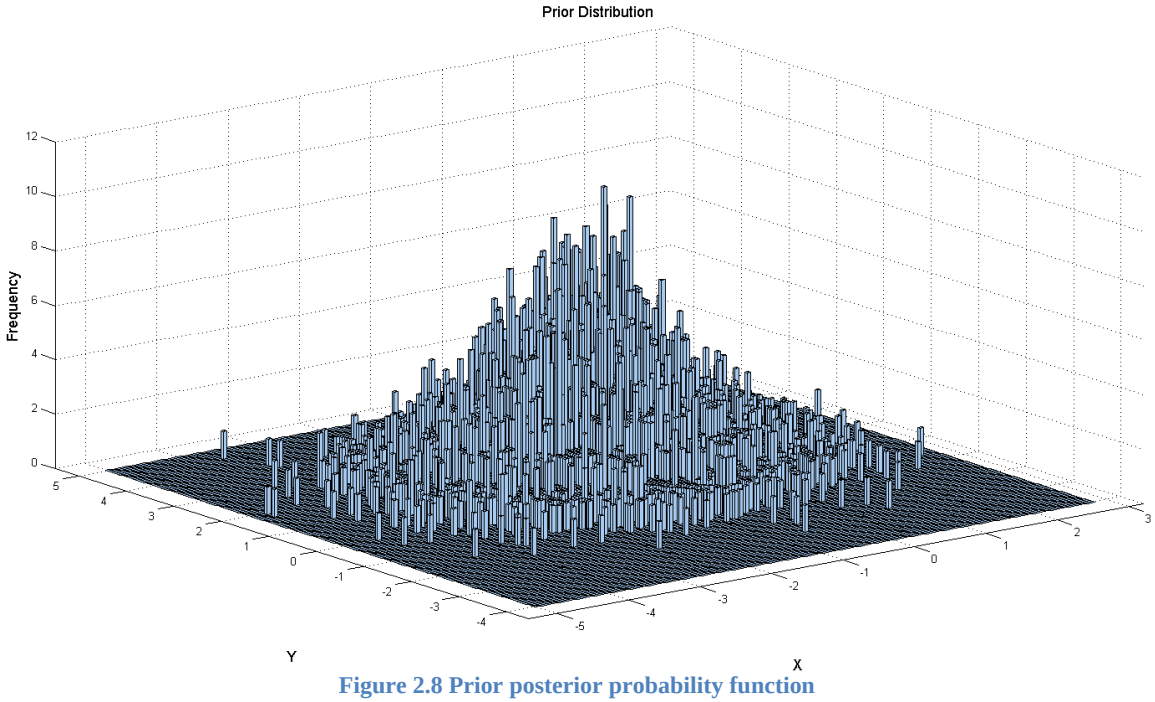


Figure 2.8 Prior posterior probability function

$$m_\infty = \begin{bmatrix} -1.9 \\ 1.4 \end{bmatrix}, C_m = \begin{bmatrix} 0.010 & 0.008 \\ 0.008 & 0.007 \end{bmatrix} \quad (45)$$

These two parameters characterise the posterior probability function, 10000 samples are selected from the probability function and are shown in figure 2.9. As it can be seen, using only two observations, the degree of confident regarding *m* dramatically is changed and it is getting very close to the actual value. The domain of 10000 samples are varied from -2.4 to -1.6 and 1.1 to 1.8 for x and y direction. These domains are much narrower than the initial domains.

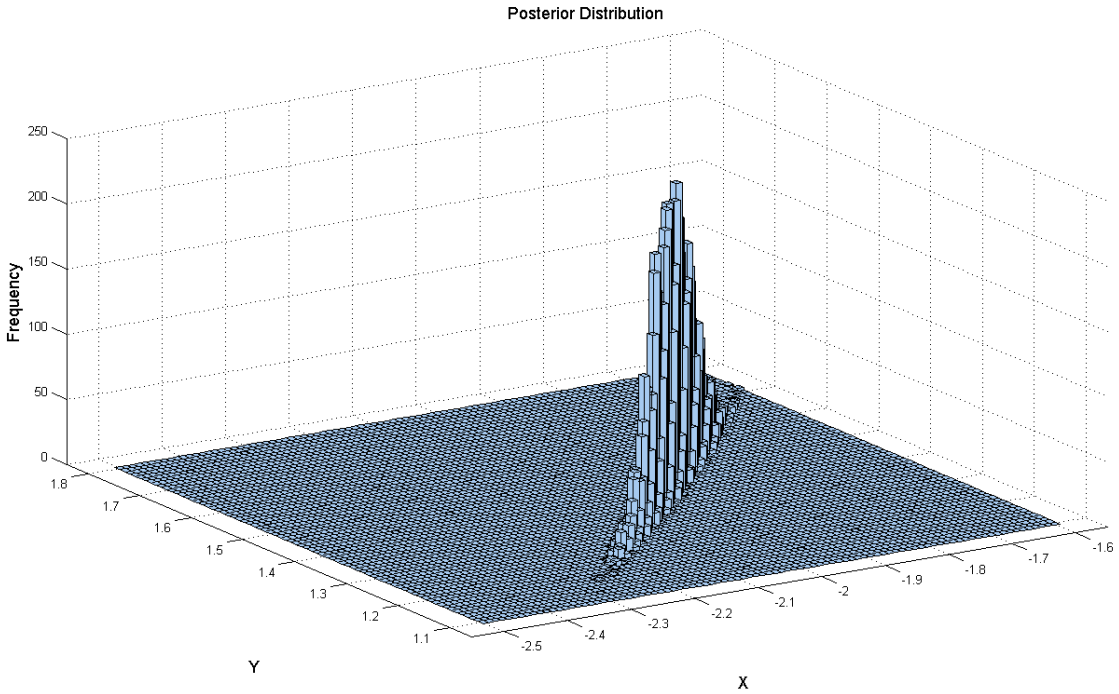


Figure 2.9 Posterior probability distribution

In nonlinear inverse problems, the posterior probability function will deviate from being Gaussian. The deviation is a function of the degree of problem nonlinearity, i.e., a higher nonlinearity causes more deviation from being Gaussian (Tarantola, 2005). These problems similar to nonlinear least squares problems are required to be solved iteratively. The objective function is defined as equation 39 and the minimum of S(m) is being sought. The MAP can be estimated using any optimisation method. In order to be able to select a set of models for uncertainty analysis purposes, the following assumptions can be made: i. the posterior probability is close to a Gaussian distribution, and ii. the posterior covariance matrix can be estimated using equation 46 in which $G_{m\infty}$ is the sensitivity matrix at m_∞ as equation 47 (Tarantola, 1987). One of the main advantages of having the posterior covariance matrix is that the square roots of the diagonal elements of $C_{m`}$ as uncertainty bars (Tarantola, 2005).

$$C_{m'} \approx C_m - C_m G^t{}_{m_\infty}\left(G_{m_\infty} C_m G^t{}_{m_\infty} + C_D\right)^{-1} G_{m_\infty} C_m \qquad (46)$$

$$G_{m_\infty} = \begin{bmatrix} \left(\frac{\partial g_1}{\partial m_1}\right)_{m_\infty} & \cdots & \left(\frac{\partial g_1}{\partial m_{N_m}}\right)_{m_\infty} \\ \vdots & \ddots & \vdots \\ \left(\frac{\partial g_{N_{obs}}}{\partial m_1}\right)_{m_\infty} & \cdots & \left(\frac{\partial g_{N_{obs}}}{\partial m_{N_m}}\right)_{m_\infty} \end{bmatrix} \quad (47)$$

In the following chapter, the advantages and drawbacks of the two famous objective function formulation methods are discussed. In the next section, Kalman filter is described. Using Kalman filter, having an optimisation step as a separate stage will be unnecessary.

2.4.4. Kalman Filter

The Bayesian approach (equation 48) can be converted to a sequential update (recursive), if the observations are measured at different time step (n). In history matching problems, observations are gathered sequentially. Thus, they can be fitted into a sequential Bayesian revision. Using equation 49, equation 48 can be switched to a sequential update in which in time step n, it is only the current probability distribution of model (n-1) and the new likelihood is required. The current distribution itself was updated by a similar combination in the previous timestep. Equation 50 to 52 shows the inductive derivation of the sequential update.

$$p(m|d_{obs}) \propto p(d_{obs}|m) \times p(m) \quad (48)$$

$$p\left(d_{obs_1}, \dots, d_{obs_n}|m\right) = p\left(d_{obs_1}|m\right) \times p\left(d_{obs_2}|d_{obs_2}, m\right) \dots p\left(d_{obs_n}|d_{obs_{n-1}}, m\right) \quad (49)$$

$$p_1(m) = p\left(m|d_{obs_1}\right) \propto p\left(d_{obs_1}|m\right) \times p(m) \quad (50)$$

$$p_2(m) = p\left(m|d_{obs_2}\right) \propto p\left(d_{obs_2}|d_{obs_1}, m\right) \times p_1(m) \quad (51)$$

.
.
.

$$p_n(m) = p\left(m|d_{obs_n}\right) \propto p\left(d_{obs_n}|d_{obs_{n-1}}, m\right) \times p_{n-1}(m) \quad (52)$$

The sequential update is the basis of Kalman filter and data assimilation methods (Welch and Bishop, 1995, Harvey, 1991).

In order to apply Kalman filter, two assumptions are taken into consideration, i- the system is linear or linearisable, and ii- modelling and measurement errors and variables have a Gaussian distribution. In this method, the posterior probability function in time step *i* is as equation 53 in which the posterior covariance matrix and mean are calculated from equation 54 and 55. *I* is an identity matrix and *K* is called Kalman gain matrix as equation 56. $\hat{H}$ is observation matrix (Hat matrix).

$$p_i(\mathrm{m}) \propto \exp\left(-\frac{1}{2}(\mathrm{m}-\mu_\mathrm{i})^\mathrm{t}{C_{\mathrm{m_i}}}^{-1}(\mathrm{m}-\mu_\mathrm{i})\right) \tag{53}$$

$$\mathrm{C_{m_i}} = \left(\mathrm{I}-\mathrm{K}\hat{\mathrm{H}}\right)\mathrm{C_{m_{i-1}}} \tag{54}$$

$$\mu_\mathrm{i} = \mu_{\mathrm{i}-1} + \mathrm{K}\left(\mathrm{d_{obs_i}} - \hat{\mathrm{H}}\mu_{\mathrm{i}-1}\right) \tag{55}$$

$$\mathrm{K} = \mathrm{C_{m_{i-1}}}\hat{\mathrm{H}}^\mathrm{t}\left(\hat{\mathrm{H}}\mathrm{C_{m_{i-1}}}\hat{\mathrm{H}}^\mathrm{t} + \mathrm{C_D}\right)^{-1} \tag{56}$$

When the number of variables is large, the computations of the matrixes will be very time-consuming. Ensemble Kalman filter (EnKF) is another approach in which the covariance matrixes are not required to be calculated, and they are substituted by sample covariance matrixes. This approach is more applicable in history matching. Since history matching is nonlinear, some modifications have been made. These modifications are explained as follows. Instead of representing the probability distributions by mean and covariance matrixes, *n* models called ensemble (*EnM*) are selected from the distribution (prior in initial step, posterior in updating stage), and then they are being updated in each time step (*i*) (equation 57). The ensemble includes the theoretical calculations. In this approach, it is required to have a same size ensemble of observations; thus, *n* observations are generated by a normal distribution from the observation (equation 58). d_{obs_i} is the actual observation and it is considered as mean of distribution and based on its covariance matrix, a set of observations are generated (equation 59).

$$EnM_i = \begin{bmatrix} m_{11_i} & \cdots & m_{n1_i} \\ \vdots & \ddots & \vdots \\ m_{1N_{m_i}} & \cdots & m_{nN_{m_i}} \\ d_{cal_{11_i}} & \cdots & d_{cal_{n1_i}} \\ \vdots & \ddots & \vdots \\ d_{cal_{1N_{obs_i}}} & \cdots & d_{cal_{nN_{obs_i}}} \end{bmatrix} \quad (57)$$

$$End_{obs_i} = \begin{bmatrix} d_{obs_{11_i}} & \cdots & d_{obs_{n1_i}} \\ \vdots & \ddots & \vdots \\ d_{obs_{1N_{obs_i}}} & \cdots & d_{obs_{nN_{obs_i}}} \end{bmatrix} \quad (58)$$

$$d_{obs_{j_i}} = \begin{bmatrix} d_{obs_{j1_i}} \\ \vdots \\ d_{obs_{jN_{obs_i}}} \end{bmatrix} = d_{obs_i} + N(0, C_{D_i}) \quad (59)$$

The following five equations are used to update the ensemble denoted by ${EnM_i}^u$. g is the flow simulation operator.

$${EnM_i}^u = {EnM_i}^p + K_i\left(End_{obs_i} - \hat{\mathrm{H}}_i {EnM_i}^p\right) \quad (60)$$

$${EnM_i}^p = g({EnM_{i-1}}^u) \quad (61)$$

$$\hat{\mathrm{H}}_i = [o\ I]_{(N_{obs})\times(N_{obs}+N_m)} \quad (62)$$

$$o = \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{bmatrix}_{(N_{obs})\times(N_m)}, I = \begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{bmatrix}_{(N_{obs})\times(N_{obs})} \quad (63)$$

$$K_i = {C_{EnM_i}}^p \hat{\mathrm{H}}_i^t \left(\hat{\mathrm{H}}_i {C_{EnM_i}}^p \hat{\mathrm{H}}_i^t + C_{D,i}\right)^{-1} \quad (64)$$

In this equation, ${C_{EnM_i}}^p$ is estimated by analysing the ensemble of models (equation 65).

$${C_{EnM_i}}^p = \frac{1}{n-1}\left({EnM_i}^p - \overline{{EnM_i}^p}\right)\left({EnM_i}^p - \overline{{EnM_i}^p}\right)^t \quad (65)$$

More details about EnKF can be found in the following review papers (Aanonsen et al., 2009, Oliver and Chen, 2010). The difficulties of Kalman filter method are discussed in the following chapter.

2.5. Optimisation

In the first step of history matching, uncertain variables (m) are defined. This step may include a reparameterisation method in which another set of parameters (r) is used as decision variables. In the second step, an objective function is formulated to distinguishing between models according to their qualities. The objective function is denoted by $S(m)$. After these two steps, the best model which satisfies the predefined conditions should be located. This step is called calibration, and finding the best model (global minimum) is a trial-error (iterative) procedure. In either of objective function formulations (probabilistic or deterministic), it is required to execute a reservoir simulation and an inverse reparameterisation calculation in each function evaluation, and hence, the fitness function evaluations are time-consuming. In order to approximate the best model properly and also expedite the procedure, an optimisation rule is required in which the global minimum of $S(m)$ (m_∞) is sought (equation 66). In this step, the inverse problems turn into an optimisation problem.

Optimisation methods provide a rule for updating the variable in all iteration. Without an optimisation rule, it is required to evaluate whole possible scenarios to find the global minimum of the objective function. Evaluating the whole possible scenarios even with a reasonably coarse discretisation (figure 2.10) of the variable space (M) is impractical, since there are a large number of variables. Thus far, many different optimisation algorithms have been developed to find the best model (the global optimum point). In this study, these methods are categorised into two groups: gradient-based and nongradient-based. In the gradient-based method, the first order and sometimes the second order (Hessian) information of $S(m)$ may be used to update m in each iteration. The nongradient-based methods are stochastic optimisers.

$$S(m_\infty) = \min_{m}\big(S(m)\big) \tag{66}$$

NOTE:
This figure/table/image has been removed to comply with copyright regulations.
It is included in the print copy of the thesis held by the University of Adelaide Library.

Figure 2.10 Discretisation of variable space (Wikipedia)

2.5.1. Gradient-based optimisation algorithms

Gradient-based algorithms have a long history in science and engineering. Hence in this study they are called classical methods as well. These methods are categorised into four classes: steepest descent methods, conjugate gradient methods, quasi-Newton, and variable metric methods (Tarantola, 1987).

2.5.1.1. Gradient descent method

Gradient (steepest) descent is a first-order optimisation method in which the variables are updated proportional to the negative of the gradient in each step, as equation 67. If the function is not derivatable, the gradient (equation 68) should be estimated numerically.

$$m_{n+1} = m_n - \mu_n \nabla S(\mathrm{m_n}) \tag{67}$$

$$\nabla S(\mathrm{m_n}) = \begin{bmatrix} \left(\frac{\partial S}{\partial m_1}\right)_{\mathrm{m_n}} \\ \vdots \\ \left(\frac{\partial S}{\partial m_{N_m}}\right)_{\mathrm{m_n}} \end{bmatrix} \tag{68}$$

μ_n is the step size which should be carefully assigned in a manner $S(m_{n+1})<S(m_n)$. Preconditioning can increase the speed of convergence for the problems with oscillations

by providing estimation for the step sizes. The starting points in these methods are usually assumed m_{prior}.

Tarantola suggested a preconditioned steepest descent for the probabilistic objective function formulation as follows (equation 69 to 74):

$$m_{n+1} = m_n - \mu_n \phi_n \tag{69}$$

$$\phi_n = \widehat{S_0}\, \gamma_n \tag{70}$$

$$\gamma_n = C_m G_n^t C_D^{-1} (g(m_n) - d_{obs}) + (m_n - m_{prior}) \tag{71}$$

$$\widehat{S_0} = (I + C_m G_0^t C_D^{-1} G_0^t)^{-1} \tag{72}$$

$$\mu_n \approx \frac{\gamma_n C_m^{-1} \phi_n}{\phi_n^t C_m^{-1} \phi_n + b_n^t C_D^{-1} b_n} \tag{73}$$

$$b_n = G_n \phi_n \tag{74}$$

2.5.1.2. Conjugate gradient descent optimisation algorithm

The conjugate gradient descent methods are also first-order optimisation methods. In these algorithms, the direction of changes is a combination of gradient descents of current step and previous steps (Fletcher, 1987), thus it is expected to have a quicker convergence. The following figure (figure 2.11) shows the convergence behaviour of a conjugate gradient descent (red line) versus a gradient descent method (green line) for a quadratic problem. As it can be seen, in a lower number of iterations, the conjugate gradient found a similar point as the gradient descent method (Wikipedia). In this algorithm also, the step sizes should be assigned properly.

A preconditioned conjugate for the probabilistic objective function formulation can be found in the following equations (equation 75 to 83) (Tarantola, 1987).

$$m_{n+1} = m_n - \mu_n \phi_n \tag{75}$$

$$\phi_n = \lambda_n + \alpha_n \phi_{n-1} \tag{76}$$

$$\lambda_n = \widehat{S_0}\,\gamma_n \tag{77}$$

$$\gamma_n = C_m G_n^t C_D^{-1}(g(m_n) - d_{obs}) + (m_n - m_{prior}) \tag{78}$$

$$\widehat{S_0} = (\mathrm{I} + C_m G_0^t C_D^{-1} G_0^t)^{-1} \tag{79}$$

$$\mu_n \approx \frac{\gamma_n C_m^{-1} \phi_\mathrm{n}}{\phi_\mathrm{n}^\mathrm{t} C_m^{-1} \phi_\mathrm{n} + \mathrm{b}_\mathrm{n}^\mathrm{t} C_D^{-1} \mathrm{b}_\mathrm{n}} \tag{80}$$

$$\mathrm{b}_\mathrm{n} = \mathrm{G}_\mathrm{n}\phi_\mathrm{n} \tag{81}$$

$$\alpha_n = \frac{\omega_n - \gamma_{n-1}^t C_m^{-1}\,\lambda_n}{\omega_{n-1}} \tag{82}$$

$$\omega_n = \gamma_n^t C_m^{-1}\,\lambda_n \tag{83}$$

Figure 2.11 Conjugate gradient descent versus gradient descent method (Wikipedia)

These preconditioned formulas (precondition steepest descent and preconditioned conjugate gradient) are achieved based on the Newton method. In Newton method, the step size is suggested to be as the inverse hessian matrix (equation 84). In high-dimensional problems, the inverse hessian matrix calculation is very computationally expensive. In the aforementioned methods, to speed up the process and also avoiding the hessian matrix computations, the step sizes are estimated by some other elements, such as S_0 and sensitivity matrix. The S_0 is assumed fixed and predefined.

$$m_{n+1} = m_n - \left(H(S(m_n))\right)^{-1} \nabla\left((S(m_n))\right) \tag{84}$$

The calculation of hessian matrix with a number of predefined terms may reduce the accuracy of approximation and also it reduces the speed of convergence. Thus, in the following methods the hessian matrixes are also estimated in each iteration, while some other terms are applied for the approximation and the *S* is not a predefined term.

2.5.1.3. Quasi-Newton

In this method the hessian matrix is approximated by equation 85. Thus, m_{n+1} will be updated as equation 86. This method is a specific form of the variable metric methods (Tarantola, 1987).

$$H_n \approx (I + C_m G_n^t C_D^{-1} G_n) \tag{85}$$

$$m_{n+1} = m_{prior} - \mu_n C_m G_n^t C_D^{-1} \left((I + G_n C_m G_n^t C_D^{-1})\right)^{-1} \dots$$

$$\dots \left((g(m_n) - d_{obs}) - G_n(m_n - m_{prior})\right) \tag{86}$$

In which $\mu_n \approx 1$. The Gauss-Newton method is a simular method to quasi-Newton method in which there is no prior information, i.e., $C_m \to \infty I$ (Tarantola, 1987). It should be mentioned that Gauss-Newton can also be used while the objective function has a prior term (Oliver and Chen, 2010).

2.5.1.4. Variable metric methods

In these methods, the *S* is not being fixed, and it is being updated in each iteration. In the beginning of the algorithms, S_n is as equation 87.

$$S_n \approx (I + C_m G_n^t C_D^{-1} G_n)^{-1} \tag{87}$$

The algorithm is as following equations (88 to 91):

$$m_{n+1} = m_n - \mu_n \phi_n \tag{88}$$

$$\phi_n = S_n \gamma_n \tag{89}$$

$$\gamma_n = C_m G_n^t C_D^{-1}(g(m_n) - d_{obs}) + (m_n - m_{prior}) \quad (90)$$

$$S_{n+1} = S_n + \delta S_n \quad (91)$$

Thus far, many different methods have been introduced for updating S_n, such as "rank one formula", "Davidon–Fletcher–Powell formula (DFP)", and "Broyden, Fletcher, Goldfarb and Shanno (BFGS)". The formula can be found in table 2.1.

Table 2.1 Hessian matrix estimation using different methods in the variable metric algorithm

Method	S_{n+1}	
Rank one formula	$S_n + \frac{u_n u_n^t C_m^{-1}}{u_n^t C_m^{-1} \delta\gamma_n}$	(92)
DFP	$S_n + \frac{\delta m_n \delta m_n^t C_m^{-1}}{\delta m_n^t C_m^{-1} \delta\gamma_n} - \frac{v_n v_n^t C_m^{-1}}{v_n^t C_m^{-1} \delta\gamma_n}$	(93)
BFGS	$\left(I - \frac{\delta m_n \delta\gamma_n^t C_m^{-1}}{\delta\gamma_n^t C_m^{-1} \delta m_n}\right) S_n \left(I - \frac{\delta m_n \delta\gamma_n^t C_m^{-1}}{\delta\gamma_n^t C_m^{-1} \delta m_n}\right) - \frac{\delta m_n \delta m_n^t C_m^{-1}}{\delta\gamma_n^t C_m^{-1} \delta m_n}$	(94)

δm_n, $\delta\gamma_n$, v_n, u_n are as follows:

$$\delta m_n = m_n - m_{n-1} \quad (95)$$

$$\delta\gamma_n = \gamma_n - \gamma_{n-1} \quad (96)$$

$$v_n = S_n \delta\gamma_n \quad (97)$$

$$u_n = \delta m_n - v_n \quad (98)$$

Among these three methods, BFGS is being used more than the others, as it approximates the S_n with a reasonable accuracy (Tarantola, 1987).

The Levenberg-Marquette algorithm is a method between gradient descent methods and the Gauss-Newton method. In the Gauss-Newton method, the hessian matrix is replaced by $(G_n^t G_n)^{-1}$, thus, modifications in m are as equation 99. The Levenberg-Marquette algorithm has a term which can provide an interpolation between Gauss-Newton and descent method, as equation 100. ζ is damping coefficient, and can speed up the process of

optimisation; a small value for it switches the algorithm to a Gauss-Newton and a large value switches the algorithm to a descent method.

$$m_{n+1} = m_n + (G_n^t G_n)^{-1} G_n^t (g(m_n) - d_{obs}) \quad (99)$$

$$m_{n+1} = m_n + ((G_n^t G_n)^{-1} + \zeta \times diag(G_n^t G_n)) G_n^t (g(m_n) - d_{obs}) \quad (100)$$

When the number of variables is huge, the BFGS algorithm needs a huge amount of memory. The limited-memory BFGS is an alternative approach in which to approximate the inverse hessian matrix, instead of storing a large matrix, only a few vectors are kept to represent the matrix.

2.5.2. Non-gradient optimisation algorithms

Another group of optimisation algorithms is stochastic methods. In these methods, search spaces are being visited randomly to escape from local minima. A local minimum is a point (m°) in which the gradient of objective function is zero, or in other word, there is an ε>0 in which for *m* in $|m-m°|<\varepsilon$, $S(m°) \leq S(m)$. Figure 2.12 shows the difference of a local and a global optimum point. The point around 1 is a local minimum, while the global minimum is an *m* around 0.3.

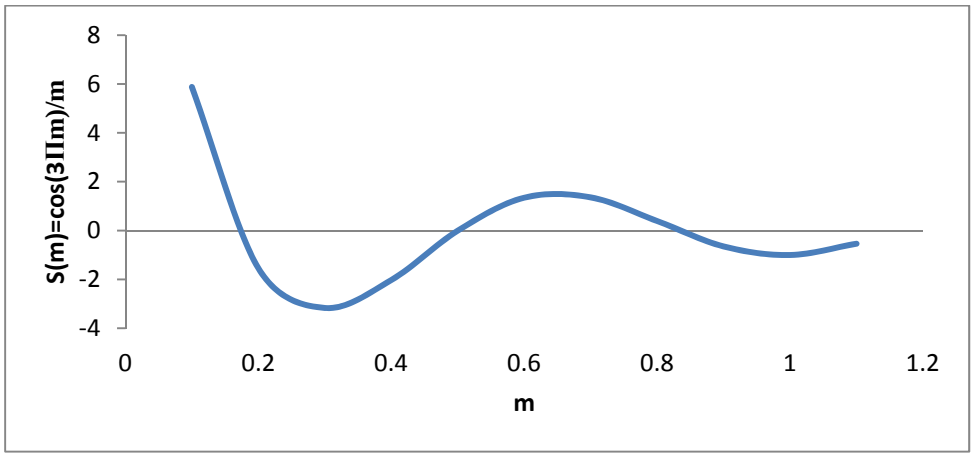


Figure 2.12 Local minimum versus global minimum

The nongradient algorithms use a stochastic search, and usually they do not need gradient calculations (derivative-free). They are iterative methods, and in each iteration, it is sought

to improve the candidate solution. These algorithms are mostly black-box optimisers and they do not need information concerning the problems. It should be mentioned that however, they have a higher chance to approximate the global optimum, there is no guarantee for it. Thus far, different stochastic optimisation algorithms have been developed to perform optimisation efficiently in terms of computation and also the accuracy of global optimum approximation.

In random search methods, first, a position (m) is selected randomly in the corresponding space, then, it is evaluated based on a fitness function (S(m)), after that, the main loop of the algorithm begins, and until the predefined stopping criteria are not met, it will be remained in the loop. In this loop, in each iteration, a new sample model is selected according to a specific stochastic rule and then, the new sample is evaluated. If the fitness value of the new model is lower than the current one ($S(m_{n+1})<S(m_n)$), the new model will be usually stored, otherwise the current model will be kept. Once the loop is terminated, the achieved model will be selected as the optimum point. This procedure is the basis of many stochastic optimisers such as Simulated Annealing (SA) (van Laarhoven and Aarts, 1987), Tabu Search (Glover, 1989, Glover, 1990), and Stochastic Gradient Descent (Gardner, 1984). In SA algorithms, in each iteration, a neighboured of m_n is checked out, and probabilistically the algorithm decides between m_n and m_{n+1}. The pseudo-code is as follows,

```
m=m0;
Fit=S(m);
i=1;
While i<StoppingCriteria
        T=tempretureupdating(i)
        mnew=neighbour(m);
        Fitnew=S(mnew)
        if Probabilityrule(m, mnew, T)> rand
                m=mnew; Fit=Fitnew;
        end
end
```

In the Tabu search algorithm, jumps are added to the algorithm to search not only the neighbours of current position, but also search other locations of space, in order to escape from trapping in local minima. The Stochastic Gradient Descent algorithm is similar to steepest descent algorithms while the gradients are not calculated entirely and they are estimated based on analysing some samples.

Improving a single solution is not a proper way of finding the global point in a nonquadratic search space even with a random search; sometimes worse solutions can redirect the algorithm to the global optimum point. Therefore, population-algorithms were established.

In these algorithms, instead of modifying a single solution, a set of solutions known as population is being modified in each iteration known as generation. The population not only includes the best ever found solution, but also within has some other solutions which are used along with the best solution to construct the new populations. They can be categorised into two famous groups: Evolutionary Algorithms (EAs) and Swarm Intelligence algorithms (SI). In EAs, the next population is constructed by only the individuals of the current population, and the individuals of the constructed population do not have interactions with each other, while in SI algorithms, the individuals have interactions with each other, and some of the individuals of a population are constructed based on a number of individual from the same population. The most famous EAs are Evolution Strategy (ES) and Genetic Algorithm, and the most famous SI algorithms are Ant Colony, Artificial Bee Colony and Particle Swarm Optimisation (PSO).

In the next sections, the two algorithms used in this study (Genetic Algorithm and Artificial Bee Colony) are explained (they are described in more details in the following chapters).

2.5.2.1. Genetic Algorithm

There are two types of GA: binary and real-coded (Haupt and Haupt, 2004). The real-coded GAs are more utilised, due to its quicker convergence ability. The GAs consist of a number of steps. The algorithm begins with a generation of a set (known as population) of candidate solutions (known as chromosomes) using an initialisation procedure. The population size and the chromosome size are defined. Then, using one of the two initialisation procedures, random and heuristic, an initial population is generated. In the random initialisation, the initial population is generated totally random via a uniform distribution, while in the heuristic initialisation, background information is used to generate the population. After the initialisation, all the chromosomes are evaluated based on a fitness function (objective function). After generating the initial chromosomes and evaluating them, the main loop of the algorithm starts. In each step (known as generation) of the main loop, a sub-set of the current population is selected using a selection operator. Usually, the selection procedure is conducted according to the fitness of chromosomes. Two operators called crossover and mutation are applied to the selected set of chromosomes and a new set of chromosomes is produced. The crossover and mutation operators are applied to the chromosomes with probability p_c and p_m, respectively. The new set generated by the selection, crossover and mutation is recombined with the current population to provide the new population. To keep the best ever found solution, the chromosome (known as elite) is migrated directly to the new population. The main loop is repeated until predefined stopping criteria are met.

There are many types of selection, crossover, and mutation operators, and each of them are different based on the type of the search space and functionality (Gwiazda, 2007). These operators carry out important roles in GAs. In order to deliver an algorithm which finds the global optimum points of different kinds of search space in a reasonable computation cost, a balance should exist between exploration and exploitation ability of the algorithm.

Explorations provide a low resolution search and exploitations provide a high resolution task. Another term which has a significant effect on the GA performance is the diversity of population. If the chromosomes of a population do not have enough diversity, it will be likely that the algorithm converges to a false optimum point or an arbitrary point. The balance and diversity should be provided by selecting appropriate operators. Thus far, many different operators have been designed. Gwiazda in his books (2007) gather different operators and describe the effectiveness and drawbacks of each approach. The most famous selection operators are 1- tournament, 2- roulette wheel, and 3- stochastic uniform. The most famous crossover operators are: 1- scattered, 2- heuristic, 3- interpolation, 4- single point, and 5- two points. The most famous mutation operators are: 1- uniform, and 2- Gaussian noise. Evolutionary Strategy algorithms are very similar to the GAs while they consist of only mutation and selection.

2.5.2.2. Artificial Bee Colony

Using the inspiration of honeybee behaviour, an algorithm for optimisation was developed by Karaboga in 2005. The colony of ABC comprises employed bees, onlookers and scouts, just like real honeybees. In the first attempt, employed bees search for food randomly and then memorise the locations. In the next attempt, an employed bee looks for a food source based on her knowledge. This means employed bees always looked for better locations around the previous one. They also share their information about food source profitability and their locations with onlooker bees. Onlookers collect information from all of the employed bees and then make decisions based on the observations. If the nectar amount increases, the number of onlookers looking for it will increase (Karaboga, 2005). This means that location(s) with the highest profitability is exploited by more onlooker bees. Scouts search randomly and usually they look for a new home or food source. An employed bee will become a scout bee, whenever they cannot extract more food from its

corresponding area (i.e. they cannot find a better location with a higher nectar) after a number of attempts.

2.5.2.3. Multi-objective optimisation

Another group of optimisation algorithm are multi-objective population-based algorithms. In multi-objective optimisation problems, there are a number of objective functions $f(x) = [f_1(x), f_2(x), \dots, f_n(x)]$ which are required to be optimised simultaneously. *x* is a *k* dimensional variable vector $x = [x_1, x_2, \dots, x_k]$. Usually, it is impracticable to find a single point which optimises all objectives together (Coello, 2000, Konak et al., 2006). A multi-objective problem can be switched to a single-objective problem, if proper weighing factors between the objectives can be determined. Usually, the point which minimises both objective functions together is out of the feasible region. In these problems, a set of solutions called noninferior or Pareto optimal solutions is introduced instead of a single point. Pareto front is located on the bottom border of feasible solutions (Haupt and Haupt, 2004) and it consists of nondominated solutions. In Pareto front, all the solutions are equally acceptable, unless supplementary information is available.

Pareto optimal is introduced by Vilferdo Pareto in nineteen century. Pareto optimal, $x^* = [x_1^*, x_2^*, \dots, x_k^*]$ means that for every x, either

$$\bigwedge_{i \in I} (f_i(x) = f_i(x^*)) \tag{101}$$

Or at least there is one i which (Coello, 2000)

$$f_i(x) > f_i(x^*) \tag{102}$$

Finding the whole optimal set is almost unachievable, since Pareto front usually consists of an infinite number of solutions (Coello et al., 2007). Thus, the target, in the multi-objective optimisation problems, is finding a set of solutions which is satisfactorily close to the

Pareto optimal set (Konak et al., 2006). After locating the set of solutions, it is possible to take out a number of solutions from the set based on post-optimisation trade-offs and keep the best solutions (Haupt and Haupt, 2004). In order to find the Pareto optimal solutions, a multi-objective optimisation is required. Population-based algorithms have a considerable robust benefit in comparison with the traditional multi-objective optimisation techniques (Coello et al., 2007). The main advantage of population-based algorithms, when they are applied to solve multi-objective optimisation problems, is the fact that they typically optimise a set of solutions. Thus, they allow the approximation of the entire Pareto front in a single algorithm run.

2.5.2.4. Comparison of the optimisation algorithms

Three optimisation methods are compared in here for a number of benchmarking functions shown in table 2.2. The benchmarking functions are a combination of simple, unimodal and convex functions to complex and multi-modal functions. This combination of benchmarking functions is selected to investigate the performance of a same algorithm on different shape of landscapes. An algorithm is successful (according to history matching requirement) which is not affected by the shape of landscape of function and also is computationally efficient. The 2-dimensional benchmarking functions are shown in the following figures. The selected optimisation algorithms are as follows: 1- BFGS as a gradient-based optimiser, 2- SA as a random search, and 3- GA as a population-based algorithm. Each of these algorithms is carried out by MATLAB (MathWork, 2011b).

The maximum of the function evaluations is set to 40,000. For the BFGS, it is required to identify an initial point; for all of them a point equal to $[1 \dots 1]_{1\times10}$ is used. For the GA, the default penalty term was set to zero to deliver a fair comparison and also the crossover and mutation operator used are heuristic (ratio 1.2) and uniform (rate 0.11).

Table 2.2 Mathematical benchmarking functions

Function	Mathematical formula (f(x))	Bounds	Global optimum	No. D
Sphere	$f(x) = \sum_{i=1}^{n} x_i^2$	-5.12 ≤ x_i ≤ 5.12	$f(x) = 0$ $for\ x_i = 0, i = 1 \dots n$	n=10
Rastrigin	$f(x) = 10n + \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i))$	-5.12 ≤ x_i ≤ 5.12	$f(x) = 0$ $for\ x_i = 0, i = 1 \dots n$	n=10
Schwefel	$f(x) = \sum_{i=1}^{n} (-x_i \sin(\sqrt{\lvert x_i \rvert}))$	-500 ≤ x_i ≤ 500	$f(x) = -418.9829n$ $for\ x_i = 420.9687, i = 1 \dots n$	n=10
Griewank	$f(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos(x_i/\sqrt{i}) + 1$	-600 ≤ x_i ≤ 600	$f(x) = 0$ $for\ x_i = 0, i = 1 \dots n$	n=10
Ackley	$f(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)) + 20 + e$	-32.76 ≤ x_i ≤ 32.76	$f(x) = 0$ $for\ x_i = 0, i = 1 \dots n$	n=10

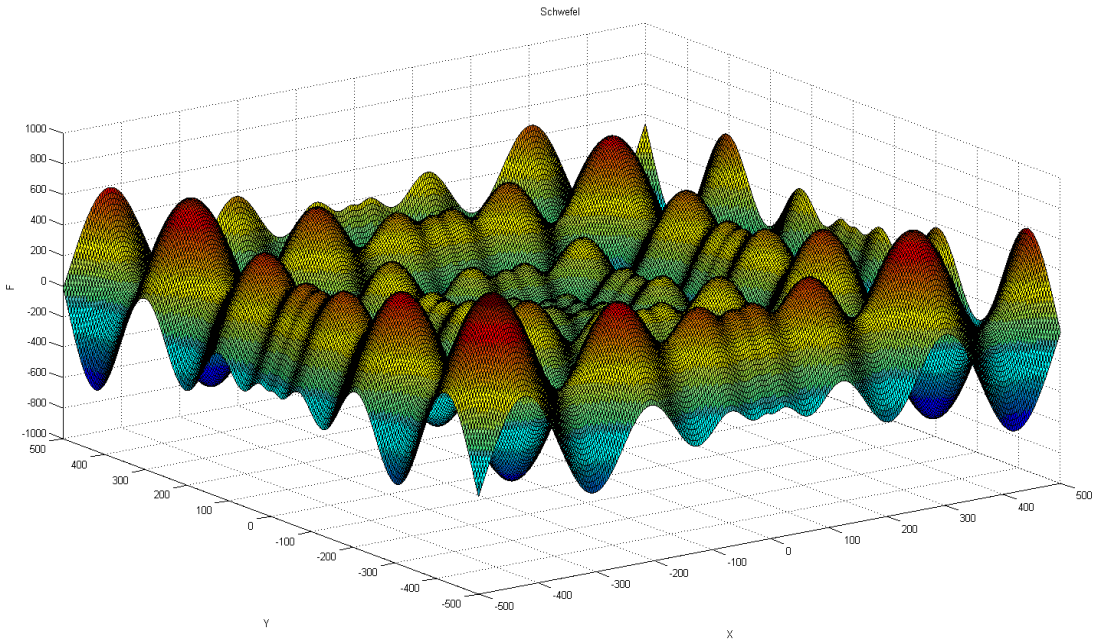


Figure 2.13 2D Schwefel function

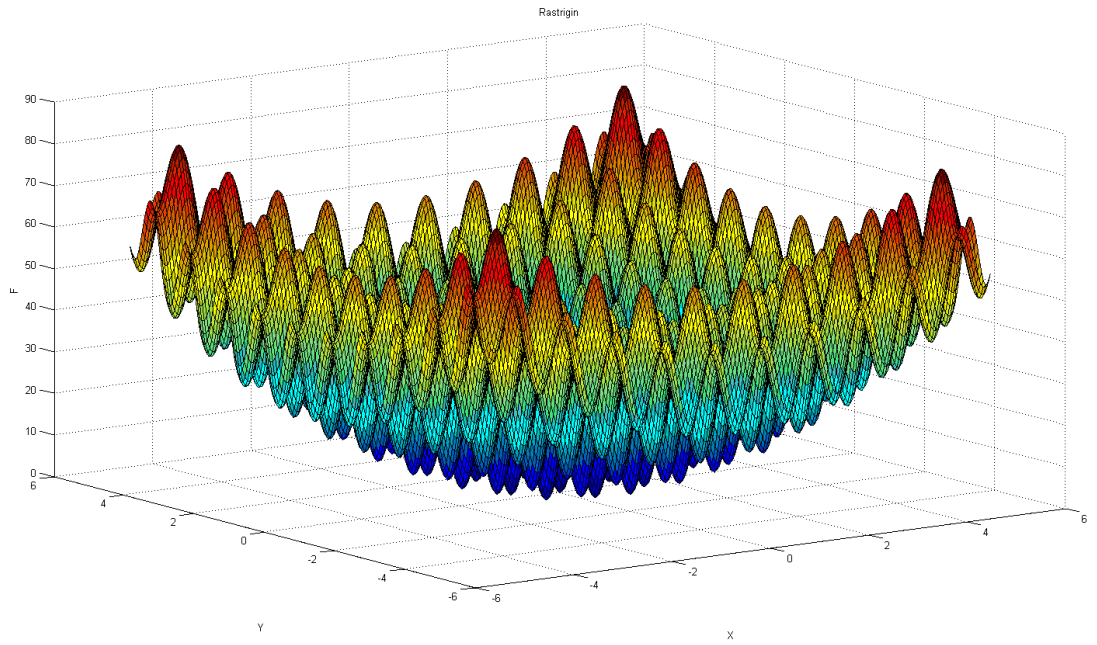


Figure 2.14 2D Rastrigin function

In order to assess the outcomes of the algorithms, two criteria are used: fitness value at achieved optimum point and distance from the real global optimum. Each method which can find a lower value for these two elements is called a better algorithm in comparison with the other two algorithms. The other element which should be taken into consideration for the assessment is the number of function evaluations. All these elements are shown in table 2.3.

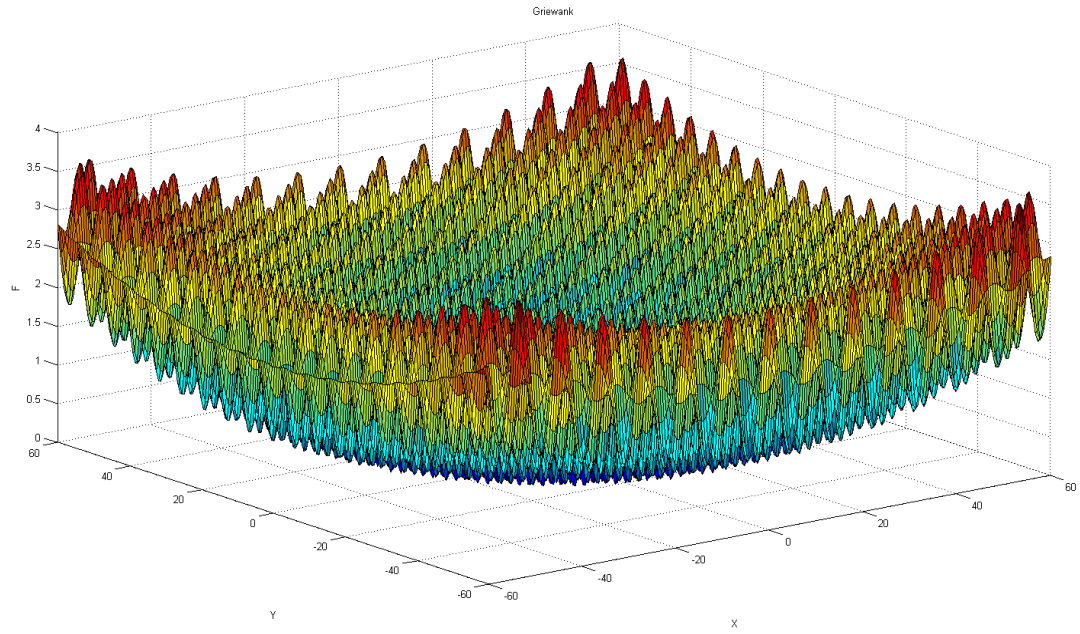


Figure 2.15 2D Griewank function

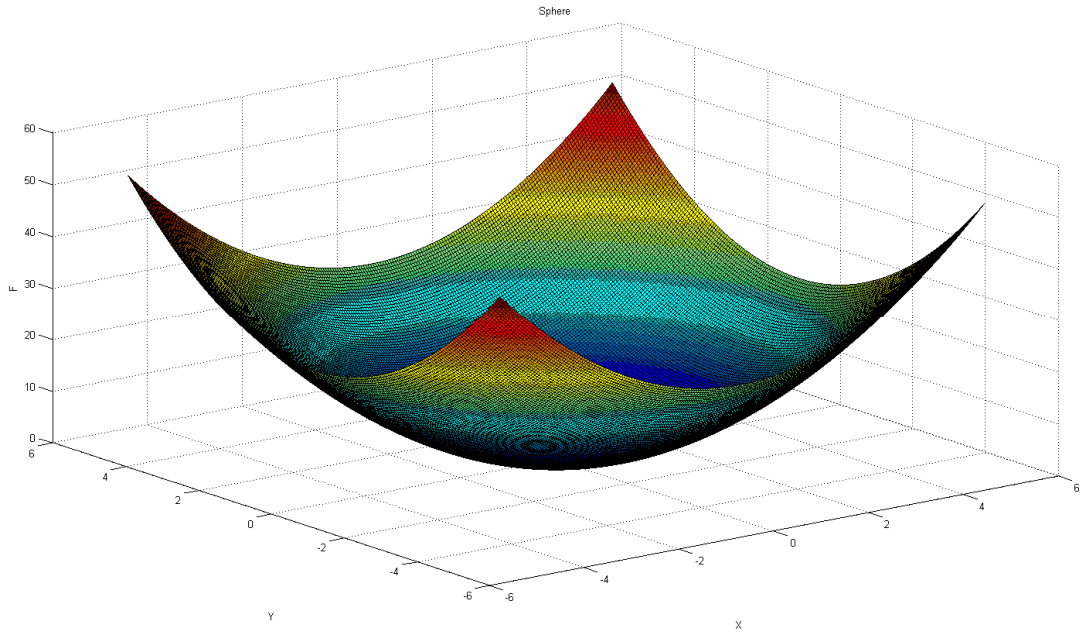


Figure 2.16 2D Sphere function

By comparing the results, a significant difference between the outcomes of the applied algorithms are observed. The first observation is that the SA could not deliver good results even for the convex function (Sphere) with 40,000 function evaluations. The second observation is that the BFGS almost found the global optimum point for Sphere, Griewank and Rastrigin, but it trapped into a local minimum for Ackley and Schwefel function. The third observation is that the GA approximated the global optimum point properly for all these five functions. The only matter of the GA is its computational time. For Sphere, Griewank and Rastrigin, the GA could find the global optimum points with 40,020 evaluations, while the BFGS approximate them with 44, 352, and 154 evaluations. The results show a significant difference for the computational time.

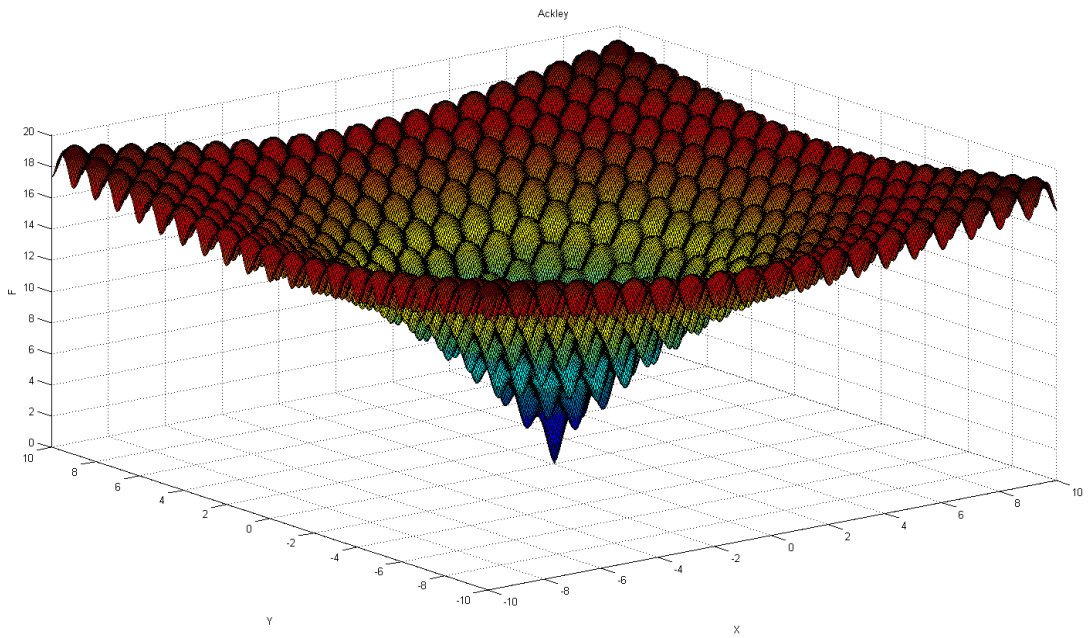


Figure 2.17 2D Ackley function

Table 2.3 Comparison of three optimisation algorithm for the benchmarking functions

Algorithm	BFGS			SA			GA		
	Fitness	Distance	No Evaluations	Fitness	Distance	No Evaluations	Fitness	Distance	No Evaluations
Sphere	0	0	44	0.31	0.31	40001	0.007	0.007	40020
Griewank	1.9e-11	1.9e-11	352	10	10	40001	4e-11	4e-11	40020
Rastrigin	0	0	154	10	10	40001	4e-14	4e-14	40020
Ackley	9.37	9.37	143	10	10	40001	1e-13	1e-13	40020
Schwefel	-39	1.7e6	88	-1024	2e6	40001	-4187	18	40020

In order to investigate whether the BFGS is capable of approximating the global optimum points of Ackley function with different initial points or not, several random points are selected, and the optimisation is carried out using them. The total evaluations should not be more than 40,000. The best optimum point which is achieved using BFGS has a fitness value equal to 18 which is far from the actual point. The GA, as a sample of population-based optimisers, showed that the shape of landscape does not have an impact on its performance and it is capable of approximating the function; the only concern is the computational costs.

Another drawback of population-based algorithms is that their performance in high-dimensional search space. For the 1000-Dimensional Sphere function, the best point found by the GA, in 50,000 function calls, has a fitness value equal to 320, while the BFGS algorithm found the global (fitness equal to zero) in 4000 evaluation calls.

The advantageous of disadvantageous along with the applications of optimisation algorithms in history matching are reviewed in the following chapter.

2.5.3. Fitness approximation

Fitness approximation, known as response surface model, meta-model, surrogate-model and proxy-modelling, is a method for overcoming the computational cost concern of time-consuming functions in optimisation problems (Jin, 2005, Cullick et al., 2006). In fitness approximation methods, the original function (OF) which is computationally expensive to be evaluated is substituted by an approximation function (AF) called proxy. This substitution expedites the process of optimisation. It is a challenging task, since a multi dimensional fitness landscape should be carefully modelled by the applied AF.

These approaches consist of the following main steps: 1- a set of samples are selected from the search domain (M) using an experimental design method, and the set of samples are

evaluated by the OF (S(m)), 2- then, a proxy model is selected and trained by the set of samples with their corresponding fitness values, and then it is validated by another set of samples, 3- afterward, the global optimum of the proxy model (AF) (the new space) is looked for by an optimisation algorithm, 4- the achieved solution, after meeting the stopping criterion, is used as the optimum point (history matched model). In this approach, the OF is utilised only in the initial stage for evaluating the samples, and it is not being used during the procedure of optimisation.

The proxy model can be different mathematical equations, such as the artificial neural network (ANN), kriging, polynomial and support vector machine. Kriging is similar to the geostatistical Kriging (see the reparameterisation section), the difference is that the space which is interpolated in this technique is usually more than two dimensional, unlike the pilot point reparameterisation in which the corresponding space is two dimensional. The artificial neural network (ANN) relates the input signals to output signals using a series of transfer functions (mathematical formulas). This method is inspired from the biological neural networks. In the problems in which the modelling the phenomena is difficult or impossible, the ANN will be a superior options. The ANN, by analysing the input and output, provides a network which can be used for the estimation. In the analyses procedure, the number of hidden layers (the form of network) and the number of neurones in each layer are fixed and a set of observations is used to train a neural network (estimate the unknown weighting factors) by backpropagation (figure 2.18). One of the most famous neural network is the feed-forward in which the output does not have interactions (cycle) with the input signals.

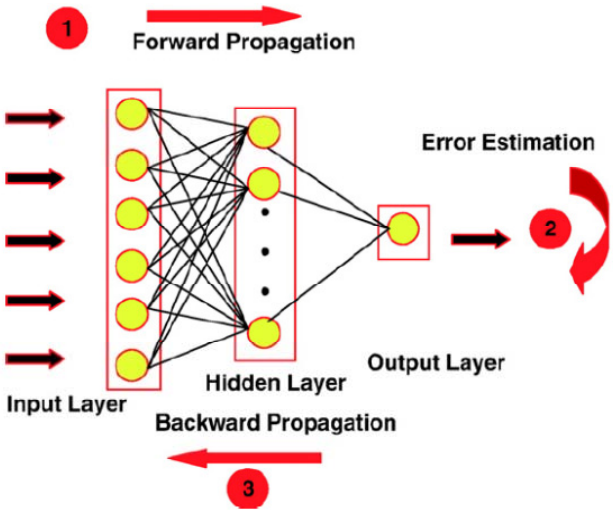


Figure 2.18 A multilayered network with a hidden layer with a number of neurones and a number of inputs and one output (Saemi et al., 2007)

A multilayer network with one layer input layer, and three hidden layers which has a one output neuron (S) has an equation as follows:

$$S(m) = \sum_{p=1}^{P} v_p f\left(\sum_{q=1}^{Q} {w_{qp}}^3 f\left(\sum_{r=1}^{R} {w_{rq}}^2 f\left(\sum_{i=1}^{N_m} {w_{ir}}^1 m_i\right)\right)\right) \quad (103)$$

In which f (activation function) usually is

$$f(x) = \frac{1}{1+e^{-ax}} \quad (104)$$

P, Q and R are the number of hidden nodes (neurones). The advantages and disadvantages of proxy-modelling techniques in history matching are discussed in the next chapter.

Chapter 3

3. Literature review

Central to this chapter is the background works in history matching. This episode presents the recent progress in different steps of history matching.

3.1. History matching

History matching is attached to reservoir modelling and simulation from almost five decades ago; the following studies are a number of first papers regarding this subject, (Jacquard, 1965, Thomas et al., 1972, Sheldon et al., 1960). Thomas et al. in 1972 developed one of the first generations of automatic history matching framework in which the Gauss-Newton and least squares are implemented as the optimisation and objective function formulation method. In their problems, there were only few gridblocks and the reservoirs were single-phase. Figure 3.1 shows the number of SPE conference and journal papers about history matching (the figure is taken from Oliver and Chen (2010)). The number of published papers regarding history matching is an evidence for its complexity and importance. By advances in computers, especially hardware and also reservoir simulation, the research efforts on this subject have been escalated.

In the following sections, the progress in each section is described individually.

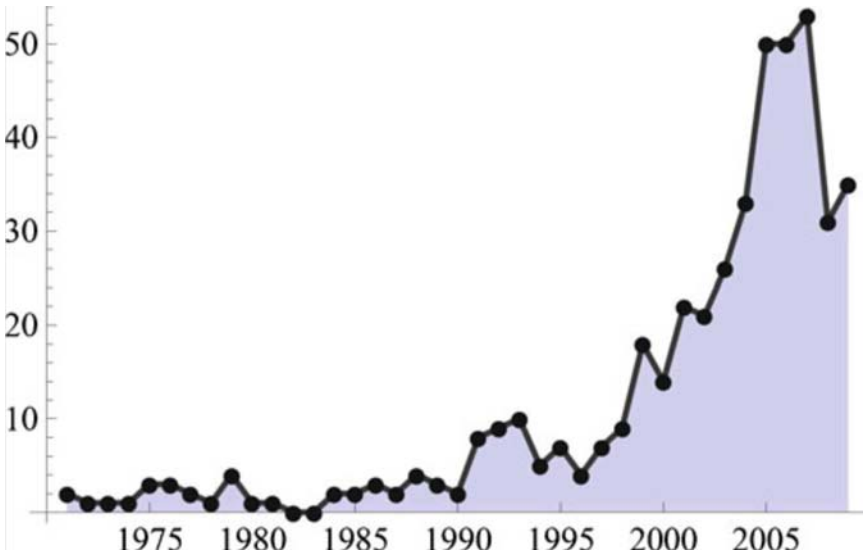


Figure 3.1 Published SPE paper concerning history matching subject (Oliver and Chen 2010)

3.2. Parameterisation and reparameterisation

The uncertain parameters are normally dissimilar for each case study, for instance, in the case study of Romero and Carter 2001, the variables are porosity, shale fraction, well skins and fault trasmissibilities, in the case study of Li et al., 2009, the variables are the relative permeability curves, in the case studies of Sarma et al., 2007, the variables are the permeabilities. The differences between the parameterisation cause to generate dissimilar problems. When the spatial properties are uncertain, a reparameterisation method is applied (Oliver and Chen, 2010).

The reparameterisation technique should be able to map a much smaller space (R) to the variables space (M) space with the enough accuracy. Thus far, several techniques are introduced, such as zonation (Gavalas et al., 1976, Jacquard, 1965, Shah et al., 1978, Grimstad et al., 2003), pilot points (Gavalas et al., 1976, De Marsily et al., 1984, Bissell et al., 1997, Romero and Carter, 2001, Doherty, 2003), spectral decomposition of prior covariance matrix (Reynolds et al., 1996, Sarma et al., 2006, Zhang et al., 2007), bicubic spline (Leo et al., 1986) wavelet transform (Lu and Horne, 2000, Sahni and Horne, 2005, Grimstad et al., 2001), and discrete cosine transform (Jafarpour and Mclaughlin, 2007, Jafarpour and McLaughlin, 2009), also a number of reparameterisation methods which use

the eigenvalues and eigenvectors of data sensitivity (Oliver and Chen, 2010); they are applied by Shah et al., 1978 and Rodrigues, 2006.

The zonation technique is the oldest reparameterisation method. In this method, the reservoir is divided into a number of coarse zones. It was applied by Jacquard in 1965, for the first time, and it has been used by other reservoir engineers several times (CHAVENT et al., 1975, Gavalas et al., 1976, Shah et al., 1978, Grimstad et al., 2001). However the reduction in the number of variables using this method increases the speed of history matching procedure dramatically, usually the final misfit is very large (Oliver and Chen, 2010). In addition, it is very important to find the optimum number of zones, before starting history matching; otherwise a high reparameterisation error will be imposed. The number can vary from 1 to N_m. Whole reservoir model can be represented by a single zone, if it is totally homogenous. In a very heterogeneous media, coarse zones cause a high reparameterisation error. To find the optimum number of zones, prior information about heterogeneity of the reservoir is needed which is not usually available. Excluding the mentioned disadvantages, this approach also suffers from several other drawbacks which are summarised in table 3.1 along with its advantages (Oliver and Chen, 2010).

Table 3.1 Advantages and disadvantages of zonation reparameterisation

Advantages	Disadvantages
Quick reduction in misfit in history matching processes	Difficulty in finding the optimal number of zones
Simple mathematics	In heterogeneous media, it is not efficient.
Carrying the main features of model	Low-resolution models
Fitting properly in a Bayesian framework	It causes discontinuities.

Pilot point reparameterisation was introduced by de Marsily et al. in 1984 for groundwater problems, and it was used for the first time in hydrocarbon problems in 1990's (Ouenes et al., 1993, Bissell et al., 1997). In this method, a number of gridblocks only are being tuned by history matching and the rest of the gridblocks are estimated by geostatistical interpolations. The pilot point method can deliver high-resolution models and also geostatistical correlations can be kept by this technique. The main difficulty of this

approach is in finding the proper location and number of pilot points. The locations of pilot points are usually fixed before carrying out any history matching. de Marsily disturbed them uniformly, Bissell et al. (1997) and LaVenue and Pickens (1992) suggested sensitivity analysis to find the best locations for pilot points, and Wen et al. 1998 located them randomly in which locations were changed in each iteration. To overcome the issue of finding the most appropriate locations for pilot point, in some studies, locations are considered as variables, and hence in these studies it was sought to find the best history matched models by changing both the value and location of pilot points (Wen et al., 2006, Romero et al., 2000, Romero and Carter, 2001, Ballester and Carter, 2007). The other issues of the pilot point reparameterisation are as follows: i. instability may occur in the solutions, and ii. extreme values might be seen in the results. These issues can be prevailed over by adding a regularisation term to the objective function (Oliver and Chen, 2010, McLaughlin and Townley, 1996).

In the pilot point technique, the variogram is the basis for estimation of the property across the model (Hohn, 1999, Olea, 1999). Therefore, a variogram model is required that accordingly needs a prior data (an inappropriate variogram may lead to incorrect history matching model) which is not always available, but one of the advantages of this approach is that the variogram parameters, including range, nugget and contribution, can be assumed unknown and considered as variables (Romero and Carter, 2001). By this consideration, having prior data for reparameterisation using this method becomes less important. It should be mentioned that the issue of finding the optimal number of variables still exists. It was suggested to increase the number of pilot points, when the misfit value could not be reduced by the current number of pilot points (LaVenue and Pickens, 1992) which is very time-consuming.

The pilot point technique, up to now, has not been used along with a Bayesian objective function formulation with an included prior term, since it is not feasible to provide prior

information regarding the best location and value of the pilot points (different combination of location and value may result in similar results). Table 3.2 summarises the main advantage and disadvantages of the pilot point reparameterisation method.

Table 3.2 Advantages and disadvantages of pilot point reparameterisation

Advantages	Disadvantages
High-resolution models	Difficulty in finding the optimal number of pilot points
It works properly in heterogeneous media	Computationally expensive
Carrying the main features of model	Not fitting properly in a Bayesian framework
Having prior data is not a must	-
Keeps the geostatistical correlations	-

The spectral decomposition of prior covariance matrix is another method for the reparameterisation, it was introduced by Reynolds et al., 1996. In this method, models are reparameterised based on eigenvalues and eigenvectors of prior covariance matrix. The main drawback of this method is that it is sensitive to the prior information, and if an accurate prior covariance matrix is not available, it may mislead the optimisation to a wrong optimum point and accordingly result into an incorrect history matched model. This method also is computationally expensive, when the dimension of m is large. Table 3.3 summarises the main advantages and disadvantages of this method.

Table 3.3 Advantages and disadvantages of spectral decomposition reparameterisation

Advantages	Disadvantages
High-resolution models	Difficulty in finding the optimal number of variables of p
It works properly in heterogeneous media	Computationally expensive
Carrying the main features of model	Needs high-quality prior information
Fitting properly in a Bayesian framework	-

The spline-based reparameterisation method was applied in history matching by Lee and Seinfeld (1987) and Makhlouf et al. (1993b). In this method, the properties which are a function of location are represented by a bicubic spline approximation. It is a kind of interpolation for 2D properties. It is similar to the pilot point technique in which unlike geostatistical correlations, a spline interpolation is used to estimate the properties between

the gridblocks. In this method, the number of coefficients should be determined before carrying out a reparameterisation. The fundamentals of spline can be found in (De Boor, 1978). This technique usually smooths out the distribution (Oliver and Chen, 2010) and also suffers from similar drawbacks to the pilot point approach (determining the proper number of gridblocks and their locations).

Another reparameterisation technique is discrete cosine transform introduced by Jafarpour and McLaughlin in (2007) for history matching problems and was used in several other studies (Jafarpour and McLaughlin, 2009, Li and Jafarpour, 2010, Jafarpour et al., 2010). A discrete cosine transform (DCT) shows a sequence of data in a summation of cosine functions. It is similar to Fourier transform while only it has cosine terms. It is widely applied in different fields, such as audio and image compression (MP3 and JPEG are the most famous example for this compression method). In the compression, the image (data) is transformed to this domain, and the corresponding coefficients are calculated and ranked. Then, the small high-frequency components are truncated, and the rest of coefficients are used for representing the model. Usually, the reparameterised models have acceptable accuracy. But, this method is used to represent a same image or signal, and the coefficients may not be able to represent another model accurately. Thus, this method also needs high-quality prior information or a sensitivity analysis on all the coefficients.

The wavelet transform as a reparameterisation method was proposed by Lu and Horne (2000). In this method, the parameters are transferred to a wavelet domain, and the coefficients are computed. Then, based on a sensitivity analysis or prior information, a number of coefficients are selected to reparameterise the model. This method also is used in several studies (Grimstad et al., 2003, Sahni and Horne, 2005, Sahni and Horne, 2006a, Sahni and Horne, 2006b). This approach also needs a high-quality initial model to be used.

There are some further reparameterisation methods in which models are reparameterised based on the eigenvalues and eigenvectors of data sensitivity (Oliver and Chen, 2010). They are applied by Shah et al., 1978 and Rodrigues, 2006. In these methods, singular values or eigenvectors of sensitivity matrix are used. These methods need sensitivity matrix calculations. Also, this reparameterisation may lead the optimisation to a local optimum, as a number of high-sensitive variables chosen based on sensitivity matrix at the initial stage are only adjusted.

Most of the reparameterisation methods were concisely explained in the previous sections. As it was seen, in the majority of the techniques, prior information or data sensitivity matrix is required. If prior information does not have a good quality, there will be an inaccuracy in the reparameterisation, and consequently there will not be any guarantee that the selected parameters are able to characterise the models properly. Thus, it is fair to expect reparameterisation errors in these techniques. Each of this method can be an appropriate technique in a particular problem, and according to the problem, the reparameterisation method should be selected, for instance, if there is a channel in the geomodel, the zonation cannot be an appropriate method; or, if there is a visible geostatistical correlation in the reservoir, a pilot point technique can be a superior choice.

Another approach which can be taken to solve this inverse problem is to allow whole parameters of m directly be changed. This approach is known as full-parameterisation. In this approach, there are no reparameterisation errors. In this direct approach, a regularisation term is usually added to the objective function to overcome the ill-posedness. Hunt et al. (2007) compared the results of direct calibration and indirect calibration for groundwater problems. They stated that a direct approach in which the objective function has a regularisation term can deliver more reliable results in comparison with the indirect methods with a reparameterisation method. It should be pointed out that the comparison of the reparameterisation methods does not fall into the scope of this study

and also it is not our aim. In this study, both reparameterisation and full-parameterisation are used according to the requirements.

3.3. Objective function formulation

The most common deterministic formulation is least squares of data misfits used in the following studies (Schulze-Riegert et al., 2003, Rotondi et al., 2006, Maschio et al., 2008, Maschio et al., 2005, Hajizadeh et al., 2011a). The least squares formulation without a regularisation term can be used when the number of variables is small. In the aforementioned studies, the number of variables was less than 40. If the number of variables is large (more than independent observations), a regularisation term should be added, otherwise the problem will be ill-posed (Oliver and Chen, 2010). Thus, a regularisation term may be added to the formulation (Tikhonov and Arsenin, 1977).

The λ (regularisation factor) should be carefully assigned otherwise, a wrong domination of one of the objectives over the other one may occur, for example, in a problems with very noisy observations, a large penalty factor for prior function causes insufficient observation fitting and in contrast, small value cause highly oscillatory solutions as a result of noise amplification (Tautenhahn and Qi-nian, 2003). The deterministic formulation with a penalty term is widely used in ground water problems (Doherty, 2003, Fienen et al., 2009), and also it is applied in a number of studies in history matching problems (Makhlouf et al., 1993a, Leo et al., 1986). The main challenge of this formulation is the determination of the proper regularisation factor, which is a time-consuming procedure.

Having a single solution from an ill-posed problem in which observations contain noise, and also there are modelling and optimisation errors is accompanied by risks, hence, the history matching procedure is usually culminated by performing uncertainty quantification. It will be more reliable, if a set of solutions with their corresponding probabilities can be presented for this kind of problems. By the set of solutions, an uncertainty analysis can be

carried out. In the probabilistic approach, it is sought to find the posterior probability distribution function ($p(m|d_{obs})$). Due to the importance of uncertainty analysis, the probabilistic approach has found more interests in history matching problems. The Bayesian framework for the first time was utilised by (Gavalas et al., 1976, Shah et al., 1978), then it is extensively used by others (Oliver, 1994, Craig et al., 1996, Hegstad and Omre, 1997, Barker et al., 2000, Zhang et al., 2005b, Liu et al., 2001, Li et al., 2001, Tavassoli et al., 2004, Gao et al., 2007, Oliver et al., 2008).

Finding the most probable model in this approach is similar to the deterministic approach. An optimisation rule is used to find the best model. An advantage of the Bayesian framework over the deterministic approach is that weighting factors (covariance matrixes) of individual elements of observed and prior data, and accordingly the regularisation factor are estimated using an assessment of measurement, modelling and prior model reliabilities. Consequently, in the Bayesian framework, the determination of a regularisation factor by numerical calculations is unnecessary. In this approach, the challenges are in the selection of the set of solutions, if the problem was linear, the posterior probability function would be Gaussian and the covariance matrix would be easily estimated. But, in history matching problems, the posterior probability function will not be Gaussian (Oliver and Chen, 2010). The deviation from Gaussian is related to the degree of nonlinearity, i.e., a higher nonlinearity causes more deviation from being Gaussian (Tarantola, 2005). In order to select a set of models, two of the methods are as following: 1- With the assumption of the posterior probability is close to a Gaussian distribution, the posterior covariance is approximated by equation 1 (Tarantola, 1987). 2- Randomised maximum likelihood (RML) which is an improved Markov chain Monte Carlo algorithm for sampling, in this approach, different samples from the initial distribution and observed data probability are selected as prior model and d_{obs}, then the optimisation is carried out for each arrangement, and according to a Metropolis-Hasting test, the results of calibrations are accepted or

rejected (Oliver et al., 1996). Thus, by the probabilistic approach, it is possible to estimate a set of solutions and also it does not need numerical calculation of the regularisation factor. In this study, the Bayesian framework is utilised to define the objective functions, due to its benefits in comparison with the deterministic formulation.

$$C_{m'} \approx C_m - C_m G^t{}_{m_\infty} \left(G_{m_\infty} C_m G^t{}_{m_\infty} + C_D\right)^{-1} G_{m_\infty} C_m \quad (1)$$

The covariance matrixes should be carefully assigned in this formulation; otherwise they may misdirect the optimisation algorithm to a false optimum point. It should be mentioned that modelling errors are difficult to be computed, thus usually it is assumed zero and C_D=C_d (Carter, 2004). It is a demanding task to accurately determine and quantify the covariance matrixes' elements. The effect of covariance matrixes' uncertainties on the reliability of achieved history matched models has not been studies before.

3.3.1. The shape of landscape

In order to choose a suitable optimisation algorithm for a problem, the shape of the objective function can provide valuable information. For example, if the shape of objective function is multimodal, gradient-based algorithms may be trapped in a local minimum and a stochastic optimiser should be utilised (Haupt and Haupt, 2004). Due to the high-dimensionality of the problem and also dissimilarities in case studies, the shape of landscape cannot be estimated accurately (Oliver and Chen, 2010). One of the beliefs about the shape of landscapes is that although measurement noise is assumed Gaussian and the regularisation terms are quadratic, the corresponding landscape of S(m)s are not quadratic (Oliver and Chen, 2010). Zhang et al. in 2003 and Tavassoli et al. in 2005 showed that their problems have multiple minima. Being non-quadratic is mostly on the account of the nonlinearity of *g* which may be enhanced by including a reparameterisation and switching a *g* operator to *h*. The effect of reparameterisation operators on the shape of *S(m)* has not been studied.

3.3.2. Data assimilation

Kalman filter is a sequential update method (Welch and Bishop, 1995, Harvey, 1991). Kalman filter is utilised widely in robotic problems, navigation and time series, and the Kalman filter or the Ensemble Kalman filter (EnKM) were applied in groundwater and hydrocarbon reservoir parameters identification from last decades (Geir et al., 2003, Reichle et al., 2002, Lodoen and Omre, 2008, Gu and Oliver, 2005, Liu and Oliver, 2005, Skjervheim et al., 2005, Lorentzen et al., 2005, Lorentzen et al., 2009, Evensen et al., 2007, NÃ¦vdal et al., 2003, Liang et al., 2009, Jung and Choe, 2010, Emerick and Reynolds, 2012). However, this method is being used extensively and has several advantages, it suffers from some drawbacks: a- underestimation, b- overshooting, c- not coping with discrete variables, d- the linearization assumption may not work in all problems, and e- being appropriate only for Gaussian distributions (Oliver and Chen, 2010).

3.4. Optimisation

In the calibration step, the history matching problem will become an optimisation problem. With the progress in the optimisation science, new algorithms are developed. From 1970 up to present, along with this progress, the applications of the new algorithms are being seen in history matching.

Steepest descent algorithms are the first methods applied in history matching (Chen et al., 1974 and Coats et al., 1970). In these methods, only first-order information of the previous step is utilised to direct the optimisation, thus they have a low speed of convergence which restricts their applications. The conjugate gradient descent algorithms are also first-order optimisation methods, but in these algorithms, the direction of changes is a combination of gradient descent of current step and previous steps, hence it has a quicker convergence in comparison with the steepest descent (Oliver and Chen, 2010). Gavalas et al., in 1976 and

Lee and Seinfeld in 1987 used a conjugate gradient method as an optimiser in history matching problems. However, several preconditioning formula have been developed to optimise the step size in the first-order algorithms, they are not very efficient.

As mentioned in chapter two, the step size is better to be the inverse of hessian matrix based on the Newton equation. With development in computer industry, the second-order algorithms (hessian calculation) found a place in history matching problems. But still, the hessian matrix computation is very time-consuming, because of high-dimensionality of the problems. Therefore, various methods have been applied in history matching to approximate the hessian matrix by some other terms.

Gauss-Newton was used in history matching in several studies (Li et al., 2003, Tan and Kalogerakis, 1992, He et al., 1997). In this method, the hessian is approximated by the sensitivity matrix. The Gauss-Newton is computationally expensive (Oliver and Chen, 2010). The Levenberg-Marquette algorithm is a method between gradient descent methods and the Gauss-Newton method, and it has a damping coefficient which can speed up the process of optimisation; a small value for this coefficient switches the algorithm to Gauss-Newton and a large value switches the algorithm to a descent method. This algorithm is applied in several studies (Zhang et al., 2003a, Vefring et al., 2006).

Another group of hessian approximation methods are variables metric algorithms, such as quasi-Newton, BFGS and DFP. In these methods, an initial guess is used as the hessian matrix, and in each iteration, it is tried to be improved. In the following studies, variable metric methods are implemented in history matching (Yang and Watson, 1988, Liu and Oliver, 2004, Dong and Oliver, 2008). When the number of variables is large, the aforementioned algorithms need a huge amount of memory. The limited-memory BFGS is an alternative approach in which to approximate the inverse hessian matrix, instead of storing a large matrix, only a few vectors are kept to represent the matrix. This method is applied in full-parameterisation problems which have a large number of variables (Zhang

et al., 2003b, Eydinov et al., 2009). The gradient-based algorithms, conjugate gradient, Gauss-Newton, Levenberg-Marquette, BFGS and limited memory BFGS, are compared with each other in several case studies (Zhang et al., 2002). The results indicate that limited memory BFGS outperform in comparison with the other algorithms.

In all the gradient-based algorithms, a number of sensitivity matrix (G) calculations are required. They cannot be calculated analytically, and usually they are approximated numerically. The most common and straightforward technique is the finite difference method. Adjoint methods are another means for G calculation. Chavent et al., 1975 and Chen et al., 1974 introduced the adjoint method for the first time. In their studies, they calculated the gradient of the objective function for each variable. After that, the adjoint was used to calculate *G* matrix instead of the calculation of gradient of *S(m)* (Li et al., 2003). The main difficulty of the adjoint methods is that it is challenging to incorporate them to various simulators and decision variables (Oliver and Chen, 2010). There are also some other methods for the approximation of sensitivity matrixes, such as simultaneous perturbation stochastic approximation (SPSA) (Spall, 1992), and making use of streamline simulators (Kulkarni and Datta-Gupta, 2000). In the SPSA method, the gradient is approximated based on only two evaluations, it has uncertainty (Oliver and Chen, 2010). The use of streamline simulation for sensitivity matrix calculations needs a further computation (streamline simulation).

Zhang et al. (2003) showed that the classical methods do not necessity converge to a local minimum and Levenberg-Marquette can escape from local minima which are located in narrow valleys. But, the gradient-based algorithms cannot always escape from the local minima, and they may get stuck in a local minimum, as they direct the optimisation only downward (Ouenes et al., 1993). Thus, another group of optimisers, stochastic algorithms, came into use from 90s. In these methods, search spaces are visited randomly to escape from the local minima. This procedure is the basis for many stochastic optimisers such as

Simulated Annealing (SA) (van Laarhoven and Aarts, 1987), Tabu Search (Glover, 1989, Glover, 1990), and Stochastic Gradient Descent (Gardner, 1984). The SA is used for history matching problems by Ouenes et al., 1993, Ouenes and Bhagavan, 1994 and Portelland and Prais, 1999. Mantica et al., 2002 developed a hybrid method in which SA is used as a chaotic optimiser to select a set of samples as initial points for gradient-based optimiser. The Tabu search was applied by Yang et al. in 2007. Neighbourhood Algorithm is another history matching algorithm which was applied by Christie et al. (2006). The mentioned algorithms cannot be categorised into global optimisers, since they have the chance of getting stuck in a local minimum. More recently, population-based algorithms came into history matching.

One of the first population-based algorithms used in history matching problems several times is genetic algorithm (Romero and Carter, 2001, Ballester and Carter, 2007, Maschio et al., 2008). Romero and Carter (2001) compared the results of GA with SA, and they showed that their designed GA in which the chromosomes are multi-dimensional can perform better than SA. In another study, GA was compared with gradient-based algorithms, including quasi-Newton and conjugate gradient; it was presented that GA delivers more reliable results (dos Santos et al., 2009). Some other population-based algorithms are also implemented in history matching problems, such as Particle Swarm (PSO) (Mohamed et al., 2010), Ant Colony (Hajizadeh et al., 2011a), Differential Evolution (Hajizadeh et al., 2010), Covariance Matrix Adaptive Evolutionary Strategy (CMA-ES) (Schulze-Riegert et al., 2009). Rwechungura et al., (2011) reviewed the application of several optimisation algorithms; they stated that non-gradient algorithms are robust methods. Zhang et al. state that the non-gradient algorithms are not efficient as gradient-based methods, due to the high computational costs. Oliver and Chen (2010) believe the non-gradient algorithms are appropriate when the number of variables is small.

Thus far, none of the stochastic optimisers have been used for large history matching problems as a direct calibrator.

Selecting a suitable optimiser for a history matching problem is a controversial matter because of dissimilarities in the case studies and also the unknown shape of landscape. Several algorithms have been studied, but still the optimisation step is an ongoing research topic. Petroleum engineers always examine latest optimisation algorithms to find the desired algorithm. The desired algorithm should have the following properties: 1- Be able to approximate the global optimum point without being influenced by the objective function specifications (number of variables and/or shape of landscape). 2- Be computationally efficient.

The artificial bee colony algorithm is one of the most-recent optimisation methods. It was developed by Karaboga et al. in 2005. This algorithm was compared with several standard algorithms including GA, particle swarm, evolutionary strategy and differential evolution algorithm, by Karaboga and Akay (2009) on numerical benchmarking functions. The results indicate that the ABC outperformed among other algorithms. The ABC has been applied for several optimisation problems so far, including, digital IIR filters (Nurhan, 2009), heat-transfer coefficient estimations (Zielonka et al., 2011), capacitated vehicle routing problems (Szeto et al., 2011) and many other problems, but it has not been studied in history matching problems.

3.4.1. Pareto optimisation

In recent years, several multi-objective algorithms have been developed based on similar concepts as single-objective population-based algorithms. In the multi-objective algorithms, the selection is based on ranking and a single execution delivers a good approximation about the Pareto front. So far, different algorithms have been developed to deal with multi-objective problems, some of these algorithms have been mentioned in

Coello et al., 2007 book: Vector Evaluated GA (VEGA) (Schaffer, 1985), Lexicographic Ordering GA (Fourman, 1985), Vector Optimized Evolution Strategy (VOES) (Kursawe, 1991), Weight-Based GA (WBGA) (Hajela and Lin, 1992), Multiple Objective GA (MOGA) (Fonseca and Fleming, 1993), Niched Pareto GA (NPGA, NPGA 2) (Horn et al., 1994, Erickson et al., 2001), Nondominated Sorting GA (NSGA, NSGA-II) (Deb et al., 2002, Srinivas and Deb, 1994), Distance-based Pareto GA (DPGA) (Osyczka and Kundu, 1995), Thermodynamical GA (TDGA) (Kita et al., 1996), Strength Pareto Evolutionary Algorithm (SPEA, SPEA2) (Zitzler and Thiele, 1999), Multi-Objective Messy GA (MOMGA-I,II,III) (Veldhuizen and Lamont, 1999, Zydallis et al., 2001, Day and Lamont, 2005), Pareto Archived ES (PAES) (Knowles and Corne, 1999, Knowles and Corne, 2000), Pareto Envelope-based Selection Algorithm (PESA, PESA II) (Corne et al., 2000), Micro GA-MOEA (μGA, μGA2) (Coello and Toscano Pulido, 2001, Toscano Pulido and Coello Coello, 2003), Multi-Objective Bayesian Optimization Algorithm (mBOA) (Laumanns and Ocenasek, 2002, Pelikan et al., 2005). In NSGA-II (Deb et al., 2002) algorithm called controlled elitist genetic algorithm, recombination is based on two terms, rank and distance of individual chromosomes. Distance is used to supply diversity in Pareto front which can improve convergence to the optimal Pareto front (MathWork, 2011a). NSGA-II is widely applied especially when the number of variables is limited (Coello et al., 2007).

Multi-objective algorithms have been used in a number of papers for history matching problems. Schulze-Riegert et al. defined four objective functions which are misfits of water-cuts of three different wells and misfits of bottomhole pressure of one of the wells. Hajizadeh et al. 2011b reported a faster convergence using the multi-objective optimisation in comparison with the single-objective optimisation. In that study, there were two objective functions made by splitting the wells into two groups. Mohamed et al. reported a faster convergence using the multi-objective optimisation. In their study, two objective

functions were defined: water rates and oil rates. The applied optimisation algorithms are as following: a SPEA was applied by Schulze-Riegert et al., 2007, a multi-objective particle swarm optimisation was applied by Mohamed et al., 2011, a differential evolution for multi-objective optimisation using Pareto ranking was applied by Hajizadeh et al., 2011b.

In the mentioned studies, a regularisation term (a prior knowledge) was not included and the objective functions consisted of different parts of the likelihood term. This separation of the likelihood term does not improve the quality of history matched models unless when the weighting factors for the separated terms are uncertain which was not stated in those studies. A single objective optimisation performs more effectual on problems in which objective functions can be related to each other with weighting factors than a multi-objective optimisation, as it minimises proportionally all the terms according to their weighting factors.

3.4.2. Proxy-modelling

Fitness approximation (proxy-modelling) is a method for overcoming the computational time in optimisation problems (Jin, 2005, Cullick et al., 2006). So far, different proxy-modelling techniques have been used in history matching problems. In the applied methods, there were two main focus areas to enhance the quality of proxy-modelling: 1- selecting a suitable proxy model, and 2- selecting the sufficient number of samples and the proper location of samples. Inappropriate proxy models or samples result in inaccurate space modelling (figure 3.2) and accordingly lead to wrong history matched models. Thus far, different types of sampling strategy and different types of proxy model are checked out. Silva et al. in 2006 used an artificial neural network as a proxy model, and they applied it for two case studies. Cullick et al. 2006 and Sampaio et al. 2009 also used an artificial neural network as a proxy model and articulated good results. Polynomial is

another proxy model, but high-quality results were not achieved by this technique (Cullick et al., 2006, Osterloh, 2008, Li and Friedmann, 2005). Osterloh (2008) attained good results by using kriging (DACE package is used for multi-dimensional interpolations (Lophaven et al., 2002)) as a proxy model; however Li and Friedmann (2005) revealed that kriging smooth out the response surface.

In addition to the proxy models, the type of sampling strategy also plays a significant role on the quality of trained proxies (Zubarev, 2009). Different experimental design methods have been suggested to select the appropriate set of samples, such as space filling designs (Latin Hypercube Sampling) and partitioning domain to subdomains for sampling (Li and Friedmann, 2005). Latin Hypercube Sampling (LHS) was reported as an effective technique for sampling (Zubarev, 2009). LHS is a statistical method which is capable of selecting samples from a multivariate distribution. It was developed and introduced by Mckay (1979). In this method, the space is discretisation with same sized Latin squares and then samples are selected in a manner in which there is only one sample in each row and each column (figure 3.3).

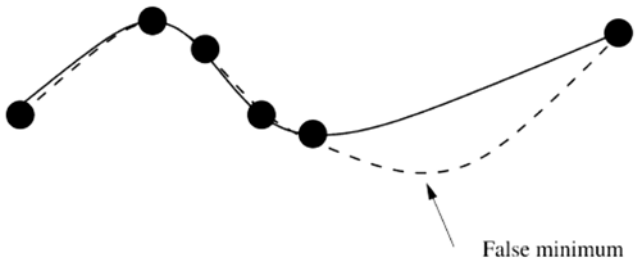


Figure 3.2 The solid line and dashed line expresses the original fitness function and the approximate function, respectively, and the dots are the available samples (Jin, 2005)

All the fitness approximation methods which have been applied in history matching use an almost similar approach called uncontrolled fitness approximation. In the uncontrolled approach, none of the evolution-control (model management) techniques is employed to control (manage) the approximation function (model). In the computer science literatures,

it has been shown that the uncontrolled fitness approximation approach has the potential of misleading of the optimisation to wrong optimum points (Jin, 2005, Yaochu et al., 2001). It is because of the limited number of samples, the multi-dimensionality of the search space and nonlinearity of the function.

Figure 3.3 Latin hypercube sample for a 2D problem (Wikipedia)

3.5. Fast simulators

In order to execute the reservoir simulation (grid-based methods), not only it needs the solutions of the inverse problem, but also requires many other data, such as fluids properties, relative permeability curves and so on. Its requirements restrict its applications when a quick forecast is sought. Also, as mentioned before, the reservoir simulation is a time-consuming function, if an economical sensitivity analysis is sough, the procedure using the grid-based reservoir simulation will be computationally expensive. Thus, engineers look for a method in order to provide a quick estimation with the minimum information. Thus far, several methods have been developed, such as Capacitance Resistance Method (CRM) (Sayarpour et al., 2007, Sayarpour et al., 2009a, Sayarpour et al., 2010, Mou et al., 2005, Yousef et al., 2006, Lake et al., 2007, Delshad et al., 2009), streamline simulation (Rust and Caudle, 1972, Peddibhotla et al., 1997, Batycky et al., 1997) and type curve matching methods (Arps, 1944).

3.5.1. Decline curve analysis

One of the traditional type curve matching methods regarding performance prediction is decline curve analysis. There are three main formulas, harmonic, exponential and hyperbolic (Guo et al., 2007). These formulas are achieved from the following equation:

$$\frac{1}{q}\frac{dq}{dt} = -bq^{d} \qquad (2)$$

Where d is between zero to one.

When d is equal to zero, the model is exponential; when d is equal to one, the model is harmonic; and when d is between zero and one ($0<d<1$), the model is hyperbolic. The b and d are estimated by analysis the historical performance of the well. After finding b and d using history matching, there will be an analytical equation which can provide flow performance (q) versus time. If the conditions which are affecting the rate are not altered by an outside effect, the equation will be useful to be applied. The normal forms of this method are not capable of predicting the future performance during secondary or tertiary recovery.

3.5.2. Streamline simulation

Streamline simulation is another method used for predicting the future performance of wells and field. In this method, pressure equations are solved implicitly, and the saturation (mass) equations are solved explicitly, unlike the grid-based simulation methods in which these two are solved together. The main advantage of this simulation method is its speed of simulation. It usually is faster than the conventional approach of simulation (Al-Najem et al., 2012). The main drawback of this method for quick forecast is that it required large amount of input data, including permeability distribution, rock and fluid prosperities, and

geomodel. Hence, in order to have a precise streamline simulation, it is required to solve a nonlinear inverse problem.

3.5.3. Capacitance resistance method

One of the most recent methods concerning field performance prediction is CRM. This method is applied in reservoir models which are under water-flooding. It performs similar to signal processing in which water injections are input signals and productions are output signals. There are two main elements between each pair of injection-production wells, resistance (well connectivity) and capacitance (fluid and rock compressibility and its initial content). A mathematical equation is developed in which the production rate is a function of injection rate of each injector with a specific weighting factor (f_{ij}) and also the reservoir pressure and productivity index (J) as equation 3. Based on equation 3, the production rate of producer j is a function of its production rate at previous time step and injection rates of all injectors at the current step.

$$q_j(t_k) = q_j(t_{k-1})e^{-\frac{t_k - t_{k-1}}{\tau_j}} + \left(1 - e^{-\frac{t_k - t_{k-1}}{\tau_j}}\right)\left(\sum_{i=1}^{No_{Injector}} f_{ij}\, I_i^k - J_j\tau_j\left(\frac{\Delta p_{wf,j}}{\Delta t_k}^k\right)\right) \quad (3)$$

Where

$$\tau = \frac{c_t V_p}{J} \quad (4)$$

C_t is total compressibility, J is productivity index, V_p is pore volume. Using only production and injection history rates, the unknown elements of this equation (f and τ) are estimated. Then, it can be applied for future performance forecasts. This model only provides prediction regarding liquid rate, in order to estimate the oil production rates, the equation should be coupled with a fractional flow. Gentil in 2005 developed a fractional flow in which its parameters also are estimated only by analysing the historical performance of wells. This method has some restrictions; the equation was derived by the

assumption of having only two phases (water and oil). Hence, this model cannot be applied for undersaturated reservoirs. Also, this model cannot predict the performance during gas-flooding.

Chapter 4

4. Reservoir characterisation using artificial bee colony optimisation

This chapter analyses the application of one of the most-recent optimisation algorithms, artificial bee colony, in history matching, for the first time. The questions, the author wishes to answer in this chapter, are:

a- Is the artificial bee colony algorithm a suitable optimiser for history matching?

b- Does it deliver better results in comparison with other optimisers?

c- Does the shape of landscape have significant influence on its performance?

The chapter begins by describing a synthetic reservoir model constructed for the assessment of the algorithm, and followed by explaining the implementation of artificial bee colony algorithm in our developed history matching framework. Afterward, the results of history matching using the artificial bee colony algorithm and three other optimisation algorithms, genetic algorithm, simulated annealing, Levenberg-Marquette, are presented. Then, the outcomes are compared with each other. In this chapter, also, the effects of reparameterisation on the shape of landscape and the performance of the optimisation algorithms are cursorily studied. This chapter is a modified and adjusted version of one of our published papers in a peer-reviewed journal, “SAYYAFZADEH, M., HAGHIGHI, M.,

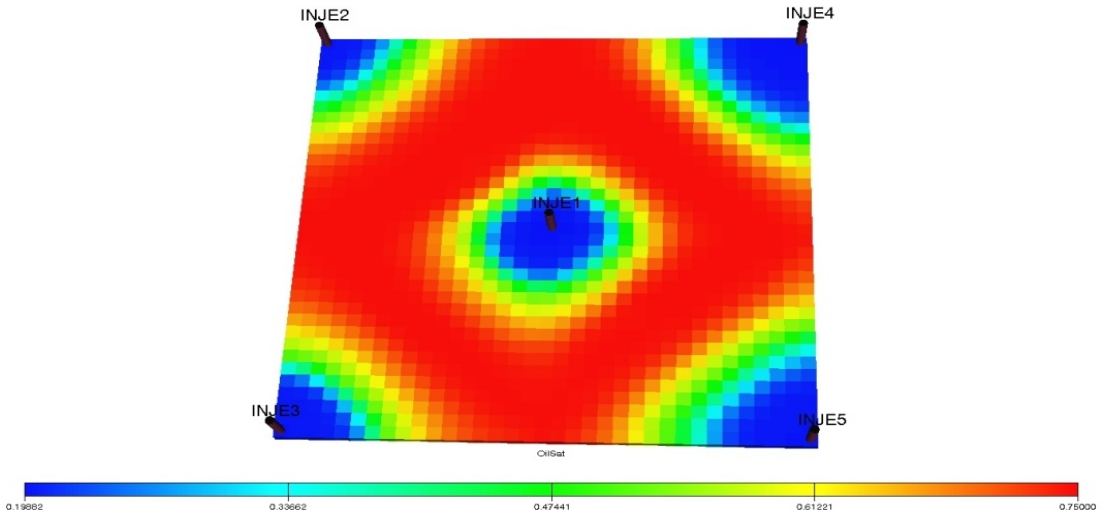
BOLOURI, K., & ARJOMAND, E. 2012. Reservoir characterization using artificial bee colony, in APPEA Journal 2012; 52:115-128".

4.1. Synthetic reservoir model

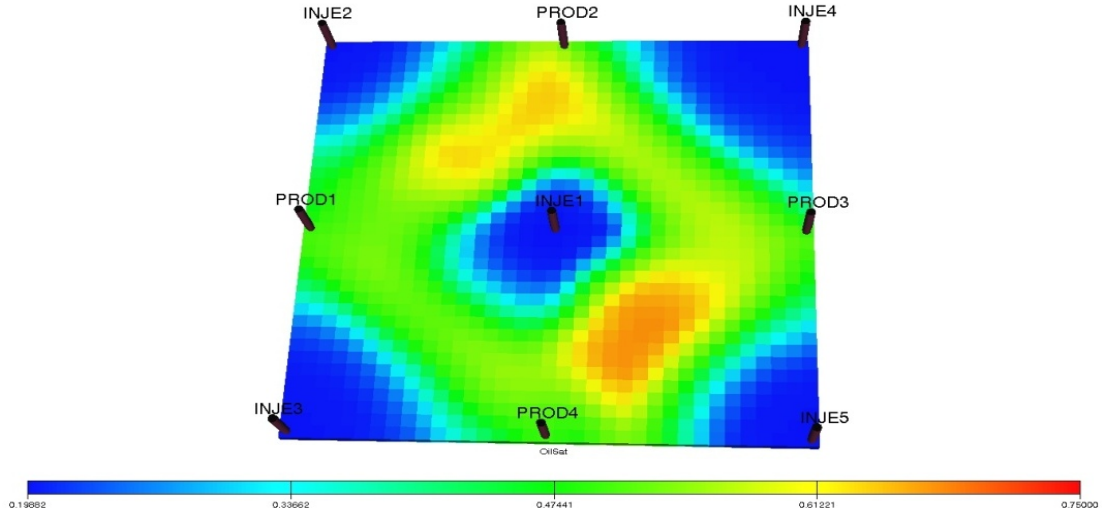
A synthetic reservoir model is constructed to evaluate the proposed optimisation method (artificial bee colony). The reservoir model consists of 35×35×3 gridblocks in x, y and z directions respectively. The permeability is a function of porosity based on a simplified Kozeny-Carman correlation (equation 1); and the permeability is the same in x and y directions. The permeability in the vertical direction is 10% of the horizontal. This case is a system with two phases (water and oil), and has a nine-spot waterflooding pattern (five injectors and four producers). The injector wells are connected to the first and second layers, and the production wells are completed only into the second layer. The map of oil saturation after 2210 days is shown in figure 4.1. The reservoir properties are summarised in table 4.1.

$$k\ (mD) = 30{,}000 \times \phi^3 \qquad (1)$$

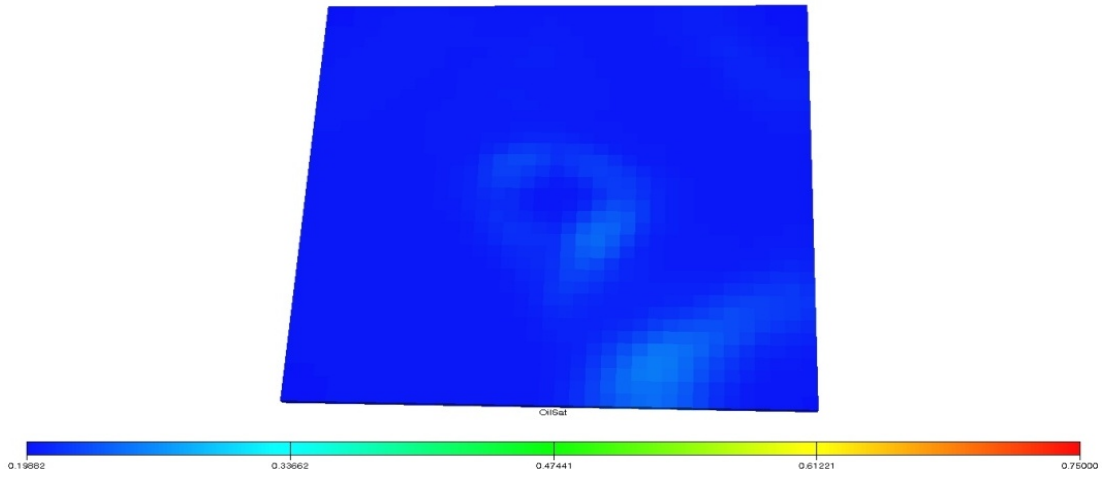
The porosity distribution of the reservoir model (reference) (m_{ref}) is generated via a geostatistical interpretation. The value for 16 gridblocks (nine gridblocks where the wells are located at and seven other gridblocks which are selected randomly) in each layer are defined manually. By analysing these 48 gridblocks data, the porosity distributions are created. An ordinary kriging performed by SGeMS package (Remy et al., 2009) is utilised for interpolating the properties between these 16 gridblocks for each layer. The permeabilities are calculated based on equation 1. The permeability distribution for each layer of the reservoir model is shown in figure 4.2.



1st layer



2nd layer



3rd layer

Figure 4.1 Oil saturation for the synthetic reservoir model for each layer

Since synthetic cases do not have any actual flow performance history, they should be simulated for a period of time to create a performance history. The history (observed vector) is generated, for the case study, by simulating the reference case for 2210 days (6

years) in 38 time steps. Its elements are well bottomhole pressures, well oil production rates, well liquid production rates, field oil production rate, field pressure and field water production rate at each time step. The total number of observed data is 760 (N_D=760). Gaussian noise with standard deviation of 2% and zero mean is added to the observed vector. In this case study, all the information of the reservoir model is assumed exactly known, excluding the porosity in every gridblocks. Thus, the porosities should be estimated (reproduced) using inverse modelling based on the observed data. The total number of variables (N_m) is 3648 (35×35×3-9×3).

Table 4.1 The synthetic reservoir properties

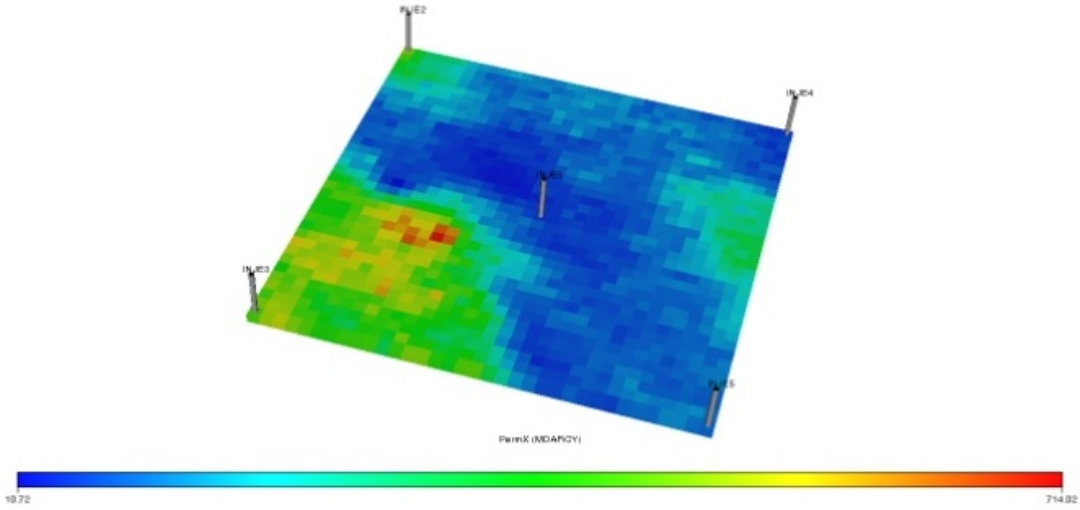
Property	Value
Mean of porosities of the 1st layer	0.167
Standard deviation of porosities of the 1st layer	0.045
Mean of porosities of the 2nd layer	0.192
Standard deviation of porosities of the 2nd layer	0.044
Mean of porosities of the 3rd layer	0.170
Standard deviation of porosities of the 3rd layer	0.037
Dimension of gridblocks in X,Y,Z direction	80, 80, 60 ft respectively
Rock compressibility	4×10^{-6} 1/psia @ 4500 psi
Oil viscosity	2 cP @ 6000 psi
Water viscosity	0.8 cP @ 4500 psi
Initial pressure	6900 psi @ 9600 ft
Reservoir top	9540 ft
Water oil contact	9660 ft

The objective function for this case study is formulated by a Bayesian framework in which the likelihood term is only considered (equation 2). As mentioned in the previous chapters, when the number of variables is large, indirect calibration is used, especially when the optimiser is a nongradient-based algorithm. Hence, this problem is reparameterised via an operator (h) to reduce the number of variables, equation 3, and consequently, the objective function is changed to equation 4. The aim is to find the global minimum of the $S'(r)$. According to the objective function formulation and the known Gaussian noise, the $S'(r)$ at the global minimum should have a value close to $N_D/2$. To approximate the global optimum (the most probable model), an optimisation algorithm should be applied.

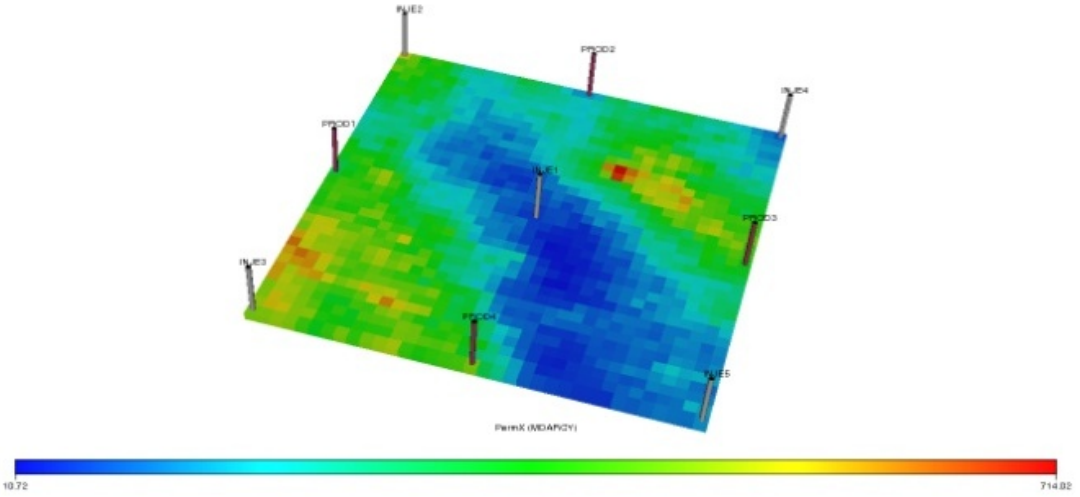
$$S(m) = \left(\frac{1}{2}(g(m) - d_{obs})^t C_D^{-1}(g(m) - d_{obs})\right) \quad (2)$$

$$m \approx h(r) \tag{3}$$

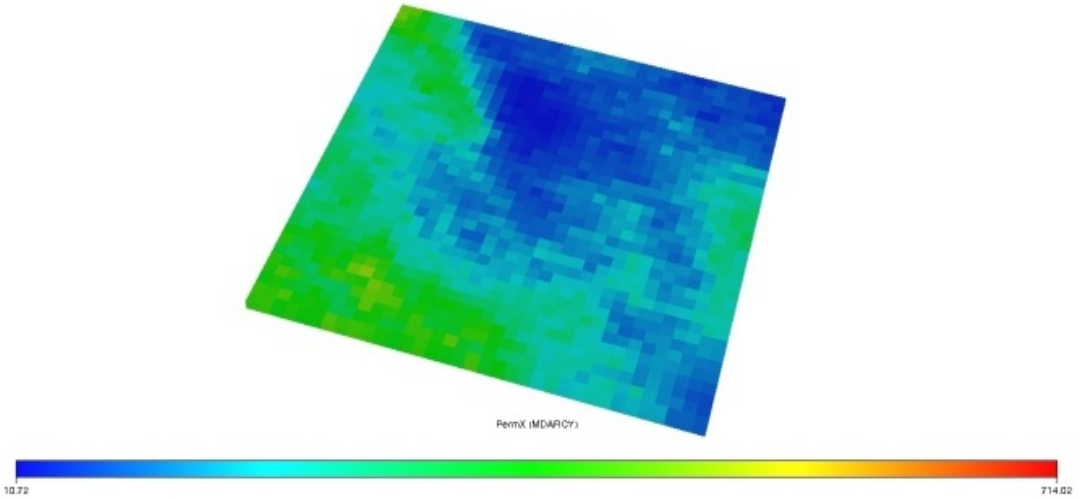
$$S'(r) = \left(\frac{1}{2}\left(\mathrm{g}\left(h(r)\right) - \mathrm{d}_{\mathrm{obs}}\right)^{\mathrm{t}} \mathrm{C}_{\mathrm{D}}{}^{-1}\left(\mathrm{g}\left(h(r)\right) - \mathrm{d}_{\mathrm{obs}}\right)\right) \tag{4}$$



1st layer



2nd layer



3rd layer

Figure 4.2 Reference permeability distribution of the synthetic reservoir model

A real case is not used in the evaluations, since 1- its corresponding true solution is not necessarily known, and so further verification of the solutions acquired by different methods is not possible or reliable, and 2- the computational costs associated with the evaluations for a real case is much higher than the evaluations for a synthetic reservoir model.

4.2. Optimisation step for the case study

Thus far, various optimisation methods, including gradient-based and nongradient-based algorithms, have been suggested to be applied in history matching; they are reviewed in the third chapter of this thesis. Each method has its own capabilities and weaknesses. Gradient-based (classical) optimisers converge much faster than nongradient-based (stochastic) optimisers, (Zhang et al., 2005a); but they have some restrictions especially in systems with multiple local minima and/or discrete decision variables. On the other hand, stochastic methods are slow in terms of convergence and cannot perform appropriately in systems with a large number of variables (Oliver and Chen, 2010), but, in order to find the global minimum in systems with many local minima and nonlinear functions, one of the best options is making use of stochastic optimisers.

The artificial bee colony (ABC) method introduced by Karaboga in 2005 is one of the most recent stochastic methods. Its applications have been studied in different fields such as, digital IIR filters (Nurhan, 2009), heat transfer coefficient (Zielonka et al., 2011) and several numerical functions (Karaboga and Akay, 2009). In this section, the application of the artificial bee colony algorithm in assisted history matching and reservoir characterisation is investigated, and its advantages and disadvantages are explored by evaluating its performance on the constructed synthetic models. The algorithm is tested out on two benchmarking functions. Then, it is implemented for the case study and its outcomes are compared with the outcomes of three conventional techniques, two stochastic

optimisers (genetic algorithm, and simulated annealing) and one classical optimiser (Levenberg-Marquette).

4.2.1. Artificial bee colony algorithm

To survive through the winters, bees have to produce adequate honey. Hence, honey bees need to explore and exploit the area around them for food as efficiently as possible, and store enough food. So as to carry out efficient exploitation and exploration, they not only memorise the locations with highest profitability, but also share their information with each other (Panigrahi et al., 2011). The profitability of a food source depends on its distance from the hive, richness and the ease of extraction (Karaboga, 2005). The main stage of efficient exploration and exploitation involves sharing information with each other through a dancing language. The bees communicate with each other by a waggle dance. In this dance, they explain the profitability of food sources, directions and distances (Panigrahi et al., 2011). By this means, they share their knowledge and consequently they are able to focus more on areas with highest profitability. Bees' exploration and exploitation of their surrounding area is one of the well-organised in nature.

The colony of ABC algorithm comprises of employed bees, onlookers and scouts, just like real honeybees. Usually half of the initial colony is assumed to be employed bees and the rest are onlooker bees. The colony size (N_C) should be defined prior to the beginning of the algorithm. Each employed bee is an agent for a solution (food source) (r). The corresponding initial solutions for the employed bees are generated randomly according to the given bounds; this step is known as initialisation. After the initialisation step, employed bees start searching for food randomly. In each attempt (cycle), employed bees look for a better food source (lower $S'(r)$) based on her knowledge, i.e., each employed bee always seeks for a better solution around her best-ever found solution. If an employed bee gets a better result in the current attempt, she will memorise the new location (solution);

otherwise she will keep the previous solution in her mind. After completing their search in each cycle, they share their information about the food source profitability ($fFitness_i$) and their locations (r_i) with onlooker bees.

Onlookers collect information from all of the employed bees and then make decisions based on the observations collected from all the employed bees to forage around the food source locations (solutions). The solutions with the highest profitability (lower $S'(r)$) are exploited by more onlooker bees according to the calculated probabilities. The probabilities are given via equation 5 and 6. r_i is an individual (solution) which is found by the employed bees. $fFitness_i$ demonstrates the profitability of r_i which is reported by an employed bee.

$$fFitness_i = \frac{1}{S'(r_i) + 1} \tag{5}$$

$$Prob_i = 0.1 + \frac{0.9 \times fFitness_i}{\left(max\left((fFitness)_i, i = 1 \dots \frac{N_C}{2}\right)\right)} \tag{6}$$

Scout bees search randomly and usually they look for a new home or food source. An employed bee will become a scout bee, whenever she cannot extract more food from its corresponding area (i.e., she cannot find a better solution with a higher nectar) after a number of attempts (limit).

ABC has a repetitive process, and in each attempt (cycle), each employed bee searches for a better fitness using their own individual knowledge and each onlooker searches for food after collecting the knowledge of the all employed bees. These cycles are repeated until the required satisfaction is gained, or one of the stopping criteria is met. The workflow of this algorithm is shown in figure 4.3. The two functions used to benchmark the capability of optimisation methods are Ackley and Schwefel.

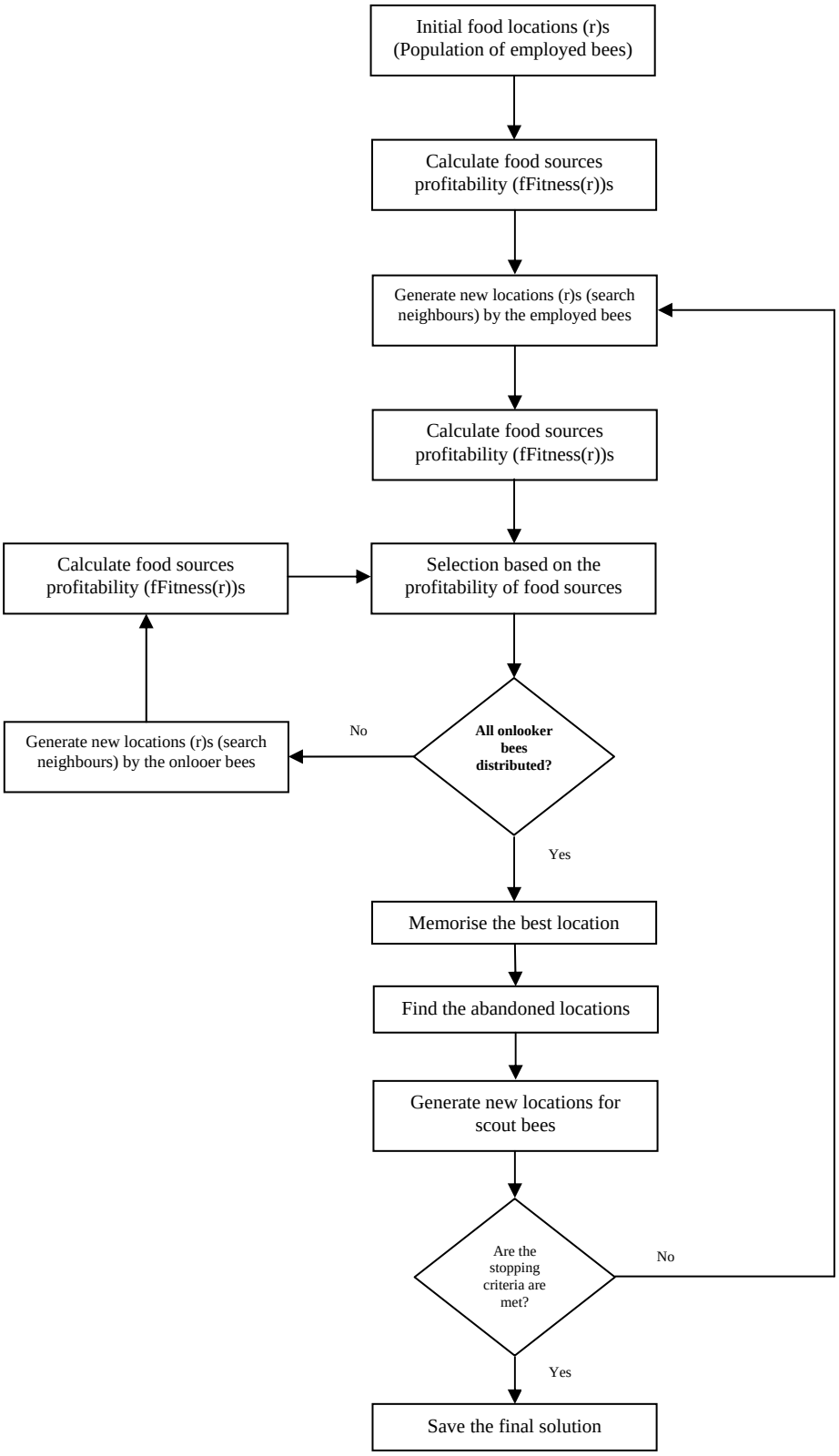


Figure 4.3 Workflow of artificial bee colony algorithm

4.2.2.1. Searching behaviour of ABC on the Ackley function

The Ackley function is a multidimensional model which is extensively used to test out optimisation methods; this function has several local minima and one global minimum. In

this study, in order to be able to draw the function and also investigate the searching behaviour, a 2-D Ackley function is used. The global minimum is (0,0) with a zero fitness value (f(0,0)=0). The function is displayed in figure 4.4 for -10<x,y<10. The function is written in equation 7, with a=20, b=0.2, c=2π (Molga and Smutnicki, 2005) and the search space is -2<x,y<2.

$$F(x,y) = -ae^{\left(-b\sqrt{\frac{1}{2}(x^2+y^2)}\right)} - e^{\frac{1}{2}(\cos(cx)+\cos(cy))} + a + e \qquad (7)$$

The ABC algorithm was run on the Ackley function with 50 iterations (cycles) and a colony size of 30, this point was found as a final solution: 3.9×10^{-10}, -3.2×10^{-10}, with a fitness of 1.4×10^{-9}, which is so close to the exact solution. The total number of evaluations (fitness calls) is 1500. The next figure (figure 4.5) exhibits the exploration and exploitation of ABC on this function. Each dot point shows a location (solution) which is evaluated either by employed, onlooker or a scout bee. It is noticeable that even though the landscape contains many local minima, ABC almost converged to the global minimum.

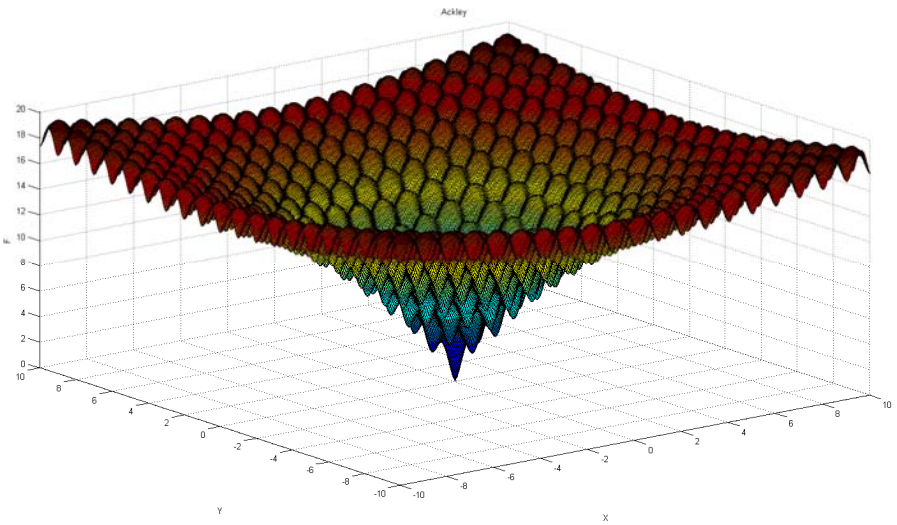


Figure 4.4 2D Ackley function

4.2.2.2. Searching behaviour of ABC on the Schwefel function

A similar procedure is used for the Schwefel function written in equation 8. For this study, x and y are between -500 and 500, and the global minimum is -837.9658 which is located

at (x y)=(420.9687 420.9687). Schwefel is a multidimensional function which is problematic even for stochastic optimisation algorithms, due to its highly rugged landscape shape (figure 4.6) (Molga and Smutnicki, 2005). ABC was run for the function with 70 iterations and colony size of 30. The following results were obtained: x,y of the solution are equal to 420.9688 and 420.9688 respectively and its fitness is -837.966 which is close to the exact solution. Figure 4.7 exhibits the convergence behaviour of ABC on Schwefel function.

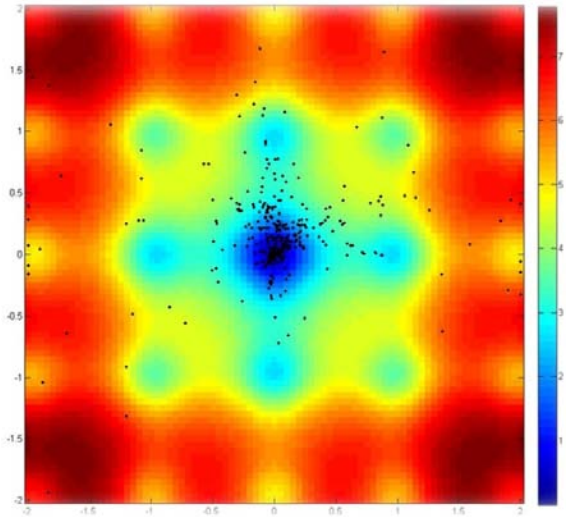


Figure 4.5 Searching behaviour of ABC on the Ackley function

$$F(x,y) = -x(\sin\left(\sqrt{|x|}\right) - y(\sin\left(\sqrt{|y|}\right) \quad (8)$$

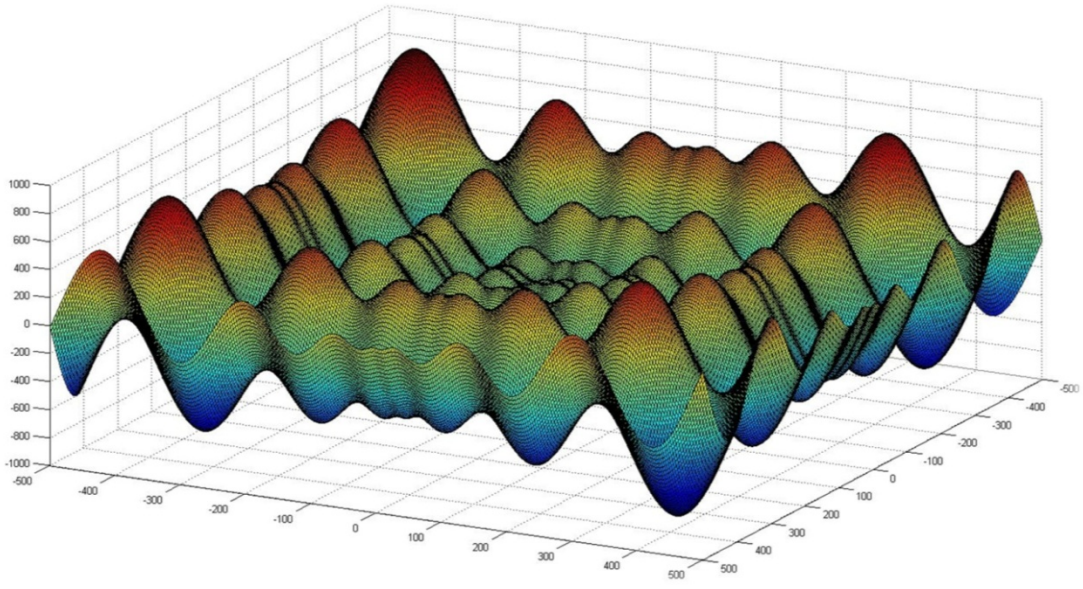


Figure 4.6 2D Schwefel function

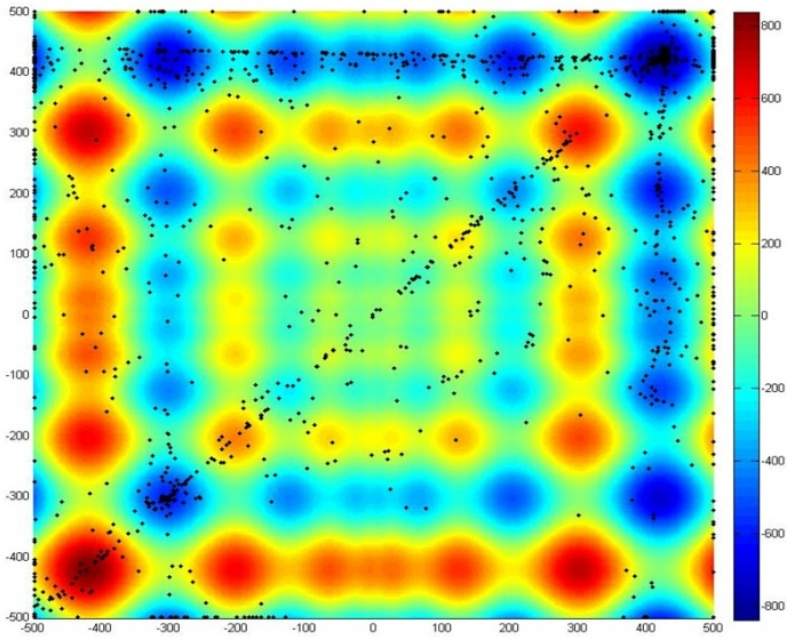


Figure 4.7 Searching behaviour of ABC on Schwefel function

4.2.3. History matching using artificial bee colony algorithm

In this section, the calibration step of history matching for the reservoir model is conducted using the artificial bee colony algorithm. It is coupled to the reservoir simulator (ECLIPSE E-100) within MATLAB (refer to appendixes for the framework development). Before carrying out the optimisation, a reparameterisation method should be selected to reduce the dimension of the search space. The pilot point technique is picked and applied, for the reason that this method is likely to be able to regenerate the reference porosity distribution, as the reference model was created itself by a geostatistical correlation. Hence, there is at least one solution (r) which can approximate the reference porosity distribution, i.e., the approximation of equation 3 around the reference model (mref) is valid. As a result, the reparameterisation error is assumed equal to zero, and the final misfit is assumed only on the account of the optimisation error which allows us to provide a high-quality assessment between the optimisation methods. It should be noted that a portion of final misfit value is also because of noise in measurements (see chapter two (2.4.1)).

Pilot points can alter the property distribution with their parameters, which are their locations and their values (in this case, porosity). A limited number of the gridblocks are described by the pilot points and the rest are calculated through geostatistical interpolation based on the pilot points and well data. In this study, a fixed number of pilot points are used to describe the reservoir and their values (porosity) and locations are considered as decision variables. Therefore, it is only necessary to find the best locations and the best values for pilot points and the rest of gridblocks are calculated through geostatistical interpolation. For this case study, it is assumed there are seven pilot points in each layer and their locations are defined in a polar coordination system. The pilot point technique has not been used in a polar coordination previously. Each pilot point is located on a specific fixed radius, but could be located at any angle. By the polar coordination and fixed radius, it is easier to distribute pilot point through the whole model and avoid having pilot point close to each other. Figure 4.8 demonstrates how pilot points can be distributed in a reservoir through a polar coordination. One pilot point should exist on the permitter of each circle. This minor alternation in the implementation of pilot point method can speed of the optimisation step, as scenarios which have pilot points close to each other will be removed.

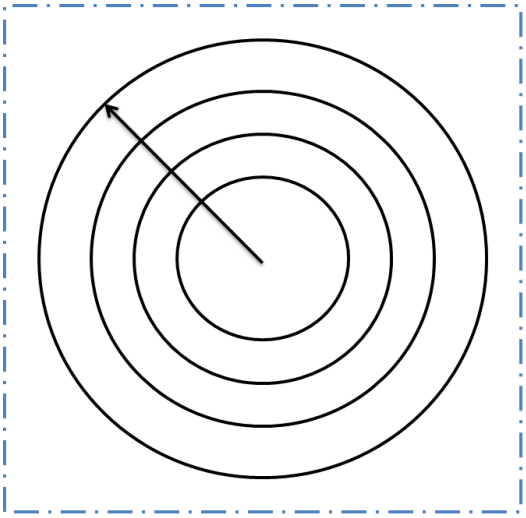


Figure 4.8 Pilot point distribution in polar coordination

In each layer, the values and angles of the seven pilot points are unknown (2×7=14). Therefore, there are totally 42 (14×3) decision variables. By history matching, the location and value of all the 21 pilot points are obtained, and the rest of the gridblocks are estimated using a geostatistical interpolation. In this study, the ordinary kriging is used for the interpolation (It should be mentioned that the reference porosity distribution was generated by allocating a specific value to 42 variables (r_{ref})). Also, it is assumed the variogram parameters are all known for further simplification of the problem which helps us to emphasis on the assessment of the quality and reliability of optimisation algorithms.

The ABC algorithm is coupled with the reservoir simulator (ECLIPSE) and the geostatistical software (SGeMS). The decision variables (r) are the aforementioned 42 elements. The algorithm should approximate the global minimum of the objective function ($S'(r)$). The mentioned options in table 4.2 are used for the algorithm. The stopping criterion is the computational costs in which ABC is not allowed to call the fitness function more than 20,000 times. $S'(r)$ is expected to be highly nonlinear, as in this function, the following calculations are being made: 1- geostatistical interpolations and 2-the simulation of fluid flow through the medium.

Table 4.2 ABC algorithm options

Colony size	20
The initial number of employed bees (the number of food sources)	10
The number of onlooker bees	10
The limit for the employed bees	100
Initialisation	Random
Stopping criteria	20,000 fitness calls (1000 cycles)

In order to assess the quality of obtained optimum points, the following criteria are used:

1- The objective function value at the achieved optimum point ($S'(r_\infty)$); it should be close enough to $N_{obs}/2$ (≈380), (a lower value for $S'(r_\infty)$ expresses a higher chance of existence for r_∞ according to the Bayesian interpretation).

2- The ability of future performance prediction; this criterion is measured by estimating cumulative produced water and oil for 1000 days subsequent to the last timestep of

history matching (2210th) for two different water injection scenarios. The actual cumulative produced water and oil for 1000 days for the two scenarios are given and shown in table 4.3. The estimated values via the obtained history matched models should be close enough to the reference values.

3- The generated porosity distribution (m_∞=h(r_∞)); the estimated porosity distribution after inverse modelling is compared with the reference distribution, based on equation 9. (Q should be close to zero)

$$Q = \frac{1}{N_m}\sum_{i=1}^{N_m}\left|\frac{(m_{\infty_i} - m_{ref_i})}{m_{ref_i}}\right| \times 100, \quad m_\infty = h(r_\infty) \tag{9}$$

History matching was carried out by estimating the global optimum of $S'(r)$ using the artificial bee colony algorithm. The ABC converged to a point which is denoted by r_∞ after meeting the stopping criteria. The objective function value at the achieved point is 503. It is almost close to the reference value (375). Figure 4.9 expresses schematically the matching quality for four producers; the left handside figures show the initial misfit (before history matching) between the reference (history[1]) and prior values (the prior were calculated via assigning the initial porosity distribution created via a geostatistical interpolation based on only the 9 gridblocks data); the right handside figures show the misfit between the reference and gained data via the ABC (after history matching).

The blue line shows the reference oil production rates, the red line shows the oil production rates predicted by the initial model and the orange line shows the oil production rates predicted by the model gained after history matching. The figures express that good matching between the simulation and history can be achieved by the ABC algorithm.

[1] The shown graphs are noise-free history, but the matching reference was the history with noise to be similar to real case studies.

The other results are shown in table 4.3 which demonstrate almost satisfactory results. Using the achieved model (r_∞), m_∞ is calculated, and the future performance for the two different scenarios is estimated.

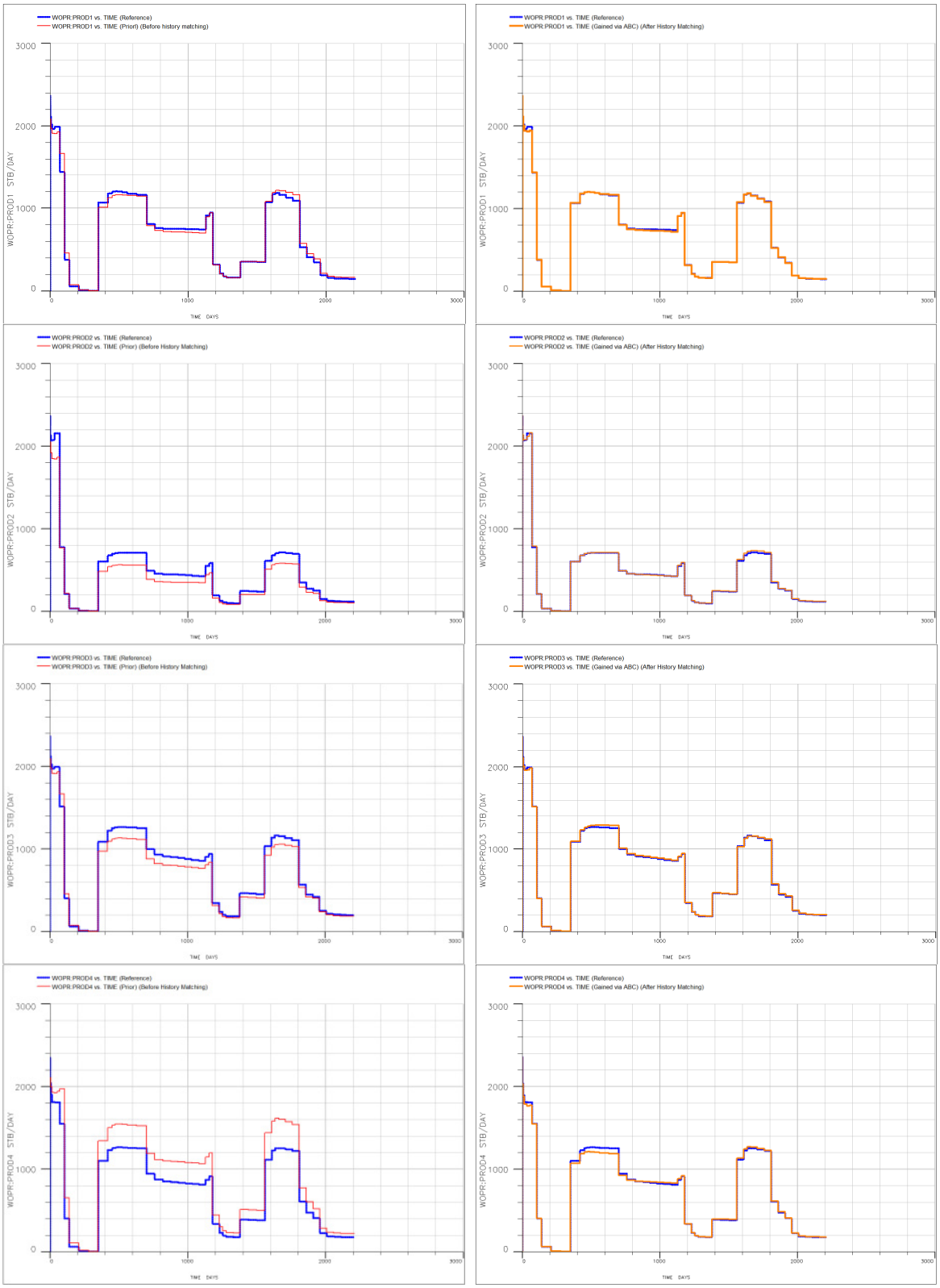


Figure 4.9 Well oil production rate for all producers, before and after history matching

The forecasting error is 1.69% and 0.53% for the first and second scenario respectively. It is fair to state that the achieved history matched model can predict the future performance with an almost 1% ((1.69+0.53)/2) difference. *Q* expresses that the gained porosity distribution using inverse modelling has 17.1% difference from the reference case. These deviations (objective function value, forecasting errors and *Q* value) from the corresponding reference case values are because of the nonlinearity of the problem and measurement noise which led into the optimisation error.

Table 4.3 The outcomes of history matching via the artificial bee colony algorithm and the reference values

	Reference (m_{ref})	**Gained solution via ABC (m_{∞})**
Objective function value ($S(m_{\infty})$) or $S'(r_{\infty})$	375	503
Cumulative produced water for 1st scenario	1.15×10^{6}	1.14×10^{6}
Cumulative produced oil for 1st scenario	7.84×10^{5}	7.99×10^{5}
Difference in the estimation of upper elements for the 1st scenario	-	1.69%
Cumulative produced water for 2nd scenario	2.14×10^{6}	2.13×10^{6}
Cumulative produced oil for 2nd scenario	1.32×10^{6}	1.33×10^{6}
Difference in the estimation of upper elements for the 2nd scenario	-	0.53%
Q	-	17.1%

It should be mentioned that the procedure was executed three times with three different seed numbers, as the algorithm is stochastic and may be affected by the random selections, and the best of these three runs in terms of objective function value are presented. The results of the other two runs can be found in table 4.4. All the three runs delivered almost similar results.

Table 4.4 The outcomes of history matching via the artificial bee colony algorithm using two other seed numbers

	Gained solution via ABC (m_{∞}) seed No.#2	**Gained solution via ABC (m_{∞}) seed No.#3**
Objective function value ($S(m_{\infty})$) or $S'(r_{\infty})$	565	521
Error in the estimation of water and oil recovery for the 1st scenario	1.29%	0.29%
Error in the estimation of water and oil recovery for the 2nd scenario	0.73%	1.49%
Q	20.4%	20.9%

The number of fitness calls to approximate r_{∞} using artificial bee colony algorithm was 20,000, and in each fitness function calls, one ECLIPSE and one SGeMS were executed. The optimisation process took almost 2 days for each seed number execution on a personal computer with the following configurations, CPU Intel i-7-2820QM and 8 GB ram. The optimisation error might be reduced by increasing maximum allowed fitness function calls,

but due to computational limitations, hardware and software, it was not possible to be checked out.

4.2.4. Comparison of the artificial bee colony with three optimisation algorithms

History matching is carried out for the same reservoir model using three other optimisation algorithms to provide a comparison between the artificial bee colony algorithm and conventional algorithms, genetic algorithm (GA), simulated annealing (SA), and Levenberg-Marquette (LM) (refer to chapter 2 for the details of these three algorithms). These algorithms are chosen, because of their diverse capabilities. GA, especially with a scattered (uniform) crossover, is a powerful tool for the exploration of search spaces, and it is a widespread population-based algorithm (Gwiazda, 2007). SA is a powerful tool for the exploitation of search space, especially when the temperature updating is followed by an exponential rule (at high temperature (in initial stage of each interval), the search is more explorative, while at low temperature (a number of step after each interval) the search is more exploitative), and it is a widespread individual-based stochastic algorithm (Rangaiah, 2001). LM is a widespread gradient-based algorithm and can deliver quick convergence; it is useful when the objective function does not have a regularisation term (Tarantola, 2005). Artificial bee colony algorithm balances the exploitation and explorations capabilities (Akay and Karaboga, 2012). GA, SA and LM are conducted by MATLAB (MathWork, 2011b) while the stated options in table 4.5 are used.

The calibration using each of the stochastic optimisers (SA and GA) was executed three times with three different seed numbers, and the mean of three runs is used for the comparison. None of the algorithms was allowed to call the fitness function more than 20,000. This computational limit and three times execution help us in delivering fair

comparison. The same criteria are used to assess the methods, and the results based on the criteria are shown in table 4.6.

Table 4.5 The options of the applied optimisation algorithms

Algorithm	Property	Value
GA	Population size	20
	Crossover operator & its probability	Scattered & p_c=0.8
	Mutation operator & its probability	Uniform with rate 0.11 & p_m=0.2
	Selection operator	Stochastic uniform
	Number of elite	1
	Migration	0
	Penalty factor	0
	Stopping criteria	Number of generation (1000)
SA	Initial point	$[1...1]_{1\times42}$
	Annealing function	Fast annealing
	Reannealing interval	50
	Temperature updating function	Exponential temperature update
	Initial temperature	20
	Acceptance probability function	Simulated annealing acceptance [2]
	Stopping criteria	Number of fitness function calls (20000)
LM	Initial point	$[1...1]_{1\times42}$
	Gradient calculation	Numerically with minimum perturbation 0.1 and maximum perturbation 15
	Stopping criteria	Function tolerance: 1×10^{-10} Maximum function calls: 20000

Table 4.6 Results of history matching using different optimisation algorithms (GA, SA, ABC and LM)

Optimisation algorithm	ABC	GA	SA	LM
Objective function value ($S(m_\infty)$) or $S'(r_\infty)$	530	613	1373	6459
Error in the estimation of water and oil recovery for the 1st scenario	1.09%	0.97%	2.99%	4.03%
Error in the estimation of water and oil recovery for the 2nd scenario	0.92%	2.01%	2.93%	3.44%
Q	19.5%	23.9%	19.1%	15.5%

The objective function value was reduced to 530 via ABC, while this value for the other two stochastic optimisers, GA and SA, was 602 and 1373 respectively. The classical optimiser (Levenberg-Marquette) could not reduce the value to less than 6,459 which indicates that it converged to a local minimum. For the pilot point approach, it is not feasible to provide a proper initial guess when their locations are assumed variable. This can be a reason why the classical method got stuck in a local minimum. It should be mentioned that if in the pilot point reparameterisation design, only the value of the pilot

[2] This function is used to provide a rule to accept or reject the achieved point in current iteration, the applied function is as follows: if the corresponding fitness value at the current point is less than the fitness value at the previous one, the current point is memorised, otherwise, the current point is selected at random according to the following equation:

$$\frac{1}{1+e^{\frac{\Delta}{\max(T)}}}$$

in which, $\Delta = S(r_i) - S(r_{i-1}), i: iteration\ number$, T is the current temperature. Thus, if the difference increases (Δ↑), the chance of selection will decrease.

point are considered as matching parameters (it means that the location of the pilot points are fixed and predefined), it will be possible to provide an initial guess for the classical optimiser. But, as mentioned in the literature review chapter, fixing the location of pilot point causes overshooting, that is why in the design, the location of pilot points are also assumed as decision variables. In these cases where decision variables are dependent, stochastic optimisers perform better. The objective function values cannot be the only means to assess the gained history matched models. The other three values (the forecasting errors and Q) are good measures to distinguish between the three methods. As it can be seen, the values gained by ABC are lower than the values achieved by the other three methods. The future forecasting errors are 1%, 1.5%, 3% and 3.75% via the obtained history matched model using ABC, GA, SA and LM, respectively. The Q values gained by all the optimisers are in an acceptable range however, the best value was achieved by LM algorithm. According to the stated values, overall, it can be concluded that ABC works better than the other three algorithms in this case study while pilot point is used as reparameterisation.

Reparameterisation transforms the M space to an R space. The R space is a function of the selected reparameterisation operator. Thus, it is fair to expect that the landscape shape of $S'(r)$ is altered by changing the reparameterisation operator (h), which results in the creation of a new objective function and consequently a new optimisation problem. In the following section, this effect is investigated. The created new objective functions (problems) that have different landscape are useful means to evaluate the performance of the ABC algorithm, one more time. As mentioned before, one of the features of a successful optimisation algorithm is that its performance is not significantly influenced by the objective function specifications and is capable of approximating the global optimum point in different problems.

4.3. Effects of reparameterisation operators on the landscape shape

In this section, the same reservoir model is reparameterised using different operators: zonation, spectral decomposition of prior covariance matrix, bicubic spline and pilot point. For each operator, history matching is carried out via the four optimisation algorithms, and the outcomes are reviewed. Also, for each reparameterisation operator, rough estimation regarding the shape of landscapes is also presented. As stated before, it is impossible to draw or study profoundly the landscape shape of the history matching problems, because of i- the high-dimensionality of the system, and ii- the very large number of possible scenarios (billions), even with a coarse discretisation. In this study, to present rough estimation, the dimension of the reparameterised space (R) is reduced to two, and the objective function values are calculated over a discretised space for the two variables.

To draw the 2-D landscape shape of the objective function ($S'(r)$) for each of the operators, first the global optimum point of the $S'(r)$ is approximated, then N_r-2 out of N_r elements of *r* are fixed to the corresponding approximated global point and the other two elements are assumed variable, after that, the $S'(r)$ values are calculated for the discretised space of these two elements. It should be mentioned that the two-dimensional landscape shapes deliver very rough estimation about the full-dimensional (N_r dimensional) landscape, and they may not be very reliable. The only conclusion which can be logically reached is that if the two-dimensional landscape shape is noisy (have multiple minima), the full-dimensional problem is expected to be noisy. But, if the 2-D landscape shape is smooth or even quadratic, any conclusion about its full-dimensional problem cannot be made.

Another way to assess the landscape shape is the outcome of the classical optimiser. If the landscape is close to quadratic and unimodal, the classical optimiser should be able to converge to the global optimum point, but if it is converged to a point which is not the global minimum, it can be stated that the landscape has likely multiple minima. It should

be mentioned that in this section, the comparison of the efficiency of the reparameterisation operators on the quality of history matching does not fall into the scope of this study; the main emphasis is on the influence of reparameterisation operator on the shape of landscape.

4.3.1. Landscape shape for the zonation operator

The reservoir is divided into 36 homogenous zones (12 zones in each layer), and the porosity in each zone forms the variables (r). The number of variables is reduced from 3648 to 36. The procedure of zonation reparameterisation was explained in the literature review chapter. In the calibration step, the global optimum point of $S'(r)$ should be approximated, while this time the reparameterisation operator (h) is the zonation method, and r consists of the elements of reparameterised model by the zonation. The same optimisation algorithms with the same options are used to find the global minimum. The results of the optimisation methods are shown in table 4.7.

Table 4.7 History matching results using the four optimisation algorithms along with the zonation reparameterisation

Optimisation algorithm	**ABC**	**GA**	**SA**	**LM**
Objective function value ($S(m_\infty)$) or $S'(r_\infty)$	2549	4240	818	2081

Among the four optimisers, SA delivered the best results. LM converged to a local minimum, and the other two stochastic optimisers could not decrease the objective function value as much as SA. In terms of the objective function value, ABC performed better than GA, and the quality of achieved model via ABC is comparable with the gained model via LM.

The solution of SA is used as the global optimum point to draw the landscape shape. 34 out of 36 elements of the approximated global optimum point by SA are fixed and equal to the corresponding elements of the point. Then, via a discretisation for the two variables, $S'(r)$ is drawn, figure 4.10. The landscape shape is smooth. Hence, the landscape shape of $S'(r)$

in the full-dimensional system (N_r) cannot be estimated by the 2-D landscape shape. But, as the LM algorithm did not converge to the global optimum point, it can be said that the landscape shape of $S'(r)$ with a zonation reparameterisation likely has at least two minimums.

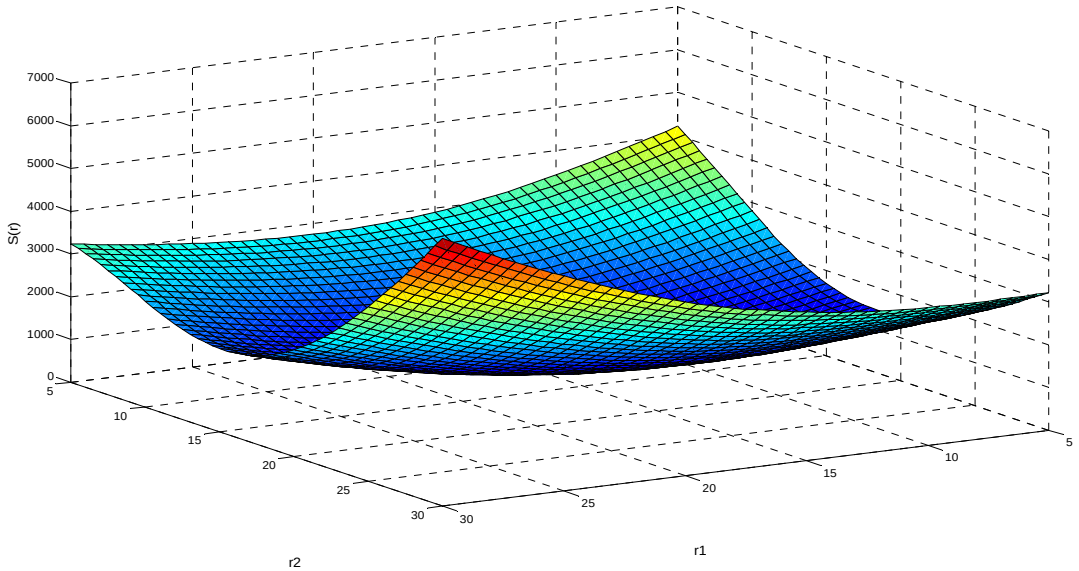


Figure 4.10 The landscape shape for the zonation reparameterisation

4.3.2. Landscape shape for the spline operator

The same procedure is used for the spline reparameterisation operator. In this reparameterisation method, the porosity in a number of gridblocks (14) are assumed as variables in each layer and the reset of gridblocks are interpolated using the cubic spline technique[3] based on the 42 variables and the 27 gridblocks where the wells are located at. The property in each layer is interpolated separately. The number of variables is reduced from 3648 to 42. In the calibration step, the global optimum point of $S'(r)$ should be approximated, while this time the reparameterisation operator (h) is the spline method, and r consists of the elements of reparameterised model by spline. The same optimisation algorithms with the same options are used to find the global minimum. The results of the optimisation methods are shown in table 4.8.

[3] For the spline interpolation, the "griddata" function of MATLAB is used.

Table 4.8 History matching results using the four optimisation algorithms along with the spline reparameterisation

Optimisation algorithm	ABC	GA	SA	LM
Objective function value ($S(\mathbf{m}_\infty)$) or $S'(\mathbf{r}_\infty)$	13,447	14,238	10,753	17,421

Among the four optimisers, SA delivered the best results in terms of objective function value. LM converged to a local minimum, and the other two stochastic optimisers could not reduce the objective function value as much as SA. In terms of the objective function value, it is fair to rank ABC as the second successful optimiser in this problem after SA.

The solution of SA is used as the global optimum point to draw the landscape shape. 40 out of 42 elements of the approximated global optimum point by SA are fixed and equal to the corresponding elements of the point. Then, via a discretisation for the other two variables, $S'(\mathrm{r})$ is drawn, figure 4.11. The landscape shape is smooth. Hence, the landscape shape of $S'(\mathrm{r})$ in the full-dimensional system (N_r) cannot be estimated by the 2-D landscape shape. But, as the LM algorithm did not converge to the global optimum point, it can be said that the landscape shape of $S'(\mathrm{r})$ with a bicubic spline reparameterisation likely has at least two minimums.

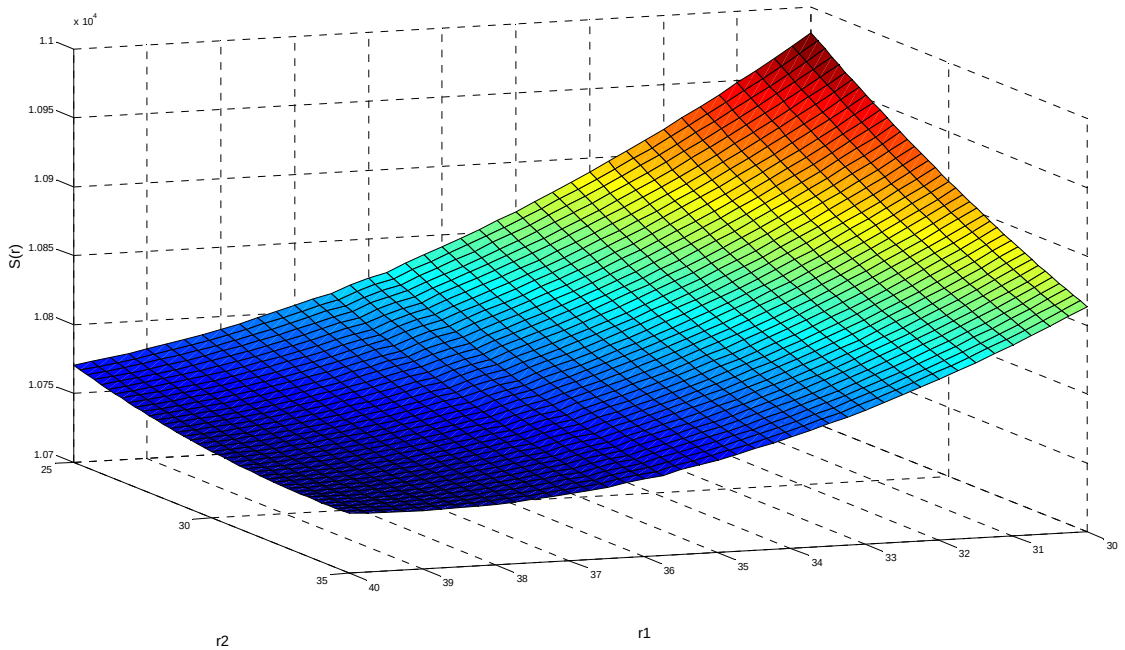


Figure 4.11 The landscape shape for the spline reparameterisation

4.3.3. Landscape shape for the spectral decomposition operator

The same procedure is used for the spectral decomposition reparameterisation operator. In this reparameterisation method, the eigenvalues and eigenvectors prior covariance matrix are calculated and ranked, then first 42 variables are selected. The prior covariance matrix was calculated by analysing a number of generated realisations for the unknown properties using a geostatistical interpolation. The procedure of the spectral decomposition was explained in the literature review chapter. Using this method, the number of variables is reduced from 3648 to 42. In the calibration step the global optimum point of $S'(r)$ should be approximated, while this time the reparameterisation operator (h) is the spectral decomposition method, and r consists of the elements of reparameterised model by spectral decomposition. The same optimisation algorithms with the same options are used to find the global minimum. The results of the optimisation methods are shown in table 4.9.

Table 4.9 History matching results using the four optimisation algorithms along with the spectral decomposition reparameterisation

Optimisation algorithm	**ABC**	**GA**	**SA**	**LM**
Objective function value ($S(m_\infty)$) or $S'(r_\infty)$	7024	8523	357,970	435,790

Among the four optimisers, ABC delivered the best results in terms of the objective function value. LM converged to a local minimum which has a very high objective function value, and the other two stochastic optimisers could not reduce the objective function value as much as ABC; the outcome of GA is much closer to ABC's outcome, but SA's outcome is not in an acceptable range (two orders of magnitude larger than the ABC's result). It is fair to rank ABC as the successful optimiser in this problem.

The solution of ABC is used as the global optimum point to draw the landscape shape. 40 out of 42 elements of the approximated global optimum point by ABC are fixed and equal to the corresponding elements of the point. Then, via a discretisation for the other two variables, $S'(r)$ is drawn, figure 4.12.

As it can be seen, the 2-D landscape shape is noisy and has several minima. Hence, it is fair to expect that the landscape shape of the full-dimensional problem of $S'(r)$ is also rugged. That is why the classical optimiser could not reduce the objective function value extensively, as it trapped into a local minimum.

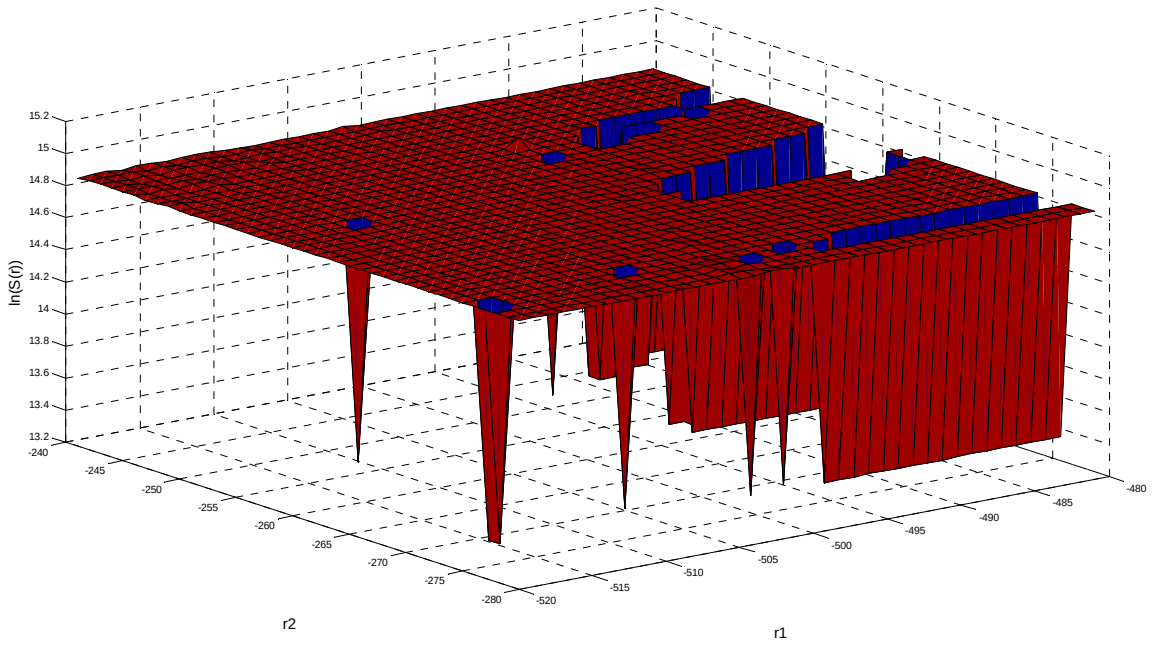


Figure 4.12 The landscape shape for the spectral decomposition reparameterisation

4.3.4. Landscape shape for the pilot point operator

The results of optimisation with a pilot point reparameterisation were shown in the previous section. Among the four optimisers, ABC delivered the best results in terms of objective function value. LM converged to a local minimum, and the other two stochastic optimisers could not reduce the objective function value as much as ABC; the outcome of GA is much closer to ABC's outcome. In terms of the objective function value, it is fair to rank ABC as the successful optimiser in this problem.

The solution of ABC is used as the global optimum point to draw the landscape shape. 40 out of 42 elements of the approximated global optimum point by ABC are fixed and equal to the corresponding elements of the point. Then, via a discretisation for the other two variables, $S'(r)$ is drawn, figure 4.13.

As it can be seen, the 2-D landscape shape is noisy and has several minima. Hence, it is fair to expect that the landscape shape of the full-dimensional problem of $S'(r)$ is also rugged. That is why the classical optimiser could not reduce the objective function value extensively, as it trapped into a local minimum.

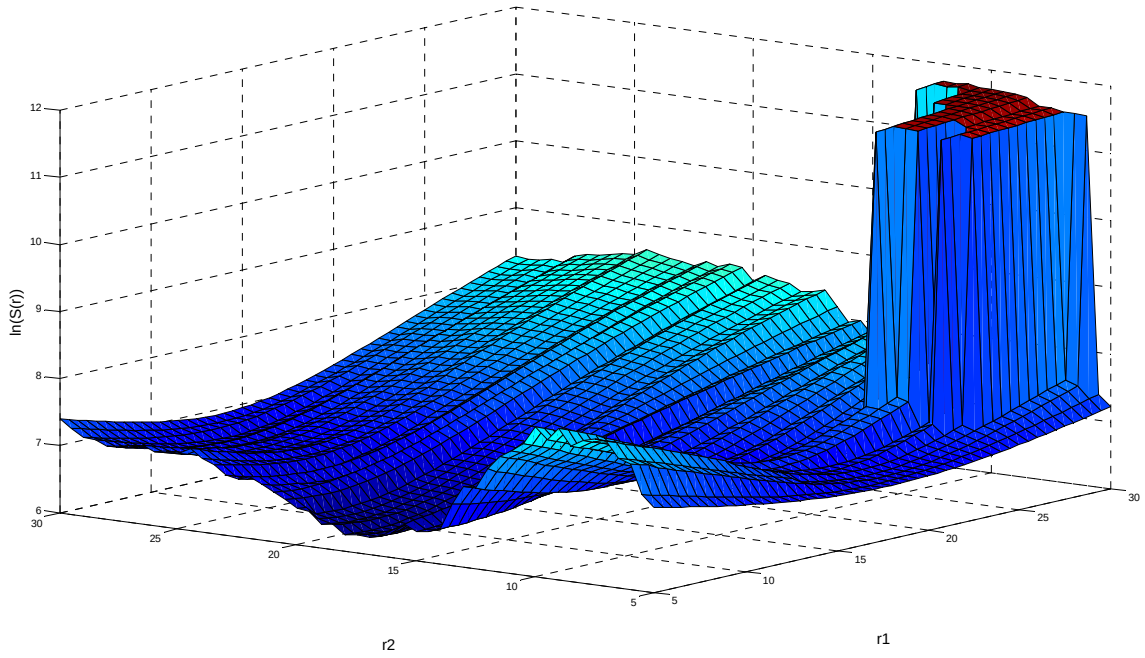


Figure 4.13 The landscape shape for the pilot point reparameterisation

The observations lead us to the following conclusions: 1- the shape of landscape is altered by changing the reparameterisation operator, and 2- the performance of artificial bee colony algorithm is not inflected extensively by the shape of landscape.

4.4. Discussion

The outcomes indicated that the artificial bee colony algorithm performs appropriately as a conductor of calibration in history matching problems. It delivered better in comparison with the three methods, genetic algorithm, simulated annealing and Levenberg-Marquette in the case study which was reparameterised accurately with a pilot point technique. These three algorithms have different searching abilities, the genetic algorithm with a scattered crossover is more explorative, the simulated annealing with an exponential temperature updating is more exploitative, and the Levenberg-Marquette has a gradient-based search. It

was not feasible to compare the ABC with more than a few algorithms, due to the computational limitations. The applied algorithms are the standard versions, and the applied options for these four algorithms are almost default options. A sensitivity analysis was carried out to modify the options, especially for GA and SA, but it cannot be stated that these are the most efficient ones, as the comprehensive sensitivity analysis is computationally expensive and almost unfeasible.

In this chapter, also, the effect of reparameterisation on the landscape shape of objective function was studied. Four widespread reparameterisation operators were studied, zonation, spline, spectral decomposition and pilot point method. 2-D landscape shape of them and the outcomes of the optimisation methods gave us rough estimations regarding the shape of landscape: 1- all the four problems likely had more than one minimum point, as the gradient-based algorithm converged to a local minimum, 2- the corresponding landscape shape of the pilot point and spectral decomposition method were rugged, as the 2-D landscape shapes were rugged, and 3- the landscape shape of spectral was highly noisy and rugged, as the simulated annealing converged to a false optimum point. According to these observations, as it was expected, it can be concluded that changing the reparameterisation operator alters the landscape shape and forms dissimilar objective functions. It should be mentioned that not only the reparameterisation operator has influence on the shape of landscape, but also the way of its implementation has effects on the shape of landscape; for instance, if in the spline method, the location of the gridblocks along with their values are assumed variable, a different landscape would be created. The locations of gridblocks in spline method cannot be assumed as variables, like pilot point approach, as this method is weak in extrapolation, and the gridblocks should be distributed in a manner in which mostly interpolations are required.

In order to endorse the effectiveness of the artificial bee colony algorithm, history matching was executed using it for the new created problems (different reparameterised

models) and compared with the outcomes of the other three optimisers for the new problems. The Levenberg-Marquette algorithm was trapped into local minima in all the problems, that is due to its gradient-based search. The genetic algorithm delivered good results in all, but in none of the problems, it was the best; it is due to its week exploitation ability. The simulated annealing was the best where the zonation and spline were used for reparameterisation, but it could not obtain good results for the other two problems, especially in the spectral decomposition problems; its poor results is due to its week exploration abilities. The artificial bee colony algorithm was the best method where pilot point and spectral decomposition were used for the reparameterisation, and in the other two problems, it delivered satisfactory results. The reason of its success may lay in its capability in balancing its searching abilities, exploration and exploitation.

To sum up, if the number of variables is low or the reparameterisation can be implemented accurately (like, pilot point in this study), stochastic optimisers which have a balance between exploration and exploitation are likely capable of approximating the best history matched model. The major drawback of ABC is the high CPU time which is a result of the slow convergence speed. Gradient-based algorithms needs less computation in comparison with stochastic algorithms, and they can be an appropriate choice when a proper initial guess is available. LM was not successful in our four problems, but it cannot be concluded that it always traps into a local minimum. Its performance significantly depends on the objective function specifications.

In the objective function formulation, a prior knowledge was not used, since the location of pilot points were assumed variable and in the other problems it was assumed there is no further information about the variables. As mentioned in the literature review, without a regularisation term in the objective function when a pilot point technique is used for reparameterisation, overshooting occurs, but in our problem, this issue did not happen,

since the location of pilot points were assumed variables and a bounded artificial bee colony was used.

The conclusive remarks of this chapter are as following:

- The artificial bee colony algorithm can deliver suitable results in history matching problems.
- The artificial bee colony algorithm performs better than the three conventional algorithms, and its performance is not significantly influenced by the shape of landscape. The success presumptively relies on having a balance between exploration and exploitation search.

Chapter 5

5. Assessment of different model management techniques in history matching problems

This chapter addresses the computational challenges of history matching problems. In the previous chapter, it was shown that stochastic optimisers that have a balance between exploration and exploitation have a high-chance of being able to approximate the high-quality history matched model. One of the major drawbacks of stochastic optimisation algorithms, generally, is their high computational time. Fitness approximation (proxy-modelling) approaches can be used to reduce computational expense. To prevent misleading of the calibration by the conventional proxy-modelling methods (uncontrolled), a model management (evolution-control) technique is proposed to be included. In this chapter, the practicality of the controlled fitness approximation approaches is studied. Also, a specific adaptive model management technique is designed. The questions, the author wishes to answer in this chapter, are:

a- Is the conventional approach of fitness approximation in history matching effectual?

b- Do the model management techniques help to increase the quality of fitness approximation?

c- Which model management technique is more efficient? Is the designed model management technique superior to the others?

This chapter begins with an introduction to the fitness approximation, and it is followed by an explanation to the uncontrolled and controlled (proposed) approaches. In the methodology section, the designed adaptive technique is described. For this study, a genetic algorithm is developed; the algorithm and its corresponding results for different benchmarking functions are presented in the GA section. In the results section, history matching using different fitness approximation approaches on PUNQ-S3 model is shown. Finally, the advantages and drawbacks of different approaches are discussed. This chapter is a modified and adjusted version of one of our published papers in a peer-reviewed journal, "SAYYAFZADEH, M. & HAGHIGHI, M. 2013. Assessment of different model management techniques in history matching problems for reservoir modelling, in APPEA Journal, 2013; 53".

5.1. Fitness approximation

Recently, an escalation in the interests of the applications of fitness approximation techniques has been observed. In fitness approximation methods, the original function (OF) which is computationally costly to be evaluated is substituted by a low-cost function known as approximation function (AF) or proxy. The proxy should be able to represent the specifications of the original function, especially having the same global optimum point. The latest advances in fitness approximation for history matching problem were presented in the literature review section. The dissimilarities of applied proxy-modelling methods in history matching methods are in the type of sampling strategy or the type of proxy model, for example artificial neural network (Cullick et al., 2006), polynomial (Li and Friedmann, 2005) and kriging (Osterloh, 2008) as proxy model, and space filling designs (Latin Hypercube Sampling) and partitioning domain (Li and Friedmann, 2005) as an experimental design (sampling strategy) .

In the following figures, 5.1 to 5.4, the results of fitness approximation on Ackley function are shown. In the first two figures, the one-dimensional function is tried to be approximated via a neural network using 20 and 200 samples. The samples are selected from the domain of the original function using a Latin Hypercube Sampling (LHS) strategy. The dot points show the samples taken from the original function. As it can be seen, the first approximation function has a low quality which is mainly due to the small number of samples, but the second approximation function can model the Ackley function appropriately. The difference between the qualities of the functions demonstrates the importance of the number of samples. If the function trained by only 20 samples is chosen for an optimisation process, a wrong optimum point (in this case, x=0.25 while the true solution is x=0) is obtained; while it can be approximated properly in the second trained function.

Figure 5.3 reveals the approximation for a two-dimensional Ackley function. The approximation function cannot model the two-dimensional Ackley function accurately via 200 samples. The global optimum using of the approximation function is [-0.32,0.13] which is not very close to the global optimum of the original function. The number of samples is increased to 20,000, and the approximation function is drawn (figure 5.4); this time, the global optimum point of the approximation function ([-0.04,0.02]) is closer to the actual optimum point. It is not adequately close to the actual global point, however a large number of samples are used. The required number of samples is expected to be significantly increased by the increase of the dimension of optimisation problem.

The procedure of the uncontrolled approach is similar to the aforementioned numerical examples. The uncontrolled fitness approximation approach has the potential of misleading of the optimisation to wrong optimum points (Jin, 2005, Yaochu et al., 2001). It is because of the limited number of samples and the multi-dimensionality of the search space.

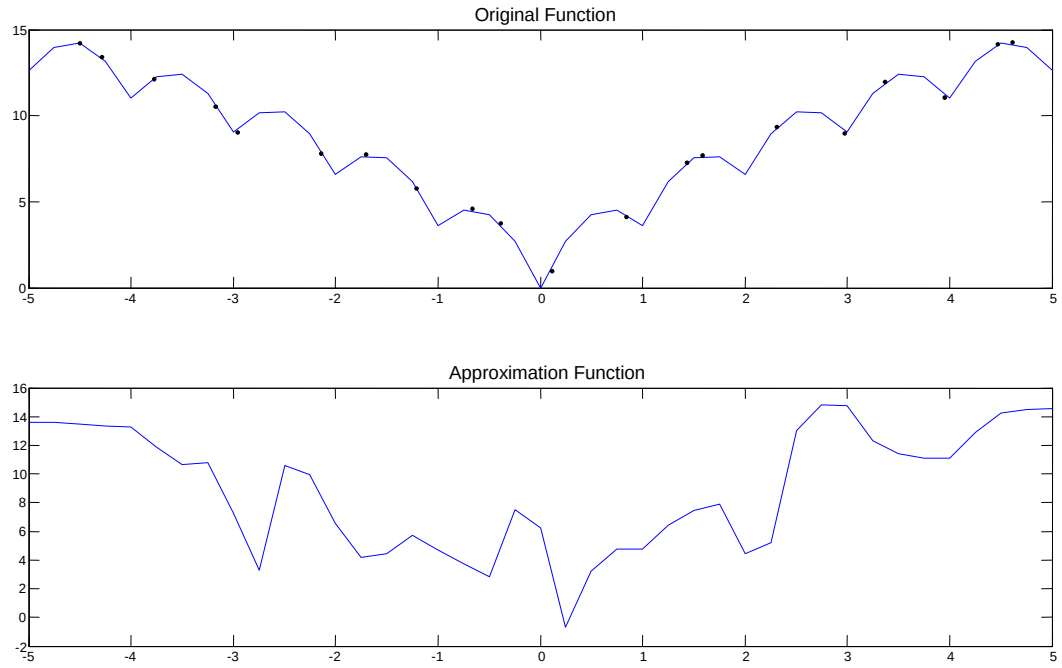


Figure 5.1 Approximation of the one-dimensional Ackley function via a neural network with 20 samples

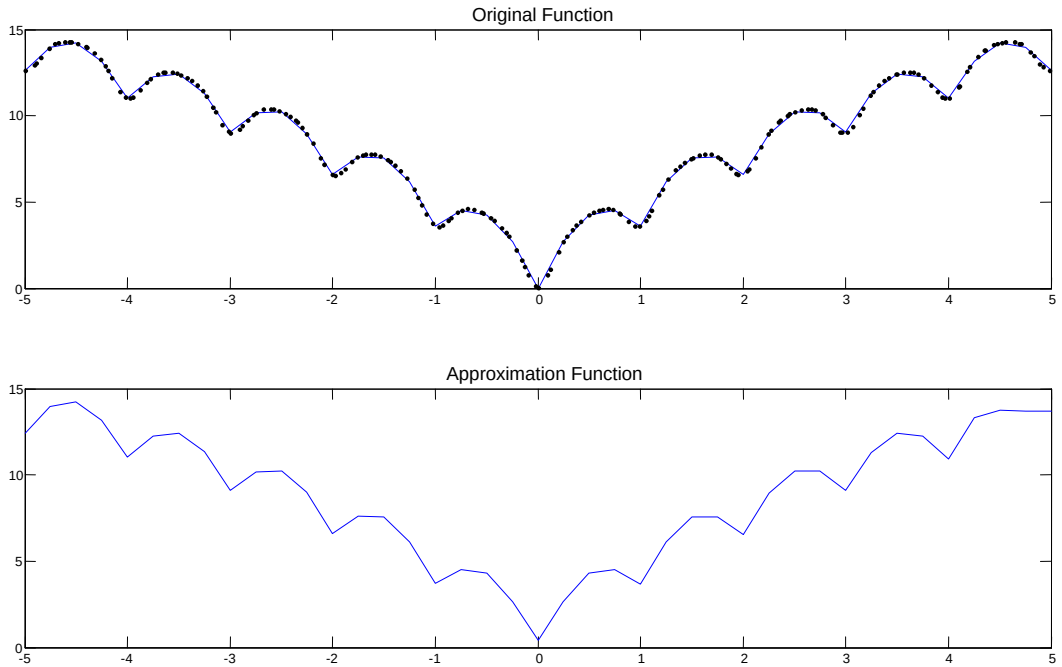


Figure 5.2 Approximation of the one-dimensional Ackley function via a neural network with 200 samples

To prevail over this concern, it is recommended to make use of the original function in conjunction with the approximation function during the optimisation. In order to employ the original function efficiently, an evolution-control technique should be employed. There are different type of evolution-control techniques: individual-based, population-based and adaptive (Jin, 2005). By employing each of these techniques, a controlled fitness

approximation is assembled which benefits from online learning. In the controlled fitness approximation methods, the main target is the approximation of the global optimum point which can be gained by emphasising on the accuracy of the approximation of the global point rather than focusing on modelling the entire model (as it can be see, in the figure 5.4, the conventional approach models the Ackley function properly, but in terms of global estimation, it is not very reliable).

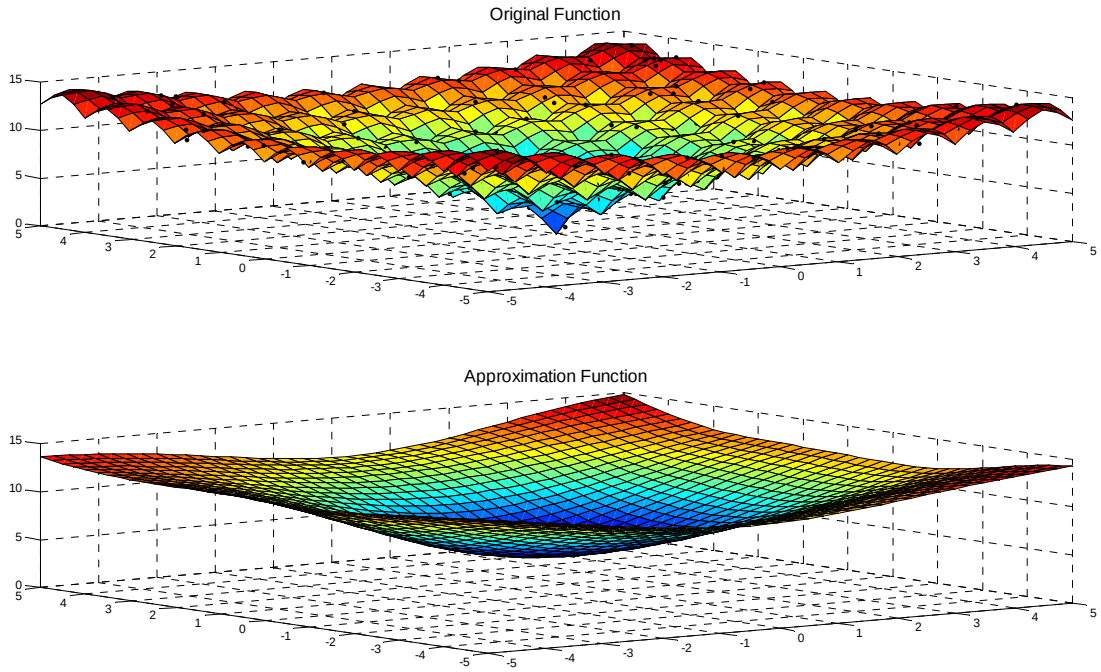


Figure 5.3 Approximation of the two-dimensional Ackley function via a neural network with 200 samples

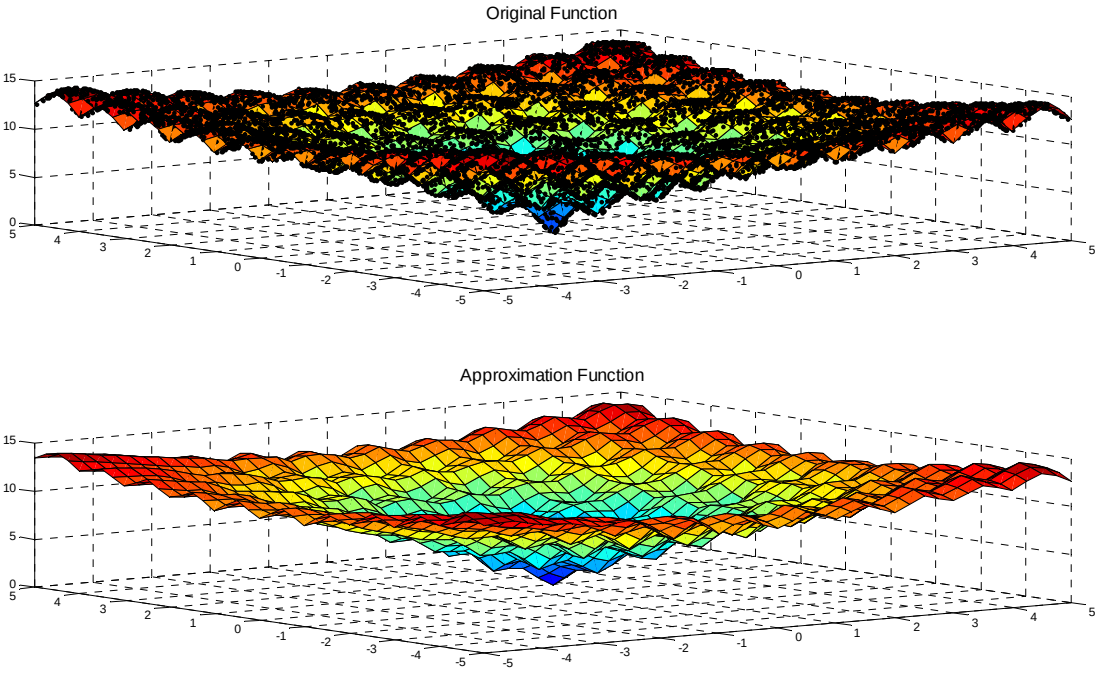


Figure 5.4 Approximation of the two-dimensional Ackley function via a neural network with 20000 samples

The importance of reduction of computation in history matching problems encouraged us to investigate the latest progress of fitness approximation in petroleum engineering discipline and also endeavour to design specific methods. In this study, the main emphasis is on the applications of three controlled approaches, and for the first time, these three approaches are applied and assessed in history matching. Also, a specific adaptive evolution-control technique based on heuristic fuzzy rules is designed.

In this study, a Genetic Algorithm (GA) with a specific crossover is designed and used as an optimiser (see section 5.3). Seeing as good results were articulated using the Artificial Neural Network (ANN) in several studies (Ramgulam et al., 2007, Sampaio et al., 2009, Silva et al., 2006), a feed-forward artificial neural network with a sigmoid hidden neurones is applied as a proxy model ($S(m) = \sum_{p=1}^{P} v_p f\left(\left(\sum_{i=1}^{N_m} w_{ir}{}^1 m_i\right)\right), f(x) = \frac{1}{1+e^{-ax}}$) (see chapter two for an introduction to the neural network). In the backpropagation, to estimate the variables of the network, the misfit is minimised by a Levenberg-Marquette algorithm. Latin Hypercube Sampling (LHS) is implemented for sampling, since LHS is known as an effectual technique (Zubarev, 2009, Osterloh, 2008). This sampling strategy was introduced by McKay et al. 1979. Using this technique, with a lower number of realisations, the space will be sampled. Assume there are N_m variables and n samples are needed. Two matrixes (P and R) are required to generate an $n \times N_m$ matrix of the samples. R is an $n \times N_m$ matrix in which each element is random number between [0-1]. P is an $n \times N_m$ matrix in which each column is a random permutation of 1 to n. The elements of the sampling matrix (V) are generated according to the following formula (Olsson and Sandberg, 2002):

$$V_{ij} = F^{-1}\left(\frac{P_{ij} - R_{ij}}{n}\right) \quad (1)$$

Where, F is the inverse of the cumulative distribution function. In this study, the probability is uniform, as we seek to select samples from all the space.

Based on the mentioned methods, the fundamentals of a framework are provided in MATLAB.

5.2. Methodology

In this section, the four different approaches of fitness approximation, a- uncontrolled fitness approximation approach (UFA) (the conventional approach in petroleum discipline), b-fitness approximation with an individual-based evolution-control approach (FAIBEC) (a random-strategy), c-fitness approximation with a population-based evolution-control approach (FAPBEC) (a random strategy), d-fitness approximation with an adaptive evolution-control approach (FAAEC), in general. Then, the designed adaptive evolution-control approach is described.

5.2.1. Uncontrolled fitness approximation approach (UFA)

The uncontrolled fitness approximation approach consists of the following steps: 1- a set of samples are selected using the LHS method (lhsdesign function carries out this task), and the set of samples are evaluated by the OF; 2- then, a neural network is trained by the set of samples with their corresponding fitness values, and then validated, (this step is carried out by the fitnet function (MathWork, 2011c)), 3- afterward, the global optimum is sought by the GA, in this process the AF (proxy) is only used to evaluate all the chromosomes (individuals) in each generation; 4- the achieved solution, after meeting the stopping criterion, is used as the optimum point (history matched model). The procedure (pseudo-code) is shown below. In this approach, the OF is utilised only in the initial stage for evaluating the samples, and it is not being used during the procedure of optimisation. Stopping criterion is the number of generations (Gen_No.Limit). One of the advantages of this approach is that computational costs are considerably reduced in comparison with the regular approach of history matching (without proxy-modelling) in which the OF is only

being used. Another benefit of using this approach over the other three fitness approximation approaches and also over the regular approach of history matching is that a lose coupling between the reservoir simulator and optimisation algorithm would be enough, as the reservoir simulator is not needed during the optimisation.

```
% Select a set of samples using LHS
S = LB+(UB-LB)×lhsdesign(NoSamples,NoVariables)
for i=1:NoSamples
    fitness (i) = OF(S(i,:));
end
% Train ANN including validation of ANN by 15% samples
Network = ANN(S,fitness);
% initial population generation
Pop = LB+(UB-LB)×rand([PopSize NoVar]);
% Evaluate the Pop using the original function
for i=1:PopSize
    Fit(i)=Network(Pop(i,:));
end
% optimization starts
Gen_No=0;
while Gen_No<Gen_No.Limit
    Gen_No=Gen_No+1;
    NewPop = GAoperators(Pop, Fit);
    Pop = NewPop;
    for i=1:PopSize
        Fit(i) = Network(Pop(i,:));
    end
    [MinFit Index] = min(Fit);
    Global = Pop(Index,:);
    FitnessGlobal = MinFit;
end
```

5.2.2. Fitness approximation with an individual-based evolution-control approach (FAIBEC)

The steps of this approach are similar to the UFA approach and the change is in the third step which is as follows: the global optimum is searched by the GA, in this process the OF and AF are both used to evaluate the new chromosomes. In this approach, the OF is utilised in the initial stage for evaluating the initial samples, and also it is being used in the process of optimisation by evaluating some of the chromosomes of each population in every generations. The rests of chromosomes are evaluated by the AF. A probability is used to specify the number of chromosomes needed to be evaluated by OF. This probability is signed by p_{IBEC} and recommended to be 50% (Yaochu et al., 2001). The chromosomes required to be evaluated by the OF are selected randomly in each population. In each specific frequent (SF) of generation number (evolution-control cycle), the ANN is

retrained by the old and new samples, and this process supplies an online learning which can increase the quality of trained proxy. The new samples are the chromosomes which are evaluated by the OF during the optimisation. The procedure (pseudo-code) is shown below. In this approach, there are two stopping criteria: 1- the number of generations and 2- the number of the OF evaluations (OFEL). Despite the computational cost reduction, the main advantage of this approach over the first one is that by this approach it is promising to be capable of redirecting the optimisation direction to the right direction, and consequently it can deliver high-quality optimum points. The main difficulty of this approach is in assigning a proper predefined p_{IBEC}. Assigning an inappropriate probability causes inefficiency of the approach. The other drawback is that parallel processing cannot be applied proficiently in this approach.

```
% Select a set of samples using LHS
S = LB+(UB-LB)×lhsdesign(NoSamples,NoVariables)
for i=1:NoSamples
    fitness (i) = OF(S (i,:));
end
% Train ANN including validation of ANN by 15% of the samples
Network=ANN(S,fitness);
% initial population generation
Pop=LB+(UB-LB)×rand([PopSize NoVar]);
% Evaluate the Pop using the original function
for i=1:PopSize
    Fit(i)=Network(Pop(i,:));
end
% optimization starts
Gen_No=0;
j=NoSamples;
pIBEC = 0.5;
while (Gen_No<Gen_No.Limit) or  (j <=OFEL)
    Gen_No= Gen_No+1;
    NewPop=GAoperators (Pop, Fit);
    Pop=NewPop;
    for i=1:PopSize
        R=rand; % random number between 0,1
            if R<= pIBEC
                Fit(i)= OF(Pop(i,:));
                j=j+1;
                S(end+1,:)=Pop(i,:);
                fitness(end+1)=Fit(i);
            else
                Fit(i)=Network(Pop(i,:));
            end
    end
    if Gen_No==SF
        Network=ANN(S,fitness);
    end
    [MinFit Index]=min(Fit);
    Global=Pop(Index,:);
    FitnessGlobal=MinFit;
end
```

5.2.3. Fitness approximation with a population-based evolution-control approach (FAPBEC)

The steps of this approach are similar to UFA approach and the only change is in the third step which is as follows: the global optimum is looked for by the GA, in this process the OF and AF are both used to evaluate all the new chromosomes. In this approach, the OF is utilised in the initial stage for evaluating the initial samples and also it is being used in the process of optimisation by evaluating the whole population (all the individuals of the population) in some of the generations. The populations in the rests of generations are evaluated by the AF. A probability is used to specify the number of populations needed to be evaluated by OF in each evolution-control cycle. This probability is signed by p_{PBEC} and recommended to be 50% (Yaochu et al., 2001). The populations required to be evaluated by the OF are selected randomly. In each specific frequent (SF) of generation numbers (evolution-control cycle), the neural network is retrained by the old and new samples, and this process supplies online learning as well. The procedure (pseudo-code) is shown below. In this approach, the stopping criteria are similar to the second approach. Despite the computational cost reduction, the main advantage of this approach over the UFA is that it is promising to be capable of redirecting the optimisation direction to the right direction, and consequently it can deliver high-quality optimum points. An advantage of this approach over the previous approach is that paralleling process can be carried out efficiently. The main difficulty is that a proper predefined p_{PBEC} cannot be determined, and it may cause inefficiency of the approach and oscillations in the optimisation process.

```
% Select a set of samples using LHS
S = LB+(UB-LB)×lhsdesign(NoSamples,NoVariables)
for i=1:NoSamples
    fitness (i) = OF(S (i,:));
end
% Train ANN including validation of ANN by 15% of the samples
Network=ANN(S,fitness);
% initial population generation
Pop=LB+(UB-LB)×rand([PopSize NoVar]);
% Evaluate the Pop using the original function
for i=1:PopSize
    Fit(i)=Network(Pop(i,:));
```

```
end
% optimization starts
Gen_No=0;
j=NoSamples;
pPBEC=0.5;
while (Gen_No<Gen_No.Limit) or  (j <=OFEL)
    Gen_No= Gen_No+1;
    NewPop=GAoperators (Pop, Fit);
    Pop=NewPop;
    R=rand; % random number between 0,1
    if R<= pPBEC
        for i=1:PopSize
            Fit(i)= OF(Pop(i,:));
            j=j+1;
            S(end+1,:)=Pop(i,:);
            fitness(end+1)=Fit(i);
        end
    else
        for i=1:PopSize
            Fit(i)=Network(Pop(i,:));
        end
    end
    if Gen_No==SF
        Network=ANN(S,fitness);
    end
    [MinFit Index]=min(Fit);
    Global=Pop(Index,:);
    FitnessGlobal=MinFit;
end
```

5.2.4. Fitness approximation with an adaptive evolution-control approach (FAAEC)

In second and third approach, the probabilities are predefined and assumed fixed in whole process. Specifying predefined probabilities may cause inefficiency, since a lower probability, in some cases, can deliver similar results, or, in some cases, a higher probability is required to control the optimisation direction, due to low-quality trained proxies. To assign a proper probability, a sensitivity analysis is needed which is time-consuming. In adaptive approaches, it is sought to update the probabilities in each evolution-control cycle, based on the fidelity of proxies, i.e., when an AF provides high-quality results, the probabilities should be reduced, and in contrast, when the AF cannot provide high-quality results, the probabilities should be increased.

In this study, a random-strategy population-based and a best-strategy individual-based technique are merged, while an adaptive probability for the number of populations evaluated by OF is used. The corresponding probability for the number of individuals needed to be evaluated by the OF in each generation is set to 0.25 or 1. It is equal to 1 in

those generations which are under a population-control, and it is equal to 0.25 in the rests of generations. The 25% chromosomes are not selected randomly; 25% of the best chromosomes are selected and reevaluated by the OF, i.e., firstly all of the chromosomes are evaluated by the AF, ranked, and the first 25% are reevaluated. The probability of the number of populations (p_{AEF}) which are needed to be entirely evaluated by the OF varies from 0.3 to 1 adaptively based on fidelity of the AF. The initial p_{AEF} is assumed 100%. Hence, an initial training is not required, and all the chromosomes in first evolution-control cycle are evaluated by the OF. The p_{AEF} will be updated based on the error of AF in each evolution-control cycle. The error of the AF is measured by equation 2 which shows the fidelity of the AF. N is the number of the OF evaluations in each evolution-control cycle. The error is updated in each evolution-control cycle. It was suggested to update the probability based on heuristic fuzzy rules (Yaochu et al., 2001), thus, the following formula is used to update the probability (equation 3). E_{max} is maximum allowed error of the AF and assumed 2.5 (this value (2.5) is picked by carrying out a sensitivity analysis). In each evolution-control cycle, the samples are used to retrain the proxy which provides an online-learning sample. After passing 90% of allowed original function evaluations, the p_{AEC} is changed to 1 to provide accurate exploitation of search space, for the reason that the AF cannot be a good tool for exploitations of search space[4]. Thus, by switching the probability to 1, the OF evaluates all the new chromosomes in last generations. The stopping criteria are similar to the second and third approach. The procedure (pseudo-code) is shown below.

```
% Select a set of samples using LHS
S = LB+(UB-LB)×lhsdesign(NoSamples,NoVariables)

for i=1:NoSamples
    fitness (i) = OF(S (i,:));
end

% initial population generation
Pop=LB+(UB-LB)×rand([PopSize NoVar]);

% Evaluate the Pop using the original function
for i=1:PopSize
```

[4] It is assumed after passing 90% of OF evaluations, the global optimum point is estimated roughly, and in the other 10% evaluations, it is sought to increase the accuracy of the approximation.

```
    Fit(i)=OF(Pop(i,:));
end
S=Pop;
fitness=Fit;
% optimization starts
Gen_No=0;
j= PopSize;
k=0;
p=0.25;
pAEC =1

while (Gen_No< Gen_No.Limit) or  (j <=OFEL)
    Gen_No= Gen_No+1;
    NewPop=GAoperators (Pop, Fit);
    Pop=NewPop;
    R=rand; % random number between 0,1
    if R<= pAEC
        for i=1:PopSize
            Fit(i)= OF(Pop(i,:));
            j=j+1;
            S(end+1,:)=Pop(i,:);
            fitness(end+1)=Fit(i);
            k=k+1;
            E(k)=(Fit(i)-Network(Pop(i,:)))^2;
        end
    else
        for i=1:PopSize
            Fit(i)=Network(Pop(i,:));
        end
        [temp Index]=rank(Fit);
        for i=1:ceil(p×PopSize)
            Fit(Index(i))=OF(Pop(Index(i),:));
            j=j+1;
            S(end+1,:)=Pop(i,:);
            fitness(end+1)=Fit(i);
            k=k+1;
        end
    end
    if mod(Gen_No,SF)==0
        Network=ANN(S,fitness);
        Ec=1/k*(sum(E));
        pAEC= pAECmin+Ec/Emax*(pAECmax- pAECmin);
        k=0;
    end
    if j>9000
       pAEC=1;
    end
    [MinFit Index]=min(Fit);
    Global=Pop(Index,:);
    FitnessGlobal=MinFit;
end
```

$$E(c) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(OF(x_i) - AF(x_i)\right)^2} \tag{2}$$

$$p_{AEC} = p_{AEC\,min} + \frac{E(c)}{E_{max}}\left(p_{AEC\,max} - p_{AEC\,min}\right) = \frac{E(c)}{2.5}(1 - 0.3) \tag{3}$$

By this adaptive evolution-control, there is always an evolution-control: in some generations there is an individual-based, and in the rests of the generations there is a population-based. This approach is expected to be more efficient than the other three approaches, since the probability is changed based on the fidelity of the AF and also the risk of misdirection can be reduced. Figure 5.5 shows the procedure of the adaptive

evolution control technique in one evolution control cylce. As it can been seen, the individuals of each population will be evaluated all by the original function or approximation function, randomly. After each evolution control cycle, the proxy will be retrained and the probability of evolution-control will be updated.

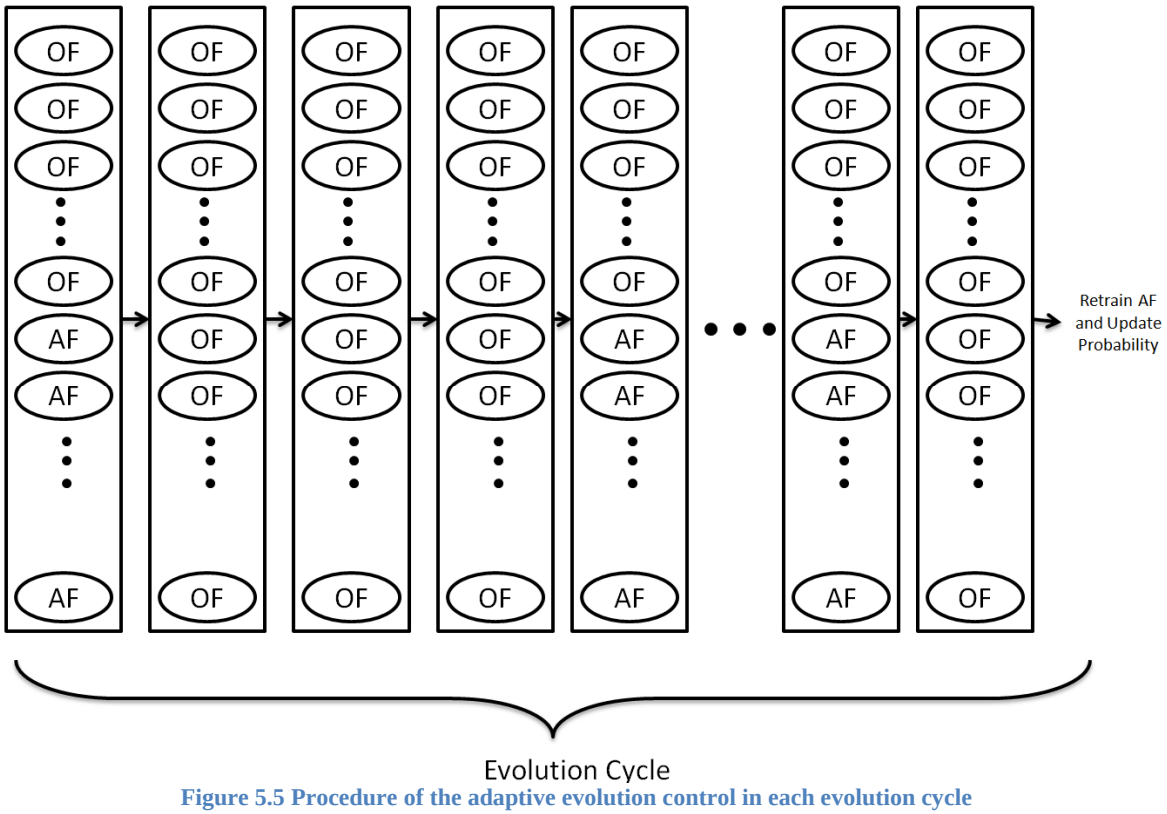


Figure 5.5 Procedure of the adaptive evolution control in each evolution cycle

The global optimum of the two-dimensional Ackley function is approximated using this approach, in order to have an initial validation of the approach. The uncontrolled approach trained by 20,000 samples reached to the following point [-0.04,0.02] ,while the designed adaptive approach, with 10,000 original function evaluation points, reached to the following point, ($[-4\times10^{-10},-6\times10^{-11}]$). It delivered significantly superior result in comparison with the conventional approach. The algorithm is also applied for the benchmarking functions (table 2.2). The following results (fitness values) are achieved with 10,000 original function evaluations, 0.0117, 0.2408, 0.0009, -4071, 0.99 respectively for Ackley, Griewank, Sphere, Schwefel and Rastrigin.

5.3. Genetic algorithm

In the nature, stronger (most adaptive) individuals tend to survive more than their peers and consequently, they can generate a more successful offspring. When selection is persistent, these individuals become universal. Based on Darwin's theory (natural evolutions), the Genetic Algorithm (GA) was inspired (Holland, 1975). It has been extensively used for optimisation problems during last decades.

The designed genetic algorithm is real-coded and consists of a number of steps: initial population generation, selection, crossover, mutation and recombination. A specific crossover is developed to keep a balance between the main capabilities, exploration and exploitation. The procedure is as follows:

1- At the start, the number of variables (N_{var}), their corresponding domains (lower bound [LB] and upper bound [UP]) and objective (fitness) function (f(x)) are defined.

2- An initial population with a specified size (N_{pop}) is generated randomly over the given domain.

3- Every chromosome of the population is evaluated based on f(x).

4- Four chromosomes are selected as parents for mating via a tournament selection with a specified tournament size (N_{tour}).

5- Five children are produced with the selected parents using a designed crossover.

6- The selection and crossover are repeated until $p_c \times N_{pop}$ chromosomes are generated; p_c is the probability of crossover operator.

7- The $p_c \times N_{pop}$ chromosomes are moved to the current population (in first generations, the current population is the initial population), and $p_c \times N_{pop}$ chromosomes of the current population are removed, these chromosomes are selected randomly. This step is called recombination.

8- $p_m \times N_{pop}$ chromosomes of the recombined population are randomly selected and mutated; the uniform mutation operator is used, i.e., a randomly chosen genome of the selected chromosomes is replaced by a random number in the corresponding variable domain. By completing this step, the new population is created.

9- The best chromosome of the current population (elite) in terms of fitness value survives by substituting it by the worst chromosomes of the new population.

10- Each chromosome of the new population is evaluated by f(x), after this step, the new population is become the current population.

11- The steps 4 to 10 are repeated till stopping criteria are met.

The custom crossover is constructed by combining different operators. The scattered (uniform) crossover is a powerful operator for exploration of solution space and its major drawback is its week exploitation capabilities (Gwiazda, 2007). The weighted interpolation is an appropriate operator for smooth landscapes and brings quick convergences to optimum in smooth landscapes (Gwiazda, 2007). Heuristic crossover has a great ability in exploitation of search space (Gwiazda, 2007). Normally in crossover operators, a child is generated by the selected parents, but in this study's crossover, each set of parents (which are four chromosomes) is used for the generation of five new chromosomes (children). Three of the children are constructed by the mentioned operators above, i.e. one chromosome (child) is generated via the scattered crossover with the probability of 50% between two of the selected chromosomes (parents), one chromosome is generated via the heuristic operator (with a ratio of 1.2) using the other two selected chromosomes, and one chromosome is generated via the weighted interpolation between the four selected chromosomes. The weighing factors for interpolations are calculated based on the fitness values of the selected chromosomes. Higher weights are allocated for the chromosomes with lower fitness values. In order to provide high-resolution exploitations, uniformly random noise with zero means and variable ranges are added to the produced

chromosomes. Noise is added to the chromosome produced by the interpolation crossover, and another noise is added to the chromosome produced by the scattered crossover. It should be mentioned that the ranges of the noise are varied by the generation number exponentially, i.e. by increase of the generation number, the ranges are reduced which accordingly causes higher-resolution exploitations.

In order to evaluate the custom crossover, five famous benchmarking functions (Sphere, Griewank, Schwefel, Ackley and Rastrigin) are utilised (Karaboga and Basturk, 2008, Molga and Smutnicki, 2005). These functions are a combination of simple, convex and unimodal, and noisy and complicated landscapes. The number of variables, the corresponding global optimum points and mathematical formulas are shown in table 5.1.

The results of the custom crossover are compared with two famous operators, scattered and heuristic on each of the benchmarking functions. The criterion of assessing is fitness value at the achieved point. Any operator which reaches to a lower value of fitness can be called a successful operator. To make a fair comparison between the operators, each optimisation is executed 7 times with different seed numbers, best, mean and worst results are reported. A same initial population and same GA options, given in table 5.2, are used by all. The outcomes are shown in table 5.3.

Table 5.1 Benchmarking functions

Function	Mathematical formula f(x)	Bounds	Global optimum x_∞	No. of variables
Sphere	$f(x) = \sum_{i=1}^{n} x_i^2$	$-5.12 \le x_i \le 5.12$	$f(x) = 0$ $for\ x_i = 0, i = 1 \dots n$	n=10
Rastrigin	$f(x) = 10n + \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i))$	$-5.12 \le x_i \le 5.12$	$f(x) = 0$ $for\ x_i = 0, i = 1 \dots n$	n=10
Schwefel	$f(x) = \sum_{i=1}^{n} (-x_i \sin(\sqrt{\lvert x_i \rvert}))$	$-500 \le x_i \le 500$	$f(x) = -418.9829n$ $for\ x_i = 420.9687, i = 1 \dots n$	n=10
Griewank	$f(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos(x_i/\sqrt{i}) + 1$	$-600 \le x_i \le 600$	$f(x) = 0$ $for\ x_i = 0, i = 1 \dots n$	n=10
Ackley	$f(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + \exp(1)$	$-32.76 \le x_i \le 32.76$	$f(x) = 0$ $for\ x_i = 0, i = 1 \dots n$	n=10

Table 5.2 GA options

Population size (N_{pop})	Stopping criteria	Initial population	Mutation & its probability	Crossover probability	Number of elites	Tournament size (N_{tour})
40	No_{gen} = 1500	Uniform	Uniform & p=0.1	p=0.9	1	4

By comparing the outcomes of different crossover operators, it can be concluded that the custom crossover provides better results, 1 to 16 orders of magnitudes improvement. The custom crossover delivers acceptable optimum points in all the functions. Figures 5.6 to 5.10 reveal the average fitness value of all 7 runs versus the generation number during optimisation via the three different crossover operators for the five benchmarking functions. The custom crossover, not only provides better final results in comparison with the other two crossovers, but also it delivers a quicker convergence.

Table 5.3 Achieved fitness values after optimisation using three crossover operators for different functions

Crossover	Fitness Value	Ackley	Sphere	Rastrigin	Griewank	Schwefel
Scattered	Worst	3.76E-01	3.03E-03	4.22E-01	1.01E+00	-4187.60
	Mean	5.64E-01	1.47E-03	2.37E-01	4.95E-01	-4188.48
	Best	7.54E-01	8.16E-04	6.01E-02	2.04E-01	-4189.27
Heuristic	Worst	3.14E-01	2.14E-03	6.55E-02	4.13E-01	-4189.01
	Mean	1.55E-01	8.61E-04	2.74E-02	1.99E-01	-4189.31
	Best	4.11E-02	1.07E-04	2.51E-03	2.35E-02	-4189.79
Custom	Worst	1.86E-08	6.15E-17	1.13E-07	3.61E-01	-4189.83
	Mean	1.04E-08	1.20E-17	1.62E-08	1.49E-01	-4189.83
	Best	1.44E-09	3.87E-19	1.55E-12	7.62E-02	-4189.83

From the figure 5.6 to 5.10, it can be found out that the slope of fitness reduction over generation number using the custom crossover is higher than the other two operators which demonstrates its strength in quick reduction of fitness. The slopes of the reduction over the first 500 generations are reported in table 5.4. The outcomes demonstrate that the custom crossover can deliver almost significant faster reduction in fitness value for different functions in comparison with the other two operators. It should be pointed out that fast reduction of misfit is not vital as the ability of approximating the global optimum, i.e., if an algorithm has the ability of approximating the global optimum properly, then it can be evaluated based on other terms such as speed of convergence.

Table 5.4 Comparison of the achieved slopes of fitness reduction in first 500 generations using different crossover operators

Functions	Scattered	Heuristic	Custom
Ackley	-0.0019	-0.0022	-0.00912
Sphere	-0.0063	-0.0072	-0.018
Rastrigin	-0.0039	-0.0031	-0.0034
Schwefel	-5.54	-5.59	-5.68
Griewank	-0.0040	-0.0045	-0.0055

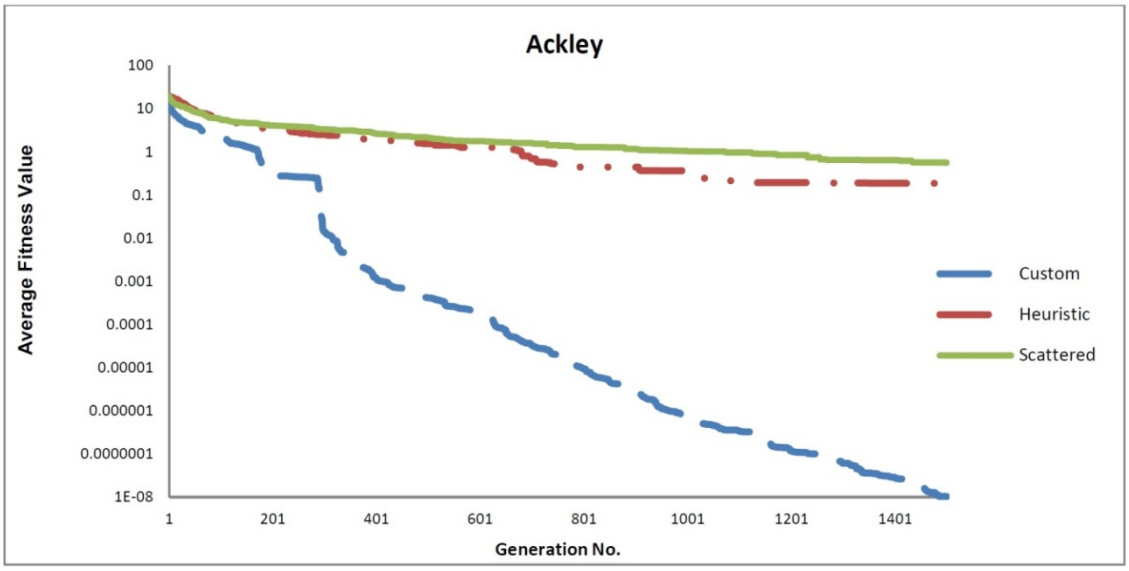


Figure 5.6 Comparison of the crossover operators on Ackley function

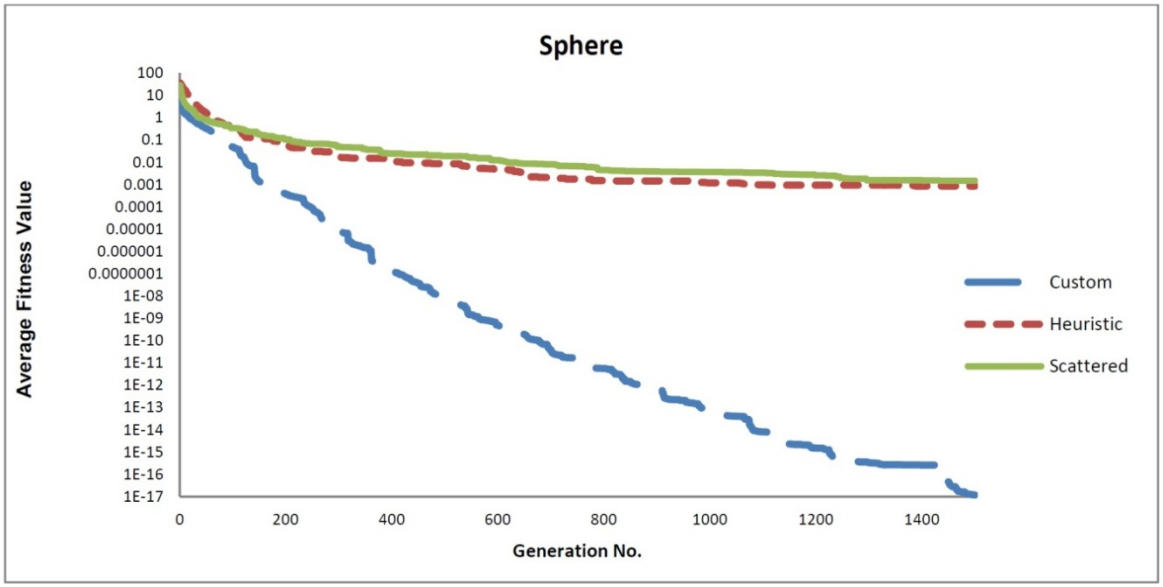


Figure 5.7 Comparison of the crossover operators on Sphere function

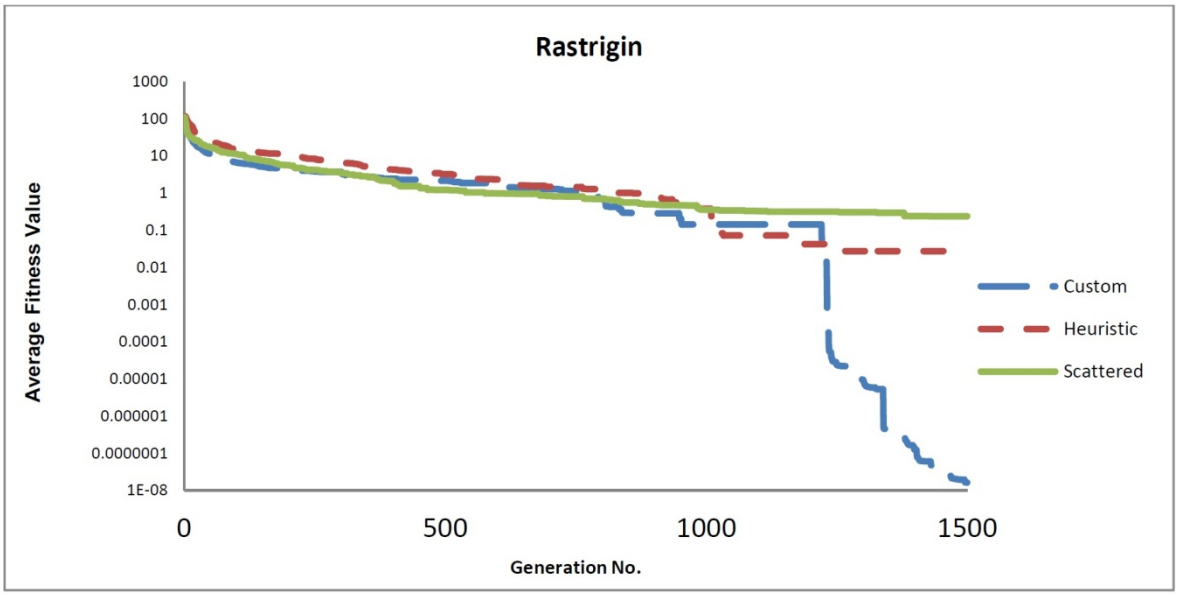


Figure 5.8 Comparison of the crossover operators on Rastrigin function

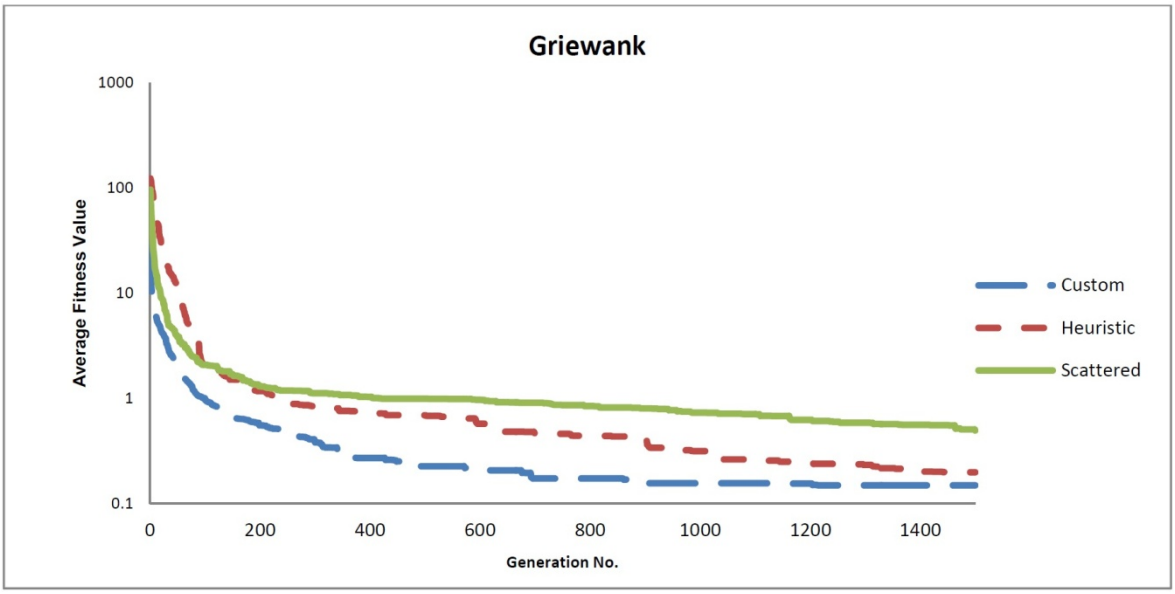


Figure 5.9 Comparison of the crossover operators on Griewank function

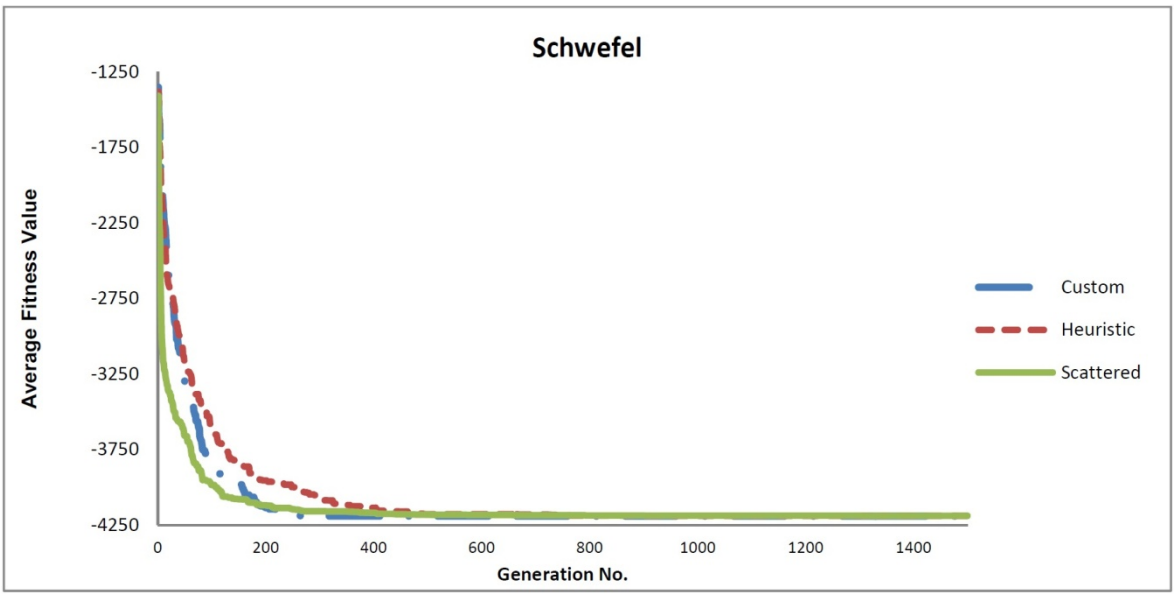


Figure 5.10 Comparison of the crossover operators on Schwefel function

The custom crossover has a great ability of exploitation conducted by the uniform noise; also, the scattered and heuristic crossover bring a thorough search of landscape by their exploration and exploitation abilities, and the weighted interpolation provided significant acceleration in converging to the optimum point. It should be stated that the improvement of optimisation is not the emphasis of this study. Any population-based optimisation algorithm can be used in these methods, but the controlled fitness approximation methods are better fit into evolutionary algorithms rather than into swarm intelligence algorithms. That is why a genetic algorithm is used. In the previous chapter, it was shown that a stochastic optimiser that has a balance between exploitation and exploration has more

chance of successfulness. Thus, the genetic algorithm is designed in a way to have a balance between the abilities. It may not be the most powerful GA.

5.4. Results

In this section, the application of the different fitness approximation approaches in a history matching problem, PUNQ-S3 reservoir model, is reviewed.

5.4.1. Case Study

PUNQ-S3 is a reservoir model widely used for benchmarking different history matching algorithms. This case was provided by Imperial Collage London University. It is a part of a real field performed by Elf Exploration Production. The reservoir model consists of 19×28×5 gridblocks and has 1761 active blocks. The reservoir is connected to a fairly strong aquifer and also, it has a gas cap. There are 6 production wells and can be seen in figure 5.11. More detail about this model can be found in the following studies (Romero et al., 2000, Hajizadeh et al., 2011a). The corresponding history for this case study consists of 1107 elements which are field cumulative gas production, cumulative field water production, cumulative field oil production, well water-cuts, well gas oil ratios, well oil rates and well bottomhole pressures for 2936 days at 42 time steps. Gaussian noise with zero mean and 2% standard deviation were added to the observed data (history). The observed data are assumed uncorrelated.

The uncertain parameters (decision variables) and their corresponding domains are shown in table 5.5. The variables are skin factors, porosities and permeabilities multipliers, aquifer parameters and connate water saturations. As it can be seen in table 5.5, there are 36 decision variables which are all continuous, the domain of all decision variables are altered to be in a same rang ([0.8-1.2]). As the variables are not the spatial property of the

reservoir (they are designed in this way, based on Zubarev et.al (2009) parameterisation), reparameterisation is not required which results in more emphasis on the fitness approximation application. The last column of the table shows the actual values (m_R) (the true solution) for each decision variable in the altered domains.

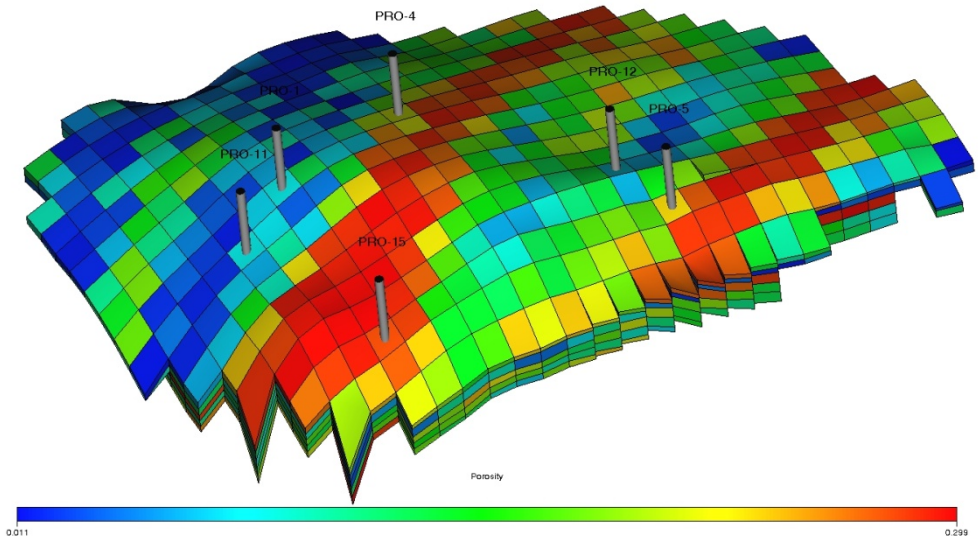


Figure 5.11 PUNQ-S3 reservoir model

The objective function is defined as equation 4, using a Bayesian inversion context by an assumption of Gaussian measurement errors. m is a column vector which its elements are the uncertain parameters. g is the forward operator in which it is required to execute ECLIPSE E100. d_{obs} is a column vector which consists of the observed data (the reference of calibration). In history matching it is required to find an m which minimises S(m). Thus, it is required to apply an optimisation algorithm to find the global optimum point, the maximum of posterior distribution (MAP), of this 36-dimensional continuous search space. The corresponding fitness value for m_R ($S(m_R)$) is 425.73. To shrink the nonlinearity of the OF, it is transformed from equation 4's form to equation 5's form (Zubarev, 2009).

$$S(m) = \sum_{i=1}^{1107} \left(\frac{g(m)_i - d_{obs_i}}{0.02 \times d_{obs_i}} \right)^2 \quad (4)$$

$$t(m) = \ln\big(S(m)\big) = \ln\left(\sum_{i=1}^{1107} \left(\frac{g(m)_i - d_{obs_i}}{0.02 \times d_{obs_i}} \right)^2 \right) \quad (5)$$

Table 5.5 Decision variables and their corresponding domains

Element	No. of variables	Domains	Altered domains	Actual solution	Actual solutions in modified domain
Porosities' multipliers in each layer	5	[0.8…1.2]	×1 [0.8…1.2]	1 for each layer	1 for each layer
Permeabilities' multipliers in x direction in each layer	5	[0.8…1.2]	×1 [0.8…1.2]	1 for each layer	1 for each layer
Permeabilities' multipliers in y direction in each layer	5	[0.8…1.2]	×1 [0.8…1.2]	1 for each layer	1 for each layer
Permeabilities' multipliers in z direction in each layer	5	[0.8…1.2]	×1 [0.8…1.2]	1 for each layer	1 for each layer
Pressure of first aquifer	1	[234bars…351bars]	/292.5 [0.8…1.2]	234 bars	0.8
Pressure of second aquifer	1	[234bars…351bars]	/292.5 [0.8…1.2]	234 bars	0.8
Permeability of first aquifer	1	[110md…165md]	/137.5 [0.8…1.2]	137.5 md	1
Permeability of second aquifer	1	[110md…165md]	/137.5 [0.8…1.2]	137.5 md	1
Connate water saturation	1	[0.149… 0.225]	×5.35 [0.8…1.2]	0.2	1.07
Skin factors	11	[-5…50]	/137.5+0.836 [0.8…1.2]	5 for all 11 variables	0.8727 for all 11 variables

In order to provide a fair comparison between all four approaches, a limit is set for computation time, i.e., once any of the approaches reaches to the limit, the optimisation will be terminated. The fitness value at the achieved optimum point $S(m_\infty)$ will be used for comparison. The computation time for training the proxy and the evaluation of each point using the proxy is assumed negligible in contrast with the computational cost of the original function evaluations. The computational cost limit is that the approach is not allowed to evaluate the original function more than a threshold (OFEL) which is set to 10,000. The required CPU time to evaluate the original fitness function is about 8 second on a personal computer with the following configurations: CPU Intel i-7-2820QM and 8 GB ram. The required CPU time on the same computer for evaluating the approximation function is about 0.02 second; that is why the corresponding computation of approximation function evaluation is assumed negligible. As mentioned before, in each run (seed number) for each method, the computation limit is 10,000 original fitness function calls; therefore each result takes almost 1 day to be calculated using the aforementioned machine.

UFA uses the all 10,000 OF evaluations in the initial stage to train the proxy; hence, the stopping criterion in this approach will be the number of generations (2000). FAIBEC and FAPBEC approaches use 1,000 OF evaluations in the initial stage to train an initial proxy

and make use of the rests (9,000) during the optimisation process, i.e., once the number of OF evaluations reaches to 10,000, the optimisation will be stopped. In the adaptive approach, an initial training is not required, thus the all 10,000 evaluations are made through the optimisation process. In FAIBEC and FAPBEC approach, the number of OF evaluations in each evolution-control cycle is fixed, but in FAAEC the number of OF evaluations is varied based on the fidelity of AF. Since the computation time is same for approaches, any approach which reaches to a lower fitness value can be called a better approach.

To provide high-quality comparison: 1- history matching using each approach is executed 5 times with different seed numbers and the average fitness values are used for the comparison, 2- same GA options given in table 5.6 are used by all, and 3- a same number of hidden neurones (16) for the ANN is used by all. To select this value for the number of hidden neurones, a sensitivity analysis has been done by evaluating 50 new samples via the proxy trained by the 10,000 samples. In figure 5.12, fitting errors calculated by equation 2 are drawn versus the number of hidden neurons. Using 12 to 25 hidden neurones almost similar performance is achieved; among them, the ANN with 16 hidden neurones has the lowest error and thus is picked[5]. It should be mentioned that the quality of fitting may be enhanced by increasing the number of hidden neurones (more than 30), by the increase of neurones, the computation of fitting will be more expensive. As in this study, the emphasis is on the effect of model management on the quality of final model, an insignificant misfit is favoured to be able to distinguish between the approaches.

Table 5.6 GA options for carry out history matching

Population size (N_{pop})	Stopping criteria	Initial population	Mutation & its probability	Crossover probability	Number of elites	Tournament size (N_{tour})
40	No_{gen} = 2,000 or OFEL=10,000	Uniform	Uniform & p=0.1	p=0.9	1	4

[5] The number of hidden neurons also could be determined by the R^2 value of validation set, but as the samples are selected randomly in each run, and also similar R^2 was obtained by each set. We decided to distinguish between them according to the calculated misfit based on the same 50 new samples.

5.4.1.1. History matching using UFA

The results of this approach are shown in table 5.7 and figure 5.13 to 5.17. The average, best and worst result in terms of fitness values are bolded in the table. In the figures, two graphs are shown: the red solid line shows the fitness value in each generation number which is calculated based on the approach calculations, and the blue dashed line shows the fitness value in each generation based on OF ($S(m_i)$). As it can be seen, the graphs in all five seed numbers are not matched on each other; it shows that the approach misdirected the optimisation. The other mean for assessment is the average fitness value based on OF ($\frac{1}{5}\sum_{i=1}^{5}\left(S\left(m_{\infty_i}\right)\right)$), it should be close enough to 425.73. This criterion is not seen for this approach. The average of the five runs is 86,697 which is too far from the fitness value at the global optimum point (425.73). The difference between each seed number is due to the random selection of the samples via ANN for training and validation which results in different approximation function and accordingly different optimum points.

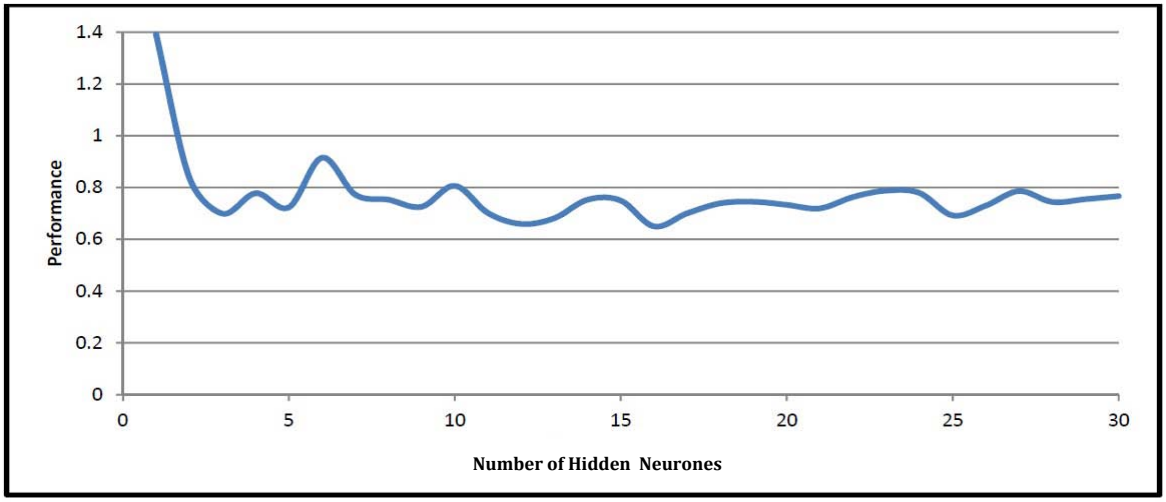


Figure 5.12 Neural network performance versus the number of hidden neurones

Table 5.7 Fitness values for the uncontrolled fitness approximation approach

Seed No.	1	2	3	4	5	Average	Best	Worst
Fitness Value at Achieved optimum point (S(m))	182,170	191,606	59,006	93,153	25,529	86,697	25,529	191,606

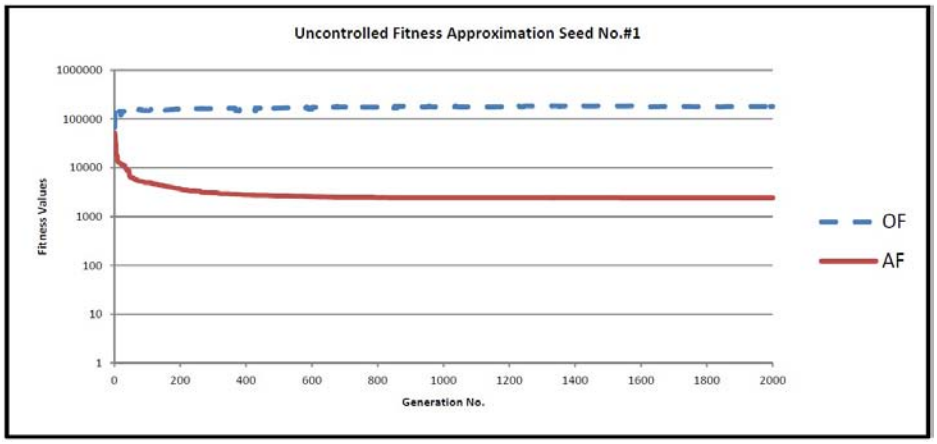


Figure 5.13 History matching using the uncontrolled fitness approximation, seed No.#1

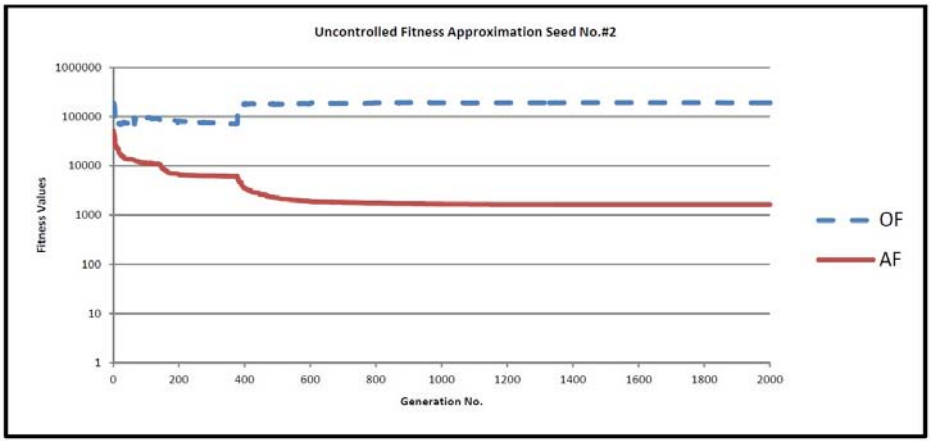


Figure 5.14 History matching using the uncontrolled fitness approximation, seed No.#2

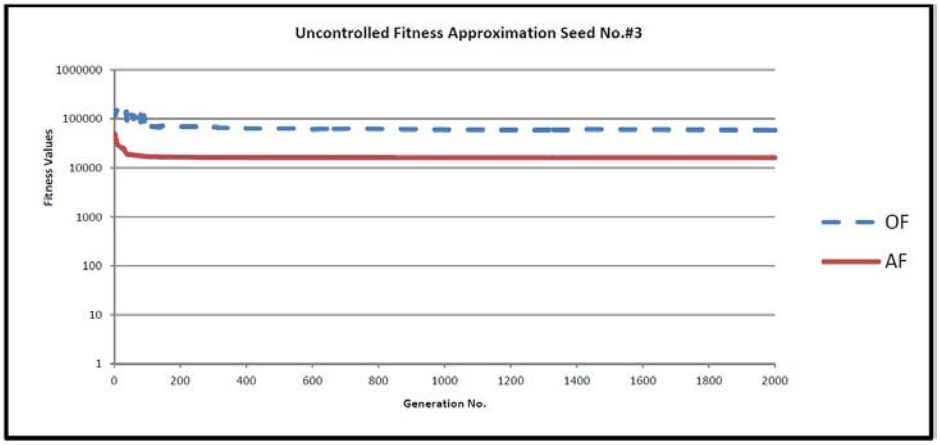


Figure 5.15 History matching using the uncontrolled fitness approximation, seed No.#3

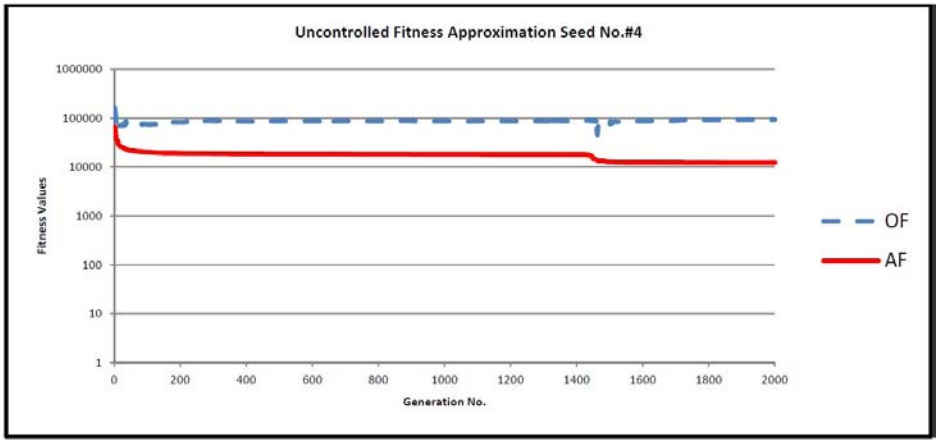


Figure 5.16 History matching using the uncontrolled fitness approximation, seed No.#4

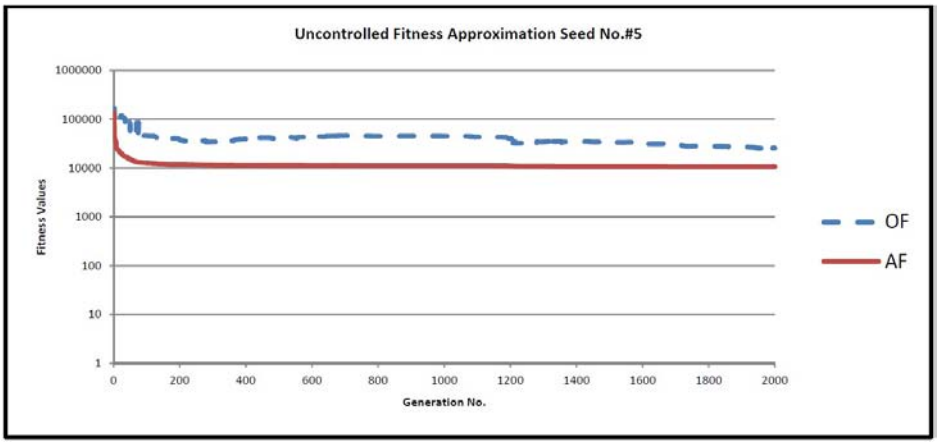


Figure 5.17 History matching using the uncontrolled fitness approximation, seed No.#5

5.4.1.2. History matching using FAIBEC

The results are shown in table 5.8 and figure 5.18 to 5.22. As it can be seen, the average fitness value (1,746) is significantly lower than the previous approach. The two graphs are acceptably matched on each other, however, a number of deviations can be seen.

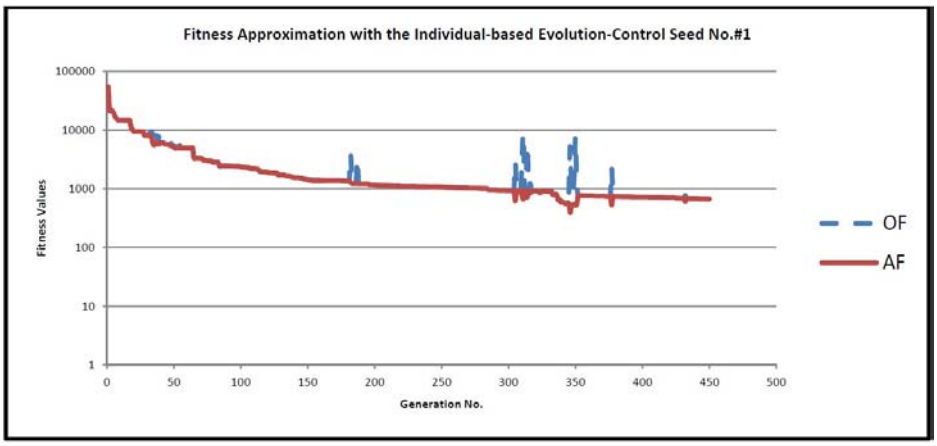


Figure 5.18 History matching using the FAIBEC seed No.1

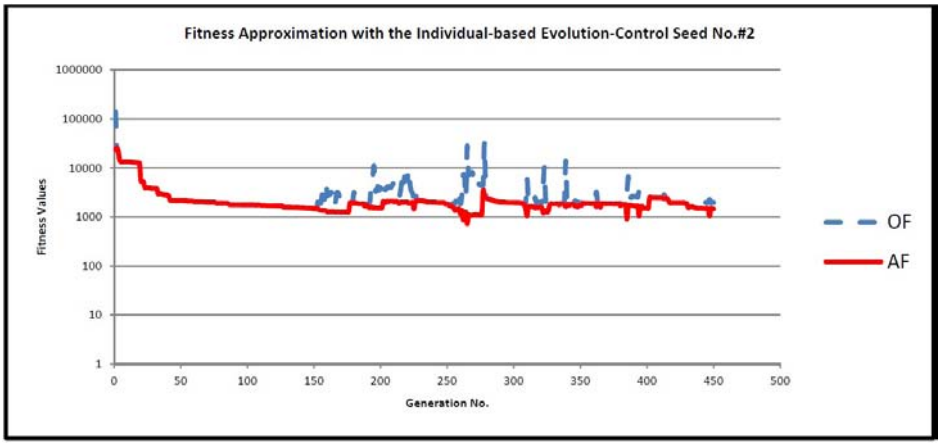


Figure 5.19 History matching using the FAIBEC seed No.2

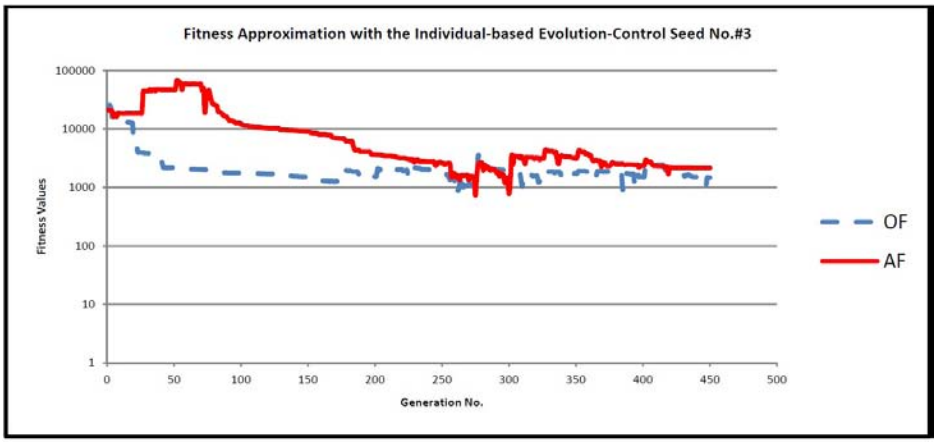


Figure 5.20 History matching using the FAIBEC seed No.3

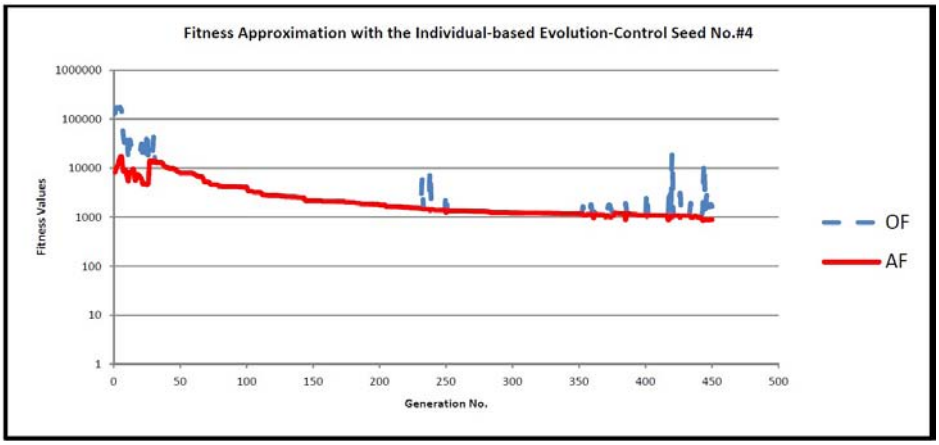


Figure 5.21 History matching using the FAIBEC seed No.4

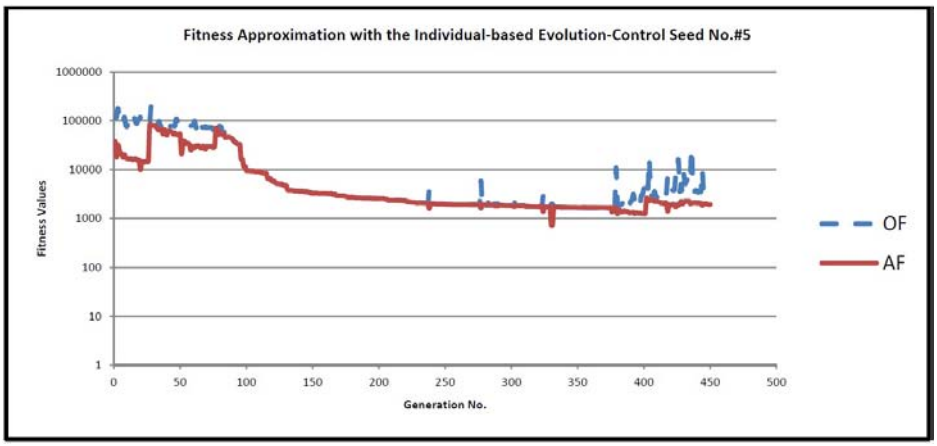


Figure 5.22 History matching using the FAIBEC seed No.5

Table 5.8 Fitness values for fitness approximation with the individual-based evolution-control approach

Seed No.	1	2	3	4	5	Average	Best	Worst
Fitness Value at Achieved optimum point (S(m))	667	1,978	2,155	1,651	3,458	1,746	667	3,458

5.4.1.3. History matching using FAPBEC

The results are shown in table 5.9 and figure 5.23 to 5.27. As it can be seen, the average fitness value (951) is significantly lower than the first approach, and considerably lower

than the second approach. The two graphs are acceptably matched on each other, however, a number of oscillations can be seen.

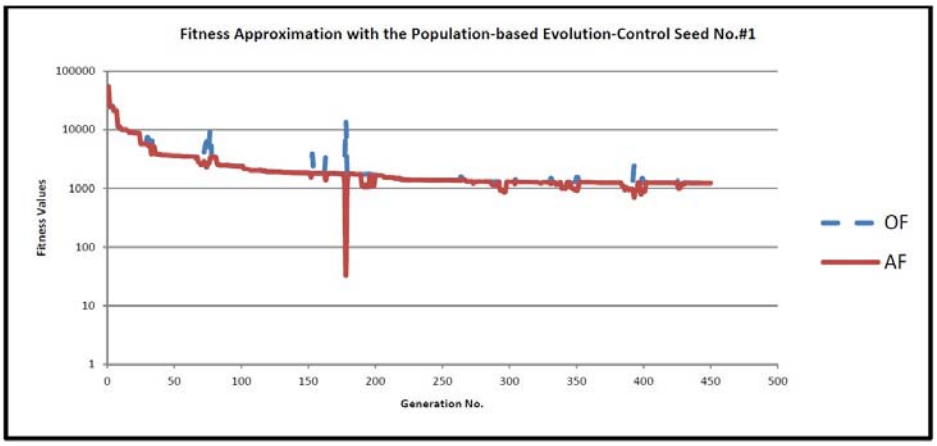


Figure 5.23 History matching using the FAPBEC seed No.1

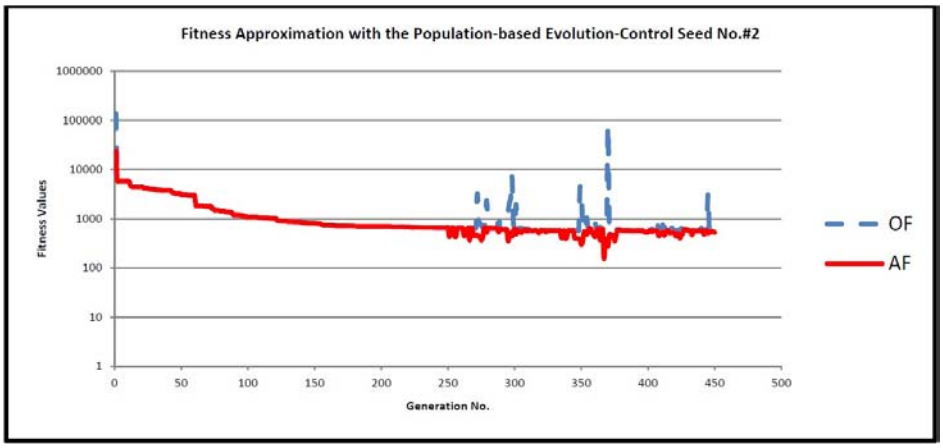


Figure 5.24 History matching using the FAPBEC seed No.2

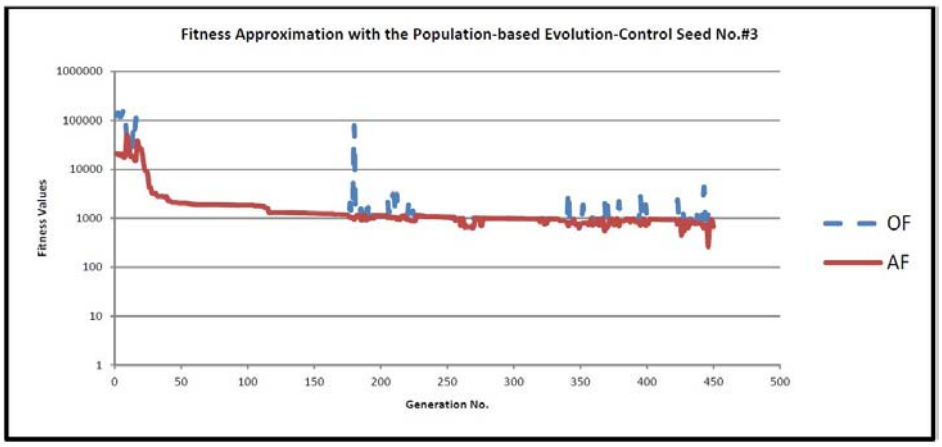


Figure 5.25 History matching using the FAPBEC seed No.3

Table 5.9 Fitness values for fitness approximation with the population-based evolution-control approach

Seed No.	1	2	3	4	5	Average	Best	Worst
Fitness Value at Achieved optimum point (S(m))	1,236	562	918	1,083	1,125	951	562	1,236

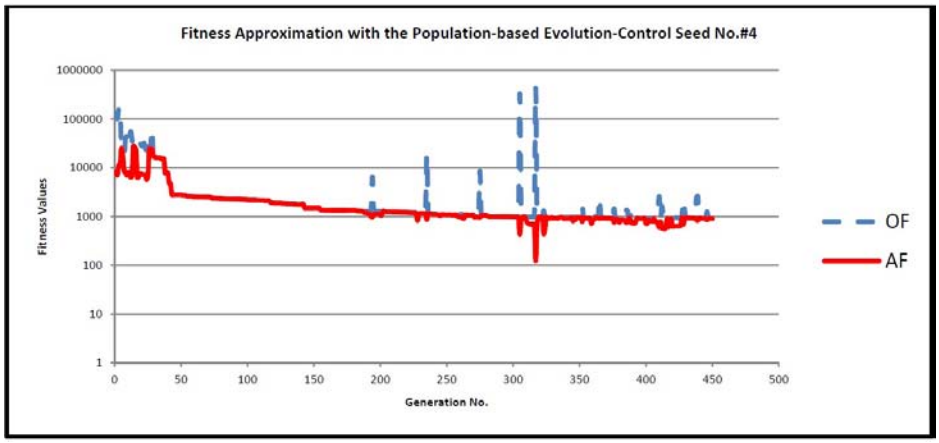


Figure 5.26 History matching using the FAPBEC seed No.4

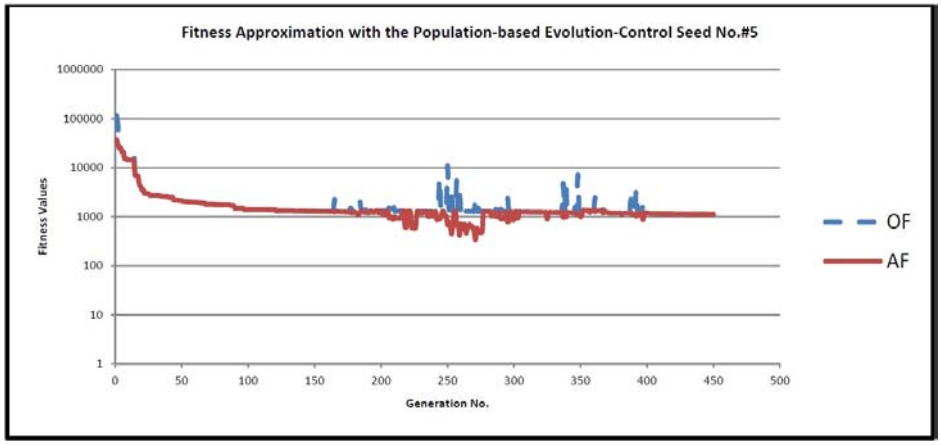


Figure 5.27 History matching using the FAPBEC seed No.5

5.4.1.4. History matching using FAAEC

The results are shown in table 5.10 and figure 5.28 to 5.32. As it can be seen, the average fitness value (841) is less than the other three approaches. Also, the two graphs are perfectly matched on each other which shows that the controlled approach could redirect the optimisation direction into a right path.

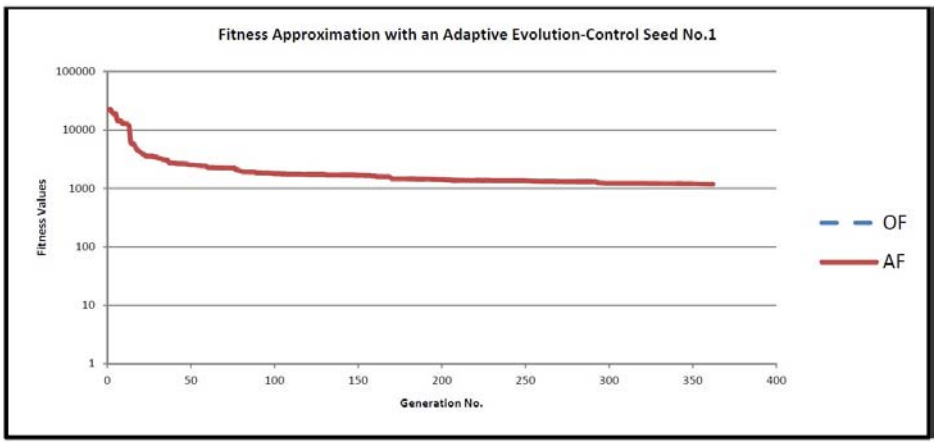


Figure 5.28 History matching using the FAAEC seed No.1

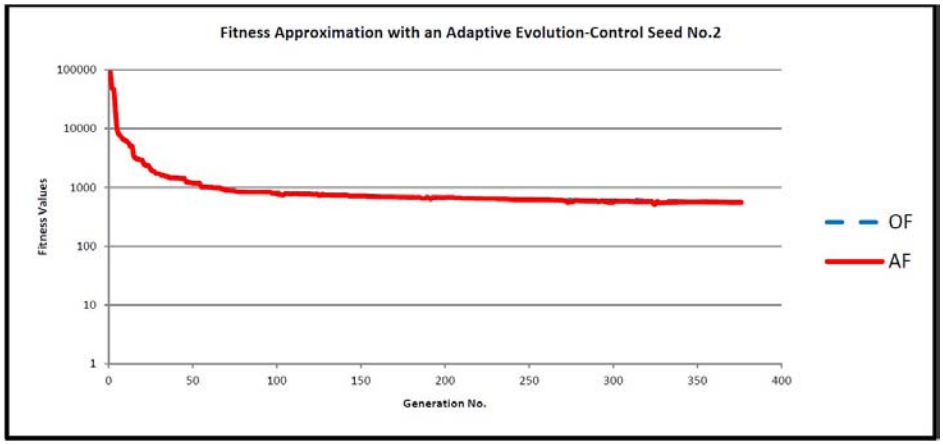


Figure 5.29 History matching using the FAAEC seed No.2

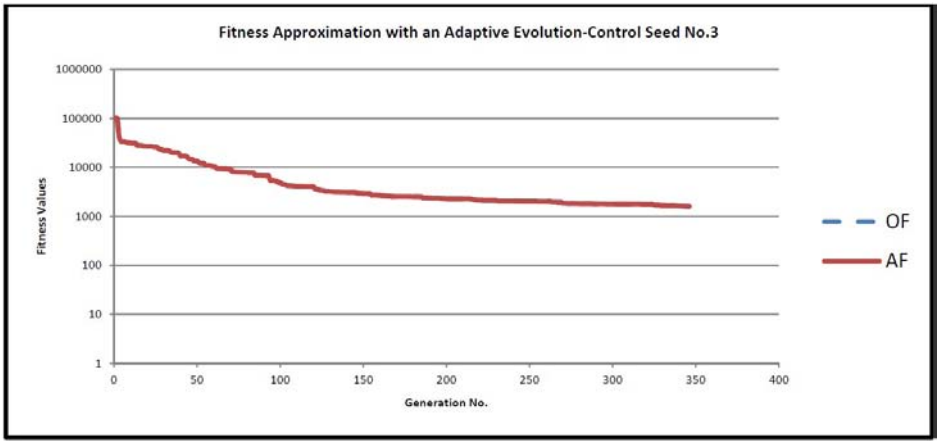


Figure 5.30 History matching using the FAAEC seed No.3

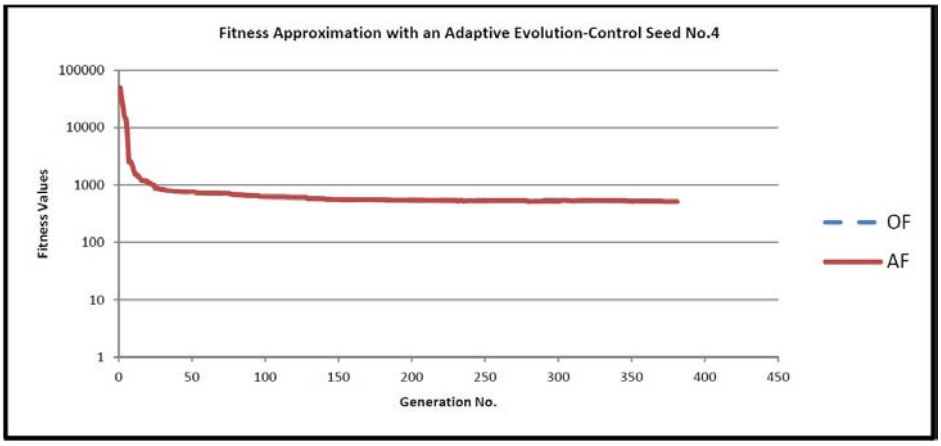


Figure 5.31 History matching using the FAAEC seed No.4

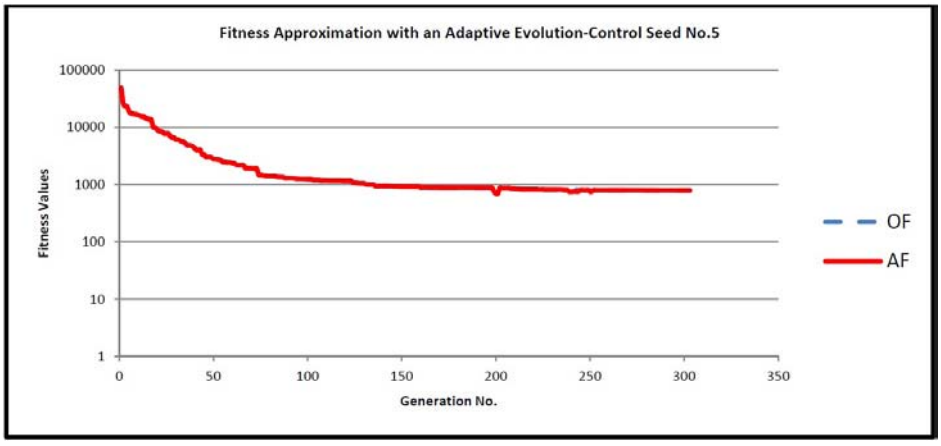


Figure 5.32 History matching using the FAAEC seed No.5

Table 5.10 Fitness values for fitness approximation with the adaptive evolution-control approach

Seed No.	1	2	3	4	5	Average	Best	Worst
Fitness Value at Achieved optimum point (S(m))	1181	553	1602	513	785	841	513	1602

The outcomes of the four approaches indicate that i- the controlled approach delivers significantly better results, and ii- between the controlled approaches, the customised adaptive approach delivers better results. Figure 5.33 shows the misfit after history matching for bottomhole pressure of producer 15, as an example for comparison of the methods. The black curve is history (true behaviour), the dashed blue curve is the simulation result achieved by the uncontrolled approach and the red curve is the simulation result achieved by the adaptive approach. As it can be seen, the obtained result by the adaptive approach are much more consistent with the history data than the outcome of the uncontrolled approach.

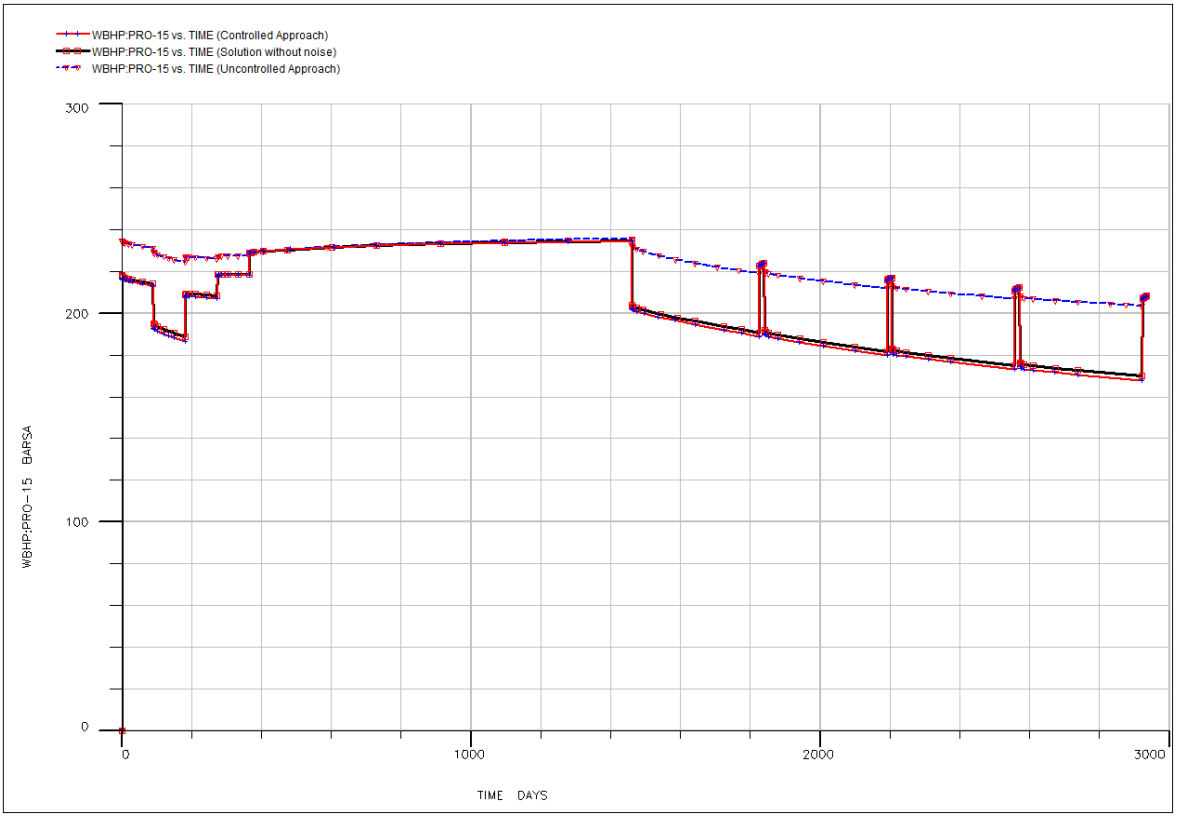


Figure 5.33 Comparison of achieved mismatch after history matching by the uncontrolled and the adaptive approach for well bottomhole pressure of producer#15

5.5. Discussion

The uncontrolled fitness approximation approach could not deliver acceptable results (average fitness value was much larger than the global fitness value), however, a large number of samples (10,000) were used to train the network. The unsuccessfulness can be either because of having offline training, an inadequate number of samples or inaccurate fitting. Increasing the number of initial samples may augment the quality of the trained proxy (Zubarev, 2009), but it makes this approach expensive and uneconomical. This approach is very sensitive to the quality of trained proxy. If the best number of hidden neurones and layers and the sufficient number of samples are not selected, the chance of trapping into a false optimum is significant. It is fair to state that this approach may misdirect the optimisation algorithm to false optimum points.

The three controlled approaches, with the same amount of computation time, articulated significantly superior results in comparison with the UFA approach. The average fitness values for these three approaches were acceptably close to the global fitness value. The success of these approaches is because of their emphasis on the approximation of the global optimum point rather than the approximation of the entire landscape, unlike UFA. In the controlled approach, the inaccuracy of the proxy model (even inaccurate fitted model) will not significantly misdirect the calibration, as the original function evaluation redirect the calibration through the way of optimisation.

Among these three approaches, the adaptive approach performed better than the others. Thus, it is fair to claim that the evolution-control was carried out more efficiently in the designed adaptive approach. It is due to its ability in exploitation of search space and having an updatable probability. Figure 5.34 shows the average fitness value for all four approaches.

One of the restrictions of fitness approximation approaches with an evolution-control is that these approaches can be employed only when a population-based algorithm is being used for optimisation.

In order to show that the effect of fitness approximation in speeding up the procedure of history matching, history matching was carried out for the same problem using the regular approach (without proxy-modelling). The regular approach of history matching, in which OF is used in whole optimisation process, reached to the same result (fitness value) as the adaptive approach with the same optimisation options, but it needed evaluating the original function 40,000 times. Thus, using the adaptive approach for history matching, **75%** computational costs can be reduced.

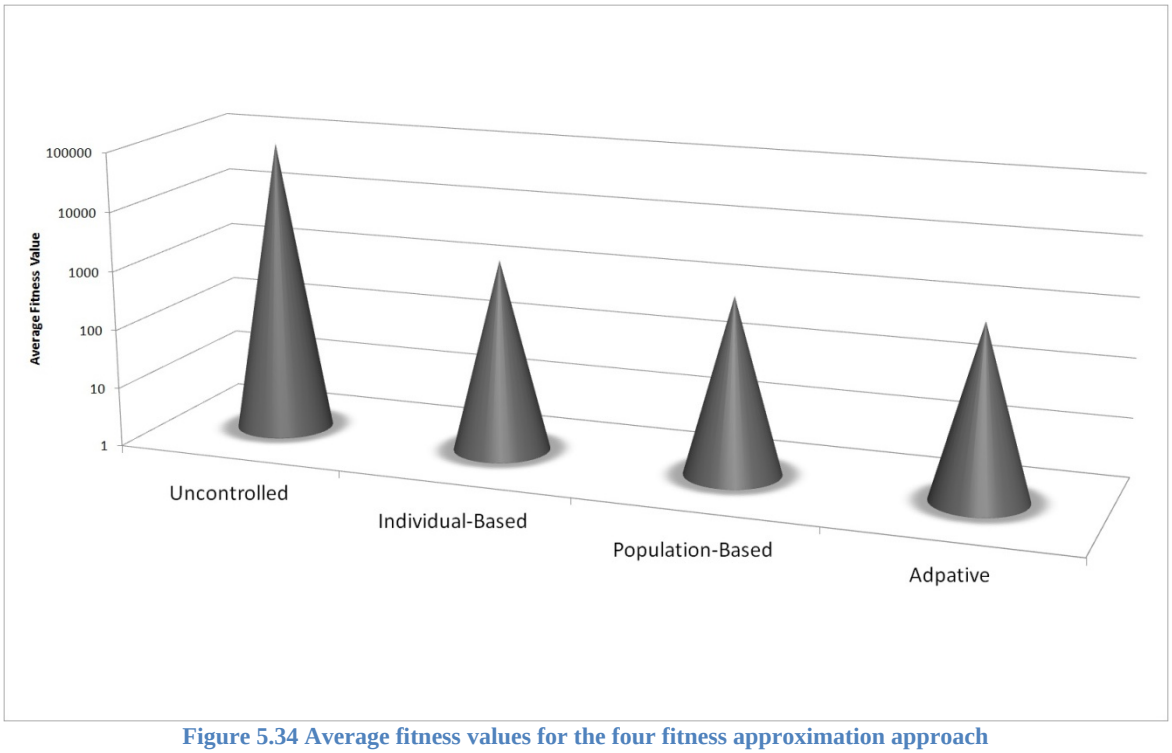


Figure 5.34 Average fitness values for the four fitness approximation approach

A genetic algorithm was implemented in this study for the optimisation, since GAs can be suitably integrated into the framework. GA can be replaced by any other population-based optimisation algorithm, preferably Evolutionary Algorithms, as they are better fit into the framework than the swarm intelligence methods. In the swarm intelligence methods, some of the individuals in every population are generated from the other individuals of the same

population; thus deciding between the individuals who should be evaluated via the approximation function is tricky and may result in inefficiency of model management. In the ABC algorithm, the employed bees are needed to be evaluated first and according to these evaluations, the onlooker bees are created. Evaluations with error (evaluations with the proxy model) for the employed bees may result in wrong decision by the onlooker bees, and evaluations with error for onlooker bees may result in inappropriate exploitation. Besides, the scout bees are not always available in every population. It should be noted that any other type of proxy or sampling strategy can be replaced by ANN. It is worth mentioning that the proxy-modelling methods should be employed, when we have computation limitations.

The conclusive remarks of this chapter are as following:

- The uncontrolled fitness approximation methods are not reliable in speeding up the history matching, and they may misdirect the optimisation algorithm to false optimum points.
- The controlled fitness approximation methods perform significantly better than the uncontrolled fitness approximation methods. It shows the constructive effects of having an evolution-control technique and online learning sampling.
- Computational costs can be reduced up to 75%, if the adaptive fitness approximation approach is used with reasonable accuracy.
- Predefined probabilities for FAPBEC and FAIBEC make the model management not as efficient as the designed adaptive method.

Chapter 6

6. Reservoir modelling using image fusion technique

This chapter addresses the challenges in high-dimensional history matching problems. In the fourth chapter, it was shown that stochastic methods are powerful algorithms in approximating the most probable models, but they suffer from high-computational costs. In the previous chapter, a fitness approximation method was introduced to prevail over the computational expenses associated with population-based algorithms. In those case studies, the number of variables was not large or the reparameterisation could be implemented in an error-free manner. But, the reparameterisation cannot always be implemented reliably. One way to remove the reparameterisation error is using a direct calibration approach, instead of indirect calibrations. In the direct calibration of large history matching problems, the optimiser should be capable of coping with a very high-dimensional search space. A number of optimisation algorithms were introduced; all the applied algorithms are gradient-based methods, (Wu et al., 1999, Li et al., 2001, Zhang et al., 2005a). Due to the nonlinearity of the inverse problem, gradient-based algorithms have the chance of getting trapped into a local minimum. On the other hand, the regular forms of the stochastic optimisers are expected to be inefficient in high-dimensional search space.

In this chapter, an inventive stochastic algorithm is introduced which can be used as a direct calibrator. To develop this history matching (optimisation) algorithm, the image

fusion and evolutionary computation concepts are blended. The questions, the author wishes to answer in this chapter, are:

a- Does the developed optimisation (history matching) algorithm deliver high-quality results?
b- Is the algorithm better than the conventional history matching algorithms either the direct or indirect approach?
c- Do the regular stochastic optimisers perform acceptably in full-parameterised history matching models?
d- Do the reparameterisation methods misdirect history matching?

In order to assess the proposed algorithm and answer the aforementioned questions, history matching is carried out for a synthetic model and also PUNQ-S3 reservoir model using the proposed algorithm and six history matching algorithms (direct and indirect). This chapter begins by an introduction to the image fusion technique. The methodology and results are described in subsequent sections. The benefits and drawbacks of the method are discussed, and followed by some conclusive remarks in the summary and discussion section. This chapter is a modified and adjusted version of one of our published papers in a SPE conference, "SAYYAFZADEH, M. & HAGHIGHI, M. 2013. High-resolution reservoir modelling using image-fusion in history matching problems, in SPE EUROPEC conference London,"

6.1. Image fusion technique

Image fusion (data fusion) is a process in which two or more images (data) are combined together to provide a single new image (data) which carries more information in comparison with the input images (data). The data fusion is being used from 1950's while the main goal was merging the images of different sensors to form a more informative

image for object identification (Wald, 1999). It is also known as synergy (Wald, 1999). The applications of image fusion can be found in various problems, such as multi-focus image fusion, satellite and medical imaging (Stathaki, 2008). The data fusion can be performed on different levels, signal, pixel, feature, and symbolic (Petrosian and Meyer, 2001). In this study, the main emphasis is on pixel level. Different mathematical techniques have been developed to form high-quality fused images (Petrosian and Meyer, 2001). These techniques can be categorised into two groups, fusion in the spatial domain and fusion in transform domains. A number of these techniques are image averaging, Laplacian pyramid based image fusion and discrete wavelet transform (Petrosian and Meyer, 2001).

The wavelet transform has found more interests in comparison with the other techniques, as a result of its advantages, including, affording directional information, not having blocking effects on pixels with a huge contrast, and having better signal to noise ratios (Nikolov et al., 2001). In this technique, input images (I_1 and I_2) are transformed into a wavelet domain, then, based on a specific rule (γ) the images are merged, and finally fused image (I_F) is transformed back to the spatial domain using an inverse wavelet (equation 1) (Nikolov et al., 2001). Merging in the wavelet domain is carried out with a rule based on the coefficients (Pajares and Manuel de la Cruz, 2004). A high-quality fusion rule typically transfers the following elements, 1- the maximum information from input images, 2- the minimum noises from them. The general concepts are explained in the following paragraphs.

$$I_F(x,y) = Wavelet^{-1}\big(\gamma(Wavelet(I_1(x,y)), Wavelet(I_2(x,y)))\big) \quad (1)$$

First, 1D wavelet transform is described, and then it is extended to 2D transform. Wavelets are created by a function (ψ) which has dilations and translations, generally, as follows:

$$\psi_{a,b}(t) = |a|^{-\frac{1}{2}}\psi\left(\frac{t-b}{a}\right) \quad (2)$$

It is sought to represent (decompose) signals (*f*), based on wavelets, and recreate it. There are two types of wavelet transform, continuous and discrete. The discrete wavelet transform (DWT) is used and favoured in image analysing. By the considering piecewise interval by a factor of 2,

$$a = 2^m \tag{3}$$

$$b = n2^m \tag{4}$$

Signal based on the wavelet is as follows,

$$f(x) = \sum_{m,n} c_{m,n}\psi_{m,n}(x) \tag{5}$$

$$\psi_{m,n}(x) = 2^{-\frac{m}{2}}\psi(2^{-m}x - n) \tag{6}$$

In order to be able to calculate the $c_{m,n}$ (coefficients), the mother wavelet (prototype) ψ should be carefully chosen.

$$c_{m,n} =< f, \psi_{m,n} >= \int \psi_{m,n}(x)f(x)dx \tag{7}$$

This concept is used for multi-resolution analysis. A signal can be represented by a series of coarser approximations (subsapces V_j) (Gao and Yan, 2011). Φ is wavelet scale function which is diagonal on the wavelet function ψ, and its inner product with the f provides accurate approximation at fixed scales. Hence, *f(x)* can be represented by the detailed information in W_1 and the approximation in V_1 which itself can be represented by a W_2 and V_2 and so on.

$$f(x) = \sum_{m=-\infty}^{M}\sum_{n=-\infty}^{\infty} c_{m,n}\psi_{j,n}(x) + \sum_{k=-\infty}^{\infty} a_{M,k}\Phi_{m,n}(x) \tag{8}$$

Approximation coefficient is as inner products of

$$a_{m,n} =< f(x), \Phi_{m,n}(x) > \tag{9}$$

Detail coefficient is as inner products of

$$c_{m,n} =< f(x), \psi_{m,n}(x) > \quad (10)$$

$$\Phi_{m,n}(x) = 2^{-\frac{m}{2}}\Phi(2^{-m}x - n) \quad (11)$$

$a_{m,n}$describes the approximations of *f* and it is calculated,

$$a_{m,n} = \sum_{k} h_{2n-k} a_{m-1,k} \quad (9)$$

In which the low pass filter is

$$h_n = 2^{\frac{1}{2}} \int \Phi(x - n)\, \Phi(2x) dx \quad (10)$$

and

$$c_{m,n} = \sum_{k} g_{2n-k} a_{m-1,k} \quad (11)$$

In which the high pass filter is

$$g_l = (-1)^l h_{1-l} \quad (12)$$

There are different algorithms for scale and wavelet base functions, such as Haar, Symlet, Daubechies and Mallat. The images are 2 dimensional. The 1D wavelet can be extended to 2D transform in which the 2 filters (h and g) are conducted in horizontal direction and followed by down-sampling in each row. Then, the outcomes are revisited in vertical direction followed by down-sampling in each column. By this procedure, in one stage process, a single image results in four images (called low-low, high-low, high-high and low-high) with a level coarser resolution (Li et al., 1995) (three detail coefficients and one scale coefficient). The images are same size matrix as the initial image. After finding the coefficients with desired level for the images which are sought to be fused, a rule should be applied to combine these coefficients efficiently. It should be mentioned that in order to fuse the images, they should be in a same level of approximation.

The coefficients (with assumption of one level coarser) of each image in each pixel should be combined together. One way is calculate the average or a weighted average for two corresponding coefficients of the two images in each pixel. By this way, the directional and contrast may be lost. The alternative way is the selection between the coefficients of the two images. The selection can be maximum, minimum, or random. In the maximum selection, the maximum of absolute coefficients are selected. This selection keeps the high contrast in each image. It delivers good results in multi-focus problems from a same object. It recovers the unclear sections by the assist of the other image. The pixel by pixel evaluation may be result in losing information, thus a courser windows can be used which may consists of 25 pixels or more. After fusing (selecting) the coefficients in wavelet domain, by an inverse wavelet, (summation), the fused image is transformed to the spatial domain.

Figure 6.1 taken from Nikolov et al. paper (2001) is an example of image fusion in multi-focus problems. The two images on top are two different images from a same subject with different camera focus centres. To form an image in which two alarm clocks are clear (not-blurred), a wavelet image fusion is used. Both images are transformed into the wavelet domain, then in the wavelet domain, the transformed images are combined, then using an inverse wavelet the new image is approximated. As it can be seen, in the fused image, the both alarm clocks are clear.

6.2. Methodology

A history matching algorithm based on the image fusion concept is developed. An example is used to demonstrate the application of image fusion in merging different porosity realisations. Similar to multifocus problems, each realisation of variables (image) may represent a section of reality (subject). To have a realisation which has more information concerning the reality, a combination of the realisations is a good candidate to be checked

out. The merged image should keep the main features which are in common in the input realisations, recover the blurred or missing section of either of input images with the support of other images, and also carry the minimum noise of the input images.

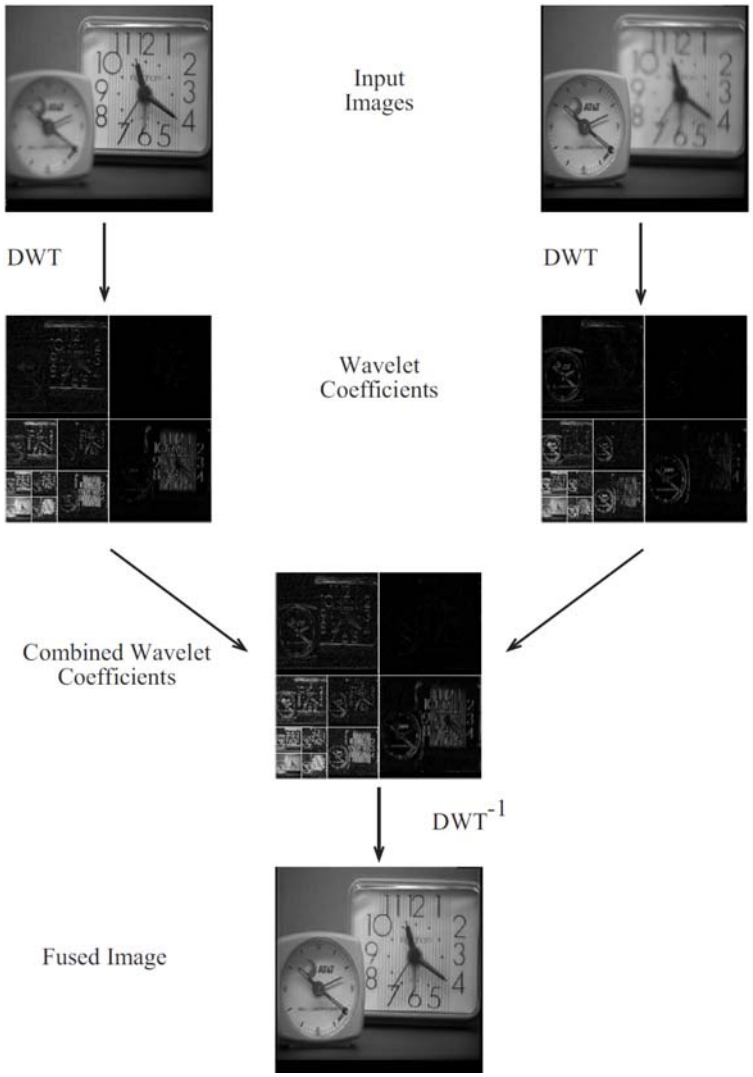


Figure 6.1 Image fusion example (Nikolov et al., 2001)

Assume figure 6.2-a and figure 6.2-b show two porosity distribution realisations. Figure 6.2-a is more informative on the section of x<80 while figure 6.2-b is more informative on the section of x>80. If these two realisations provide an average match of simulation data with the history data (for instance, the right hand side provides a good match for the wells located on x<80 and a poor match for the wells located on x>80, and the left hand side provides a poor match for the wells located on x<80 and a good match for the wells located on x>80), it is likely that the third realisation (figure 6.2-c) become a more reliable

distribution. The third image can be intelligently obtained by fusing the two images using the discrete wavelet image fusion.

Based on this concept and using stochastic modelling, the algorithm is developed. The designed algorithm is similar to the general form of GA (Haupt and Haupt, 2004). Therefore, similar terminologies are used to explain the algorithm. The algorithm begins with a generation of a set (known as population) of candidate solutions (known as chromosomes) using an initialisation procedure. The population size and the chromosome size are defined. Then, using one of the two initialisation procedures, random and heuristic, an initial population is generated. After initialisation, all the chromosomes are evaluated based on a fitness function (objective function). The task of the fitness function is to distinguish between the chromosomes based on their goodness. After generating the initial chromosomes and evaluating them, the main loop of the algorithm starts. In each step (known as generation) of the main loop, a sub-set of the current population is selected using a selection operator. Usually, the selection procedure is conduced according to the fitness of chromosomes. Two operators called crossover and mutation are applied to the selected set of chromosomes and a new set of chromosomes is produced. The crossover and mutation operators are applied to the chromosomes with probability p_c and p_m, respectively. The new set generated by the selection, crossover and mutation is recombined with the current population to provide the new population. To keep the best ever found solution, the chromosome (known as elite) is migrated directly to the new population. The main loop is repeated until predefined stopping criteria are met.

There are many types of selection, crossover, and mutation operators in the regular types of GA, and each of them are different based on the type of the search space and functionality (Gwiazda, 2007). The crossover operator performs an important role in GAs. Most of the crossover operators select the features (genomes) randomly from the parents, therefore the produced chromosome (kid) may not carry the best features of each parent in each mating

process (due to the randomness), and consequently the produced chromosomes are not always better than parents (the chance of having a kid better than the corresponding parents depends on the crossover operator). Although, this random selection of genomes using the crossover provides a thoroughly search of landscapes, it causes the low speed of convergence in GAs, especially in very high-dimensional search spaces. In this study, it is sought to reduce this randomness and provide a more intelligent mating operator which increases the probability of producing a fitting kid from parents. The diversity of population in each generation is another element which should be taken into account. To provide the diversity and also to prevent getting stuck in local minimum, a specific mutation operator is designed and also randomness is added to the crossover operator.

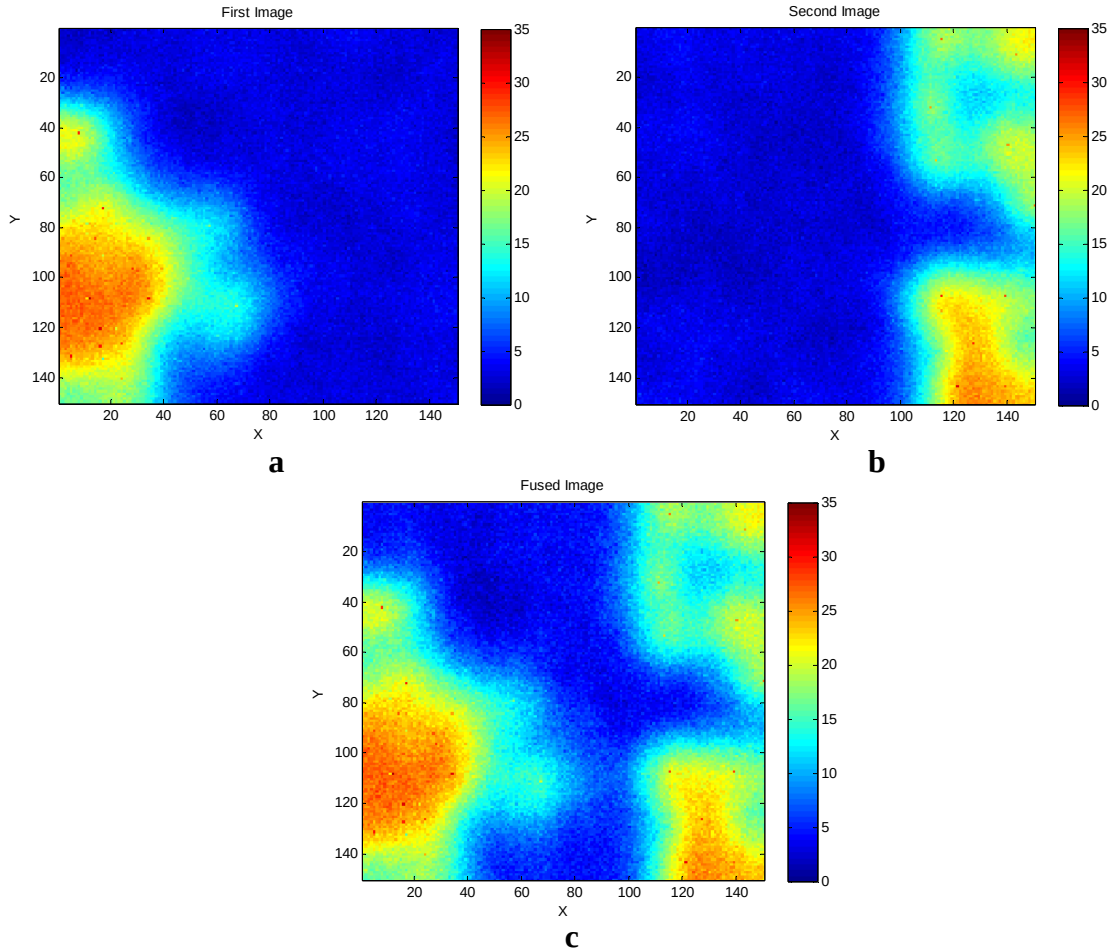


Figure 6.2 Porosity realisations merging by wavelet image-fusion

The designed algorithm has some key differences with the conventional GAs. As mentioned before, one of the main advantages of wavelet image fusion is the capability of keeping the directional information, thus, instead of considering the chromosomes one

dimensional with a length equal to the number of variables, they are considered multidimensional in the proposed algorithm. They are 2D, if the reservoir is single layer and a single property is required to be matched; otherwise they are 3D in which the size of third dimension is equal to number of properties multiply by the number of layers. Figure 6.3 shows a 2D chromosome, it is porosity distribution for a single layer reservoir, and each cell is the corresponding porosity for the reservoir simulator. As the chromosomes are n-dimensional, the population is (n+1)-dimensional.

For the initialisation step, if there is no prior information about the variables which is not normally the case, the initial population will be generated randomly over the given bounds (random initialization). Usually, a prior information is available which are geostatistical estimations (Journel and Huijbregts, 1978). Using this information, a heuristic initialisation can be done which speeds up the optimisation process. The crossover operator is the discrete wavelet image fusion with specified options; the image fusion carries out with "wfusimg" function (MathWork, 2011d). By selecting parents using tournament selection, the kid is created by combining the parents using the mentioned crossover.

The mutation is performed with this procedure: a 2-D rectangular section (with size X,Y) is randomly selected over one of the properties in one of the layer, and then it is substituted with a same size rectangular which has a homogenous value, and the value is selected randomly over the bounds (LB and UB). The mutation is similar to the zonation reparameterisation. Figure 6.4 shows the same chromosome (figure 6.3) which has been mutated. The workflow of the developed algorithm is shown in figure 6.5. The pseudo-code of the algorithm is as following.

```
%---------------- initial population generation
InPop=PopGeneration(NoPop,NoVar);
pop=InPop;
S=size(Pop);
for i=1:NoPop
     fitness(i)=evaluation(pop(i,:,:,:));
end
%------------- start of the main loop
while i<NoGen
    for i=1:p_c×NoPop
        parents = tournament(pop,TourSize);
```

```
        kids(i,:,:,:) = ImageFusion(parents(1,:,:,:),parents(2,:,:,:),options);
    end
    NewPop = recombine(kids,pop);
    for i=1:NoPop
        R=rand;
        if R<p_m
            j=ceil(S(2)×rand);
            IX=ceil(rand×(LX-X));
            IY=ceil(rand×(LY-Y));
            NewPop (i,j,IX:IX+X-1,IY:IY+Y-1) = ones([X Y])×(LB+rand×(UB-LB);
        end
    end
    index=sort(fitness);
    elite = pop(index(1),:,:,:);
    pop=NewPop;
    for i=1:NoPop
       fitness(i)=evaluation (pop(i,:,:,:));
    end
    index=sort(fitness);
    pop(index(end),:,:,:)=elite;
end
Global=pop(index(1),:,:,:);
```

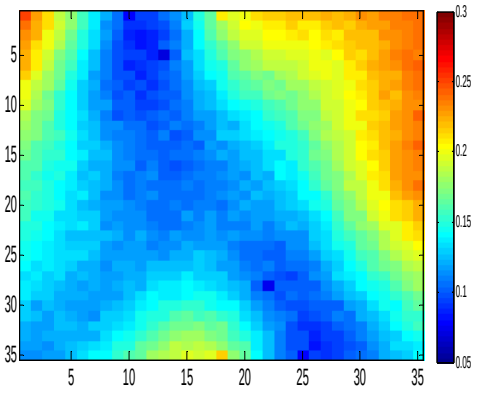


Figure 6.3 A 2-dimensional chromosome

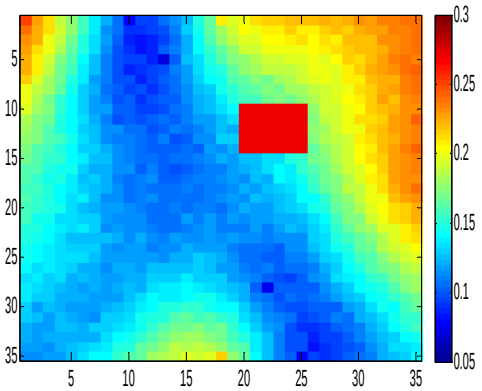


Figure 6.4 The 2-dimensional chromosome after mutation

6.3. Results

History matching is carried out using the designed algorithm for two different reservoir models, a synthetic and PUNQ-S3, and its outcomes are compared with the results of six methods shown in table 6.1. In these two case studies, the variables are the spatial property of the reservoir, therefore a large number of variables are sought to be adjusted. The last two methods (5 and 6) are not regular methods for history matching particularly with a

full-parameterised model, and thus far, the author has not found any stochastic algorithms to be examined for history matching problems without reparameterisation. The two stochastic optimisers (GA and ABC) are employed to investigate their applications in history matching problems with full-parameterised models. The outcomes of these two optimisers are compared with the gradient-based algorithm's result.

In order to execute history matching automatically, flow simulator (ECLIPSE-E100) (Schlumberger, 2010) is coupled to the coded proposed algorithm within MATLAB (MathWork, 2011b). The interface between MATLAB and SGeMS is also used.

Table 6.1 Applied approaches for comparison

Method	Reparameterisation method	Optimisation method
1	Zonation	LBFGS
2	Spectral decomposition of prior covariance	LBFGS
3	Pilot point	GA
4	Full-parameterisation	LBFGS
5	Full-parameterisation	GA
6	Full-parameterisation	ABC

6.3.1. Synthetic Reservoir Model

A synthetic model was constructed with 35×35×1 gridblocks in x, y and z directions respectively. The sizes of gridblock in x, y and z directions are 80, 80 and 100 ft. The porosities in gridblocks are assumed unknown while the permeability is a function of porosity as equation 2. The permeability is isotropy in the horizontal directions, while there is a vertical anisotropy (the permeability in the vertical direction is 10 percent of the horizontal direction). This reservoir is two-phase (oil and water) system. The well pattern is 9-spot with 5 injectors and 4 producers. The drive mechanism is mostly water injection. The location of wells and the horizontal permeability distribution of the reference case are shown in figure 6.6.

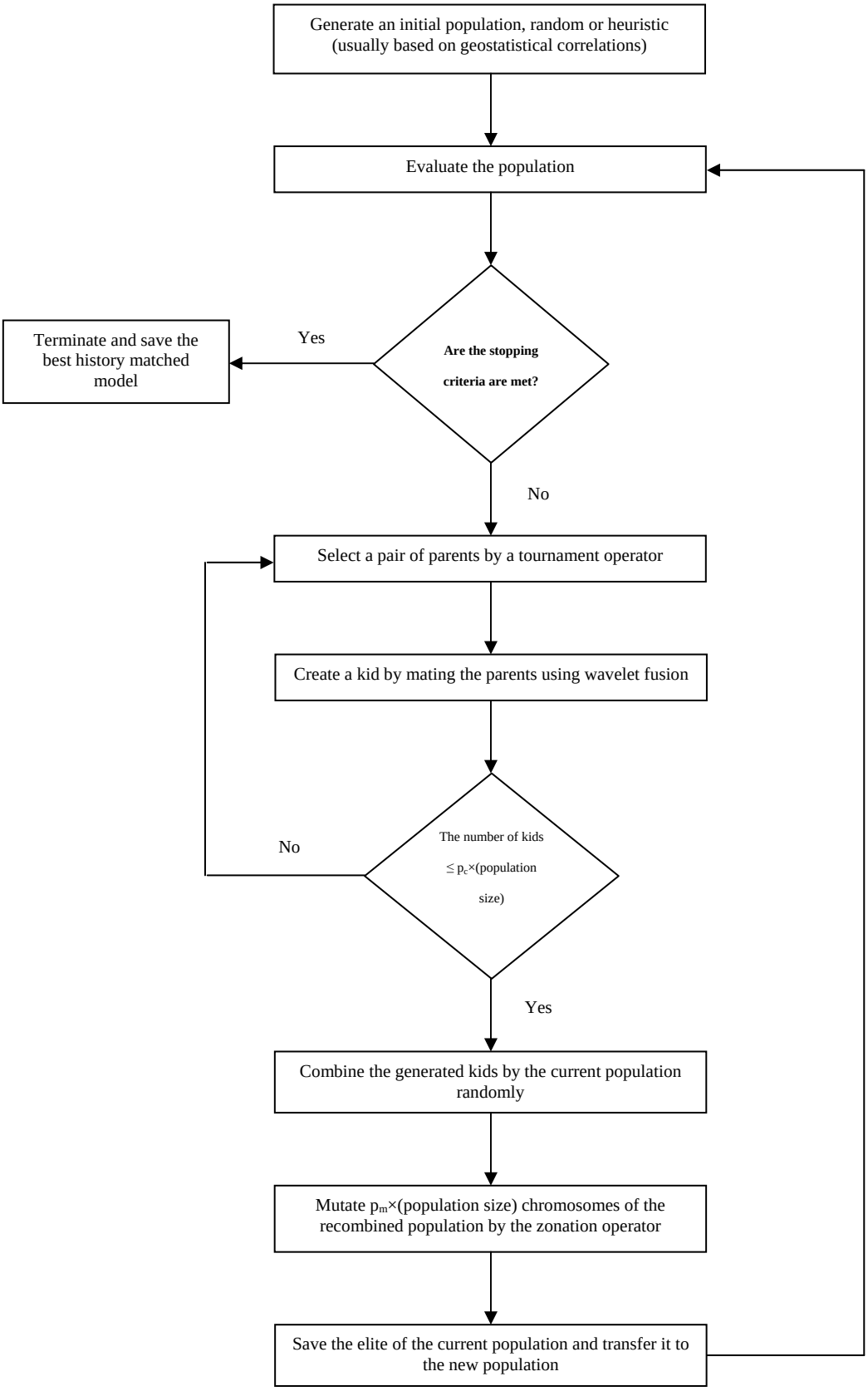


Figure 6.5 Workflow of the developed history match algorithm

The reference case was simulated for 1860 days in 313 time steps to generate observed data, and its elements are well bottomhole pressures, well oil production rates, well liquid

production rates, field oil production rates and field water production rates. The total number of non-zero observed data is 5149 (the number of independent observed data are less than 5149, since the field data are correlated with the well data). Gaussian noise with zero mean and standard deviation of one percent was added to the simulated data. There are 9 wells in the system and the porosities in the corresponding gridblocks are considered known. Therefore, the number of variables is 1216.

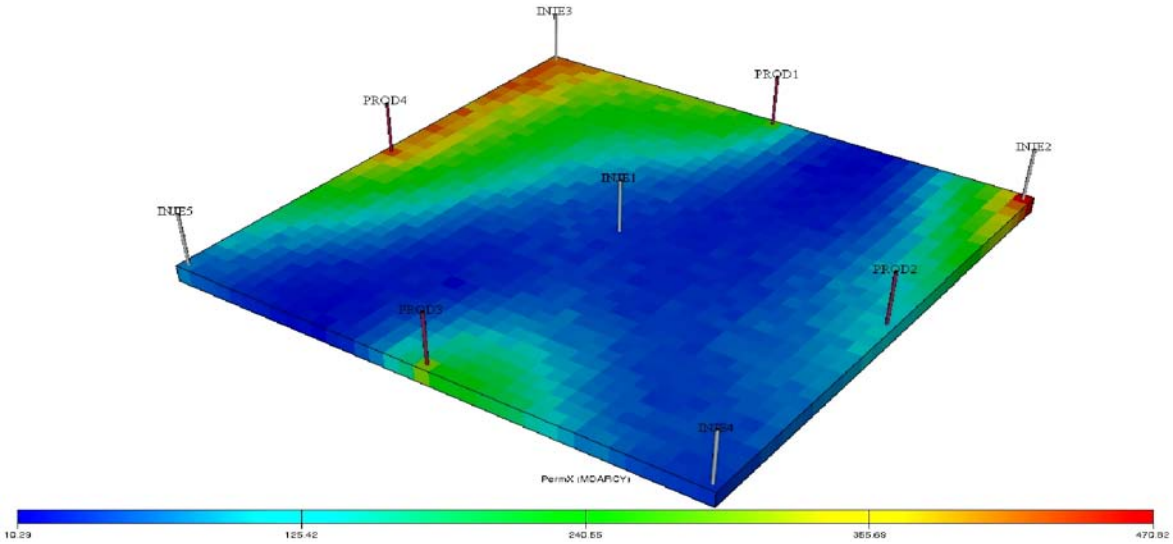


Figure 6.6 The permeability distribution of the reference case and the well locations

The goal of history matching is the estimation of the 1216 variables using the observed data. The objective function is defined as equation 3 (multiply by a negative one (-1)), or as equation 4 (multiply by a negative one (-1)), if the full-parameterised model is being calibrated, or a reparameterised model is being calibrated, respectively. In either of definitions, only the likelihood term is taken into account to investigate overshooting. The fitness function value is equal to approximately (N_D/2) (2642) at the actual solution (the reference model (m_R)). As the observed data was generated via the simulation, the flow modelling error is zero.

$$k(mD) = 30000\phi^3 \tag{13}$$

$$\mathrm{p}(\mathrm{m}|\mathrm{d}_{\mathrm{obs}}) \propto -\mathrm{S}(\mathrm{m}) = \left(-\frac{1}{2}(\mathrm{g}(\mathrm{m}) - \mathrm{d}_{\mathrm{obs}})^{\mathrm{T}}\mathrm{C}_{\mathrm{D}}{}^{-1}(\mathrm{g}(\mathrm{m}) - \mathrm{d}_{\mathrm{obs}})\right) \tag{14}$$

$$p(m|d_{obs}) \approx p'(r|d_{obs}) \propto -S'(r) = \left(-\frac{1}{2}(f(r) - d_{obs})^T {C_{D'}}^{-1}(f(r) - d_{obs})\right) \quad (15)$$

All the six methods and the proposed algorithm are utilised to solve this inverse problem. To distinguish between the methods four criteria are used:

1- The fitness value at the obtained point after history matching ($S(m_\infty)$) (it should be close enough to 2642),

2- The $Q(m_\infty)$ value which shows the quality of the obtained history matched model quantitatively (equation 5),

3- The accuracy of forecasting for two different scenarios based on equation 6 ($P_1(m_\infty)$ and $P_2(m_\infty)$), and

4- The computational costs (the number of fitness function calls).

The two scenarios are i- the performance prediction for five years with a same injection rates scenario as the last timestep, ii- the performance prediction for five years with a dissimilar injection rates scenario. In order to show the quality of prediction quantitatively, equation 6 is used. In this equation, V_R is a vector which consists of the corresponding performance of m_R, and V_C is a vector which consists of the corresponding performance of m_∞. Any method which delivers the minimum of S, Q and Ps with a reasonable computational cost can be called a successful method. It should be mentioned that due to computational cost restrictions, a limit is set for the number of fitness function calls (20,000 times), i.e., none of the methods is allowed to evaluate the fitness function more than 20,000 times even for the sensitivity vector calculation.

$$Q(m) = \sqrt{\frac{1}{N_m}\sum_{i=1}^{N_m}\left(m_{\infty_i} - m_{R_i}\right)^2} \quad (16)$$

$$P(m) = \sqrt{\frac{1}{N_{elements}} \sum_{i=1}^{N_{elements}} \left(V_{Ri} - V_{Ci}(m)\right)^2} \qquad (17)$$

6.3.1.1. Zonation reparameterisation with LBFGS

In this method, the reservoir first is divided into 25 zones (7×7 gridblocks), and it is assumed the properties are homogenous in each zone. Using this reparameterisation, the number of variables is significantly reduced (from 1216 to 25). Then, a LBFGS algorithm (D. Kroon 2010) is utilised to find the (r_∞). The sensitivity vector is computed numerically in each iteration. The initial point is assumed uniform and equal to the mean of the 9 gridblocks value. r_∞ is found with 1117 function calls in 35 iterations, and the corresponding fitness function ($S(h(r_\infty))$) is 7478. m_∞ along with m_R are drawn in figure 6.7. The optimisation in this method needs the minimum computational costs. The computation still can be reduced, if a proper initial guess and optimal options for LBFGS are set.

The achieved model only illustrates the main features of the reservoir with a low resolution. The fitness value along with $P(m_\infty)$ and $Q(m_\infty)$ are reliable proofs for the reparameterisation error. The difference of m_∞ and m_R based on equation 5 is 4.2, and the $P(m_\infty)$ is equal to 4.43 and 2.36 for first and second scenario respectively. According to the fact that the optimisation error is negligible for this approach[6], it is fair to relate the inaccuracy of the achieved history matching model to the reparameterisation error. To reduce the history matching error in this method, it is required to increase the number of zones which at the extreme case will be the full-parameterisation. Also, the error may be slightly reduced by making use of a stochastic optimiser, if the computation costs are not the concern.

[6] This case with zonation reparameterisation was also executed by a Simulated Annealing algorithm; the achieved result was slightly better than the obtained result via LBFGS; the obtained fitness value was 6207. The result of LBFGS is reported, as it needs less computation.

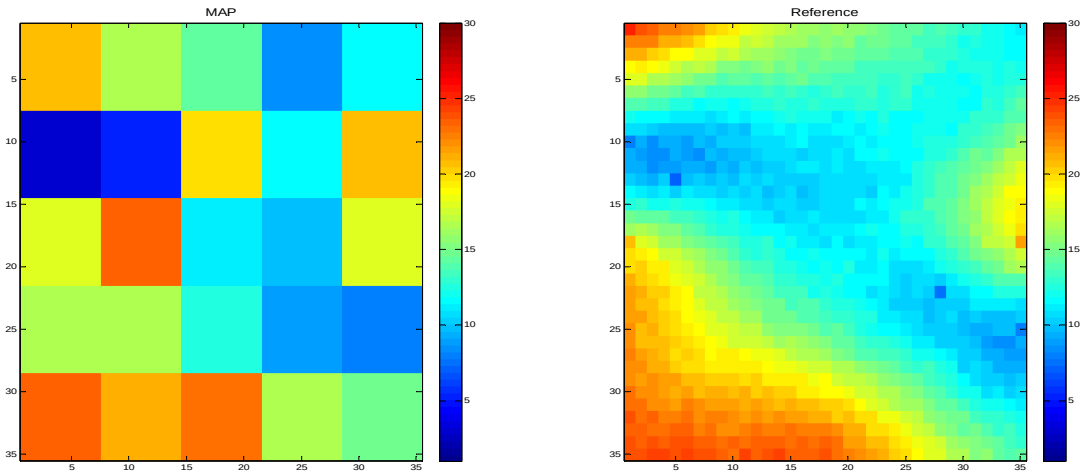


Figure 6.7 The result of zonation reparameterisation with the LBFGS and the reference porosity distribution

6.3.1.2. Spectral decomposition of prior covariance matrix reparameterisation with LBFGS

In this method, a prior covariance and a prior model (m_{prior}) are required; therefore, SGeMS by the available 9 point data generates a set of prior models. The best realisation in terms of fitness value is saved as the prior model and shown in figure 6.8 ($S(m_{prior})=8.2\times10^{5}$). The first 100 eigen values are used for reparameterisation. The number of variables is reduced from 1216 to 100. LBFGS algorithm is utilised to find the (r_{∞}). The initial point (r_{prior}) is calculated by reparameterising m_{prior}. r_{∞} is found with 3002 function calls in 28 iterations, the corresponding fitness function ($S(h(r_{\infty}))$) is 4857. m_{∞} along with m_R are drawn in figure 6.8. In terms of computational cost, this method is slightly more expensive than the previous method (the fitness function has been evaluated 3002 times).

This method has a better performance, in terms of fitness value, $P(m_{\infty})$ and resolution, in comparison with the previous method. The $P(m_{\infty})$ is equal to 2.67 and 2.12 for first and second scenario respectively. But, in this method, numerous extreme values are seen in the m_{∞}, which cause a high value for $Q(m_{\infty})$ (6.06). It is fair to expect that the reparameterisation error is dominant compared to the optimisation error[7]. In order to

[7] This case was also executed via a stochastic optimiser, and the achieved optimum point was slightly better than the obtained result via LBFGS (the obtained fitness value was 4177). The output of the LBFGS was presented, as it needs less computation.

reduce the history matching error in this method, it is required to increase the number of eigenvalues which at the extreme case will be again the full-parameterisation. Another restriction of this method is that an accurate prior covariance matrix is required which is not always available. Also, the error may be slightly reduced by making use of a stochastic optimiser, if the computation costs are not the concern.

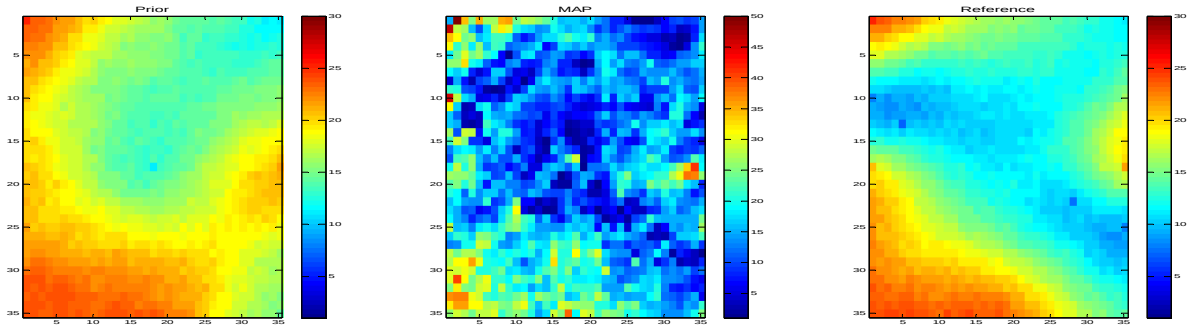


Figure 6.8 The result of the spectral decomposition of prior covariance matrix reparameterisation with the LBFGS, the reference porosity distribution and prior model

6.3.1.3. Pilot point reparameterisation with a genetic algorithm

In this method, the pilot point reparameterisation is used (De Marsily et al., 1984). A limited number of the 1225 gridblocks are described by pilot points, and the rests are calculated through ordinary kriging based on the pilot points and well data. In this study, a fixed number of pilot points (11 points) are used to describe the reservoir. Their values (porosity) and locations (x,y) are considered as decision variables. Also, it is assumed that variogram parameters, nugget, range, contribution and anisotropy angels (Kelkar and Perez, 2002) are unknown as well. The total number of variables is reduced to 38 (11×3+5) via this reparameterisation. The locations of pilot points are not fixed to prevent sensitivity analysis for locating the points. When the location and value of pilot points are both considered as variables, a discrete and multimodal search space will be formed which makes the optimisation process challenging. A GA carried out with the mentioned options in table 6.2, is used to find the r_∞. r_∞ is found with 10,000 function calls in 500 generations. Scattered crossover is used, as the locations of pilot point are discrete variables, and also in

this function, exploration is more important rather than exploitation (refer to the fourth chapter).

The corresponding fitness function ($S(h(r_\infty))$) which is higher than the previous methods is 14,536. m_∞ along with m_R are drawn in figure 6.9, and Q is equal to 3.17. In terms of computation, this method is more expensive than the previous methods. It has the enough resolution and also has a lower Q, but, in terms of fitness value, computational cost and $P(m_\infty)$s, this method is not as good as the pervious methods ($P(m_\infty)$ is equal to 7.58 and 3.32 for the first and second scenario respectively). In this method, it is difficult to state that the optimisation error is negligible compared to the reparameterisation error, thus the reason of misfit (history matching error) is considered as a combination of the reparameterisation and optimisation error. In order to reduce these errors, it is required to have an optimal number of pilot points and a larger number of GA generations.

Table 6.2 GA options

Options	Population size	Crossover operator	Selection operator	Mutation	Stopping criteria	Crossover probability	Number of Elites
	20	Scattered	Stochastic Uniform	Uniform with Rate=0.11	Generation number (500)	0.8	1

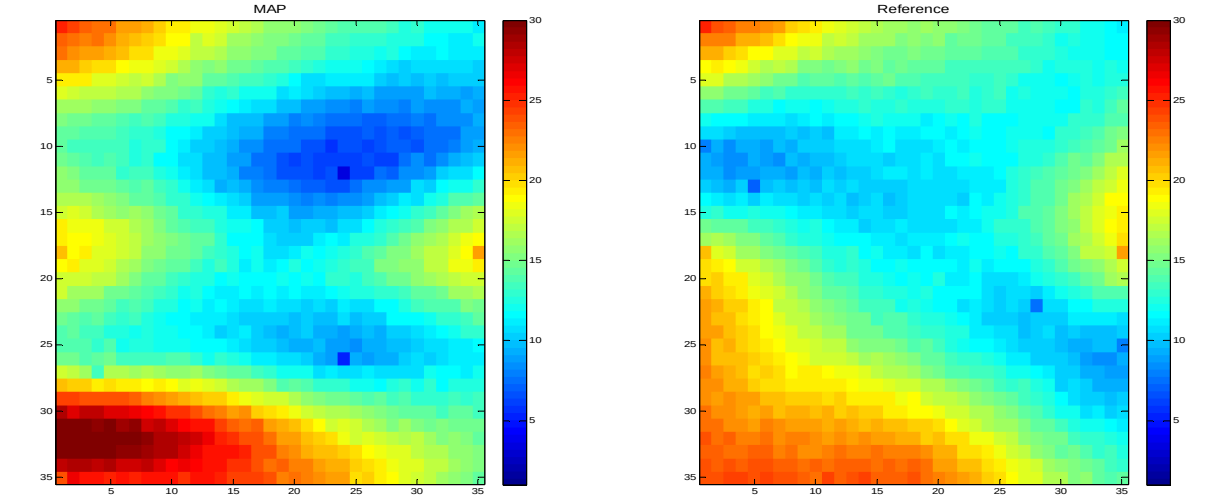


Figure 6.9 The result of the pilot point reparameterisation with the GA and the reference porosity distribution

6.3.1.4. Full-parameterisation with LBFGS

In this method, all the variables are directly calibrated using a LBFSG algorithm which a gradient-based algorithm. The sensitivity vector is computed numerically in each iteration.

m_{prior} is used as the initial point. m_{∞} is found with 19880 function calls in 16 iterations, and the corresponding fitness function ($S(m_{\infty})$) is 90334. m_{∞} along with m_R are drawn in figure 6.10. Extreme values are seen in the gridblocks where wells are located. The optimisation in this method reached to the computational cost limit, while the fitness value is not close enough to the actual value. The low speed of convergence is due to the nonlinearity of the problem. This model has a high resolution and a zero reparameterisation error, but the optimisation error became the main concern. The fitness value along with $P(m_{\infty})$ and $Q(m_{\infty})$ are reliable proofs for the optimisation error. $Q(m_{\infty})$ and $P(m_{\infty})$ are equal to 4.18, 7.39 and 12.89 for first and second scenario respectively. To reduce the history matching error in this method, it is required to increase the number of iterations which will be time-consuming; there is no guarantee that by increasing the iteration, a better result can be achieved, since the algorithm may converge to a false or local optimum point.

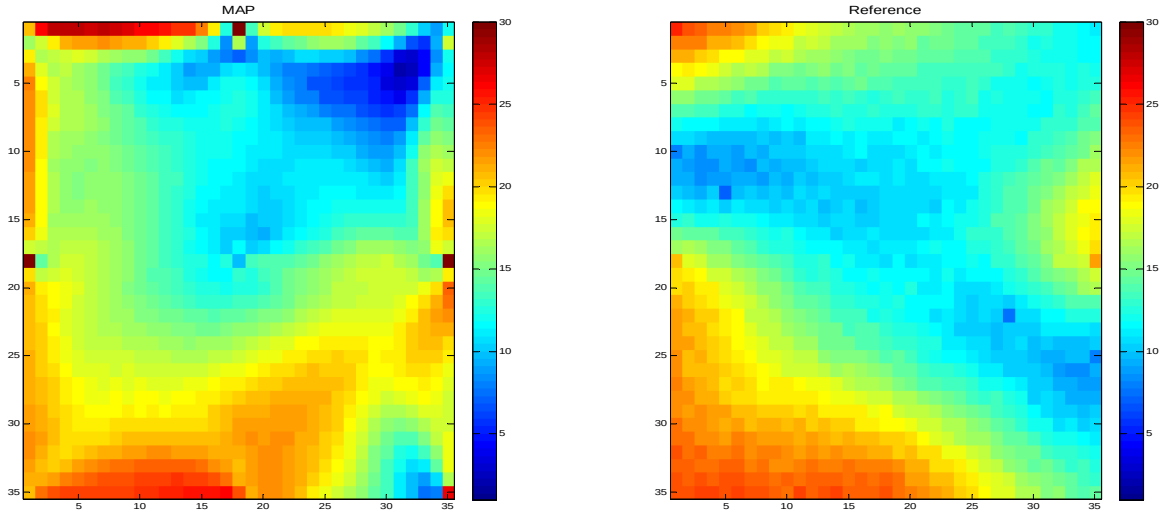


Figure 6.10 The result of the full-parameterisation with the LBFGS and the reference porosity distribution

6.3.1.5. Full-parameterisation with GA

In this method, all the variables are directly calibrated using a conventional GA algorithm. The GA options are similar to table 6.2, but the population size is 40, and a heuristic initialisation (instead of a random initialisation) is used to reduce the computational costs. 40 of the generated distributions for the second method are used as the initial population. m_{∞} is found with 20000 (maximum allowable) function calls in 500 generations, and the

corresponding fitness function ($S(m_\infty)$) is 19512. m_∞ along with m_R are drawn in figure 6.11. Extreme values are seen. The optimisation in this approach reached to the limit, while the fitness value is not close enough to the actual value. The fitness value is lower than the previous method, but it is not as good as the methods with a reparameterisation. The low speed of convergence is due to the random search and high-dimensionality of search space. The achieved model has a high resolution and a zero reparameterisation error, but the optimisation error caused the low-quality model. The fitness value along with $P(m_\infty)$ and $Q(m_\infty)$ are reliable proofs for the optimisation error. $Q(m_\infty)$ and $P(m_\infty)$ are equal to 8.80, 6.19 and 6.49 for first and second scenario respectively. To reduce the history matching error in this method, it is required to increase dramatically the number of generations which will be very time-consuming.

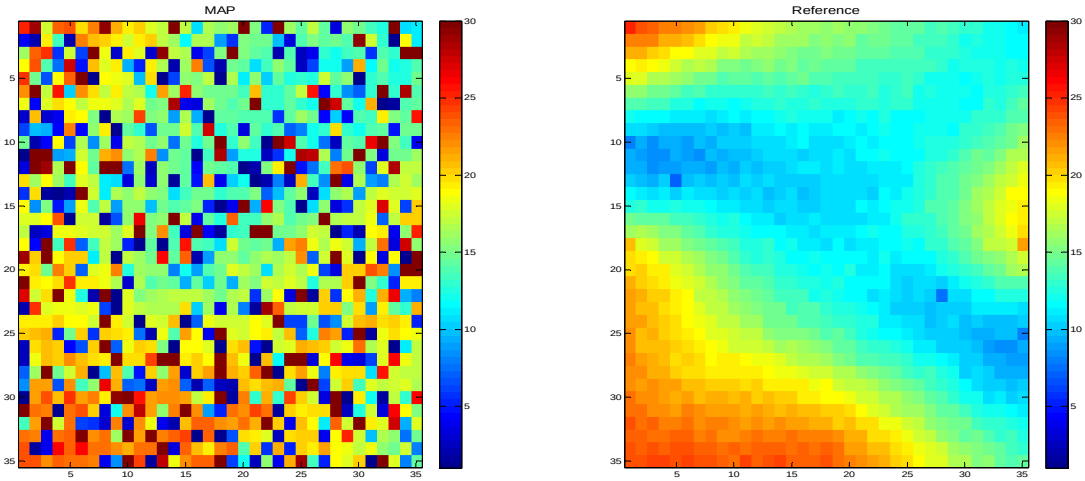


Figure 6.11 The result of the full-parameterisation with the GA and the reference porosity distribution

6.3.1.6. Full-parameterisation with ABC

In this method, all the variables are directly calibrated using an ABC algorithm. The colony size is 40, and a heuristic initialisation (instead of random initialisation) is used to reduce the computational costs. 20 of the generated distributions for the second method are used as the initial colony for employed bees. m_∞ is found with 20000 function calls in 500 iterations, and the corresponding fitness function ($S(m_\infty)$) is 639,394. m_∞ along with m_R are drawn in figure 6.12. m_∞ is similar to the m_{prior}, it shows that this algorithm performs very

inefficient in high-dimensional search. The optimisation in this method reached to the computation cost limit, while the fitness value has not been close enough to the actual value. The fitness value is higher than the previous methods. The low speed of convergence is due to random and mostly one-dimensional search (employed and onlooker bees altered the individuals through a random change in only one-dimension). The achieved model is high-resolution and has a zero reparameterisation error, but the optimisation error is very high, even more that the pervious method. The fitness value along with $P(m_\infty)$ and $Q(m_\infty)$ are reliable proofs for the optimisation error. $Q(m_\infty)$ and $P(m_\infty)$ are equal to 4.31, 20.78 and 28.83 for first and second scenario respectively. To reduce the history matching error in this method, it is required to increase dramatically the number of iterations which will be very time-consuming (even more expensive than the previous method, as the ABC preference is a balance of exploration and exploitation rather than exploration, in contrast to GA (refer to chapter 4).

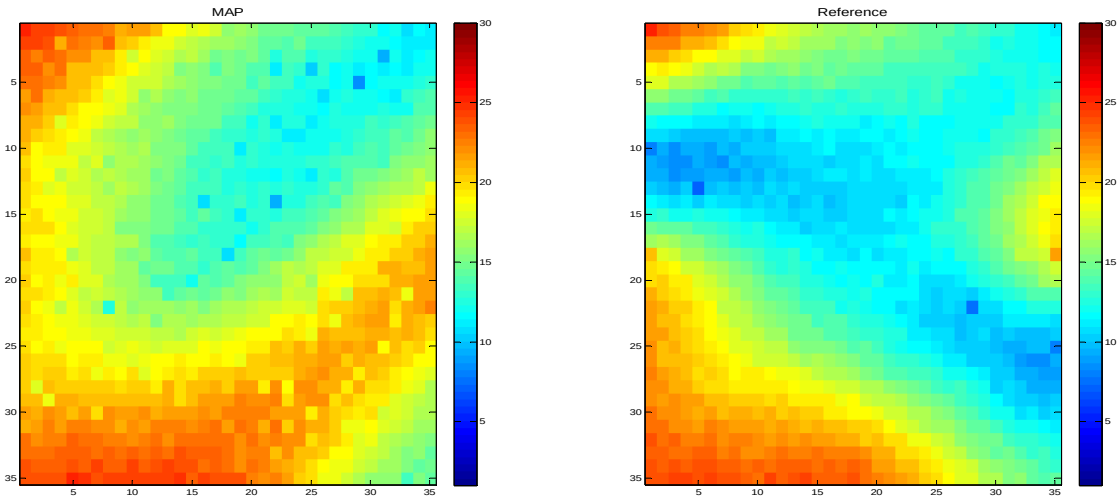


Figure 6.12 The result of the full-parameterisation with the ABC and the reference porosity distribution

6.3.1.7. Full-parameterisation with the proposed method

In this method, all the variables are directly calibrated using the proposed algorithm. The applied options can be found in table 6.3 and 6.4. For the mutation, rectangular with three different sizes are used: 5×5, 4×8 and 8×4; in each mutation, one of them is randomly selected. A heuristic initialisation is used to reduce the computational costs. 15 of the

generated distributions for the second approach are used as the initial population. m_∞ is found with 15000 function calls in 1000 generations, and the corresponding fitness function ($S(m_\infty)$) is 3000 which is significantly lower than all the six methods. m_∞ along with m_R are drawn in figure 6.13. The achieved model has a high resolution and a zero reparameterisation error, and also the optimisation error is plausibly low. All the values of m_∞ are in the actual bound and overshooting has not occurred. The fitness value along with $P(m_\infty)$ and $Q(m_\infty)$ are reliable proofs for the low optimisation error. $Q(m_\infty)$ and $P(m_\infty)$ are equal to 2.29, 0.96 and 0.91 respectively. Figure 6.14 shows the fitness value in each generation. In the first 200 generations, a very fast reduction in the fitness value can be seen which shows the ability of this algorithm.

Table 6.3 The proposed approach options

GA Options	Population size	Stopping criteria	Crossover	Mutation	Selection	Initialization	Chromosomes
	15	Number of generations 1000	Image-Fusion, Prob.: 0.9	Customised Prob.: 0.4	Tournament size 4	Heuristic	2 dimensional

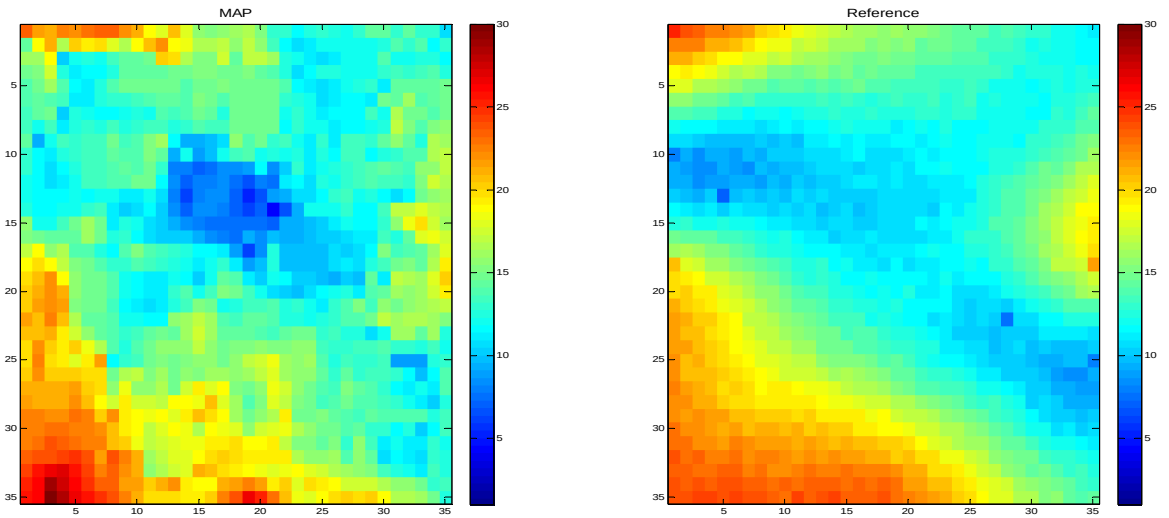


Figure 6.13 The result of the full-parameterisation with the proposed method and the reference porosity distribution

Table 6.4 The image fusion operator options

Wavelet Options	Wavelet Family	Level	Approximation	Details
	Symlet 4	5	Random	Random

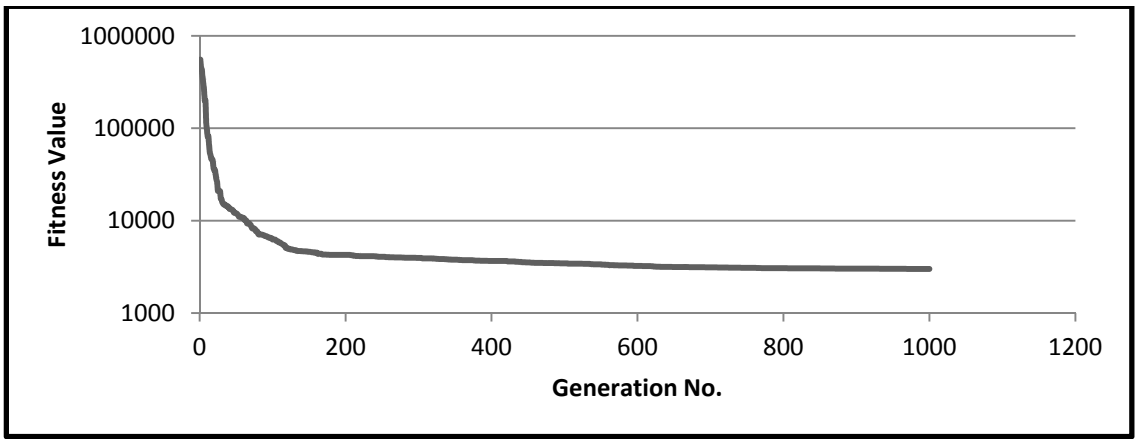


Figure 6.14 Fitness value at each generation for the 1st case study using the proposed method

6.3.1.8. Comparison

The outcomes are summarised in figure 6.15 to 6.17 which show the fitness value, the difference of the achieved model with the reference model (Q), and the accuracy of future predictions (P) for each method, respectively. In the first three methods, it is seen that they cannot provide very high-quality history matched models, which is mostly due to the reparameterisation errors. The main difficulty of these three methods is the determination of the optimal number of the elements and the reparameterisation designs. Among these three methods, the spectral decomposition slightly performed better[8].

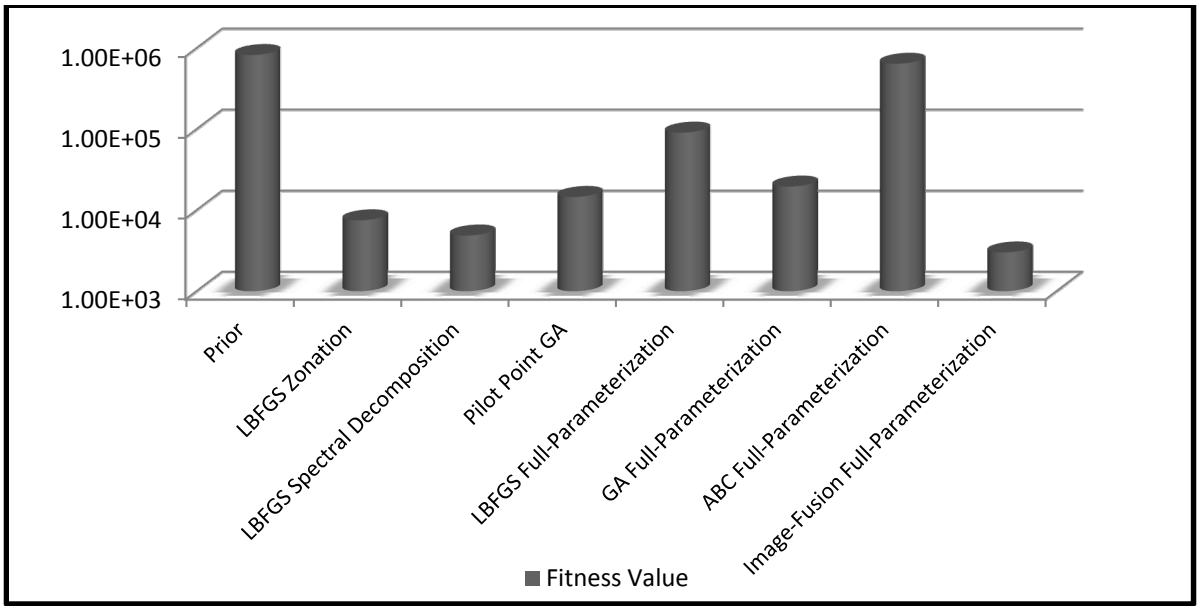


Figure 6.15 Fitness value at achieved models using each of the approaches for the 1st case study

[8] It should be pointed out that there will be a reparameterisation error in any method, and it is as a result of the fact that a lower number of variables (reparameterised model) cannot carry the same amount of information as full-parameterisation.

None of the 4th to 6th approach delivered good results (even the results are worse than the first three methods), although, the reparameterisation error was zero for these three methods. The low quality of the models is on account of optimisation errors. The optimisation errors may be reduced, if the fitness function is called much more times. Due to computation limitation (hardware and software), it was not possible to find the global point using the 4th-6th approach. A same limit was set for the seven methods to provide a fair assessment. The proposed approach delivered a superior model in comparison with all the six methods which can be on account of its low optimisation error.

The execution of each fitness function calls took around 8 seconds on a personal computer with the following configurations: CPU Intel i-7-2820QM and 8 GB ram, it mean that in the methods that the fitness was called 20,000, history matching was finished around 2 days. The computation for each fitness function call was slightly higher for the methods with a reparameterisation, as an inverse reparameterisation calculation in each fitness function call is required.

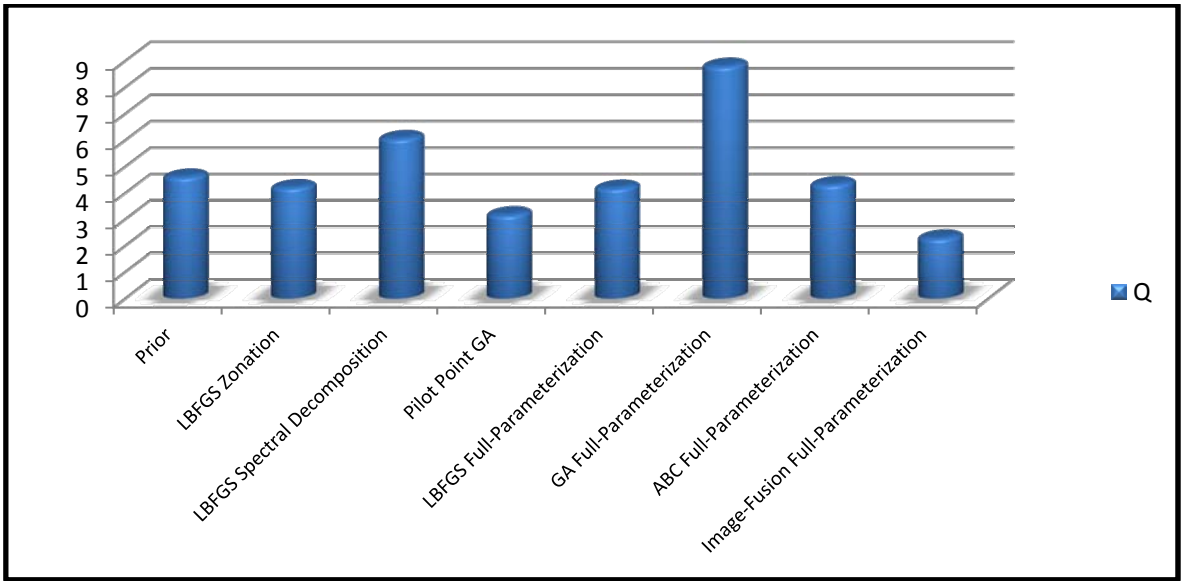


Figure 6.16 Q value at achieved models using each of the approaches for the 1st case study

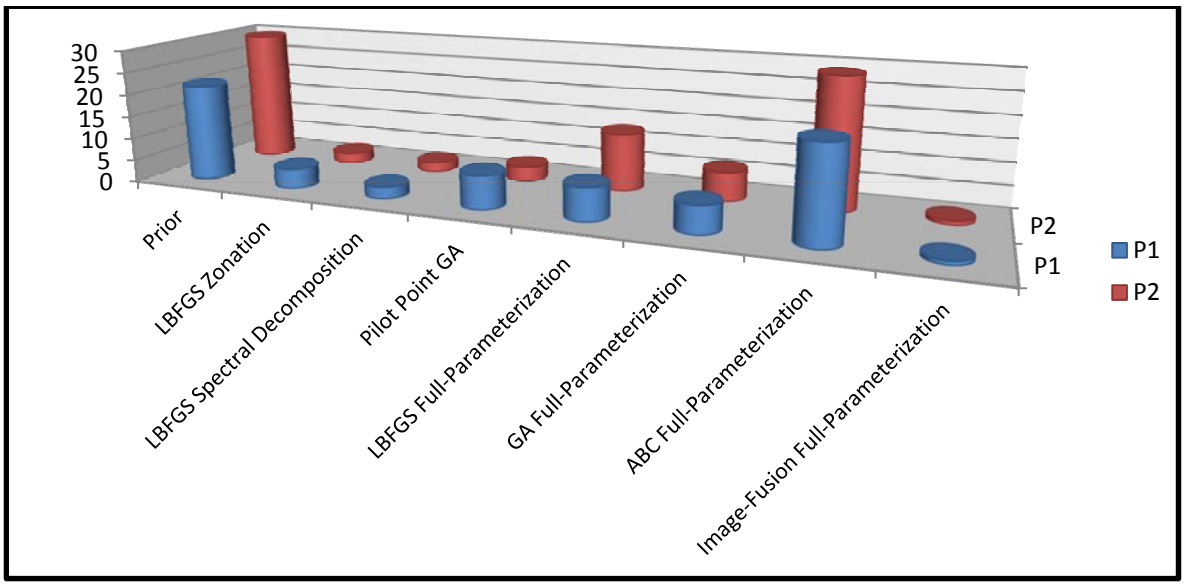


Figure 6.17 P values at achieved models using each of the approaches for the 1st case study

6.3.2. PUNQ-S3 Reservoir Model

For the second case study, PUNQ-S3 reservoir model is used. PUNQ-S3 is a reservoir model widely used for benchmarking different history matching methods. This case was provided by Imperial Collage London University. It is a part of a real field performed by Elf Exploration Production. The reservoir model consists of 19×28×5 gridblocks and has 1761 active blocks. The reservoir is connected to a fairly strong aquifer and also, it has a gas cap. There are 6 production wells and can be seen in figure 6.18. More details about this model can be found in Romero et al. 2000 paper. There are some modifications in parts of the problem. The modifications are as following, 1- the horizontal permeabilities are assumed isotropic and a function of porosity, as equation 2; 2- the vertical permeability is assumed 25 percent of the horizontal permeability; and 3- the number of timesteps are increased to 165 which include 10 years performance history. The reference model (figure 6.17) is simulated to generate a history according to the timesteps. The total number of observed data elements is 4455 elements which are cumulative field gas production, cumulative field water production, cumulative field oil production, well water-cuts, well gas oil ratios, well oil rates and well bottomhole pressures. Gaussian noise with a zero

mean and one percent standard deviation were added to the observed data (history). The goal of history matching is the estimation of the 1761 variables using the observed data. The objective function is defined as equation 3 (multiply by a negative one (-1)) and only the likelihood term is taken into account. The fitness function value is equal to approximately ($N_D/2$) (1923) at the actual solution (the reference model (m_R)).

To solve this inverse problem, only two methods are used: 1-the proposed algorithm, and 2- a full-parameterisation with a LBFGS algorithm. To distinguish between the two applied methods the following two criteria are used: 1- the fitness values at the achieved models (MAPs), 2- the cumulative produced oil for the 5 years via the existing wells, 3- the cumulative produced oil for the 5 years via the existing and three infill wells. The reference values for the second and third criterion are **1,181,886** and **1,340,497** standard cubic meter. It should be mentioned that due to computational cost restrictions, a limit is set for the number of fitness function calls (20,000 times). To generate a prior model for the LBFGS algorithm and also to generate a set of prior models for the heuristic initialisation of the proposed approach, 20 realisations are generated via a sequential Gaussian simulation in which the 6 well data are used. m_{prior} is the best model among the all 20 realisations in terms of fitness function ($S(m_{prior})=1.92\times10^7$)

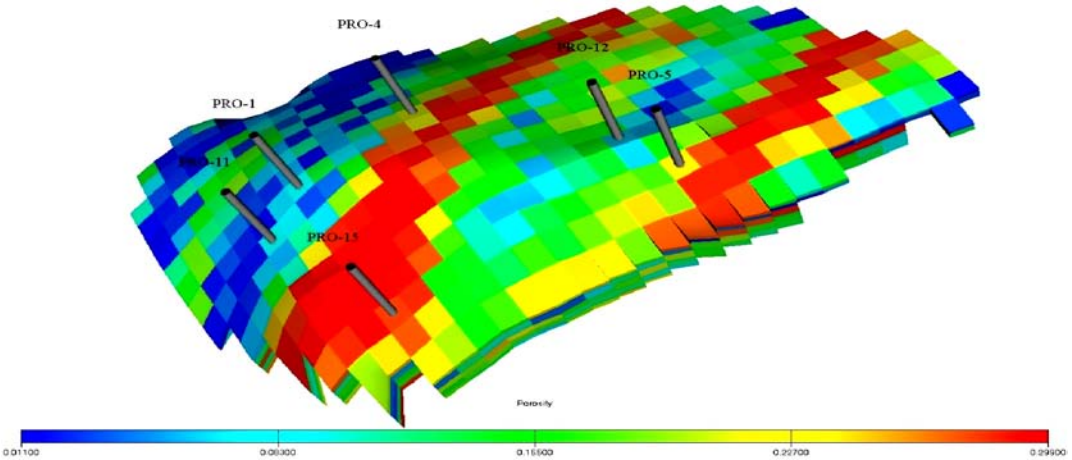


Figure 6.18 The porosity distribution of the reference case and the well locations (2nd case study)

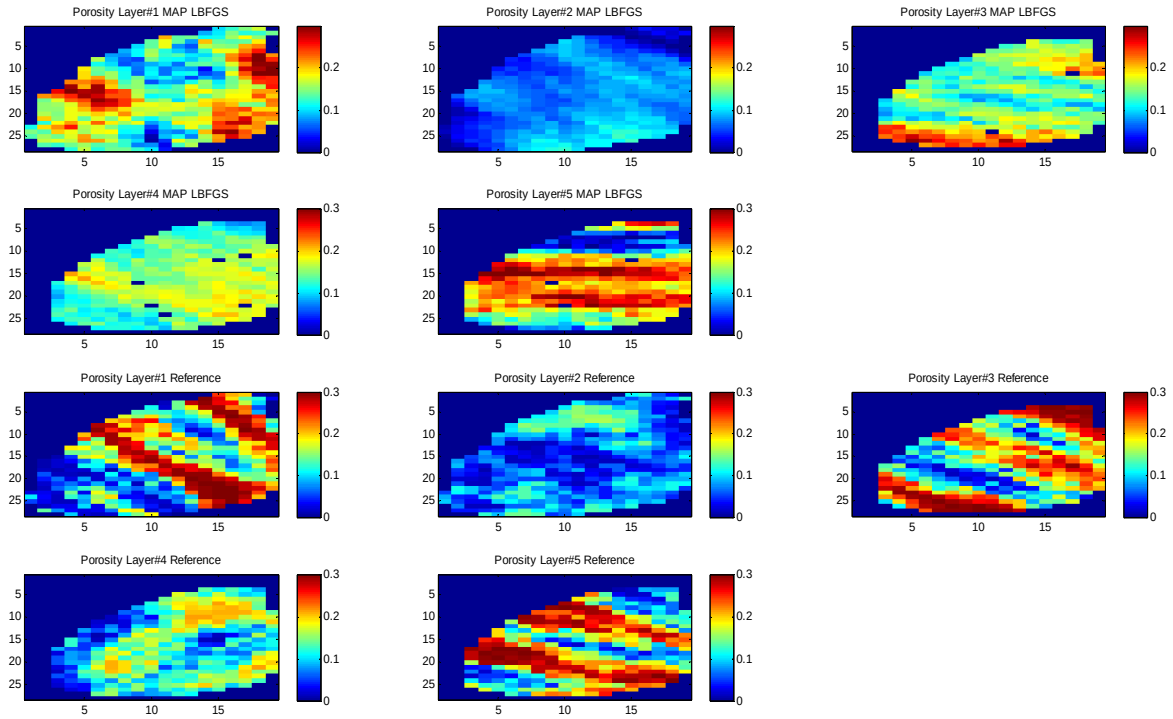


Figure 6.19 Achieved porosity distribution for each layer of PUNQ-S3 using the LBFGS approach along with the reference porosity distribution

6.3.2.1. Full-parameterisation with LBFGS

In this method, all the variables are directly calibrated using a LBFSG algorithm. The sensitivity vector is computed numerically in each iteration. m_{prior} is used as the initial point. m_{∞} is found with 17920 function calls in 8 iterations, and the corresponding fitness function ($S(m_{\infty})$) is 8.6×10^{6}. m_{∞} along with m_R are drawn in figure 6.19. Using the history matched model, the cumulative produced oil for five years via the existing wells is **1,786,109** standard cubic meter, and the cumulative produced oil for five years via the existing and the infill wells is **1,935,520** standard cubic meter.

6.3.2.2. Full parameterisation with the proposed approach

In this method, all the variables are directly calibrated using the proposed algorithm. The applied options is similar to table 6.3 and 6.4, the only difference is in the dimensional of the chromosomes, as in this case study the porosities of 5 layers are needed to be calibrated the chromosomes and population are 3D and 4D respectively. Also, the number of generations is increased to 1300. For the mutation, rectangular with three different sizes

are used: 4×4, 2×5 and 5×2; in each mutation, one of them is randomly selected. A heuristic initialisation is used to reduce the computational costs. 15 of the generated distributions are used as the initial population. m_∞ is found with 19500 function calls in 1300 generations, and the corresponding fitness function ($S(m_\infty)$) is 2.3×10^4 which is lower than the previous method. m_∞ along with m_R are drawn in figure 6.20. The achieved model has a high resolution and a zero reparameterisation error, and also the closeness of the predicted values for both scenarios to the reference values are proofs for the plausibly low optimisation error. Using the history matched model, the cumulative produced oil for five years via the existing wells is **1,160,252** standard cubic meter, and the cumulative produced oil for five years via the existing and the infill wells is **1,288,752** standard cubic meter. All the values of m_∞ are in the actual bound and overshooting has not occurred.

6.3.2.3. Comparison

Not only the achieved fitness value of the image fusion method is lower than the LBFGS, but also the predicted cumulative produced oil values using the image fusion method are much closer to the reference values. Thus, it is fair to mention that the proposed method is working properly for PUNQ-S3 as well. The CPU-time for each method was around 2 days on the same personal computer. Figure 6.21 shows the field oil production total which are obtained by simulation of the reference model (black curve), the model achieved by image-fusion (green curve) and the model achieved by LBFGS (red curve). The difference in forecasting the future demonstrates the success of the image-fusion in comparison with LBFGS algorithm.

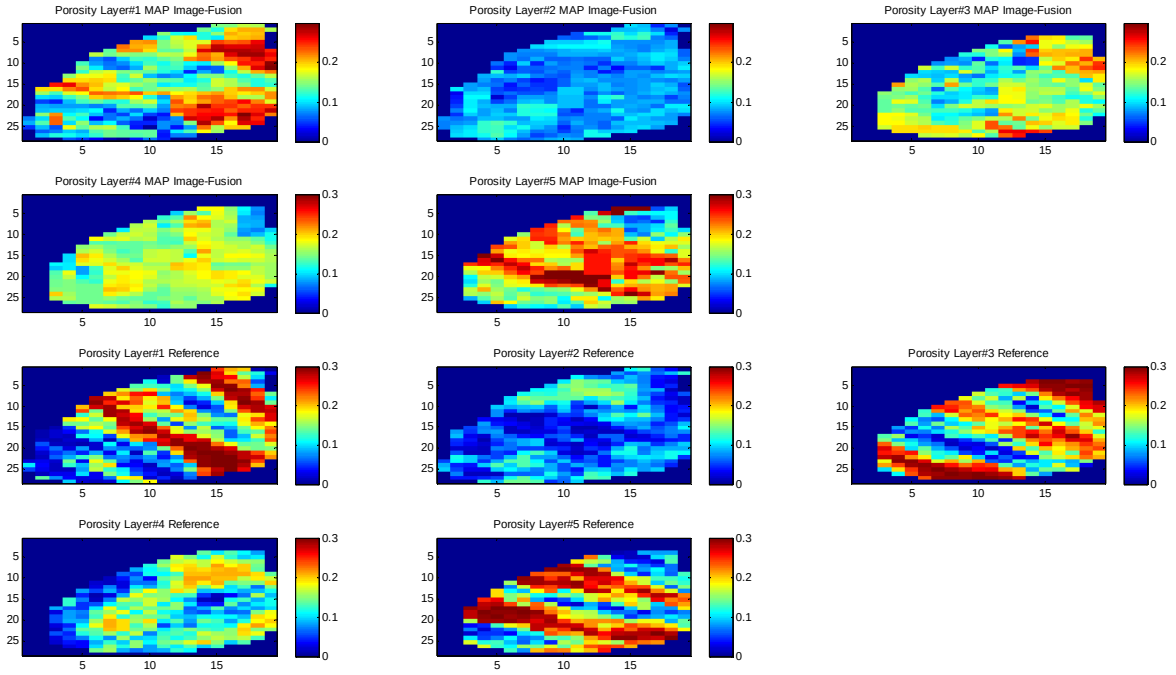


Figure 6.20 Achieved porosity distribution for each layer of PUNQ-S3 using the proposed approach along with the reference porosity distribution

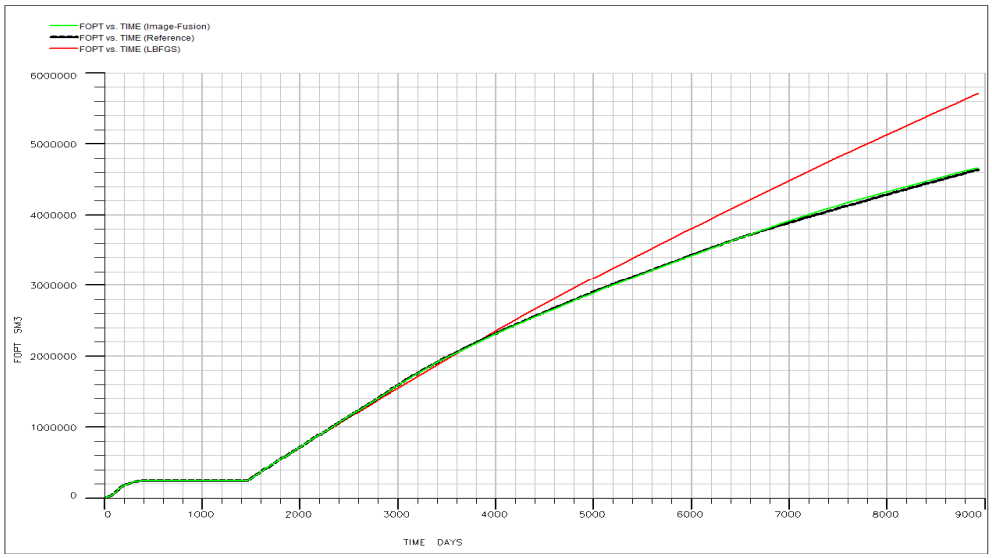


Figure 6.21 Field oil production total for 9000 days

6.4. Discussion

It was shown that the optimisation algorithm constructed from the blend of wavelet image fusion and evolutionary algorithms is able to deal with high-dimensional nonlinear problems. The results of the developed algorithm were superior in comparison with the outcomes of the six algorithms (indirect and direct). It did not converge to a false or a local minimum and also any premature convergence did not happen which can be put on the account of its intelligence mating and mutation operator.

The three indirect methods converged to high-quality reparameterised models in a reasonable computation time, but when the models were transformed back to the main domain, a difference was seen between them and the reference model. The differences are mainly in consequence of reparameterisation error. Among these three methods, the LBFGS with spectral decomposition was performed better. In chapter four, the pilot point technique performed better, as the required number of pilot point was already known and also the variogram parameters were assumed known. Here, the spectral decomposition algorithm was better than the spectral of chapter four, as a larger number of elements were taken into consideration and the prior covariance was more certain.

The three direct methods did not deliver high-quality model that is on account of optimisation errors. Due to the nonlinearity of the inverse problem, and the high-dimensionality of the system, the optimisation algorithms were not very efficient. Among these three algorithms, LBFGS was performed slightly better which expresses the drawbacks of the regular stochastic algorithms in high-dimensional history matching problems. Heuristic initialisation was used for ABC and GA. They would converge to worse solutions, if random initialisation was used. Among these six methods, the indirect methods could deliver better results.

For the assessment, only a number of methods were utilised to validate the algorithm. In these methods, specified options were used. For example, in the GA approach, a specific crossover and mutation operator were used, in the LBFGS approach, the sensitivity vectors were calculated numerically, and in the reparameterisation approaches, a specified number for the elements was used to describe the model. It cannot be stated that all the applied options are the most efficient ones. For the image fusion, a Symlet discrete wavelet transform with level 5 is used, a sensitivity analysis on the effect of image fusion rules on the quality of the algorithm assured us to use this option.

The conclusive remarks of this chapter are as following:

- The developed method has a higher chance to approximate the best history matched model.
- It has an acceptable speed of convergence, even in comparison with classical optimisers.
- Discrete variables can be taken into consideration (discrete variables need their own crossover and mutation operator, and also the shape of chromosomes should be changed to multibranched chromosomes).
- It can cope with a large number of variables, thus, it provides high-resolution reservoir models.
- The image fusion technique can keep the directional information, thus this algorithm is expected to deliver high-quality results for reservoir modelling.
- The algorithm can be executed on a parallel manner on supercomputers.
- Due to its fast reduction in the initial stage, it can be run only for the first generations and after a sharp reduction of fitness value, the algorithm can be substituted with another evolutionary algorithm which has stronger exploitation ability or with a classical optimiser for further reduction of fitness value.
- Among the conventional history matching algorithms, the indirect calibration methods are more efficient than the conventional direct methods, in high-dimensional history matching problems.
- The regular stochastic optimisers are inefficient in high-dimensional search spaces

Chapter 7

7. Regularisation in history matching problems using Pareto front and Bayesian framework

The main emphasis of this chapter is on the role of objective function formulation on the outcomes of history matching. In the previous chapters, several history matching methods were reviewed with the focus on the optimisation and reparameterisation step. Obtaining high-quality history matched models using any of the history matching algorithms is subjected not only to the performance of optimisation and reparameterisation algorithm but also it depends on the reliability of objective function formulation.

In history matching problems, to overcome the ill-posedness and also to stabilise the inverse problem, supplementary information is usually added to the objective function as a penalty term. The procedure of altering the objective function to accomplish a proper compromise between two conflicting terms, (penalty and fitting) is called regularisation (Stark, 1987). Regularisation plays a crucial role in the reliability of the formulation, and it should be carefully implemented into the formulation, especially for highly noisy observation data or underdetermined problems.

Tikhonov method is a widespread deterministic objective function formulation approach. In this formulation, the objective function comprises of two terms (functions), fitting and penalty term which are related to each other using a regularisation factor. The major

challenge of Tikhonov method is the determination of a proper regularisation factor (Li and Jafarpour, 2010) which can be varied from zero to infinity. An inappropriate regularisation factor can lead to an incorrect domination of one of the functions over the other one and accordingly results into a false optimum point. A large weighting factor for penalty term causes inaccurate observation fitting and in contrast, small value can cause highly oscillatory solutions as a result of noise amplification (Tautenhahn and Qi-nian, 2003). So far, several numerical calculation approaches have been developed to determine the proper regularisation factor, including L-curve (Hansen, 2001), generalised cross validation (GCV) (Golub et al., 1979), and by iterative methods (Doherty, 2003). These approaches are time-consuming for computationally expensive problems (Li and Jafarpour, 2010), such as history matching problems.

The most widespread approach of objective function formulation in history matching problems is Bayesian framework which within includes a regularisation. In the Bayesian framework, To find the maximum of posterior probability function (MAP), usually a decision variable is sought which minimises the summation of likelihood and prior functions (S(m)). These two functions are related to each other using two covariance matrixes (C_D and C_m). These covariance matrixes are the weighting factors for the individual elements, and also their ratio plays the task of a regularisation factor.

The quantification of the aforementioned factors is a complicated task. Therefore, the weighting factors may have uncertainties in some case studies. In this chapter, the influence of covariance matrixes' uncertainties on the reliability of models is investigated, and also an approach for reducing the risks in decision making in these particular problems is reviewed. The approach was recommended in a general form for probabilistic inverse problems (Tarantola, 2005) (the author has not found the approach in history matching problems, before). This approach is based on random selection, thus it is expected to be

computationally expensive. Therefore, an alternative approach is also developed and studied in this chapter.

A Pareto optimisation is proposed, for the first time, in which likelihood and prior functions are considered as two separated objective functions. Specific post-optimisation trade-off rules are designed to select the proper results among the Pareto solutions. The questions, the author wishes to answer in this chapter, are:

a- Do the uncertainties in covariance matrixes significantly mislead the optimisation direction to improper models?
b- Can the Pareto optimisation fulfil the requirements in delivering high-quality results for the problems with uncertain covariance matrixes?
c- Can the post-optimisation trade-off role take out the suitable solutions among the Pareto?
d- Does the proposed approach perform better than the conventional approach in terms of computation and the quality of results?

This chapter begins by an introduction to the Pareto optimisation and followed by the explanation of the methodologies (conventional and proposed). In the results section, a linear numerical example is designed and used to investigate the effect of covariance matrixes. In the numerical example, the proposed and the conventional approach are compared with each other. Afterward, the effects of uncertainty of covariance matrixes on the quality of history matching are investigated using PUNQ-S3 model. To evaluate the effectiveness of the proposed approach, its outcomes are compared with the outcomes of the random selection (Monte Carlo) approach in terms of the required computation time and the quality of achieved solutions. The benefits and drawbacks of the methods are discussed, and followed by some conclusive remarks in the summary and discussion section. This chapter is a modified and adjusted version of one of our published papers in a

SPE conference, "SAYYAFZADEH, M., HAGHIGHI, M. & CARTER, N. 2012, Regularization in history matching using multi-objective genetic algorithm and Bayesian framework."

7.1. Pareto optimisation

In multi-objective optimisation problems, there are a number of objective functions $f(m) = [f_1(m), f_2(m), \dots, f_n(m)]', f: M \to R^n$ which are required to be optimised simultaneously. m is a N_m dimensional column vector $m = [m_1, m_2, \dots, m_{Nm}]^t$ whose elements are the variables. Usually, it is impossible to find a single point which meet the conditions of (minimises) all objective functions together (Coello, 2000, Konak et al., 2006), i.e., a single point cannot be the absolute minimum of all the objective functions. Figure 7.1 reveals a system with two objectives (f1 and f2) and its feasible region. Usually, the point which minimises both objective functions together is out of the feasible region. In these problems, a set of solutions called noninferior or Pareto optimal solutions is approximated instead of a single point. Pareto front is located on the bottom border of feasible solutions (Haupt and Haupt, 2004) and it consists of non-dominated solutions. In the Pareto front, all the solutions are equally acceptable, unless supplementary information is available.

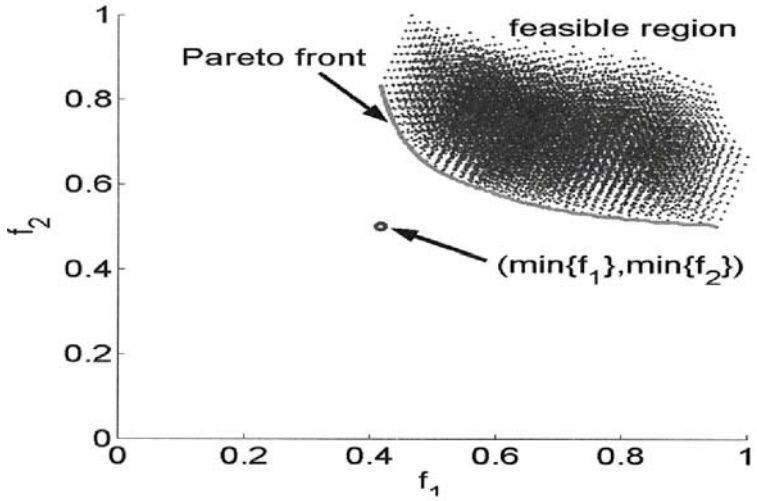


Figure 7.1 Feasible solution and Pareto front for a two-objective problem (Haupt and Haupt, 2004)

Pareto optimal is introduced by Vilferdo Pareto in nineteen century. Pareto optimal, $m^* = [m_1^*, m_2^*, \dots, m_{Nm}^*]$` means that for every m, either

$$\bigwedge_{i \in I} (f_i(m) = f_i(m^*)) \quad (1)$$

Or at least there is one i which (Coello, 2000)

$$f_i(m) > f_i(m^*) \quad (2)$$

Pareto optimal solutions are a set of non-dominated solutions, and their corresponding evaluations are named optimal front. Finding the whole optimal set is almost unachievable, since 1- Pareto front usually consists of an infinite number of solutions, and 2- optimisation error always exists (Coello et al., 2007). Thus, the target, in the multi-objective optimisation problems, is finding a set of solutions which is satisfactorily close to the Pareto optimal set (Konak et al., 2006). After locating the set of solution, usually a number of solutions are taken out from the set based on post-optimisation trade-offs for decision making (Haupt and Haupt, 2004). This step should be carefully designed. It should be mentioned that if the proper weighing factors between the objectives can be assigned, it is more efficient to make use of a single-objective optimisation.

In order to find the Pareto optimal solutions, a multi-objective optimisation is required. Different methods can be applied to solve multi-objective optimisation problems, among these methods, population-based algorithms have considerable robust benefits in comparison with the traditional multi-objective optimisation techniques (Coello et al., 2007). The main advantage of these algorithms is the fact that they optimise a set of solutions (called population), which allows the computation of Pareto front in a single algorithm run. In addition, due to their stochastic search abilities, they are likely to be able to approximate Pareto fronts even in nonlinear functions. To implement the population-based algorithms, a Pareto ranking scheme is used, instead of the regular selection

operators in single-objective algorithms. So far, different ranking algorithms have been developed; these algorithms are reviewed in the literature review chapter.

In several studies, multi-objective optimisation algorithms were used in history matching problems; these studies are reviewed in the third chapter. In all these studies, the likelihood term is split up into several objective functions. In those studies, the objective function did not have a regularisation term. Separating the likelihood term (or any single-objective) into several objective functions needs considerations. If the weighting factors for the individual elements of the likelihood are uncertain, splitting the objective function can be useful, otherwise a single-objective optimisation can be more effectual, since a single point will be obtained which minimises proportionally all the objective functions according to the predefined weighting factors, and additionally, it does not need any post-optimisation trade-off. In addition, in those studies, a specific trade-off rule was not designed and it was not stated that the weighting factors are uncertain.

In this study, a different approach is taken to the application of multi-objective optimisation into history matching problems. It is used for regularisation purposes, since the ratio of the covariance matrixes play a crucial role on the reliability of results. To perform the multi-objective optimisation, a multi-objective genetic algorithm which makes use of a modified NSGA-II (Deb et al., 2002) (called controlled elitist genetic algorithm) is used[9]. In this algorithm, recombination is based on two terms, rank and distance of individual chromosomes. The distance is used to supply diversity in Pareto front which can improve convergence to the optimal Pareto front (MathWork, 2011a). NSGA-II is widely applied especially when the number of variables is limited (Coello et al., 2007). It is fair to state that any type of Pareto optimisation method can be implemented for this study. The comparison of the multi-objective optimisation algorithms does not fall within the scope of this study.

[9] gamultiobj function is used MATHWORK, T. 2011a. gamultiobj.

7.2. Methodology

In the Bayesian formulation, the objective function is the posterior probability function which is a function of two conflicting terms: prior and likelihood (equation 3). The prior function (second term) expresses the deviation from the initial gauss (usually provided by geologists). This function is quadratic and its minimum is the initial (prior) model (m_{prior}). The likelihood function (first term) indicates the quality of fitting of the simulation over history data. It is a nonlinear and non-quadratic function. Its minimum is the maximum likelihood and denoted by $m_{likelihood}$. $m_{likelihood}$ is different from MAP when there is a prior knowledge term into the objective function formulation. To approximate mlikelihood, a nonlinear inverse problem should be solved. To approximate MAPs (the maximum of posterior), a summation of these two functions which are related to each other via the weighting factors are minimised using a single-objective optimisation (refer to chapter 2 for derivation). MAP has a closer distance from the initial guess than $m_{likelihood}$, when the prior knowledge has a larger weighting factor than the likelihood, and it is closer to the maximum likelihood, if the prior knowledge has a smaller weighting factor.

$$S(m) = \frac{1}{2}(d_{obs} - g(m))^t C_D^{-1}(d_{obs} - g(m)) + \frac{1}{2}\left(m - m_{prior}\right)^t C_M^{-1}\left(m - m_{prior}\right) \quad (3)$$

Therefore, achieving appropriate history matched model(s) is subjected to the reliability of the covariance matrixes (weighting factors). To define the covariance matrixes properly, it is required to estimate and quantify several elements, including measurement error of observed data, simulation error (flow modelling error, PVT analysis error, reparameterisation error, and upscaling error), and the uncertainty of prior model. It is not an effortless task. Hence, sometimes the elements of covariance matrixes may have considerable uncertainties. In these conditions, the Monte Carlo approach can be taken

which as following. This approach is called conventional, although this approach has not been studied previously in history matching problems.

7.2.1. The conventional method

A set of covariance matrixes are randomly selected from the estimated ranges according to their probability functions, and the MAP is approximated for each of the selected covariance matrixes by solving the corresponding inverse problem (equation 3). Then, an uncertainty analysis is carried out using the obtained MAPs (Tarantola, 2005). In order to deliver a high-quality uncertainly analysis, it is required to carry out the procedure with a large number of different arrangements. But, due to having a computationally expensive function in history matching problems; it is impossible to evaluate a large number of scenarios, especially when the optimiser is a stochastic method. It is also possible to make use of the mentioned methods for regularisation factor determination, such as L-curve and GCV, as an alternative of the conventional approach. But, they also suffer from being computationally expensive.

7.2.2. The Pareto method

The computation issue of the Monte Carlo approach conducted us to develop an alternative approach. Where the weighting factors between two or more objective functions are unknown or uncertain, it is better to locate the non-dominated solutions (Pareto) via a multi-objective optimisation for the individual objective functions, then using some criteria, pick a number of solutions among the Pareto front.

In this approach, in order to maximise the posterior probability function, prior and likelihood probability functions are considered as two independent objective functions and their corresponding Pareto front is being approximated. A single point (utopia) is rarely available which minimises two independent functions together, unless the prior model

(m_{prior}) is equal to $m_{likelihood}$. By making use of a Pareto (multi-objective) optimisation in history matching problems while the two terms of Bayesian formulation are the two objective functions, a set of solutions named the Pareto front is provided regardless the weighting factors between prior and likelihood. With any covariance matrix, same Pareto solutions will be achieved and inaccuracies in the covariance matrixes cannot allow one objective function to be dominated over the other one. In this approach, it is not required to run several single-objective optimisations to generate a set of solutions for uncertainty analysis, and a single multi-objective execution is enough to find a set of solutions for uncertainty analysis. Hence, it is expected to be computationally efficient.

In figure 7.2, a rough estimation for the feasible region and Pareto front (shown by red line) of history matching objective functions which generally is expected to be achieved is shown. The Pareto shows that in those solutions in which the prior knowledge is minimum, a high-quality matching between history data and observed data will not be obtained; and in those solutions in which the likelihood is minimum, there will be a significant deviation from the initial model. However, these solutions are all non-dominated solutions, they are not all acceptable results and should not be utilised for uncertainty analysis. Thus, it is vital to select a number of solutions among the entire Pareto front via a specific rule called post-optimisation trade-off, in order to enhance the quality of uncertainty analysis.

If there is no further information about the covariance matrixes (the covariance matrixes are completely unknown), the whole solutions on the Pareto front should be used for uncertainty analysis, otherwise a number of solutions can be taken out from the set based on post-optimisation trade-off criteria.

If the prior model has high accuracy and observed or simulation data has considerable errors, it is better to carry out an uncertainty analysis using the solutions on the left handside of the Pareto. If the prior model has low accuracy and observed and simulation

data has high accuracy, it is better to carry out an uncertainty analysis using the solutions on the right hand side of the Pareto. If the prior model and observed and simulation data, all have enough accuracy, or if all these information have low accuracy, it is better to carry out an uncertainty analysis using the solutions on the middle of the Pareto.

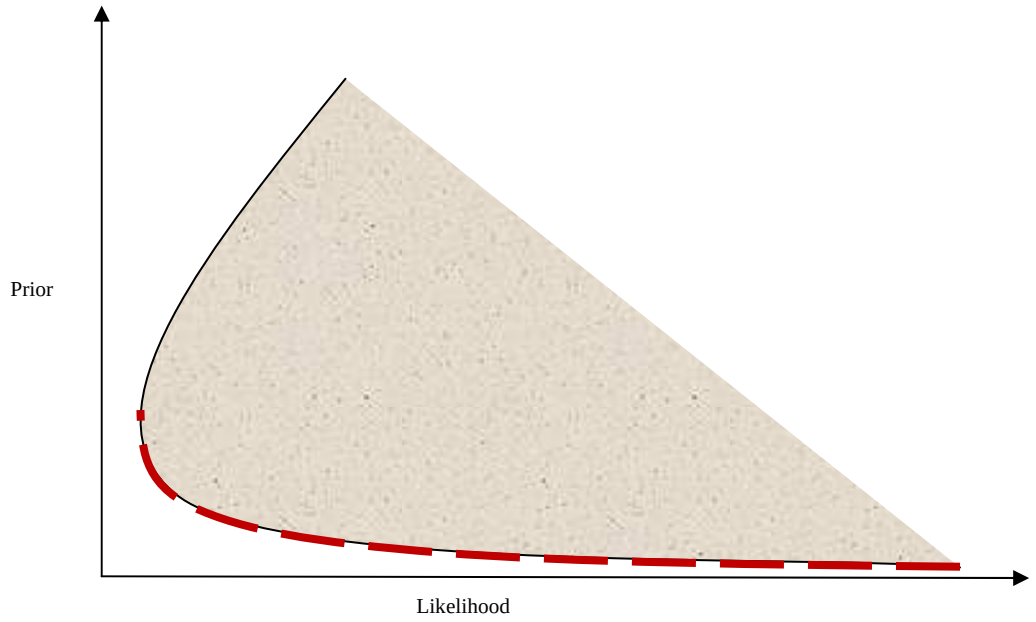


Figure 7.2 Feasible region and Pareto front of history matching problems considering prior and likelihood function as two separate objectives

The selection (post-optimisation trade-off) is a tricky task, and instead of doing it qualitatively and schematically, it needs a quantitative criterion. For this reason, the following rules are designed. In this study, it is assumed C_D is uncertain, since the modelling errors (Carter, 2004) and measurement noise cannot be estimated accurately. C_M is considered acceptably known with the assumption of accurate parameterisation. If C_M is also uncertain, the procedure will be similar and only the first rule should not be implemented for filtering the Pareto solutions.

i- Known C_m along with m_{prior} provides estimation regarding the distance of the location of the true solution in N_M dimensional space. The diagonal elements (standard deviations) of C_m indicate the maximum distance of each corresponding element of MAP from the elements of m_{prior}. Hence, the first trade-off can be made according to the prior

knowledge function. The probability that the maximum distance of each element of MAP from its corresponding element of m_{prior} is equal to the corresponding diagonal element of C_M (standard deviation) is 68%. This probability is equal to 90%, if the maximum distance increases to 1.644854 times of its corresponding diagonal element of C_M. Therefore, the probability that the MAP has a prior knowledge function value less than $N_m/2$ is 68% (equation 4 to 6), and this probability increases to 90% if the MAP are traded-off according to the following formula, $Prior(m) < \frac{N_m}{2} \times (1.644854)^2$ (equation 7). These probabilities are calculated according to the normal distribution concepts.

$$Prior(m) = \frac{1}{2}\left(m - m_{prior}\right)^t C_M^{-1}\left(m - m_{prior}\right) \tag{4}$$

$$Prior(m) = \frac{1}{2}\sum_{i=1}^{N_M}\left(\frac{m_i - m_{prior_i}}{\sigma_{ii}}\right)^2 \tag{5}$$

$$if\ max\left|MAP_i - m_{prior_i}\right| \approx \sigma_{ii} \rightarrow\ Prior(MAP) = \frac{1}{2}\sum_{i=1}^{N_M}\left(\frac{MAP_i - m_{prior_i}}{\sigma_{ii}}\right)^2 \leq \frac{N_m}{2} \tag{6}$$

$$if\ max\left|MAP_i - m_{prior_i}\right| \approx 1.644854\sigma_{ii} \rightarrow\ Prior(MAP) \leq \frac{N_m}{2} \times (1.644854)^2 \tag{7}$$

Thus, according to one of the aforementioned criteria (equation 6 or 7), the first trade-off can be applied. Solutions which have a prior knowledge value less than $\frac{N_m}{2} \times (1.644854)^2$ or $\frac{N_m}{2}$ will be taken out from the entire Pareto. It is better to use equation 7, since if the solutions are trade-off according to equation 6, there will be a chance of 32% that the true solution to be omitted; while this probably decreases to 10%, if the distance is increased to 1.644854 standard deviations.

ii- The second trade-off rule is based to the range of the diagonal elements of C_D, these elements (standard deviations) are varied from the square of $a\% \times d_{obs}$ to the square of $b\% \times d_{obs}$. In this trade-off, it is required to approximate the maximum likelihood ($m_{likelihood}$)

first, and then estimate the maximum and minimum distance of the MAPs for different C_Ds from the m_{prior} or $m_{likelihood}$ in N_M dimensional space. In order to find the distances, the MAPs should be estimated which is required to solve the inverse problem for different scenario of C_D. It is time-consuming and not advantageous. If we wanted to carry out history matching several times with different C_Ds, we would utilise the Monte Carlo approach. Hence, to solve this issue, the Taylor series approximation is used. Using the following formulas, it is possible to approximate the MAP for each set of covariance matrixes without solving the inverse problem for different C_Ds (and C_Ms, if C_M is also uncertain). Using equation 13, the MAPs are approximated. The derivation is as following:

Around the $m_{likelihood}$, it is assumed a first order Taylor series is enough to approximate the forward function.

$$g(m) \approx g(m_{likelihood}) + \nabla g(m_{likelihood})(m - m_{likelihood}) \quad (8)$$

If equation 8 is incorporated into equation 3:

$$S(m) = \frac{1}{2}\left(d_{obs} - g(m_{likelihood}) + \nabla g(m_{likelihood})(m - m_{likelihood})\right)^t C_D^{-1} \ldots$$

$$\ldots \left(d_{obs} - g(m_{likelihood}) + \nabla g(m_{likelihood})(m - m_{likelihood})\right) \ldots$$

$$\ldots + \frac{1}{2}\left(m - m_{prior}\right)^t C_M^{-1}\left(m - m_{prior}\right) \quad (9)$$

At $m_{likelihood}$,:

$$d_{obs} \approx g(m_{likelihood}) \quad (10)$$

Therefore, equation 9 will be as equation 11.

$$S(m) \approx \frac{1}{2}\left(\nabla g(m_{likelihood})(m - m_{likelihood})\right)^t C_D^{-1}\left(\nabla g(m_{likelihood})(m - m_{likelihood})\right) \ldots$$

$$\ldots + \frac{1}{2}\left(m - m_{prior}\right)^t C_M^{-1}\left(m - m_{prior}\right) \quad (11)$$

The solution (MAP) of equation 11 is as equation 13 which is a function of C_D, C_M, $m_{likelihood}$ and m_{prior}, it is achieved by equalling the derivation of S(m) to zero and solving the corresponding equation (equation 12).

$$\frac{\partial S(m)}{\partial m} = 0, and\ solve \tag{12}$$

$$MAP \approx m_{prior} + C_M\big(\nabla g(m)\big)^t(\nabla g(m) C_M \nabla g(m)^t + C_D)^{-1}\Big(\nabla g(m)\big(m_{lik.} - m_{prior}\big)\Big) \tag{13}$$

$\nabla g(m)$ is the gradient of forward operator which is $N_{dobv} \times N_M$ dimensional matrix, as equation 14 and 15.

$$\nabla g(m) = \begin{bmatrix} \frac{\partial g_1}{\partial m_1} & \cdots & \frac{\partial g_1}{\partial m_{N_m}} \\ \vdots & \ddots & \vdots \\ \frac{\partial g_{N_{dobs}}}{\partial m_1} & \cdots & \frac{\partial g_{N_{dobs}}}{\partial m_{N_m}} \end{bmatrix} \tag{14}$$

$$\frac{\partial g_l}{\partial m_k} \approx \frac{g_l\big(m_1, \dots, m_k + \Delta m_k, \dots, m_{N_m}\big) - g_l\big(m_1, \dots, m_k, \dots, m_{N_m}\big)}{\Delta m_k} \tag{15}$$

Using equation 13, it is possible to approximate the solution for each set of C_D and C_M analytically without solving its inverse problem. Thus, to carry out the second trade-off rule, many different scenarios (C_Ds and C_Ms if it is uncertain) are selected randomly from the given ranges and their MAPs are approximated using this equation. Then, the distance of the calculated MAPs from the m_{prior} is estimated, and afterward according to the estimated distances (minimum and maximum), the Pareto solutions are filtered. If the C_M is also uncertain, different scenario of C_M and C_D will be picked randomly and their corresponding MAPs will be approximated using equation 13; then the maximum and minimum distance from m_{prior} will be estimated.

In the non-quadratic problems, the equation 13 is not very reliable. That is why the distance is calculated. If it could deliver accurate results, the calculated MAPs would be

used directly for uncertainty analysis, instead of estimating the Pareto and trading-off according to this rule. Equation 13 can deliver an acceptable estimation regarding the distance of the MAP from m_{prior}, as the problem is close to quadratic around the $m_{likelihood}$ and hence the approximation can be reasonable for distance estimation. Equation 13 in the linear problem works precisely, since the error of numerical calculation of the gradient, and the deviation from being non-quadratic is zero (the Taylor approximation is exact).

7.3. Results

In this section, two examples are used to investigate the influence of the uncertainties of covariance matrixes on the quality of the results. Then, for each problem, an uncertainty analysis is performed using the two approaches (the conventional and the proposed). The Pareto approach is compared with the Monte Carlo (random selection) approach in these two examples. The first example is a numerical model which is a linear inverse problem and the second example is a history matching problem.

7.3.1. Numerical example

In this example, a column vector (m = [a b c d e f]`) should be estimated from an observed data. The column vector consists of the coefficients of equation 16. The forward problem is linear (equation 17) in which x is as equation 18. In order to generate an observed data, a set of values is specified for the unknown vector (m) as equation 19, and called reference (m_{ref}). The observed data is generated by putting the reference value into equation 20 and adding Gaussian noise with zero mean and standard deviation of 10 percents as measurement error. The observed data is assumed uncorrelated. Figure 7.3 shows two graphs, the black graph is the observed data with 10% Gaussian noise while the blue graph is the data without noise. The fitting criterion is the black graph, since no one knows the exact data in real problems. A prior model (m_{prior}) is created by adding Gaussian noise with

a zero mean and 25 percent standard deviation to m_{ref}. m_{prior} is as equation 21, and its distance from the reference is 3.78 according to equation 22. The prior knowledge covariance matrix is diagonal and the elements on the diagonal are $(0.25\times m_{prior})^2$.

The goal is finding the most probable model (m) from the given prior model (m_{prior}) and the observed data (black graph). The objective function is formulated as equation 3. In this study, to investigate the effect of covariance matrixes, it is assumed that observed covariance matrix is uncertain and its diagonal element can be any value from 5% to 40% of the observed data denoted by R; the correct value is 10% (the modelling error is zero, as the reference observed data is constructed by the forward problem). The covariance matrix of observed data will be as equation 23. In the first part of this example, the effect of the uncertain covariance matrix on the outcome of the problem is shown. The second part explains how the problem can be dealt with and the problem is solved by the proposed approach, and the conventional approach; the outcomes are compared with each other.

$$y(x) = a.x + b.sin\left(\frac{x}{2}\right) + c.cos(x^2) + d.\frac{1}{x} + e.\ln(x) + f \tag{16}$$

$$G = \begin{bmatrix} x_1 & \sin(\frac{x_1}{2}) & \cos(x_1^2) & \frac{1}{x_1} & \ln(x_1) & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{100} & \sin(\frac{x_{100}}{2}) & \cos(x_{100}^2) & \frac{1}{x_{100}} & \ln(x_{100}) & 1 \end{bmatrix}_{100\times 6} \tag{17}$$

$$x = [x_1, x_2, \cdots, x_{100}] = [1, 1.1, \cdots, 10.9]_{1\times 100} \tag{18}$$

$$m_{ref} = \begin{bmatrix} 2 \\ -8 \\ 2 \\ 4 \\ 8 \\ 10 \end{bmatrix}_{6\times 1} \tag{19}$$

$$d_{cal} = Gm \tag{20}$$

$$m_{prior} = \begin{bmatrix} 2.39 \\ -6.43 \\ 2.69 \\ 3.81 \\ 4.88 \\ 8.82 \end{bmatrix} \tag{21}$$

$$Distance(m) = \left(\sum_{i=1}^{N_m} \left(m_i - m_{ref_i} \right)^2 \right)^{\frac{1}{2}} \tag{22}$$

$$C_D = C_d = \begin{bmatrix} \left(Rd_{obs_1}\right)^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \left(Rd_{obs_{N_{dobs}}}\right)^2 \end{bmatrix} \tag{23}$$

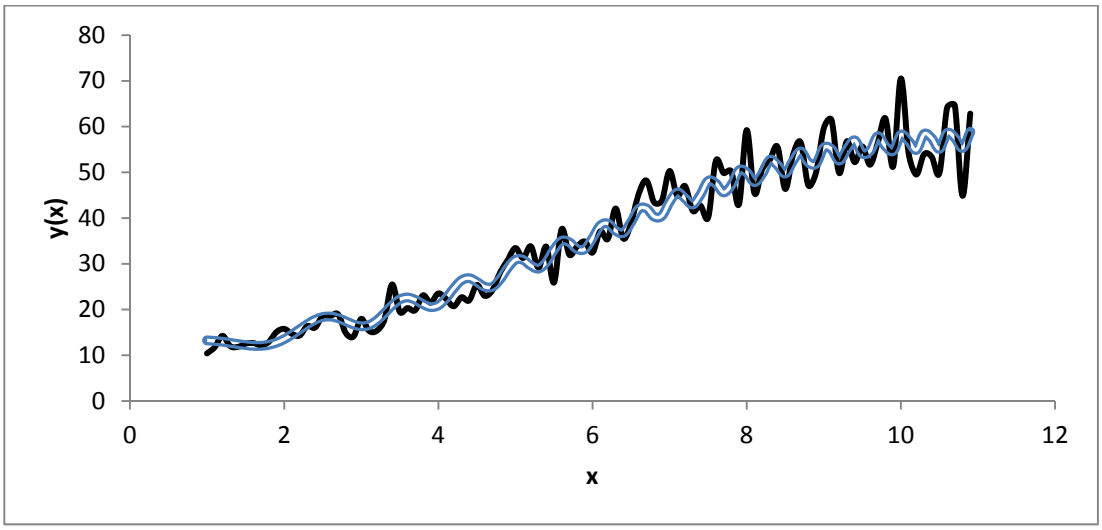


Figure 7.3 The observed data with and without noise for the numerical example

Table 7.1 shows the solutions of the problem by assuming different Rs. The first column from left hand side shows the different values which are given to R from the range. The middle column is the corresponding difference of achieved solution (m_∞), with a regularisation term consideration from the reference based on equation 24. The last column is the corresponding difference of achieved solution (m_∞), without a regularisation term consideration from the reference based on equation 24.

The first observation is that the solutions achieved by considering a regularisation term are significantly superior in comparison with the achieved solutions without any regularisation term; it shows the importance of having a regularisation term in noisy observed data. The

second observation is that by closing to the true R (10%), the solutions are improved. The difference proves the influence of covariance matrixes.

The solutions are equally acceptable, since there is no further information to discriminate between the solutions. All these solutions are the most probable model for the corresponding selected R. Thus, in these problems, a single R should not be selected, since it may lead to an incorrect model and an uncertainty analysis is required.

Table 7.1 The solution of the numerical example for different covariance matrixes

R	Difference of solutions achieved with regularisation from the reference	Difference of solutions achieved without regularisation from the reference
5%	13.69%	77.7%
10%	11.44%	77.7%
20%	14.48%	77.7%
30%	17.08%	77.7%
40%	18.26%	77.7%

$$Difference(m) = \frac{1}{N_M}\sum_{i=1}^{N_M}\left|\frac{m_i - m_{ref_i}}{m_{ref_i}}\right| \times 100 \qquad (24)$$

According to the procedure of a Monte Carlo approach, i.500 different Rs are selected randomly from the given range [5% - 40%], ii. the corresponding solution for each R is estimated (solve the inverse problem, equation 3), iii. an uncertainty analysis is carried out via the achieved solutions. The cumulative probability distribution of differences is drawn in figure 7.4. It demonstrates that the achieved solutions have difference from the reference model from 10.5% to 18.5%. A decision with 90% confident should expect maximum 17.9% error. This problem is linear and has analytical solution, therefore the step 2 in which the inverse problem is solved 500 times is not time-consuming, but in the nonlinear inverse problem, it is required to carry out the optimisation step 500 times.

This problem is also solved by the Pareto approach. In order to define the Pareto front, a multi-objective optimisation id carried out with the mentioned options in table 7.2. The Pareto front is approximated using the algorithm which consists of 500 solutions. It is

shown in figure 7.5. It expresses the front acceptably, however some deviations can be seen.

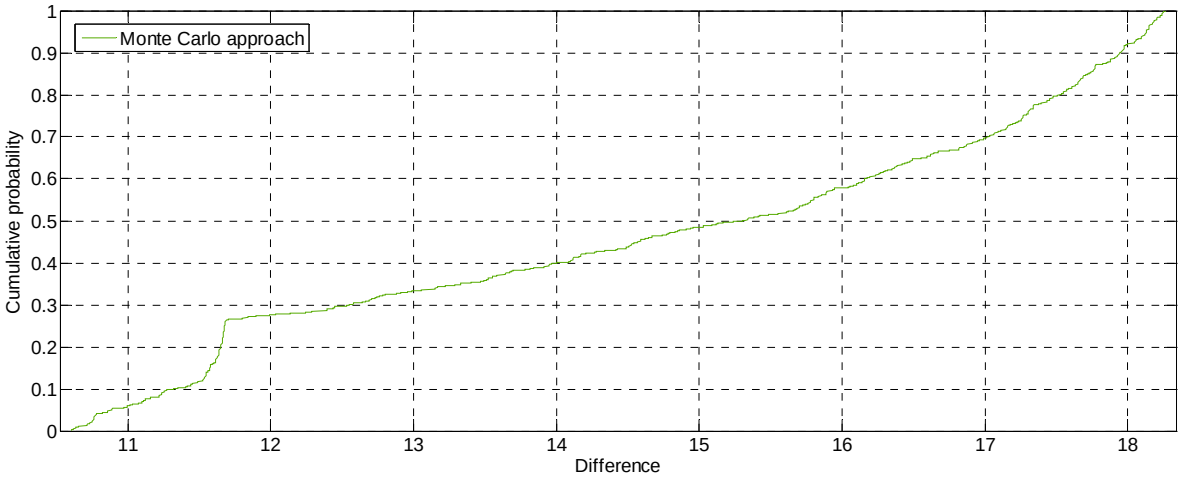


Figure 7.4 The cumulative probability distribution achieved by the conventional approach

In the next step, the post-optimisation trade-off rules should be applied. The first rule is the prior knowledge function should not have a value higher than $(1.644854)2\times N_m/2 = 8.1166$. The solutions picked by this rule are shown by blue dot points in figure 7.6 on the front.

Table 7.2 Multi-objective genetic algorithm options

Options	Objectives	Population size	Crossover operator	Selection operator	Mutation	Stopping criteria	Crossover probability	Pareto front population fraction
	Second term of eq. 3 First term of eq. 3	1000	Heuristic Ratio=1.2	Tournament size=4	Uniform Rate=0.11	Generation number (2000)	0.8	0.35

Now, the second trade-off rule should be applied on the selected solutions. In order to apply the second rule, the MAPs are approximated using equation 13 for 1000 different C_Ds which are chosen randomly within the domain. The distance of the 1000 approximated MAPs from m_{prior} are calculated via prior knowledge function. The minimum and maximum distances according to the prior knowledge function are 0.34 and 6.39. As in the linear problems the equation 13 works accurately, this domain [0.34-6.39] without any modification is used to filter some other solutions from the Pareto. Figure 7.7 shows the solutions picked by the second rule. The block spheres are the final solutions.

Using the remained solutions, the cumulative probability of differences is calculated (figure 7.8). The blue graph expresses the outcome of the solutions of Pareto approach, and

the green graph shows the outcome of the solutions of the Monte Carlo approach. As it can be seen, the difference is obtained by the proposed approach on most of the probabilities is lower, for instance, if a decision with 90% confidence is sought by the Pareto approach, the maximum difference will be 17.25%, while this value is 17.9% with the conventional approach. Hence, it is fair to state that the proposed approach is able to deliver high-quality results.

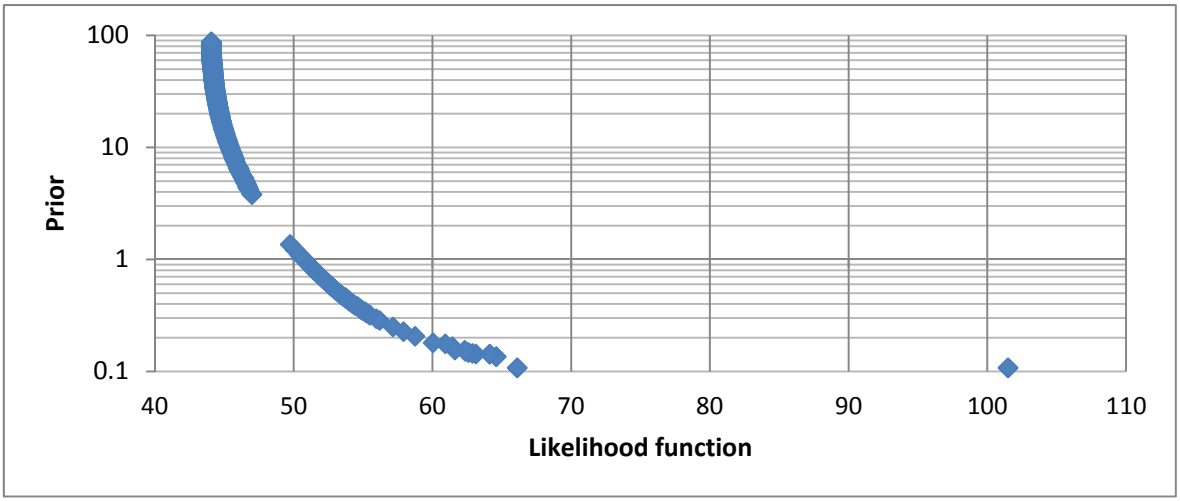


Figure 7.5 Pareto front of the numerical example

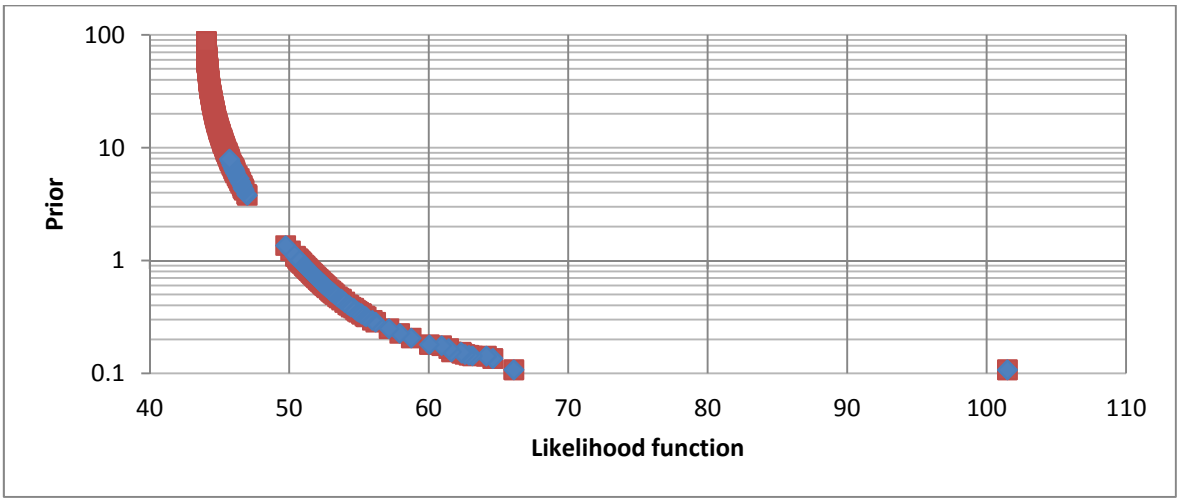


Figure 7.6 Pareto front of the numerical example with first trade-off rule

The main advantage of this approach in comparison with the conventional approach is that in order to achieve the set of solutions using the proposed approach, only a single run of multi-objective optimisation is enough while, the set of solutions using the conventional approach needs 500 runs of single-objective optimisation[10]. Therefore, it can be said that

[10] As mentioned before, in this example, as it is linear, the execution of optimisation is not required to find the MAPs in the conventional approach. Thus, the computation advantage is not valid in the linear problems.

not only acceptable uncertainty analysis can be achieved (even slightly better than the conventional) but also the computation is reduced dramatically.

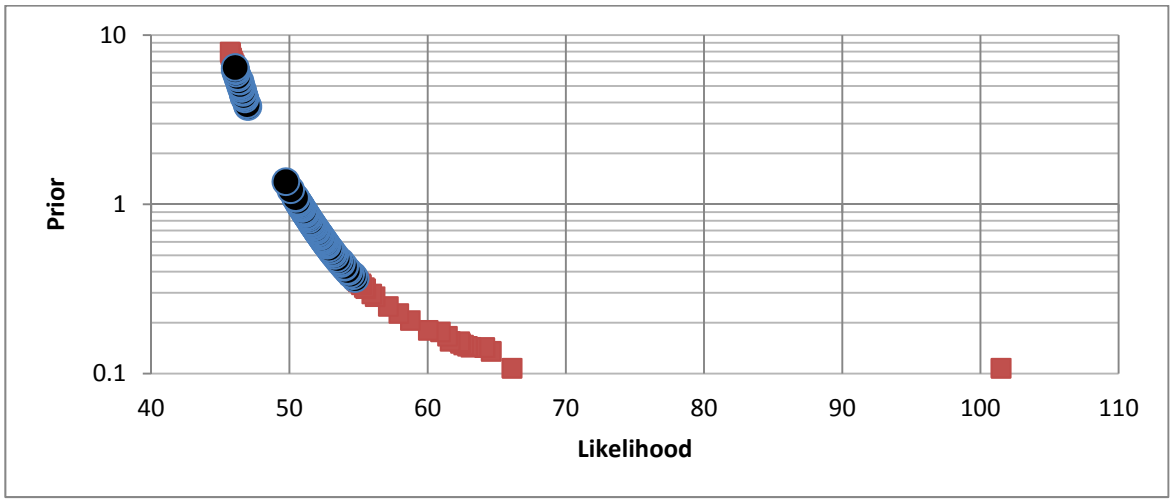


Figure 7.7 Pareto front of the numerical example with first and second trade-off rule

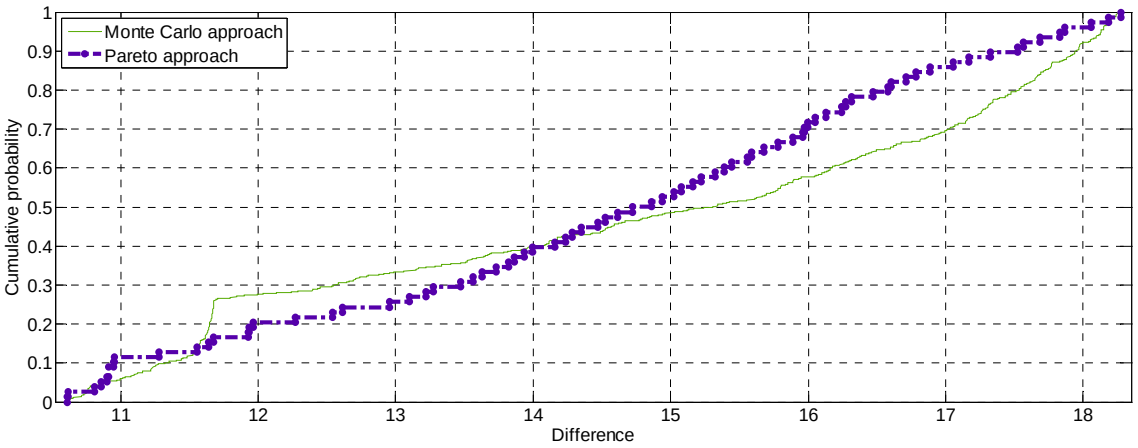


Figure 7.8 The cumulative probability distribution achieved by the conventional and the proposed approach

In the following section, the methods are compared with each other on a history matching problem.

7.3.2. History matching on PUNQ-S3 model

For the second example, PUNQ-S3 reservoir model is used. The reservoir model consists of 19×28×5 gridblocks and has 1761 active blocks. There are some modifications in some parts of the problem to switch the emphasis on the uncertainty of covariance matrixes. The modifications are as following, 1- the horizontal permeabilities are assumed isotropic and a function of porosity as equation 25 in which the coefficient (c_1) and the power (γ) are

uncertain (matching parameters); 2- the vertical permeability is assumed a function of the horizontal permeability as equation 26 in which the coefficient (c_2) is uncertain (matching parameter); and 3- the reference porosity distribution in each layer is generated using bicubic spline. The porosities in gridblocks in which wells are located at are assumed known and the porosity in five gridblocks in each layer are used as matching parameters. Using these 11 gridblocks data, the reference porosity distribution in each layer is created (figure 7.9) by an interpolation via the bicubic spline. The locations of the five gridblocks are fixed and they are shown by black dot points in figure 7.9. The white dot points show the gridblocks in which wells are located at.

$$k_{Arial}(mD) = 1000 c_1 \phi^{\frac{\gamma}{10}} \tag{25}$$

$$k_{Arial}(mD) = \frac{k_{Vertical}(mD)}{c_2} \tag{26}$$

The reference model is simulated to generate history data. The total number of observed data (history) is 1107 elements for 42 time steps (include 2936 days performance history). These data are cumulative field gas production, cumulative field water production, cumulative field oil production, well water-cuts, well gas oil ratios, well oil rates and well bottomhole pressures. Gaussian noise with a zero mean and standard deviation equal to 20 percent of observed vector were added to the observed data (history).

The matching parameters are the 25 porosities (5 gridblocks in each layer) and 3 coefficients of equation 26 and 27. As the reference model was created using allocating a specific value to these matching parameters, the parameterisation and reparameterisation error is zero, and therefore, there is no concern about the validity of reparameterisation. The true solution (reference which was used to generate the history), and the prior model generated by adding Gaussian noise with zero mean and 20% to the solution can be found in table 7.3.

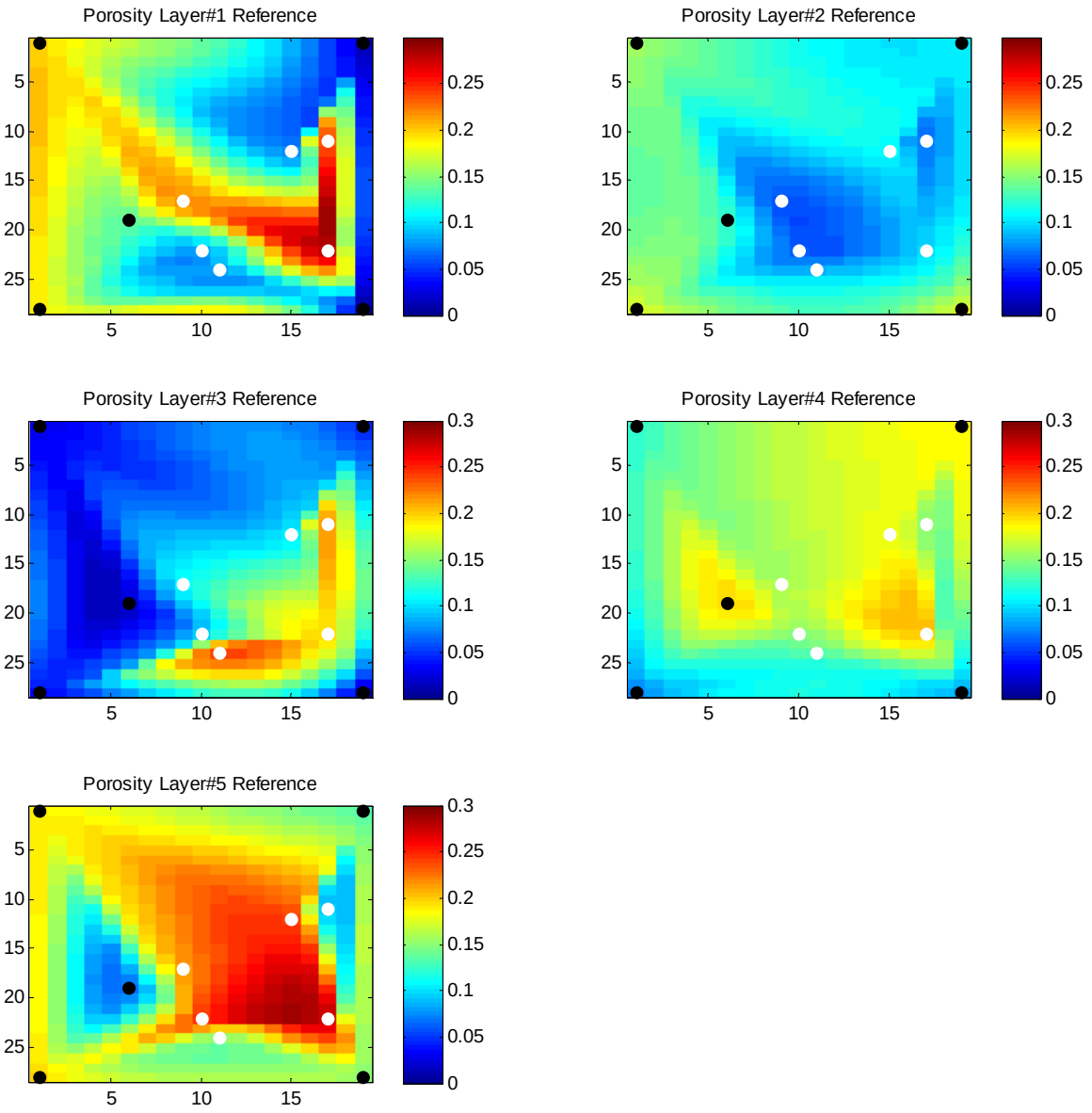


Figure 7.9 Reference porosity distribution of PUNQ-S3 generated by bicubic spline

The goal is to estimate the most probable model using the observed data and the prior model. The objective function is defined as equation 3. To investigate the effect of covariance matrixes on the reliability of the obtained history matched models, C_D which is a diagonal matrix is assumed uncertain and the domain of its elements (standard deviations) is uniform and from the square of $5\% \times d_{obsi}$ to the square of $40\% \times d_{obsi}$. C_D should include the measurement and modelling errors, in this case study, as the history was generated by simulating the case study, the modelling errors are zero, but in the real problems, they are not zero and also they are not quantifiable. That is the reason why C_D is assumed unknown in this study and the effect of its uncertainty is investigated.

In order to assess the obtained history matched models the following two criteria are used, 1- ultimate oil recovery via the existing producers over 16.5 years is estimated using the obtained history matched model and the value is compared with the reference value (2,841,904 standard cubic meters); 2- the ultimate oil recovery via the existing wells and six infill wells over 16.5 years is estimated using the obtained history matched model and the value is compared with the reference value (3,611,786 standard cubic meters). Any history matched model which can deliver the closest values to the aforementioned reference ultimate oil recoveries for both scenarios is the better model.

Table 7.3 Reference and prior model

Variable	Reference	Prior
Porosity in gridblock (x=1, y=1) 1st layer	19.7829	26.22665
Porosity in gridblock (x=19, y=1) 1st layer	1.338925	1.466267
Porosity in gridblock (x=1, y=28) 1st layer	18.78797	19.94273
Porosity in gridblock (x=19, y=28) 1st layer	0.363551	0.380664
Porosity in gridblock (x=6 y=19) 1st layer	13.67677	16.50523
Porosity in gridblock (x=1, y=1) 2nd layer	15.67473	14.63738
Porosity in gridblock (x=19, y=1) 2nd layer	10.68275	9.588425
Porosity in gridblock (x=1, y=28) 2nd layer	17.70719	19.59075
Porosity in gridblock (x=19,y=28) 2nd layer	17.9801	16.87799
Porosity in gridblock (x=6 y=19) 2nd layer	12.51875	14.75166
Porosity in gridblock (x=1, y=1) 3rd layer	2.75738	1.950091
Porosity in gridblock (x=19, y=1) 3rd layer	4.356032	5.51157
Porosity in gridblock (x=1, y=28) 3rd layer	3.642822	3.110519
Porosity in gridblock (x=19,y=28) 3rd layer	0.836397	0.876302
Porosity in gridblock (x=6 y=19) 3rd layer	2.138833	1.93869
Porosity in gridblock (x=1, y=1) 4th layer	12.32887	14.08837
Porosity in gridblock (x=19, y=1) 4th layer	18.79322	13.46536
Porosity in gridblock (x=1, y=28) 4th layer	7.089115	5.575881
Porosity in gridblock (x=19,y=28) 4th layer	8.212582	10.89379
Porosity in gridblock (x=6 y=19) 4th layer	19.68699	15.84649
Porosity in gridblock (x=1, y=1) 5th layer	18.91158	16.27016
Porosity in gridblock (x=19, y=1) 5th layer	13.53289	13.70156
Porosity in gridblock (x=1, y=28) 5th layer	19.76605	16.6086
Porosity in gridblock (x=19,y=28) 5th layer	15.33663	15.78662
Porosity in gridblock (x=6 y=19) 5th layer	6.733985	8.317719
C1	30	30.4121
γ	30	40.35712
C2	10	7.10052

Eight different C_Ds are selected from the given domain and history matching is carried out for each of them. The matching parameters are the stated 28 variables, and the objective function is as equation 3. The optimisation is carried out using the designed GA (refer to chapter 5) with the mentioned options in table 7.4. The optimisation for each problem (C_D) is executed twice with two different seed number and the average of the two runs is reported, in order to deliver fair comparison.

Table 7.4 Single-objective GA options

Population size (N_{pop})	Stopping criteria	Initial population	Mutation & its probability	Crossover probability	Number of elites	Tournament size (N_{tour})
32	No_{gen} = 350	Uniform	Uniform & p=0.1	p=0.9	1	4

The ultimate oil recoveries for the two scenarios are estimated using the achieved history matched models and their differences from the reference value can be found in table 7.5.

Table 7.5 The results of the conventional approach for the PUNQ-S3 model

R	Difference of estimated UOR using obtained model from the reference for 1st scenario	Difference of estimated UOR using obtained model from the reference for 2nd scenario	Average Difference of the two scenario
5%	4.58%	4.95%	4.76%
10%	5.97%	8.42%	7.19%
15%	5.47%	7.12%	6.29%
20%	4.49%	7.86%	6.17%
25%	7.04%	9.15%	8.09%
30%	6.24%	6.18%	6.20%
35%	2.99%	6.15%	4.57%
40%	4.54%	9.52%	7.02%

As it can be seen, the difference is varied from 4.57% to 8.09% according to the selected C_D which indicates the influence of the covariance matrix (regularisation factor) in the quality of obtained models. Therefore, an uncertainty analysis is required.

To provide an uncertainty analysis, this problem is solved with the two different approaches (Monte Carlo and Pareto). For the Monte Carlo approach, due to computation limitation, only the eight different C_Ds are applied for uncertainty analysis. To carry out the Monte Carlo approach for only the eight different C_Ds and obtain models, the fitness function should be called 179,200 (8×2×32×350) times which needed 16.5 days CPU-time on a personal computer with the following configurations, CPU Intel i-7-2820QM and 8 GB ram. The cumulative probability of average differences using these eight solutions can be seen in figure 7.12.

Also, the problem is solved by the Pareto approach. To execute the Pareto optimisation, two multi-objective genetic algorithms with the two different options (table 7.6 and table 7.7) are used. The first multi-objective GA had a random initialisation while the second one had a heuristic initialisation. For the heuristic initialisation, the chromosomes of the

initial population were generated by adding Gaussian noise with zero mean and covariance equal to C_M to m_{prior}. The heuristic initialisation helps in delivering quicker convergence; however, it may result in premature convergence. Thus, its solutions are combined with the solutions of another multi-objective genetic algorithm which uses a random initialisation. The achieved Pareto solutions of these two runs which had different seed numbers are put together. The achieved solutions are shown in figure 7.10. They consist of 120 solutions.

Table 7.6 1st multi-objective genetic algorithm options

Options	Objectives	Population size	Crossover operator	Selection operator	Mutation	Stopping criteria	Crossover probability	Pareto front population fraction
	Second term of eq. 3 First term of eq. 3	170	Intermediate ratio:1.0	Tournament size=4	Uniform Rate=0.11	Generation number (250)	0.8	0.45

Table 7.7 2nd multi-objective genetic algorithm options

Options	Objectives	Population size	Crossover operator	Selection operator	Mutation	Stopping criteria	Crossover probability	Pareto front population fraction
	Second term of eq. 3 First term of eq. 3	120	Heuristic ratio:1.2	Tournament size=4	Uniform Rate=0.11	Generation number (150)	0.8	0.35

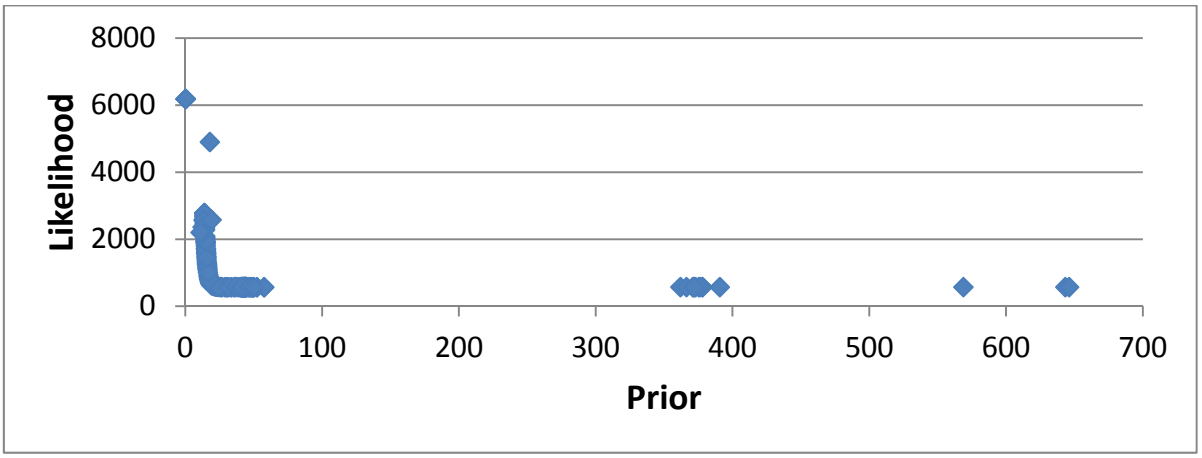


Figure 7.10 Pareto front for PUNQ-S3 reservoir

The two post-optimisation trade-off rules are applied to pick a set of solutions among the Pareto front. The first rule is as equation 27.

$$Prior(m) \leq \left(\frac{N_m}{2} \times (1.644854)^2\right); N_m = 28 \rightarrow \ Prior(m) \leq 37.87 \qquad (27)$$

The second rule is similar to the numerical example; 500 different C_Ds are picked and the solution is approximated using equation 13. Then, their distances from m_{prior} according to prior knowledge function are calculated. The minimum distance is 34 and the maximum

distance is 108. As this example is nonlinear, it is expected that the equation 13 and also gradient calculation have error, therefore the range ([34-108]) is enhanced by 50%. Hence, the second rule is as equation 28.

$$34 \times 0.5 < Prior(m) < 108 \times 1.5 \quad (28)$$

According to these two rules, 26 solutions will remain. They are shown by blue points in figure 7.11.

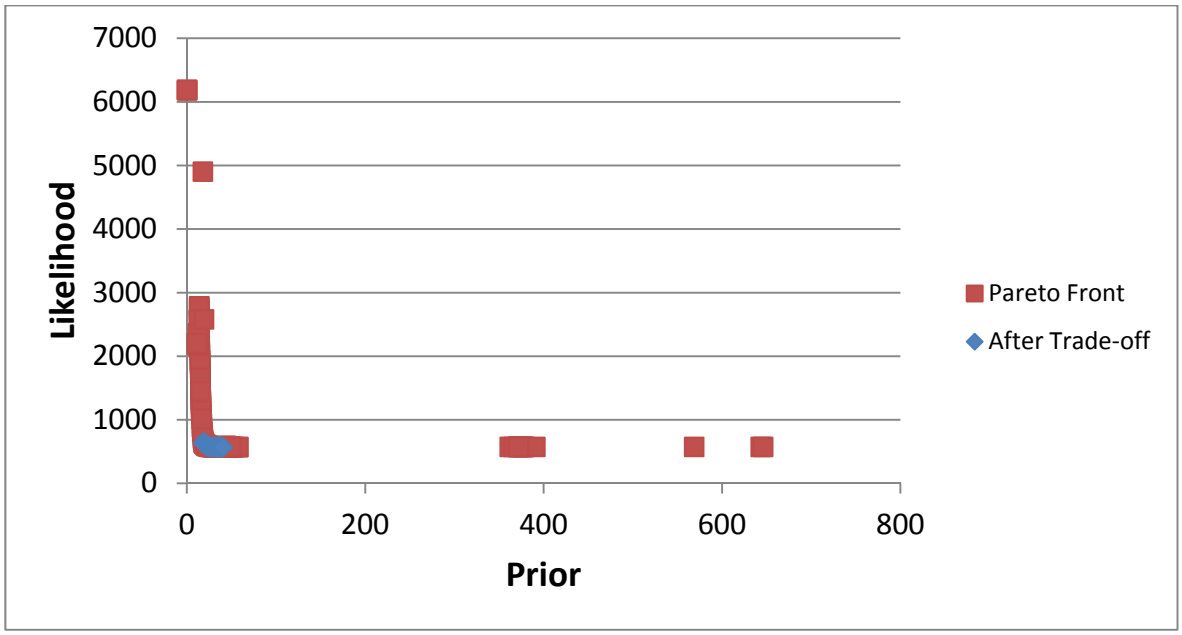


Figure 7.11 Pareto front for PUNQ-S3 and the solutions after trade-off

Using the remained solutions, the ultimate oil recoveries for the two scenarios are estimated, and the cumulative probability of average differences of the estimations from the reference values is drawn in figure 7.12. The green graph shows the solutions of Pareto approach while the blue graph expresses the solutions of Monte Carlo approach. As it can be seen, the forecasting can be made with more certainty by the solutions of Pareto approach. Solutions with less than 2% difference are seen among its remained solutions while the minimum difference is 4.5% with the conventional approach.

The computation of the Pareto approach required ***5.5 days*** CPU-time on the same personal computer, while the conventional approach needed ***16.5 days*** CPU-time. Hence, it is fair to state that the proposed approach not only needs less computation, but also it can deliver

better results in comparison with the conventional approach. It should be mentioned that the conventional approach perhaps will be able to deliver better results, if a large number of scenarios are investigated. If the trade-off rules were not applied, there would be solutions with 52% difference. That indicates the importance of trade-off rules.

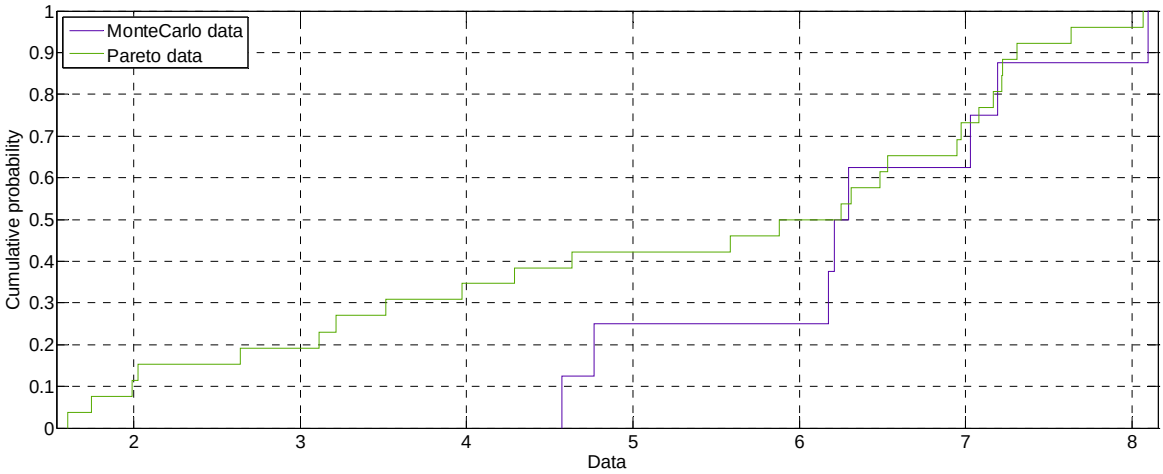


Figure 7.12 Cumulative probability of average difference of forecasting of two approaches (Monte Carlo and Pareto)

7.4. Discussion

In this chapter, it was shown that covariance matrixes play a significant role on the reliability of history matching. Also, it was demonstrated that in those problems where the covariance matrixes are uncertain, like the examples of this chapter, a set of solutions should be provided for delivering an uncertainty analysis. To achieve a set of solutions by the conventional approach, it was required to solve the inverse problem with many different arrangements of covariance matrixes. This approach was time-consuming, and it was shown that if a large number of arrangements are not picked, there will be high risks in decision making. In the proposed method, the Pareto front for likelihood versus prior function was approximated using a single run of a multi-objective genetic algorithm. Using the designed post optimisation trade-off rules, a number of solutions were taken out from the Pareto front. The outcomes of these approaches expressed that the proposed approach is more efficient in terms of the quality of solutions and also computation.

The proposed approach has also a drawback; the multi-objective genetic algorithm should be utilised in conjunction with a reparameterisation operator, as the regular crossover operators do not work properly in high-dimensional search spaces. The applied multi-objective algorithm is extensively used for Pareto optimisation. The algorithm can be replaced by any other multi-objective optimisation algorithms, and this replacement does not influence the approach. It is worth mentioning that where the covariance matrixes are certainly known, these approaches are not efficient, and it is better to use the regular methods, like the previous chapters.

The conclusive remarks of this chapter are as following:

- Covariance matrixes should be carefully assigned, and inaccuracy in covariance matrixes may result into wrong history matched models.
- If the covariance matrixes are uncertain, a set of solutions should be estimated to be able to carry out an uncertainty analysis.
- Pareto approach (proposed) delivers better than the random selection approach (conventional), in terms of estimating the set of solutions and also the required CPU-time.
- The designed post-optimisation trade-off rules perform properly in filtering the solutions.

Chapter 8

8. Application of transfer functions in providing quick estimation of future performance

The main focus of this chapter is on developing a new fast simulator that does not require the expensive history matching procedures and provides quick forecasts. Thus far, various simulators from simple (fast) to complex ones have been developed and utilised to deliver predictions regarding reservoir performance. The choice of each of these methods depends on the available data and the level of desirable accuracy (Ertekin et al., 2001a). However, grid-based simulation (Aziz and Settari, 2002, Fanchi, 2001) and streamline simulation (Rust and Caudle, 1972) are accurate methods, they are time-consuming (especially grid-based simulators) and large quantities of data are required. All the data, especially spatial distributions are not always accessible; hence expensive history matching is required (the new advances and several robust algorithms for history matching were investigated in the previous chapters).

Sometimes, a quick overview with reasonable accuracy is adequate, or sometimes, all the essential data for a grid-based simulation are not accessible or computable with history matching. In these occasions, a fast simulator which has reasonable accuracy is sought. Methods like Decline Curve Analysis (DCA) (Arps, 1944, Baker et al., 2003) and the Capacitance Resistance Model (CRM) (Sayarpour et al., 2009a, Sayarpour et al., 2010)

supply quick forecasts and also require the minimum data to simulate reservoirs, but their outputs are not very reliable in many cases, particularly secondary recovery and gas-flooding respectively. Therefore, researchers look for a method which needs minimum data and has reasonable accuracy.

In this chapter, a new fast simulator is presented to forecast quickly the performance of oil reservoirs during water and gas (miscible and immiscible) injection based on Transfer Function (TF). In this method, it is assumed a reservoir consists of a combination of TFs. The suitable order and arrangement of TFs are sought by examining several cases. The method is combined to a fractional flow model to predict oil production rates. For water-flooding problems, the chosen fractional flow is Gentil model, and for gas-flooding, a specific fractional flow is developed. The questions, the author wishes to answer in this chapter, are:

a- Can the proposed method provide reliable forecasts for the future performance of oil reservoirs?
b- Can the fractional flow models estimate oil performance reliably?
c- What are the restrictions of this method?

This chapter begins by a brief introduction to the transfer functions and followed by the explanation of the methodologies. In the methodology section, water-flooding modelling and gas-flooding modelling are described separately. In the results section, different cases are employed to validate the derived equations. The benefits and drawbacks of the methods are discussed, and followed by some conclusive remarks in the summary and discussion section. This chapter is a modified and adjusted version of two of our published papers in a peer-reviewed journal and a SPE conference: “SAYYAFZADEH, M., POURAFSHARY, P., HAGHIGHI, M. & RASHIDI, F. 2011, Application of transfer functions to model water injection in hydrocarbon reservoir, Journal of Petroleum Science and Engineering, 78, 139-148” and “SAYYAFZADEH, M., MAMGHADERI, A., POURAFSHARI, P. &

HAGHIGHI, M. 2011. A New Method to Forecast Reservoir Performance during Immiscible and Miscible Gas-Flooding via Transfer Functions Approach, SPE Asia Pacific Oil and Gas Conference and Exhibition, Jakarta, Indonesia: Society of Petroleum Engineers" (and also submitted to Journal of Petroleum Science and Technology).

8.1. Transfer functions

Any system can be assumed as a black box which has some input and some response signals. Transfer Function (TF) demonstrates the relation between input and output signals usually in Laplace domain (Buckley, 1964, Coughanowr et al., 1965, Towill, 1970) for the black box, figure 8.1. The most common TFs are the first and second-order and also the lag function. The majority of engineering systems can be modelled via a combination of these TFs. If the TF and input signal are signed by G(s) and X(s) respectively, the output signals, Y(s), are represented by equation 1 (Mikles and Fikar, 2007). s demonstrates Laplace domain and t reveals time domain.

$$Y(s) = G(s).X(s) \tag{1}$$

Figure 8.1 Transfer function

Table 8.1 shows the main input functions and their Laplace transforms which are used in this method.

Table 8.1 Input functions

Function type	Original function	Laplace Transform
Unit function	$F(t) = u(t-a) = \begin{cases} 0 & t < a \\ 1 & t > a \end{cases}$	$\frac{1}{S}$
Impulse Function	$F(t) = \delta(t) = \begin{cases} \frac{1}{\varepsilon} & 0 < t < \varepsilon \\ 0 & t > \varepsilon \end{cases}, \varepsilon \to 0$	1

8.1.1. First-order transfer function

Many engineering systems are modelled by a first-order TF, for example, RC circuits, tank modelling in process control, etc. In these systems, the transfer function is as equation 2:

$$\frac{Output\ Signal}{Input\ Signal} = \frac{K_p}{\tau s+1} \tag{2}$$

Kp demonstrates a fraction of the input signal which has influence on the output signal and/or the unit conversion and τ is the time constant. The output signals of first-order for different input signals are as follows:

Unit function

$$X(t) = Au(t) \rightarrow X(s) = \frac{A}{s} \rightarrow Y(s) = \frac{A}{s(\tau s + 1)} \rightarrow Y(t) = A\left(1 - e^{-\frac{t}{\tau}}\right) \tag{3}$$

Impulse function

$$X(t) = \delta(t) \rightarrow X(s) = 1 \rightarrow Y(s) = \frac{1}{\tau s + 1} \rightarrow Y(t) = \frac{1}{\tau} e^{-\frac{t}{\tau}} \tag{4}$$

8.1.2. Second-order transfer function

If the mechanical sections or the fluid flow in the system are accelerated, the transfer function is expressed by a second-order as:

$$\frac{Y(s)}{\mathrm{X(s)}} = \frac{\mathrm{Kp}}{\tau^2 \mathrm{s}^2 + 2\tau\xi \mathrm{s} + 1} \tag{5}$$

where τ and ξ reveal the time constant and the damping coefficient, respectively. The output signals of second-order TF for step and impulse function input signals are as follows:

Step function

$$X(t) = Au(t) \rightarrow X(s) = \frac{A}{s} \rightarrow Y(s) = \frac{\mathrm{Kp}}{s(\tau^2 s^2 + 2\tau\xi s + 1)} \rightarrow \cdots$$

$$\text{if } \xi < 1, \rightarrow \mathrm{Y(t)} = \mathrm{Kp}\left[1 - \frac{1}{\sqrt{1-\xi^2}} e^{-\frac{\xi t}{\tau}} \sin\left(\sqrt{1-\xi^2}\frac{t}{\tau} + \tan^{-1}\frac{\sqrt{1-\xi^2}}{\xi}\right)\right] \quad (6)$$

$$\text{if } \xi = 1, \rightarrow \ \mathrm{Y(t)} = \mathrm{Kp}\left[1 - \left(1 + \frac{t}{\tau}\right) e^{-\frac{t}{\tau}}\right] \quad (7)$$

$$\text{if } \xi > 1, \rightarrow \mathrm{Y(t)} = \mathrm{Kp}\left[1 - \frac{1}{\sqrt{\xi^2 - 1}} e^{-\frac{\xi t}{\tau}} \cosh\left(\sqrt{\xi^2 - 1}\frac{t}{\tau} + \frac{\xi}{\sqrt{\xi^2 - 1}} \sinh\sqrt{\xi^2 - 1}\frac{t}{\tau}\right)\right] \quad (8)$$

Impulse function

$$X(t) = \delta(t) \rightarrow X(s) = 1 \rightarrow Y(s) = \frac{\mathrm{Kp}}{(\tau^2 s^2 + 2\tau\xi s + 1)} \rightarrow \cdots$$

$$\text{if } \xi < 1, \rightarrow \mathrm{Y(t)} = \mathrm{Kp}\left[\frac{1}{\tau\sqrt{1-\xi^2}} e^{-\frac{\xi t}{\tau}} \sin\left(\sqrt{1-\xi^2}\frac{t}{\tau}\right)\right] \quad (9)$$

$$\text{if } \xi = 1, \rightarrow \mathrm{Y(t)} = \mathrm{Kp}\left[\left(\frac{t}{\tau^2}\right) e^{-\frac{t}{\tau}}\right] \quad (10)$$

$$\text{if } \xi > 1, \rightarrow \mathrm{Y(t)} = \mathrm{Kp}\left[\frac{1}{\tau\sqrt{\xi^2 - 1}} e^{-\frac{\xi t}{\tau}} \sinh\left(\sqrt{\xi^2 - 1}\frac{t}{\tau}\right)\right] \quad (11)$$

8.1.3. Lag transfer function

Another function used in the control systems is the lag function which indicates the time lag. The following equation shows the lag transfer function. t_d denotes the time lag in equation 12.

$$\frac{Y(s)}{\mathrm{X(s)}} = e^{-t_d s} \quad (12)$$

The output of systems can be a combination of responses of different transfer functions to different input signals as shown in figure 8.2 in which $Y(s)=G_1X_1+G_2X_2+...+G_nX_n$. As all the calculations are in Laplace domain, inverse Laplace method is operated to calculate the output in the time domain.

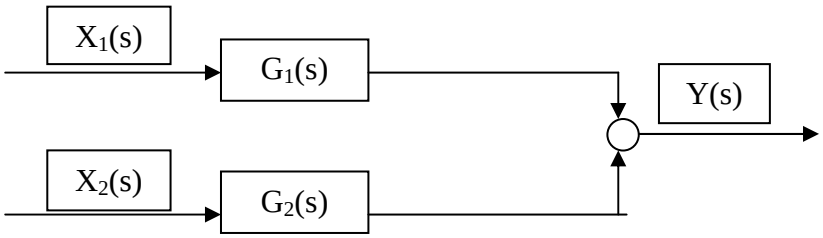


Figure 8.2 Combination of transfer functions

8.2. Methodology

In this section, it is described how transfer functions can be applied to model water-flooding and gas-flooding. These two are explained in individual subsections.

8.2.1. Water-flooding simulation by transfer function method

During water-flooding, production from each well is assumed due to two main mechanisms: the primary pressure of the reservoir and the injection drive. Hence, these mechanisms formulate the input signals, while the production rate represents the output signal. Once the appropriate transfer function is selected, the estimation of the future production for any injection rates (input signals) is feasible. The well production behaviour for input and output signals is modelled by a set of transfer functions to explain the injection influences and the reservoir pressure drive individually. Since the influence of the water drive on primary recovery is negligible and they act independently on production, the outputs of each transfer functions are added up. Different groups of transfer functions were checked to develop a combination which has physical meaning and also acceptable mathematical results.

The behaviour throughout water injection demonstrates that a combination of the first-order transfer functions simulates the water-flooding with tolerable error. First-order transfer functions usually are applied for those systems which are not accelerated such as flow in tank and Resistance Capacitance (RC) circuits. Therefore, a set of first-order TF is applied to model water-flooding due to the assumption of the reservoir as a tank. Also due to the depletion behaviour of reservoir during primary recovery, again a first-order TF is used to model primary production with plausible certainty.

For the output signals, it is assumed that the production of each well does not have much influence on the neighbouring wells. Hence, it is possible to simulate each production well separately with the same set of transfer functions. As discussed above, production is due to two main mechanisms: primary and secondary recovery. The primary recovery input signal is an ideal impulse, because when a well is completed in a reservoir (opened), the bottomhole pressure of the well acts like an initial impulse to the pressure profile of the reservoir. For the secondary recovery, step functions are applied to model the changes in water injection rates. It should be noted that there are a number of injection wells in almost all oil fields. Hence, the system consists of a series of first-order transfer functions with different input signals which are an ideal impulse signal and different step function signals for each injection well, for instance, if there are n injectors in the reservoir, the system for each production well consists of n+1 transfer functions, one for primary recovery modelling and n for the modelling of each injector well. By these functions, the effect of each injector on the total production of each well can be modelled. A lag function is also added to each section to illustrate the lag between the injection and its effect on the production profile. The system is shown in figure 8.3.

The output signal which is the liquid production rate of well j versus time is as

$$output(s) = \sum_{i=1}^{1+NoInjectors} G_i(s).Input_i(s) = \sum_{i=1}^{1+NoInjectors} \frac{Kp_i}{\tau_i s + 1}.Input_i(s) \quad (13)$$

$$P_j(t) = output(t) = \mathcal{L}^{-1}\big(output(s)\big) \tag{14}$$

$$P_j(t) = \frac{Kp_1}{\tau_1} e^{-\frac{t}{\tau_1}} + \sum_{i=1}^{n} A_i Kp_{i+1} \left(1 - e^{-\frac{t-a_i-t_{d_i}}{\tau_{i+1}}}\right) u\big(t - a_i - t_{d_i}\big) \tag{15}$$

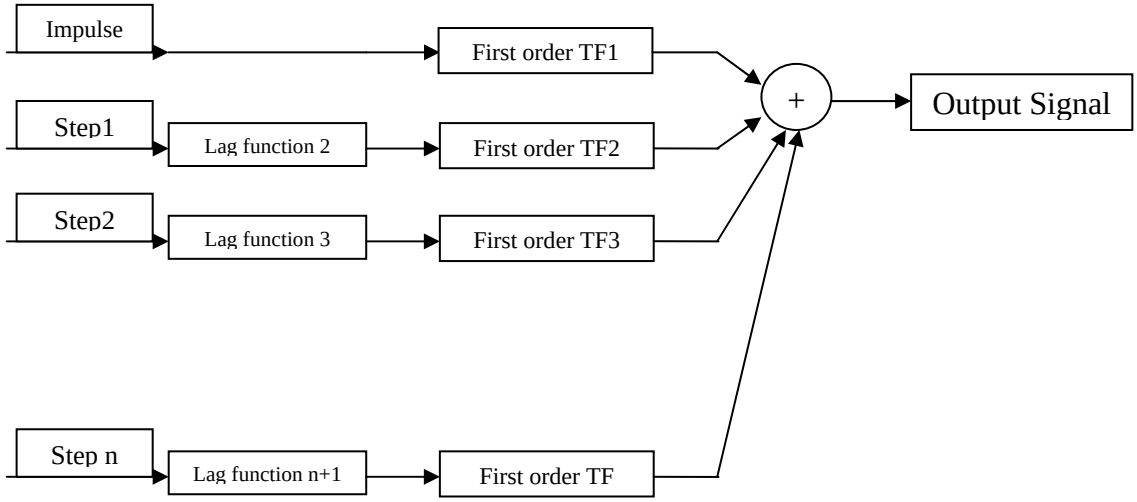


Figure 8.3 Water-flooding transfer function model

In equation 15, the first term on the right hand side shows the effect of reservoir pressure on the production rate of well number j. For this term, the input is $X_1(t)=\delta(t)$ and the transfer function is written as $G_1(s) = \frac{Kp_1}{\tau_1 s+1}$. The second term shows the effect of injection rates on two transfer functions which are a first-order and a lag function. In this case, the input function for injector i is $X_i(t)=A_i u(t-a_i)$, if it is assumed that just a change happened in injection rate, and the transfer function is $G_i(s) = \frac{Kp_{i+1} e^{-t_{d_i} s}}{\tau_{i+1} s+1}$. If the injection rates are changed a couple of times, the final solution will be different from equation 15, but it will be easily estimated through equation 13 and 14. In the results section, it is shown.

In the above equation, τ_1 is a time constant which is a function of depletion time of reservoir. The bigger τ_1 means that a longer time is needed for the depletion of primary stage, which demonstrates that there is higher pore volume or lower permeability around the producer j. Kp_1 is a function of initial production rate. Kp_{i+1} shows the connectivity of injection well i to the production well j. It is a function of the permeability of the section between the injector and producer and also the well locations and distances. τ_{i+1} reveals the

required time for the effect of the injection of well i to be seen by the producer. Time constants are proportional to the pore volume, total compressibility (C_t) and flow resistivity (1/k). It is obvious that if there is not any compressibility, the influence of injection on production can be seen instantly. On the other hand, if the pore volume is low, the producer needs less time to receive the influence of the injector. Also, if there is a higher transmissibility, less time is needed to receive the influence of the injector on the producer. t_{di} shows the lag time.

Equation 15 should be constructed for each producer well separately. As it is seen in the equation, there are still some unknown parameters like Kp_1,.. Kp_{n+1}, τ_1,.. τ_{n+1}, t_{d1}..t_{dn} which should be identified. These are the matching parameters, and they are different from case study to case study. To estimate these parameters, a history of production and injection rates is used. As the number of variable is reasonable, the Generalized Reduced Gradient (GRG2) nonlinear optimisation method is employed to minimise the error in history matching and estimate the model parameters. Firstly, an initial guess is used then using the initial guess, the production is calculated after that the result should be compared with history. In this way, the difference of estimation and history is defined as an error. After history matching, the unknown parameters are estimated and using those parameters, it will be simple to predict the reservoir performance for any input other signals. These parameters are different for each case, but equation 15 is extendable for any cases (for forecasting the performance of any oil well under water-flooding recovery). History matching of this method is not expensive, as it has an analytical solution.

The current method forecasts the production of total fluids (water and oil). Hence a fractional flow model is needed to estimate the oil flow rate.

8.2.1.1. Fractional flow for water-flooding

There are different methods to predict water-cut profile such as the Buckley–Leveret model. In most fractional flow methods, water-cut usually is a function of saturation. The TF approach does not provide saturation distribution; hence the empirical methods such as Gentil model (equation 16) should be used to estimate oil performance. Gentil in 2005 introduced the power law relation of cumulative injected water and oil fraction. The model was developed by Lake in 2007. In the Gentil model, the water-cut is not a direct function of saturation. Fraction of oil, f_o is calculated as

$$f_o(t) = \frac{1}{1 + F_{wo}} = \frac{1}{1 + \alpha W_i^{\beta}} \quad (16)$$

Where α and β are Gentil constants which are calculated by history matching of production and injection data; W_i is the cumulative water injection. There are two unknown parameters, α and β, which are calculated by history matching. These parameters are matching parameters which are different for any case. Due to the difficulty of estimation of parameter in a power low relation, their logarithms are used. Similar to the TF method, the same approach is utilised. Production and injection history is used to calculate α and β. However, α and β are referred as dimensionless (Lake et al., 2007, Sayarpour et al., 2009b), at least α should have dimension because of dimension of W_i which is L^3. Figure 8.4 shows the algorithm and summarises the calculation for this study.

Using this combination (TF method and Gentil model), it is possible to estimate the future performance of oil wells under water-injection condition. However, the equations are derived based on some assumptions, they can be extended to any other oil wells with reasonable accuracy.

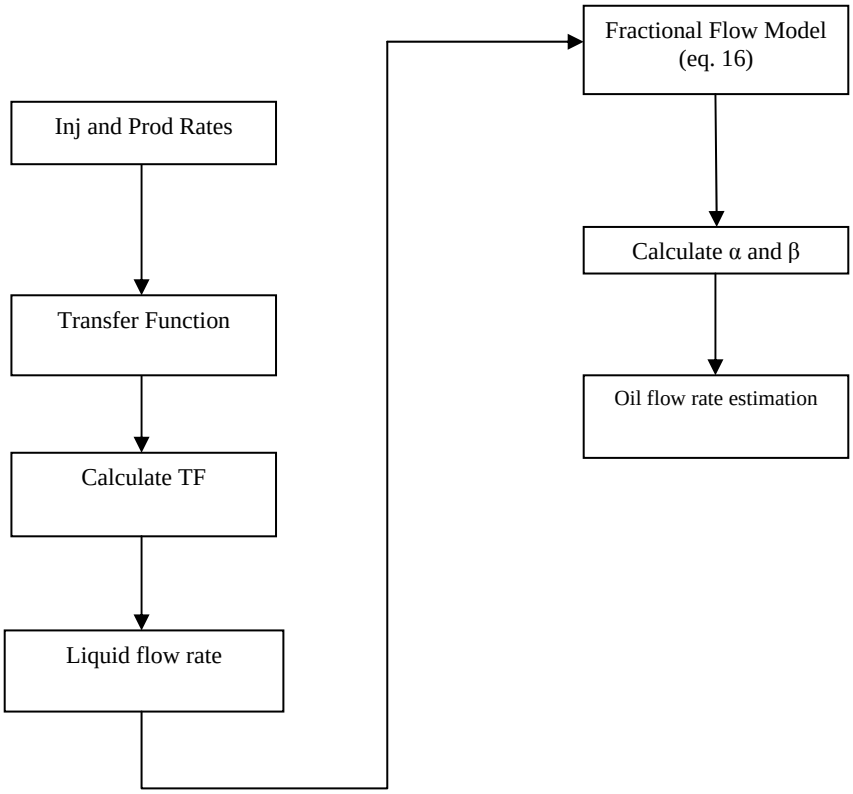


Figure 8.4 Transfer function approach algorithm for water-flooding

8.2.2. Gas-flooding simulation using transfer function method

In this section, it is explained how this approach can be used to forecast oil well performance during gas-flooding. The production rate constructs the output signal of system. As mentioned before, output signals can be the summation of outcomes of different input signals to different transfer functions. During the gas-flooding, production from well is governed by two main mechanisms: the initial pressure of reservoir and the gas injection drive, thus, these formulate the input signals. The primary recovery input signal is an ideal impulse, because when a well is completed in a reservoir, the bottomhole pressure of the well acts like an impulse to the pressure profile of the reservoir. For secondary recovery, step functions are applied to model changes in gas injection rates. It is desirable to seek the appropriate transfer functions to relate the output and input signals. It is taken for granted that the influence of the gas injection drive on natural drive is insignificant and they act independently on production. In addition, it is assumed the

production of each well does not have much influence on the neighbouring wells. Hence, it is possible to simulate each production well individually. In gas-flooding modelling, mass rates are used as signals instead of volume rate, because the volume of gas is much more than oil volume; and the changes in oil rate cannot be observed.

In order to achieve the appropriate transfer function set, different groups of transfer functions were checked to develop a combination which has physical meaning and also accurate mathematical results. The behaviour of the reservoir throughout the gas injection demonstrates that a combination of the first and second-order transfer functions simulates the gas-flooding with tolerable error. Second-order transfer functions usually are applied for those systems which are accelerated such as RCL circuit and accelerometer. In immiscible gas-flooding, due to the high compressibility of gas, the gas is compressed behind oil front until reaching the producer and then decompressed. Hence, it can be assumed the system is accelerated and a second-order transfer function is the best choice to model the change in gas injection rate. Due to the depletion behaviour of reservoir during primary recovery, a first-order TF is used to model primary production with plausible certainty.

It should be considered that there are a number of injection wells in almost all oil fields. Hence, the input signal consists of an ideal impulse signal and a set of step signals for different injectors. The transfer function also consists of a first-order function which shows the effect of the reservoir primary depletion and a set of second-order step functions to show the effects of different gas injectors. It means that if there are n injectors in the reservoir, the system consists of $n+1$ transfer functions. By these functions, the effect of each injector on the total production of a well can be modelled. A lag function is also added to each section to illustrate the lag between the injection and its effect on the production profile. The system is shown in figure 8.5.

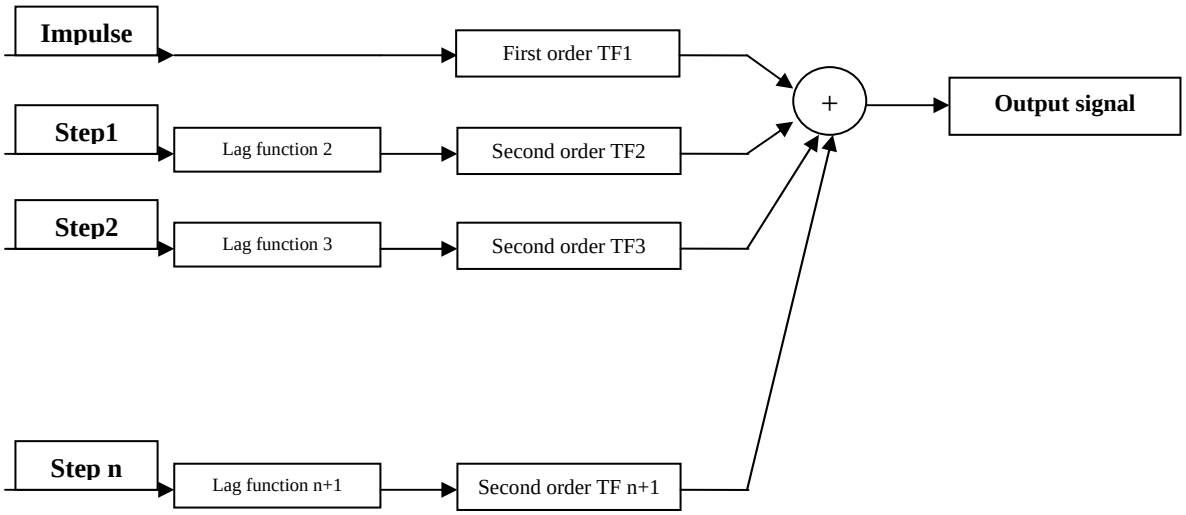


Figure 8.5 Gas-flooding transfer function model

Based on the figure 8.5, the output signal for producer j in a reservoir under gas-flooding condition is expressed as

$$Output(s) = \sum_{i=1}^{1+NoInjectors} G_i(s).Input_i(s) = \ldots$$

$$\ldots \frac{Kp_1}{\tau_1 s + 1}.Input_1(s) + \sum_{i=2}^{1+NoInjectors} \frac{\mathrm{K_{pi}} e^{-t_{d_i} s}}{\tau_i{}^2 s^2 + 2\tau_i \xi_i s + 1}.Input_i(s) \quad (17)$$

$$P_j(t) = Output(t) = \mathcal{L}^{-1}\big(Output(s)\big) \quad (18)$$

$$P_j(\mathrm{t}) = \frac{K_{p1}}{\tau_1} e^{-\frac{t}{\tau_1}} + \cdots$$

$$\ldots \sum_{\mathrm{i}=1}^{\mathrm{n}} A_i \mathrm{K_{pi+1}} \left[1 - \frac{1}{\sqrt{1-\xi_{\mathrm{i+1}}{}^2}} \mathrm{e}^{-\frac{\xi_{\mathrm{i+1}}(\mathrm{t}-t_{d_i})}{\tau_{i+1}}} \mathrm{Sin}\left(\sqrt{1-\xi_{\mathrm{i+1}}{}^2}\,\frac{\mathrm{t}-t_{d_i}}{\tau_{i+1}} + \tan^{-1}\frac{\sqrt{1-\xi_{\mathrm{i+1}}{}^2}}{\xi_{\mathrm{i+1}}}\right)\right] \ldots$$

$$\ldots u\big(t - a_i - t_{d_i}\big) \quad (19)$$

In equation 19, the first term in the right hand side shows the effect of reservoir pressure on the production rate. For this term, the input is $X_1(t) = \delta(t)$ and the transfer function is written as $G_1(s) = \frac{Kp_1}{\tau_1 s+1}$. The second term shows the effect of the injection rate on the production profile which can be shown by a second-order and a lag transfer function. In this case, the input function for injector i is $X_i(t) = A_i u(t - a_i)$ and the transfer function is $G_i(s) = \frac{\mathrm{K_{pi+1}} e^{-t_{d_i} s}}{\tau_{\mathrm{i+1}}{}^2 \mathrm{s}^2 + 2\tau_{\mathrm{i+1}} \xi_{\mathrm{i+1}} \mathrm{s} + 1}$. The changes in injection rates can be in any types.

In the above equation, τ_1 is a time constant which demonstrates the reservoir depletion time. It is a function of pore volume, well productivity index and total compressibility (Lake et al., 2007). Kp_1 is a function of initial pressure. Kp_{i+1} explains the connectivity of injector well *i* to the producer well *j*, therefore it is function of well locations and the permeability between the wells. τ_{i+1} shows the needed time for the producer to observe the effect of injection of well *i*. t_{d_i} shows the lag time for each producer. a_i shows the time of change in the rate of injector i. $\xi_{i+1}$is damping coefficient, once $\xi > 1$, the system is called overdamped, and once $\xi = 1$, it is called critical damping. In these two situations, oscillating is not seen. Bigger ξ demonstrates later equilibrium. If $\xi < 1$, it is called underdamped, in this condition, system oscillates. Lower ξ leads to more fluctuation until equilibrium. Equation 19 should be constructed for each producer well separately. There are some unknown parameters such as $Kp_1 .. Kp_{n+1}, \tau_1, .. \tau_{n+1}, t_{d_1} .. t_{d_n}$, $\xi_2 \dots \xi_{n+1}$ which must be identified to be able predict future performance.

In order to identify these parameters, the history of production and injection rates is used. Injection and production mass rates are only requirements for developing the model which shows the simplicity of approach. As the number of variable is reasonable, the Generalized Reduced Gradient (GRG2) nonlinear optimisation method is employed to minimise the error in history matching and estimate the model parameters. After history matching, the unknown parameters would be estimated and using those parameters, the prediction of reservoir performance for any input signals would be feasible. These parameters are different for each case, but the equation 19 is extendable for any cases. History matching for this method is also simple, as it has analytical solution.

8.2.2.1. Fraction flow for gas-flooding

The output signal is the mass production flow rate for each well. Hence, to estimate oil production rate, a fractional flow model should be coupled with the approach. There are some conventional methods to calculate phase fractions in reservoir simulations which need flash calculations in compositional models. The flash calculations determine phase properties and the amount of phases for a given temperature, pressure, and overall composition. There are two basic formulations for the flash calculations, minimisation of the Gibbs free energy and the solution of fugacity equations. These formulations leads to a general equation for a two/three phase systems (multiphase Rachford-Rice equation, 1952). In the grid-based simulation methods to find the fractional flow, it is adequate to solve material balance equations to find phase rates separately and calculate fractional flow values (Ertekin et al., 2001b).

The aforementioned methods are not appropriate in the fast simulators, because of lack of sufficient information to solve the relative equations. Hence, it is necessary to find an appropriate approach for the fractional flow calculation. Gentil (2005) did a similar work for water-flooding process and derived an approach to the find fractional flow as a function of cumulative fluid produced in water-flooding .

In this section, a new method to find an appropriate fractional flow model in gas-flooding process is introduced. The oil production regression model is based on the assumption of linear relationship between the logarithms of oil fraction [log (f_o)] and the cumulative injected gas [G_i]. When injection and production are in balance:

$$G_i \approx N_p + G_p = Q_p \tag{20}$$

Where,

G_i: Cumulative gas injected is approximated by the total fluids produced
N_p: cumulative oil produced
G_p: cumulative gas produced
Q_p: cumulative total fluids produced

Once the injection-production balance is achieved on a pattern-by-pattern basis, then expression (equation 20) can be applied for each producer. Using above assumption;

$$Log(f_o) = A + B(Q_p) \quad (21)$$

Using this approach in this format:

$$f_o = ab^{Q_p} \quad (22)$$

"Data fit version 9.0.x by Oakdale Engineering" is utilised to find the model (equation 22). Since this approach is obtained in a steady-state condition, hence it should be applied only for after breakthrough time.

Using this combination (TF method and developed fractional flow model), it is possible to estimate the future performance of oil wells under gas injection conditions. However, the equations are derived based on some assumptions, they can be extended to any other oil wells with a reasonable accuracy.

8.3. Results

In this section, different synthetic reservoirs are used to verify the developed method. The results section is divided into two subsections in which the developed method for water-flooding modelling and gas-flooding modelling are separately studied. Water-flooding in the synthetic reservoirs are modelled with the method and the results for the fluid production rates are compared to the grid-based simulations results (ECLIPSE-E100). Gas-flooding processes in different synthetic cases are modelled with the proposed method and the results are compared with grid-based simulation results (CMG).

8.3.1. Water-flooding model verification

Four difference reservoir models, from homogenous to complex and heterogeneous models are used to investigate the validity of the proposed method in modelling water-injection.

8.3.1.1. Case study#1: single injector/single producer

The simplest case is a reservoir with only one injector and one producer as shown in figure 8.6. Table 8.2 shows the main rock and fluid properties. The first case is totally homogenous. In this example, production in the first 300 days is due to natural depletion of the reservoir. In the next 300 days, there is an injection rate equal to 795 cu.m/day (5000 bbl/day) and the injection is stopped after that. The input signals can be written as

$$X_2(t) = 795u(t-300) - 795u(t-600) \rightarrow Injection\ Function$$

$$X_1(t) = \delta(t) \rightarrow Primary\ Recovery\ Function$$

The total production rate of prod #1 is modelled using equation 15. The first term on the right hand side in the following equation shows the production due to primary recovery and the other terms show the effect of injection on the production profile.

$$P_1(t) = \frac{Kp_1}{\tau_1} e^{-\frac{t}{\tau_1}} + 795Kp_2\left(1 - e^{-\frac{t-300-t_d}{\tau_2}}\right)u(t-300-t_{d2}) - 795Kp_2\left(1 - e^{-\frac{t-600-t_d}{\tau_2}}\right)\ldots$$

$$\ldots u(t-600-t_{d2}) \qquad (23)$$

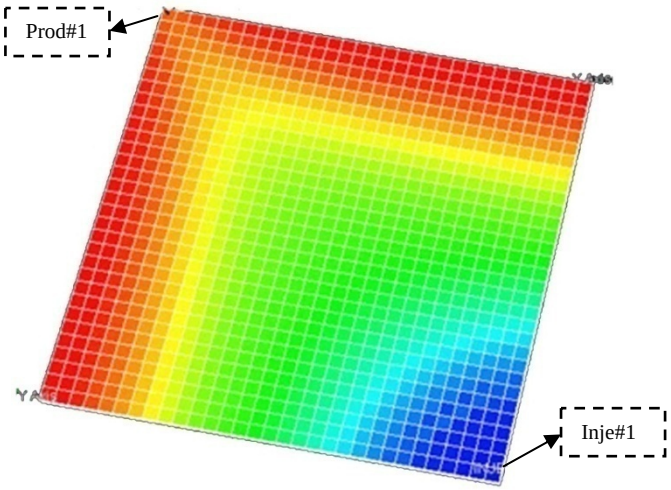


Figure 8.6 Oil saturation for case#1

The total production rate as a function of time (P(t)) is known, while the unknown parameters are Kp_2, Kp_1, τ_1, τ_2, t_d. The production and injection histories are used to calculate the unknown parameters. It should be noted that in the real cases, the input data for the model can be obtained from the history of field injection and production rates.

Here, numerical simulation results are assumed as histories. Microsoft Excel Solver is employed to calculate the unknown parameters by matching the obtained production rate from the model and the real production rate. It should be mentioned that there are some constraints for these parameters: Kp_2 (well connectivities) is less than one as it shows the fraction of the injection which reaches to the producer. Also, none of the parameters (neither connectivity nor time constants) can have negative values.

Table 8.2 The property of fluid and rock for case#1

Permeability in X-Direction	$\mathbf{5.92 \times 10^{-14}}\ \boldsymbol{m^2}$(60 md)
Permeability in Y-Direction	$\mathbf{5.92 \times 10^{-14}}\ \boldsymbol{m^2}$(60 md)
Permeability in Z-Direction	$\mathbf{1.48 \times 10^{-14}}\ \boldsymbol{m^2}$(15 md)
Porosity	0.15
No. of Grids in X,Y,Z Direction	31,31,10
Dimension of Grids in X,Y,Z Direction	30.48, 30.48, 3.48 m
Rock Compressibility	0.000058 1/bar
Reservoir Depth	1828.8 m
Phases	Oil and Water
Oil Viscosity	2 cp @ 413 bar
Initial Pressure	376.5 bar @ 1851 m

Figure 8.7 and table 8.3 show the results for this case and also the comparison to the grid-based simulator results. The results demonstrate a good agreement with each other. t_d shows that there is 10 days lag for the results and Kp_2 demonstrates there is almost 84% connectivity between the injector and the producer as it is expected.

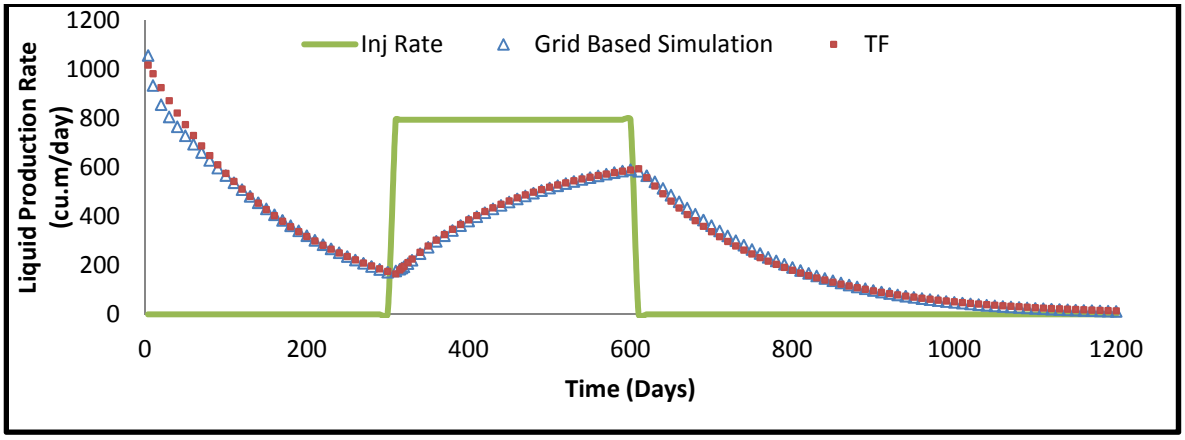


Figure 8.7 Comparison of transfer function simulator and grid-based simulator for case#1

Table 8.3 The parameters of model for case#1

$K_{p1}/\ \boldsymbol{\tau_1}$	$\boldsymbol{\tau_1}$	K_{p2}	$\boldsymbol{\tau_2}$	t_{d2}
6559.9	168.45	0.839	158.731	-10

As already mentioned, the model predicts the liquid performance. The Gentil model is used to determine oil production rate as well. The Gentil model has two unknown parameters, α and β, which are calculated using history matching. Fractional flow matching is done just for this example; more details for the fractional model matching can be found in the literature. Table 8.4 reveals the Gentil parameters for the case 1. Figure 8.8 illustrates a good match between the grid-based simulation data and Gentil model.

Table 8.4 The parameters of Gentil model for case#1

α	β
-2.6×10^{-10}	1.99

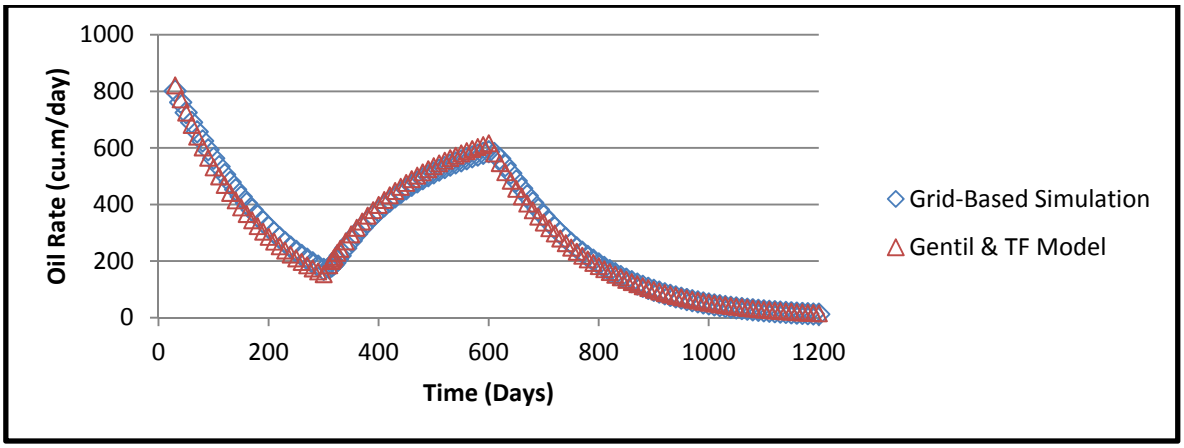


Figure 8.8 Oil prediction using Gentil model and TF and its difference from the grid-based simulator for case#1

Now, it is possible to use the confirmed model to study the performance of different scenarios of injection to choose the best plan for injection rates.

8.3.1.2. Case study#2: 2 injectors/2 producers with anisotropy

In this case, two injectors and two producers are used to define the influences of two injectors on the producers and the connectivity between wells. Figure 8.9 shows the location of wells and also the oil saturation profile after the injection. Table 8.5 reveals the rock and fluid properties. The permeability in y-direction is five times more in comparison with x-direction ($K_y=5K_x$). The output signal for each producer well can be written as the

following equation which includes the primary recovery effect and effects of two injectors:

$$Y(s) = X1(s)G1(s) + X2(s)G2(s) + X3(s)G3(s)$$

The injector #1 was shut for the first 300 days, and then its rate was changed to 1113 cu.m/day (7000 bbl/day). After that, it shut again for 200 days and finally it opened with the injection rate 1431 cu.m/day (9000 bbl/day). The injector well 2 was shut for the first 600 days then its rate was changed to 1113 cu.m/day (7000 bbl/day) for 200 days and at the end the injection rate was 158 cu.m/day (1000 bbl/day). The injection models are as follows:

$$X_1(t) = \delta(t) \qquad (Primary\ Recovery\ Function)$$

$$X_2(t) = 1113u(t-300) - 1113u(t-600) + 1431u(t-800)$$

$$X_3(t) = 1113u(t-600) - 954u(t-800)$$

Hence,

$$P_1(t) = \frac{Kp_1}{\tau_1} e^{-\frac{t}{\tau_1}} + 1113Kp_2\left(1 - e^{-\frac{t-300-t_{d2}}{\tau_2}}\right)u(t-300-t_{d2}) - \cdots$$

$$1113Kp_2\left(1 - e^{-\frac{t-600-t_{d2}}{\tau_2}}\right)u(t-600-t_{d2}) + \cdots$$

$$1431Kp_2\left(1 - e^{-\frac{t-800-t_{d2}}{\tau_2}}\right)u(t-800-t_{d2}) + \cdots$$

$$1113Kp_3\left(1 - e^{-\frac{t-600-t_{d3}}{\tau_3}}\right)u(t-600-t_{d3}) - \cdots$$

$$954Kp_3\left(1 - e^{-\frac{t-800-t_{d3}}{\tau_3}}\right)u(t-800-t_{d3})$$

A similar approach is used to determine the unknown parameters. The results are shown in figures 8.10 and 8.11 and table 8.6. As expected there is more connectivity between injector 2 and producer 1 because of permeability anisotropy and the distance between the wells. Well connectivity is a key parameter which provides the influence of injected water

on each producer. Also time constant shows that how long it takes for a producer to sense the effect of each injector during water-flooding.

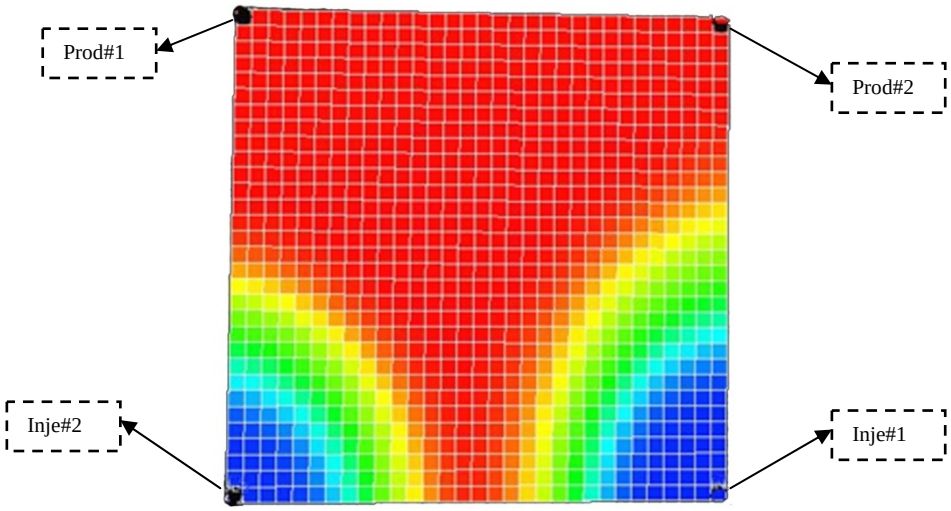


Figure 8.9 The well locations and oil saturation for case#2

Table 8.5 Fluid and rock properties for case#2

Permeability in X-Direction	$\mathbf{2.96 \times 10^{-14}\ m^2}$(30 md)
Permeability in Y-Direction	$\mathbf{14.81 \times 10^{-14}\ m^2}$(150 md)
Permeability in Z-Direction	$\mathbf{1.481 \times 10^{-14}\ m^2}$(15 md)
Porosity	0.15
No. of Grids in X,Y,Z Direction	31,31,10
Dimension of Grids in X,Y,Z	30.48, 30.48, 3.48 m
Rock Compressibility	0.000058 1/bar
Reservoir Depth	1828.8 m
Phases	Oil and Water
Oil Viscosity	2 cp @ 413 bar
Initial Pressure	376.5 bar @ 1851 m

Table 8.6 Model parameters for case#2

Prod#1					
K_{p1}/ $\boldsymbol{\tau_1}$	τ_1	K_{p2}	τ_2	t_{d2}	
9999.105	63.03	0.36	84.66	-8.5	INJ#1
		K_{p3}	τ_3	t_{d3}	
		0.514	61.76	0	INJ#2
Prod#2					
K_{p1}/ $\boldsymbol{\tau_1}$	τ_1	K_{p2}	τ_2	t_{d2}	
7999.988	67.38	0.509	47.671	0	INJ#1
		K_{p3}	τ_3	t_{d3}	
		0.360	65.707	-10	INJ#2

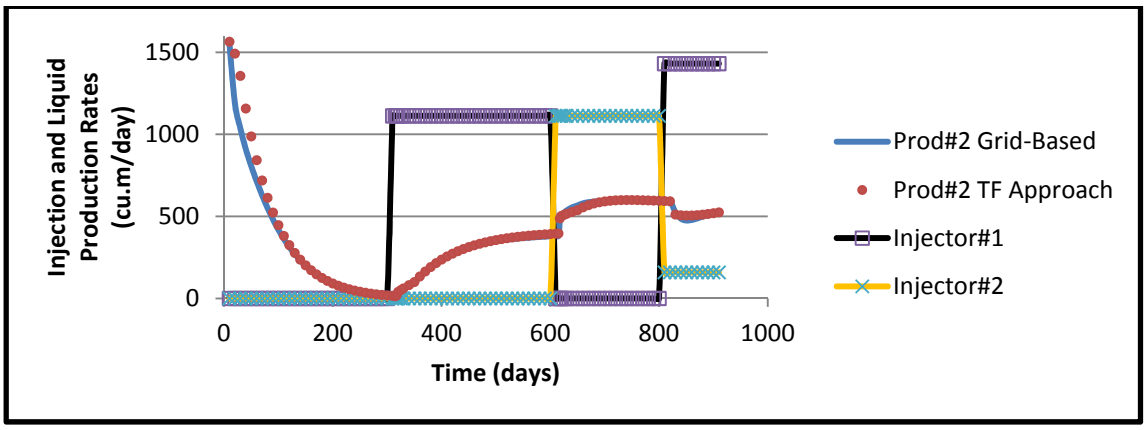


Figure 8.10 Comparison of transfer function simulator and grid-based simulation for Prod#2 of case#2

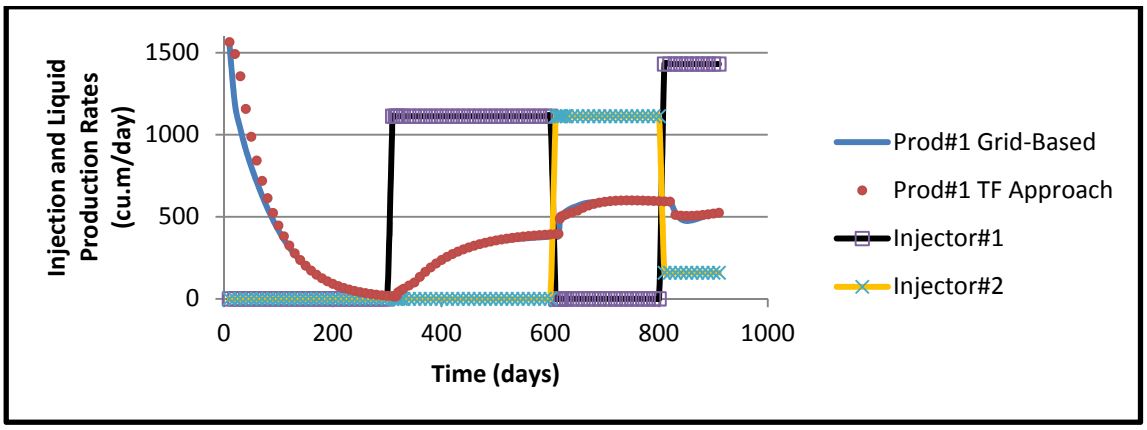


Figure 8.11 Comparison of transfer function simulator and grid-based simulation for Prod#1 of case#2

8.3.1.3. Case study#3: 9-spot heterogeneous

In this case, rock and fluid properties are same to case 2 which are shown in Table 8.5. The only difference is permeability distribution. One quarter of the reservoir has different permeability (180md in x and y direction) from the other parts (20md in x and y direction) as shown in Figure 8.12. In this case, there are 5 injectors and 4 producers which 4 wells are located in the high permeable zone and the rest are located in the low-permeable zone. Equation 15 is also used for this case to model production rate for each well. Injection profile for injector wells are shown in figure 8.13.

Figure 8.14 shows the comparison of the model and a grid-based simulation result for the Producer 4 as an example. It shows a good consistency with the numerical simulation for a complicated case with heterogeneity. Table 8.7 demonstrates the TF parameters for the case 3. As expected, connectivities between the Injector 1 to Producer 2 and 4 (0.40) are larger than the connectivities between the Injector 1 to Producer 1 and 3 (0.05), due to the

shape of permeability distribution. As mentioned before, time constant is a function of pore volume, permeability distribution between each injector/ producer pair and total compressibility. In this case, porosity and compressibility are equal in each section of the reservoir. Hence, the value of time constant is less for high permeable zone as the influence of the injection reaches to the producer more quickly.

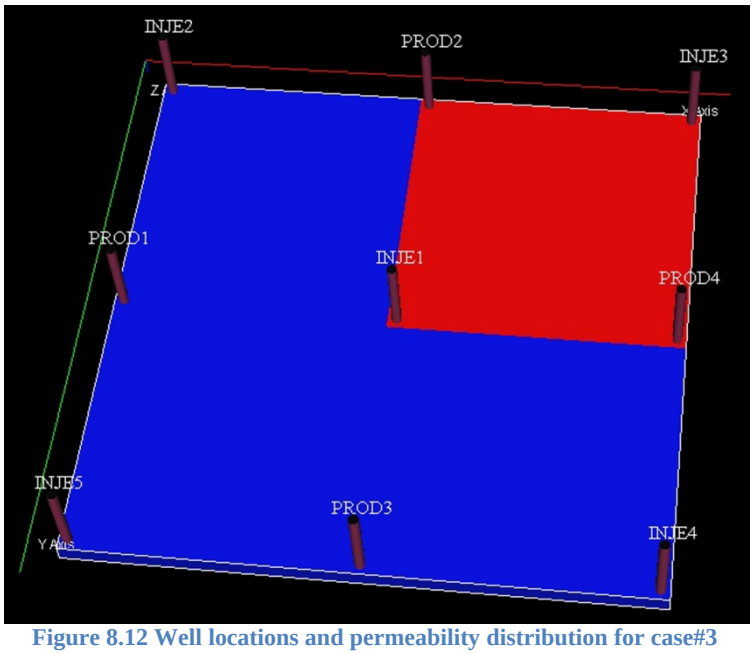


Figure 8.12 Well locations and permeability distribution for case#3

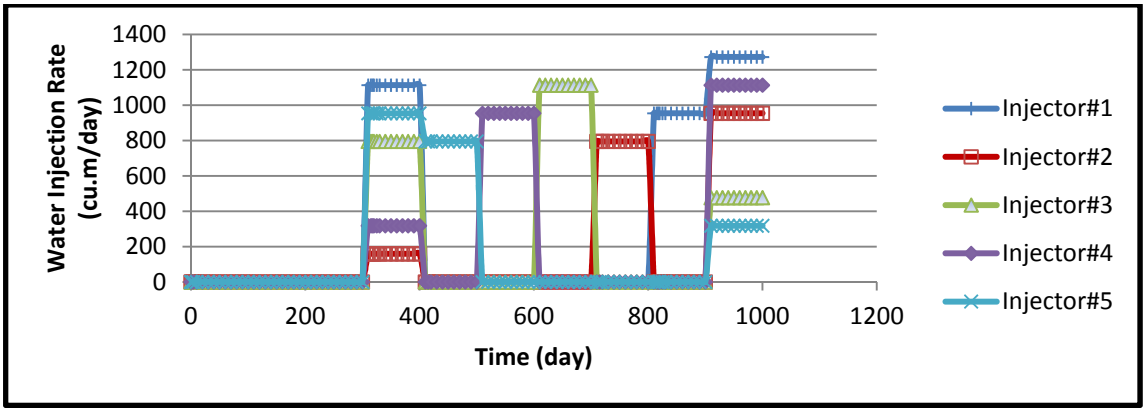


Figure 8.13 Injection profile for case#3

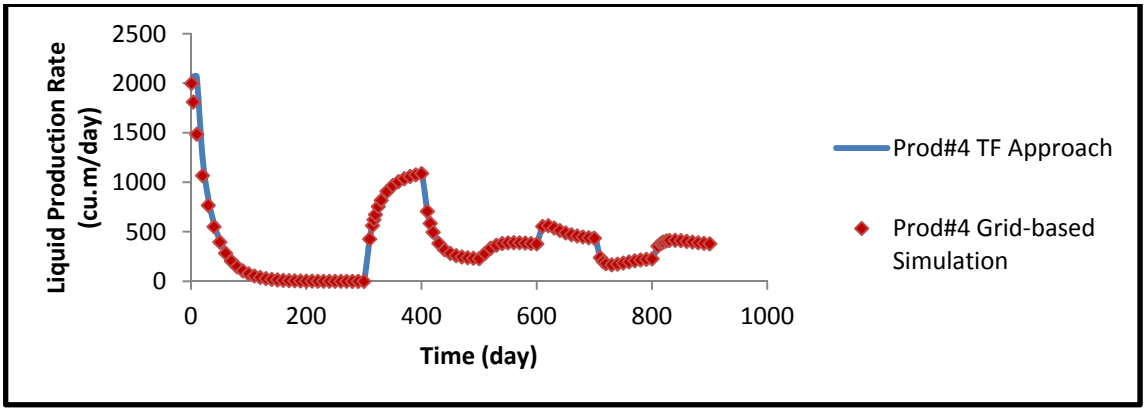


Figure 8.14 Comparison of TF simulator and grid-based simulation for Prod#4 in case#3

Table 8.7 Parameters for case#3

	Inj#1		Inj#2		Inj#3		Inj#4		Inj#5	
Well No.	Kp2	τ2	Kp3	τ3	Kp4	τ4	Kp5	τ5	Kp6	τ6
Prod#1	0.054	28.57	0.160	25.97	0.030	15.90	0.054	50.61	0.195	33.69
Prod#2	0.395	22.64	0.392	22.22	0.369	9.234	0.258	49.46	0.258	53.52
Prod#3	0.047	26.00	0.061	68.62	0.03	29.88	0.163	26.49	0.196	30.97
Prod#4	0.377	18.11	0.308	39.30	0.377	11.45	0.359	25.29	0.280	60.47
	Sum		Sum		Sum		Sum		Sum	
	0.874		0.923		0.809		0.835		0.929	

8.3.1.4. Case study#4: heterogeneous with faults

The last case is a synthetic reservoir which is more complicated and similar to the reality in comparison to previous cases. It has three faults, five water injectors and eight oil producers. The permeability in x and y direction are equal and shown in Figure 8.15. The reservoir is divided to 35, 35 and 5 grids in x, y and z direction respectively. Oil and water flow in the reservoir. The average porosity is around 20% and the porosity for each grid is between 10% and 31%. The average permeability in x and y direction is around 100 md and it deviates between 1.5 and 500 md. Initial pressure is 356 bar at 778 m depth. Rock compressibility is 0.0000058 1/bar at 310 bar. Water viscosity is equal to 0.8 cp at 310 bar. The injection profiles for all injectors can be seen in figure 8.16. In this case, there are more fluctuations in injection rates compare to previous cases.

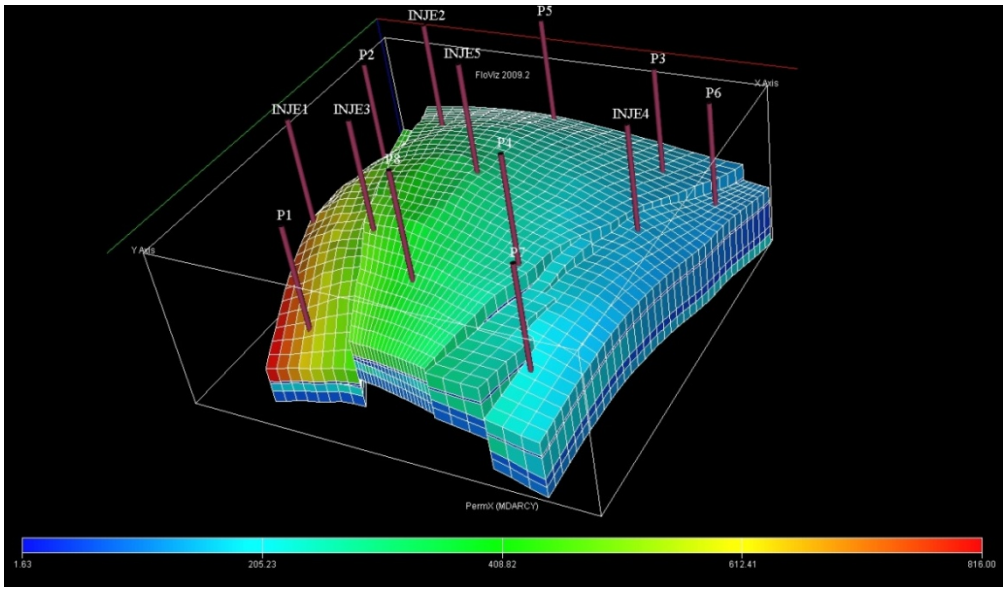


Figure 8.15 Permeability distribution and well locations for case#4

The same procedure is used to estimate the TF unknown parameters. In this case, fault trasmissibilities and permeability distribution are the main parameters which have the most influence on TF parameters especially the well connectivities. Some of the well connectivities are close to zero due to discontinuities in the reservoir and well distances. The estimated parameters are shown in Table 8.8 and 8.9. There is a good agreement between the estimated parameters and the physical condition of reservoir.

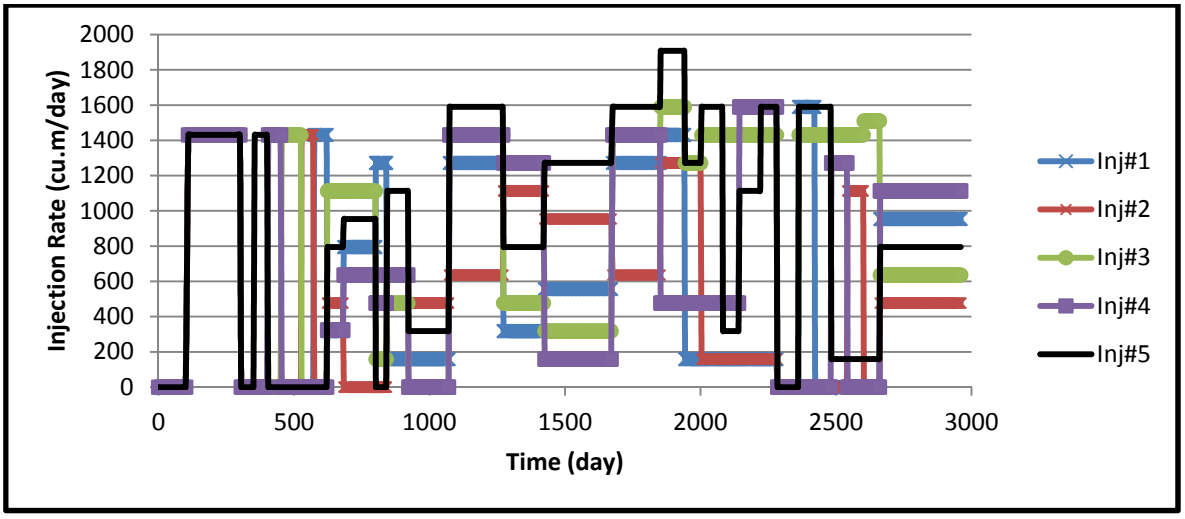


Figure 8.16 Injection profile for injector for case#4

However the case 4 is complicated and has fault and distributions of permeability and porosity, a good match can be seen for production wells in figures 8.17 to 8.24.

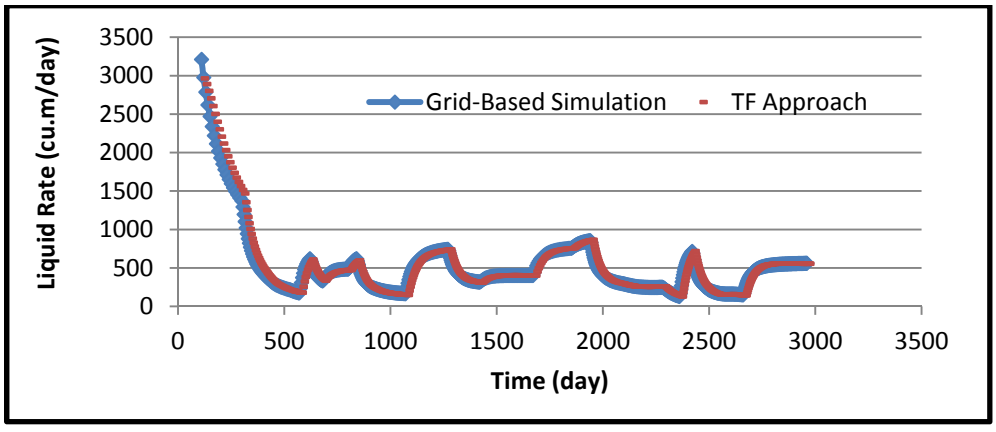


Figure 8.17 Comparison of TF simulator and grid-based simulator for Prod#1 case#4

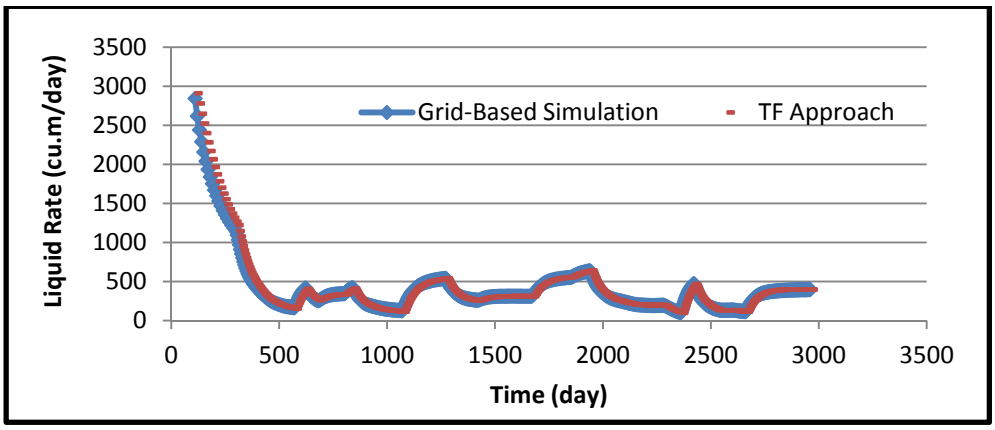


Figure 8.18 Comparison of TF simulator and grid-based simulator for Prod#2 case#4

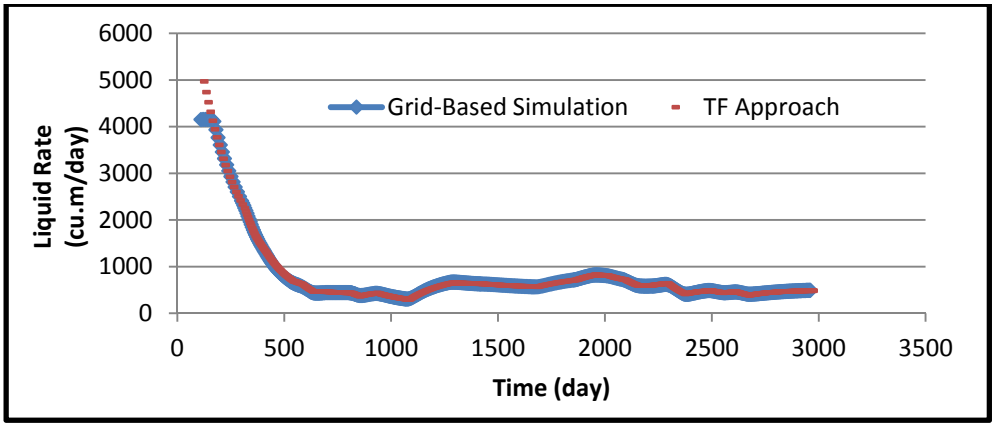


Figure 8.19 Comparison of TF simulator and grid-based simulator for Prod#3 case#4

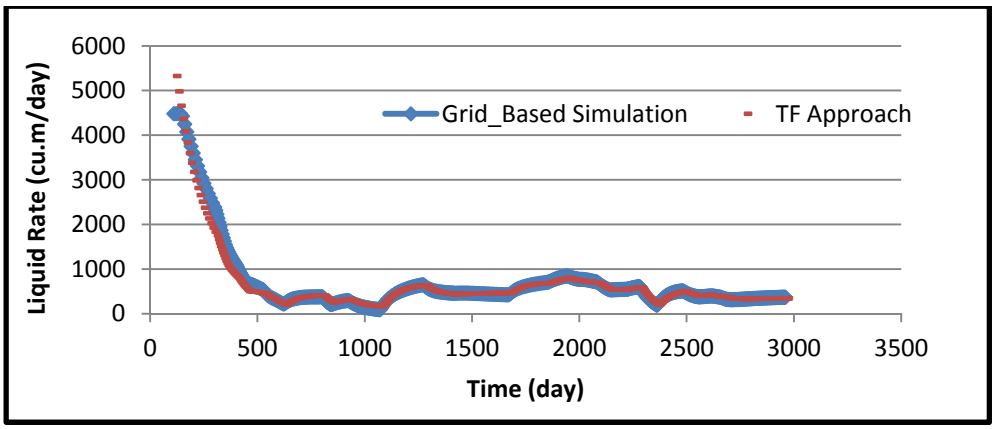


Figure 8.20 Comparison of TF simulator and grid-based simulator for Prod#4 case#4

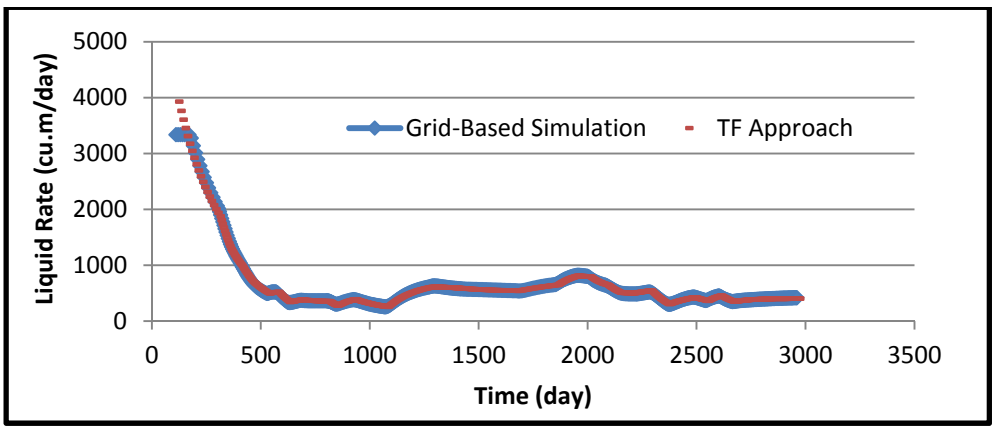


Figure 8.21 Comparison of TF simulator and grid-based simulator for Prod#5 case#4

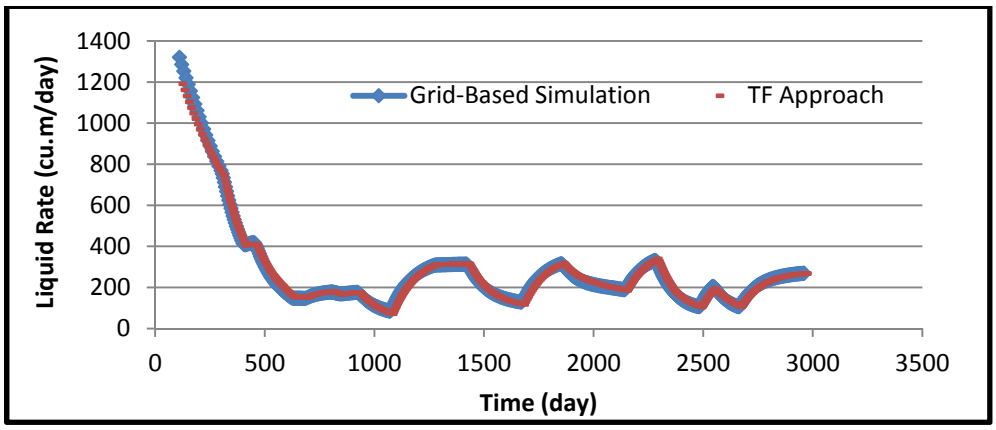


Figure 8.22 Comparison of TF simulator and grid-based simulator for Prod#6 case#4

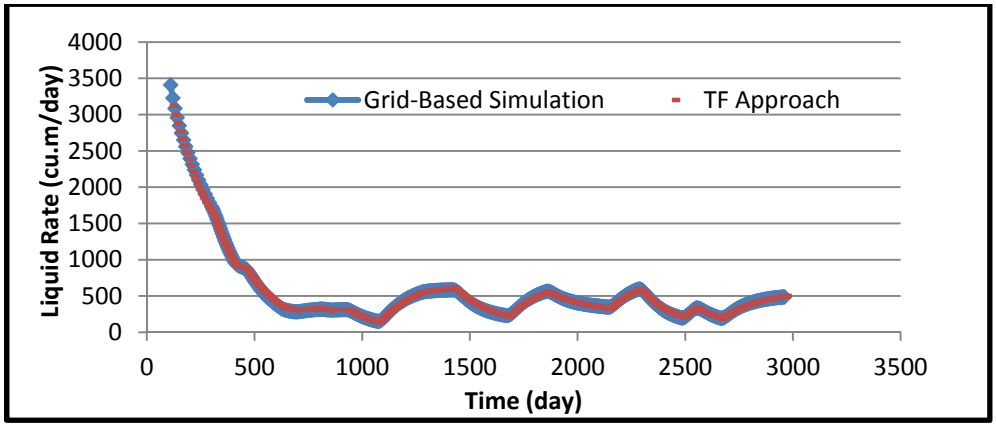


Figure 8.23 Comparison of TF simulator and grid-based simulator for Prod#7 case#4

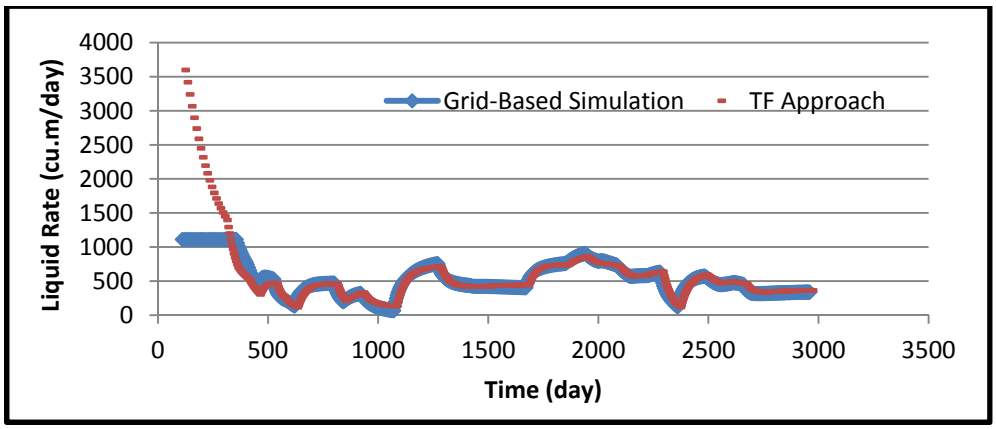


Figure 8.24 Comparison of TF simulator and grid-based simulator for Prod#8 case#4

To sum up, it was demonstrated in the four examples that the derived method is capable of delivering high-quality results (forecasts) during water-injection. By estimating the matching parameters which are different case to case according to the properties of each reservoir, the developed model can predict the future performance of liquid and oil production rapidly with reasonable accurately. Due to its rapid estimation, this method is

suitable to be employed to find the most economical injection rates. It should be mentioned that, the production and injection histories are needed to build the model.

8.3.2. Gas-flooding model verification

Six difference reservoir models, from homogenous to complex and heterogeneous models are used to investigate the validity of the proposed method in modelling gas injection. These cases include immiscible and miscible flooding. In the last two cases, the proposed fractional flow model is applied to match oil production as well.

8.3.2.1. Case study#1: immiscible gas-flooding in a homogenous reservoir with one injector and twp producers

The first case is a simple one with one injector and two producers as it is shown in the figure 8.25. Table 8.8 states the main rock and fluid properties. The injected gas is Nitrogen and it is an immiscible flooding. In this example, production in the first 1100 days is due to natural depletion of the reservoir. In the next 1100 days, there is an injection rate equal to 47000Kg/day. The input function can be written as follows:

$$X_1(t) = \delta(t) \rightarrow Primary\ Recovery\ Function$$

$$X_2(t) = 47000u(t-1100) \rightarrow Injection\ Function$$

Therefore, the total production rate of Prod 1 is written using equation 19:

$$P_1(t) = \frac{\mathrm{K_{p1}}}{\tau_1} e^{-\frac{t}{\tau_1}} + \cdots$$

$$47000\mathrm{K_{p2}}\left[1 - \frac{1}{\sqrt{1-\xi_2{}^2}} \mathrm{e}^{-\frac{\xi_2(\mathrm{t}-t_{d_1})}{\tau_2}} \sin\left(\sqrt{1-\xi_2{}^2}\,\frac{(\mathrm{t}-t_{d_1})}{\tau_2} + \tan^{-1}\frac{\sqrt{1-\xi_2{}^2}}{\xi_2}\right)\right] u(t-1100-t_{d_1})$$

A history for production and injection rates is generated by a grid-based simulator (CMG). This history is used as the input and output signals to calculate the unknown parameters of the model. These parameters are Kp_2, Kp_1, τ_1, τ_2, t_{d1}, ξ_2, ξ_3.

Table 8.8 Fluid and rock properties for case#1

Property	Value
Permeability in X-Direction	30 md
Permeability in Y-Direction	30 md
Permeability in Z-Direction	30 md
Porosity	0.20
No. of Grids in X,Y,Z Direction	8,8,1
Dimension of Grids in X,Y,Z Direction	100,100,10 m
Rock Compressibility	0.000004 1/psi
Reservoir Depth	6000 ft
Phases	oil, water and gas
Initial Pressure	17000 KPa

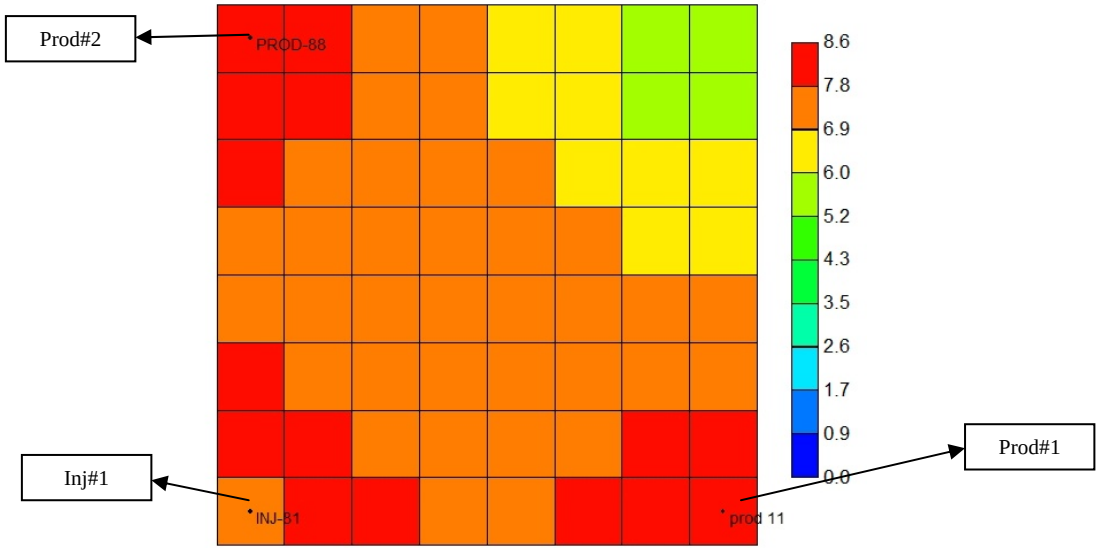


Figure 8.25 Well locations and Inter Facial Tension (IFT) for case#1

Microsoft Excel Solver is employed to estimate the unknown parameters using history matching. It should be noted that there are some constrains for parameters: Kp_{i+1} should be less than one as it shows the fraction of the injection which reaches to the producer. Also, all of the parameters such as connectivity values and time constants should have positive value. Table 8.9 shows the calculated parameters for this case. Finally, after the calculation of unknown parameters, the estimation of the production rate by the model will be possible for any injection rates. The results for this case are compared with the grid-based simulator

results which are shown in figure 8.26. The results demonstrate a good agreement. Since this case is symmetric and homogeneous, the results for both producers are the same and only producer 1 is shown in the figure.

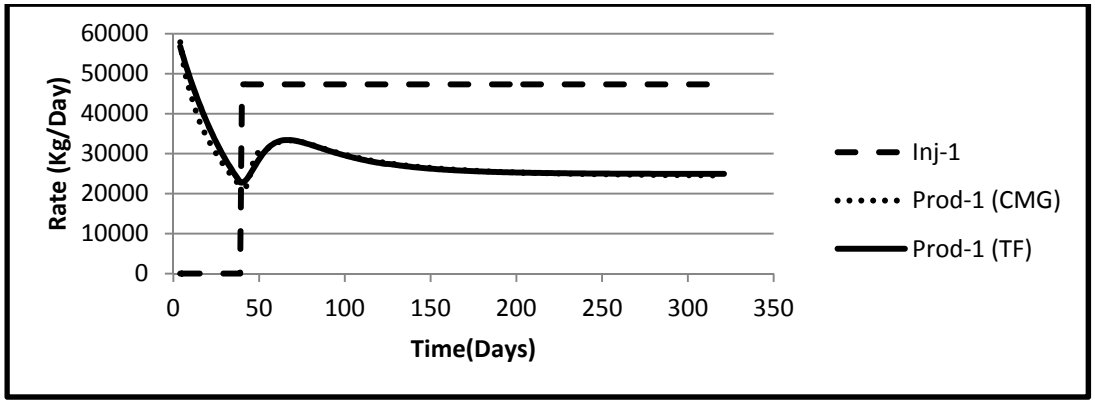


Figure 8.26 Comparison between TF and grid-based simulator results for Case#1 for Prod#1 and the injection profile

Table 8.9 The parameter of Case#1

Well	Kp_1/τ_1	τ_1	Kp_2	ξ	τ_2
Prod#1	59284.8	1153.5	0.52	0.985	226.0
Prod#2	59281.7	1145.7	0.52	0.99	225.1

As it is shown in the Table 8.9, the summation of well connectivities is exceed one slightly, it may be because of optimisation error.

8.3.2.2. Case#2: immiscible gas-flooding in an anisotropy reservoir with one injector and two producers

The second case is similar to the previous one, the injection profile has more fluctuations and there is anisotropy in the reservoir. The permeability in x-direction is 10md and in y-direction is 40md. The anisotropy is created; in order to investigate well connectivities. Figure 8.27 shows the injection profile, and the input signal is:

$$X2(t) = 17750u(t-1116) + 17750u(t-2942) - 11830u(t-4768)$$
$$- 23660u(t-6594) + 23660u(t-6960)$$

Similar approach is used to calculate the unknown parameters. The results are shown in figure 8.27 and table 8.10. However, it is more complicated, still a good match can be seen.

As it is expected, the connectivity between injector 1 with producer 2 is higher than connectivity between injector 1 and producer 1 which is due to anisotropy in permeability distribution.

Table 8.10 Parameters of Equations for Case#2

Well	Kp_1/τ_1	τ_1	Kp_2	ξ	τ_2
Prod#1	42994.5	1387.6	0.42	0.99	352.2
Prod#2	42995.8	1347.3	0.63	0.99	274.3

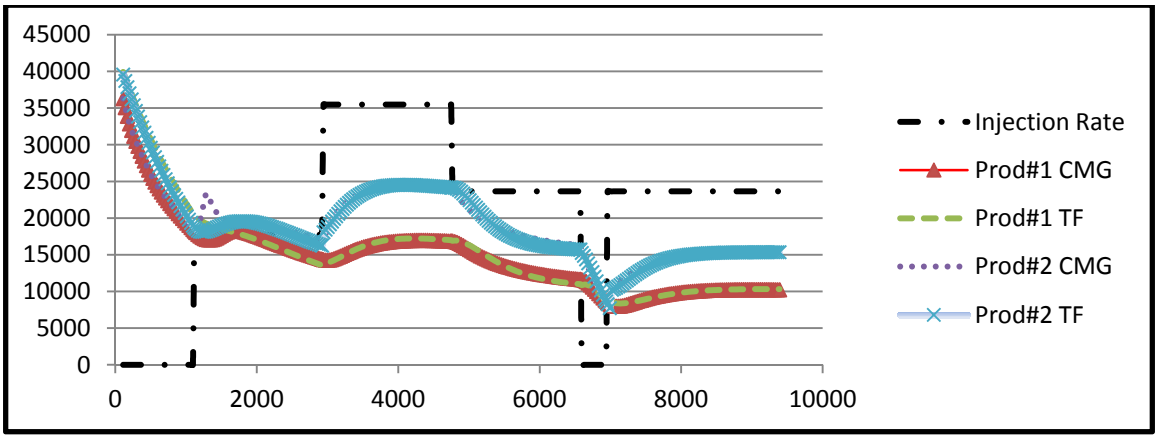


Figure 8.27 Comparison between TF and grid-based simulator results for Case#2 and the injection profile

8.3.2.3. Case study#3: miscible gas-flooding in a homogenous reservoir with one injector and two producers

In this case, the performance of miscible carbon dioxide gas-flooding in a reservoir using TF has been investigated. The properties are same to the previous cases. Figure 8.28 shows well location and IFT of the field. Miscibility can be shown by IFT profile while gas-flooding occurred.

Similar to the previous cases, the connectivity factors between injector and producers were calculated. The unknown parameters are shown in table 8.11. Figure 8.29 shows the comparison of TF model and the grid-based simulation result. Due to homogeneity the result of one well is only shown because both are similar.

Table 8.11 Parameters of model for Case#3

Well	Kp_1/τ_1	τ_1	Kp_2	ξ	τ_2
Prod#1	59280.0	1029.6	0.52	0.96	518.2
Prod#2	59283.1	1045.1	0.52	0.97	517.1

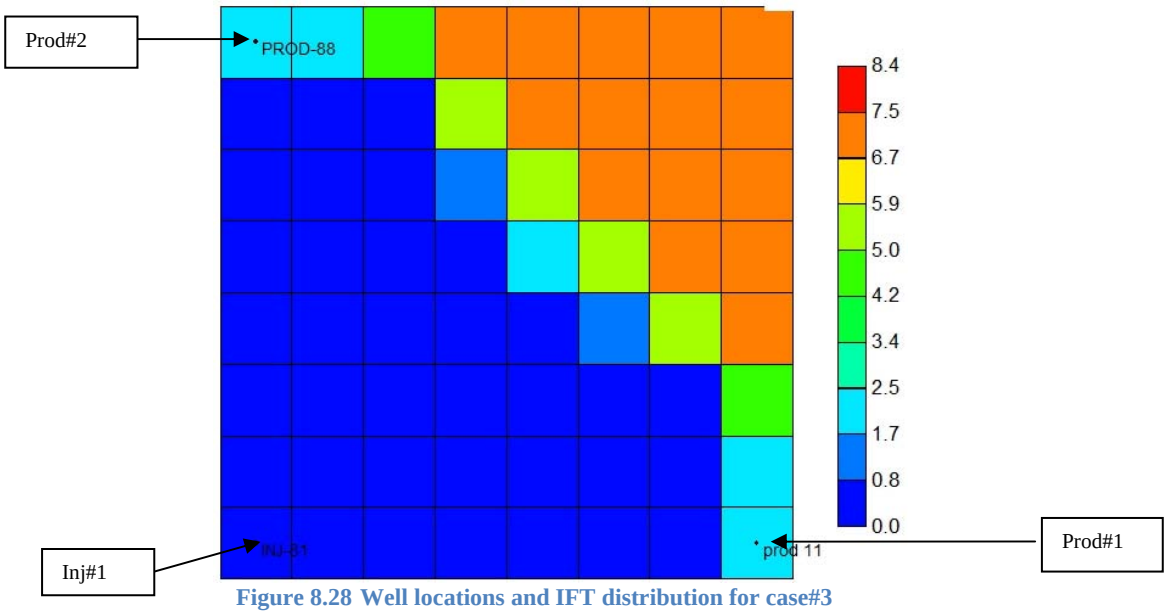


Figure 8.28 Well locations and IFT distribution for case#3

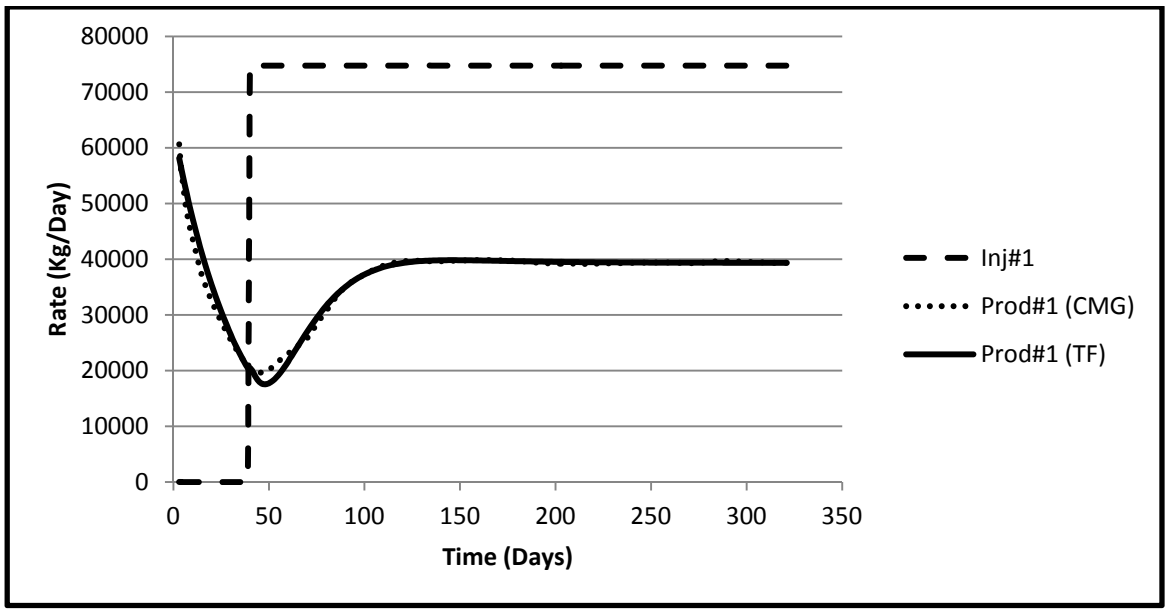


Figure 8.29 Comparison of TF simulation and grid-based simulation in Case 3 for Prod#1

As this case is a miscible flooding, there is higher value for the time constant which demonstrates later observation of influence of injector on the producer compare to the first case which is immiscible flooding. As it can be seen, the jump which was seen in case study 1 (the immiscible flooding) does not happen in miscible flooding, because the gas is solved into the liquid and will not remain in gas phase and compressed behind the front.

8.3.2.4. Case study#4: miscible gas-flooding in a heterogeneous reservoir with one injector and four producers

The properties of this reservoir are shown in table 8.12. Permeability in y-direction and z-direction is equal to 10md and there is heterogeneity in x-dir. Well locations and the permeability distribution in x-direction are shown in figure 8.30. Carbon dioxide is injected via the following input signal:

$$X_2(t) = 47000u(t - 1100)$$

Table 8.12 Fluid and rock properties for case 4

Property	Value
Permeability in x-direction	10 md
Permeability in y-direction	10 md
Permeability in z-direction	25 md
Porosity	0.20
No. of Grids in X,Y,Z Direction	11,11,1
Dimension of Grids in X,Y,Z Direction	100,100,10 m
Rock Compressibility	0.000004 1/psi
Reservoir Depth	6000 ft
Phases	oil, water and gas
Initial Pressure	17000 KPa

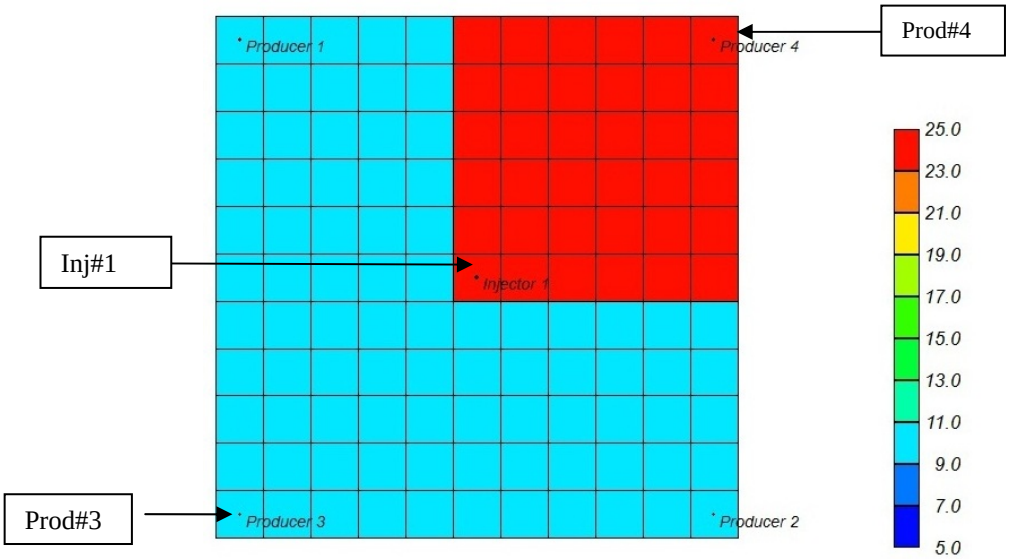


Figure 8.30 Well locations and x-dir permeability for case 4

The connectivity factors between injector and all producers were calculated in this case. The unknown parameters are shown in table 8.13. The results for two wells 3 and 4 are

compared with the grid-based simulation results in figure 8.31. As expected, there is more connectivity between injector with producer 4 because of permeability distribution.

Table 8.13 Parameters of model for case 4

Well	Kp_1/τ_1	τ_1	Kp_2	ξ	τ_2
Prod#1	148452.3	4501.2	0.24	0.77	1502.0
Prod#2	148436.1	4503.9	0.23	0.78	1500.7
Prod#3	129939.9	4522.1	0.2058	0.7868	1477.966
Prod#4	299977.3	4428.35	0.4154	0.7217	1822.021

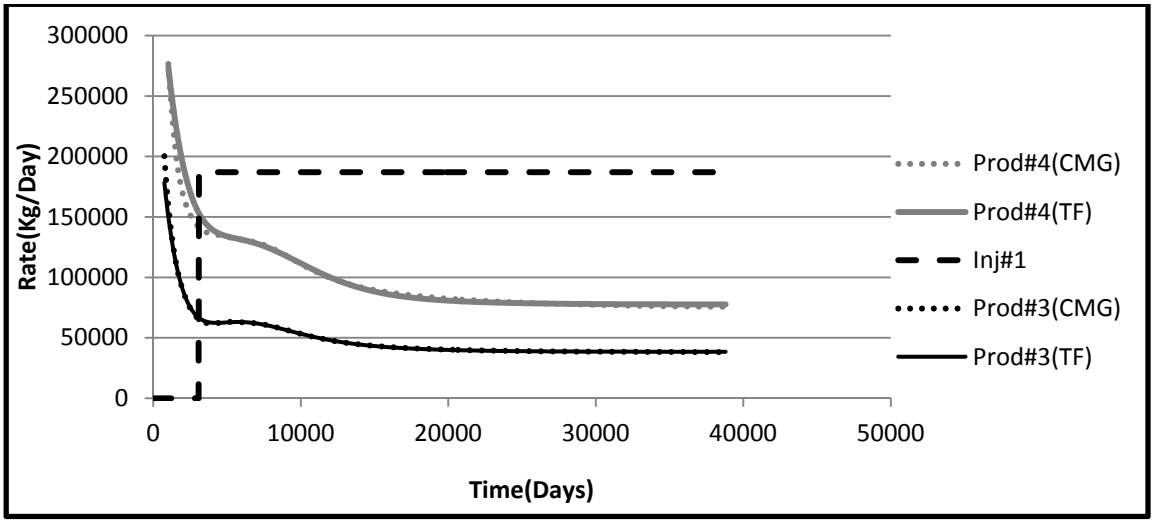


Figure 8.31 Comparison of TF simulator and Grid-based simulation for case#4

8.3.2.5. Case study#5: immiscible gas-flooding in a heterogeneous reservoir with five injectors and four producers

In this case, the performance of fractional flow model is investigated for a 9-spot reservoir. The locations of the wells are shown in figure 8.32. N_2 is flooded into the reservoir. Its rate is similar to case 1. Reservoir properties are similar to case 4. Since the producer wells are symmetrical and the reservoir is homogeneous, thus TF models are the same for all four wells. The unknown parameters are calculated and the results for producer 1 are shown in table 8.14. The results of TF model along with the outcomes of a grid-based simulation is shown in figure 8.33.

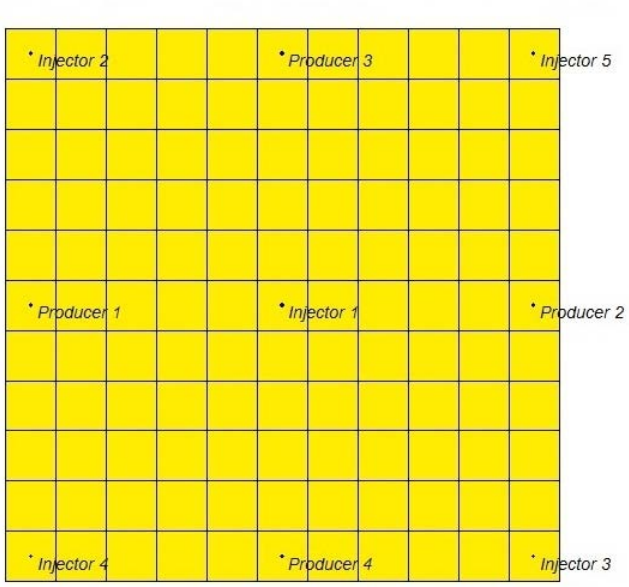


Figure 8.32 well locations for case 5

Table 8.14 Parameters of model for case 5

Well	Kp_1/τ_1	τ_1	Kp_2	ξ	τ_2
Prod#1	79000.330	1400.0708	0.3339	0.99	200.1246

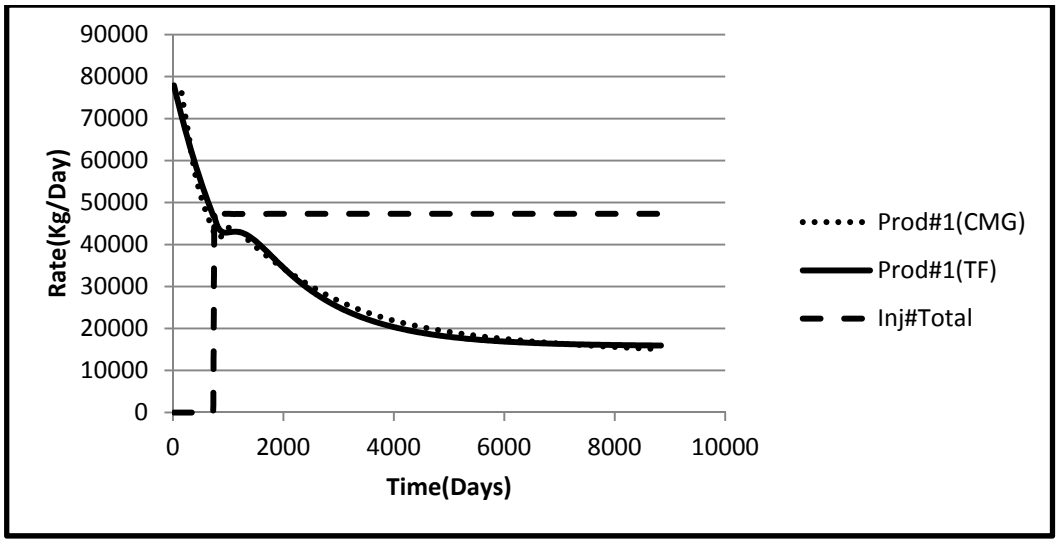


Figure 8.33 Comparison of transfer function simulation and grid-based simulation for case 5

By estimating the unknown parameters of the developed model, it is possible to predict the performance of flow production rate. To forecast the performance of oil production, the fractional flow model is applied for this case and the results after the breakthrough time for producer 1 are shown in Table 8.15. A comparison between oil production results from a

grid-based simulation and the introduced fractional flow model combined with TF model is shown in figure 8.34.

Table 8.15 Fractional flow parameters for case 5

Well	a	b
Prod#1	1.40941033	0.99999972

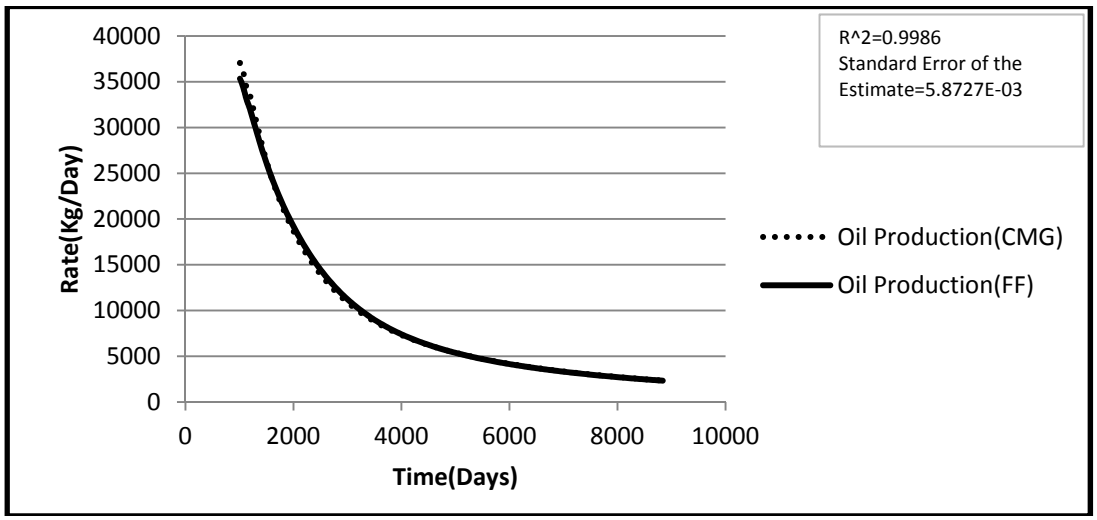


Figure 8.34 Comparison of fractional flow model combined with TF and grid-based simulation for Case 5

It should be mentioned that in the fractional flow (FF) model, it is assumed the breakthrough time is known and this model is useful when the system is in a balance situation of its life. When gas mobility is too large, the oil fraction trend will be interrupted and a two part curve will be created. Each part should be modelled by a separate fractional flow model (FF). Figure 8.35 shows such oil fraction model for a reservoir with a large thickness, while N_2 is flooded in a high rate. N_2 high mobility in comparison with low mobility oil in reservoir leads in a first reduction in oil fraction (first part of the curve) and after that, when oil rates in producer well increase with time, the curve will be interrupted and then second decline occurred. It means that oil fraction (in logarithmic scale) is not a linear function of cumulative fluid produced and its profile has two lines with different slopes. Therefore, using a unique fractional flow model for all after breakthrough time life of reservoir may causes inaccuracy in oil production calculations. This inaccuracy level is investigated in following case.

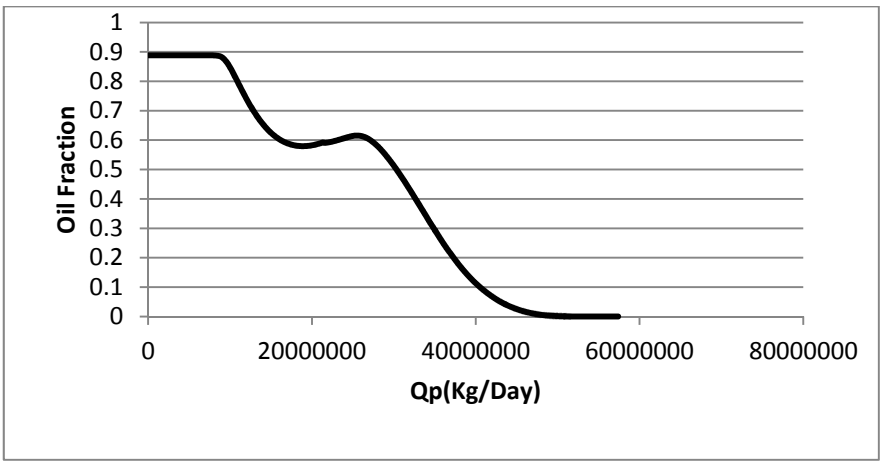


Figure 8.35 Interruption in oil fraction curve in high thickness reservoir

8.3.2.6. Case study#6: immiscible gas flooding in a homogeneous reservoir with two injectors and three producers

In this case study with above situation, only one fractional flow model is applied. The properties of the reservoir are shown in table 8.16. There are two injectors with a total injection rate of N_2 equals to 40000 Kg/Day. Three producer wells produce in a constant production rate equal to 133.33m^3/Day. The location of the wells is shown in figure 8.36. TF simulation was applied for this case and the results for producer 1 are shown in table 8.17. Total production prediction by the TF model in comparison with grid-based simulation results for producer 1 is shown in figure 8.37. Then, the FF model is applied and results are shown in table 8.18. A comparison between this model and a grid-based model for producer is shown in figure 8.38. It is obvious that there is an interruption in oil production curve. Before oil decline starts, the breakthrough occurred and FF model in a little time after the breakthrough has not enough accuracy. It shows that it is better to apply two separate FF model for each section of oil fraction curve.

Table 8.16 Parameters of model for case 6

Well	Kp_1/τ_1	τ_1	Kp_2	ξ	τ_2
Prod#1	79000.330	1400.0708	.3339	0.99	200.1246

Table 8.17 Fluid and rock properties for case 6

Permeability in X-Direction	15 md
Permeability in Y-Direction	15 md
Permeability in Z-Direction	15 md
Porosity	0.20
No. of Grids in X,Y,Z Direction	7,7,1
Dimension of Grids in X,Y,Z Direction	100,100,100 m
Rock Compressibility	0.000004
Reservoir Depth	6000 ft
Phases	Oil, Water and Gas
Initial Pressure	17000 KPa

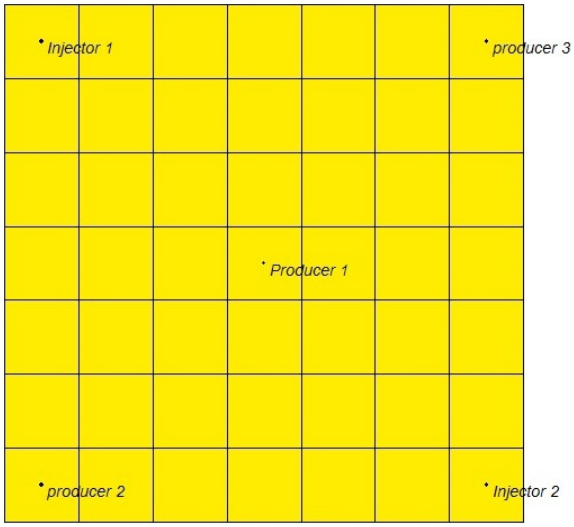


Figure 8.36 Well locations for case 6

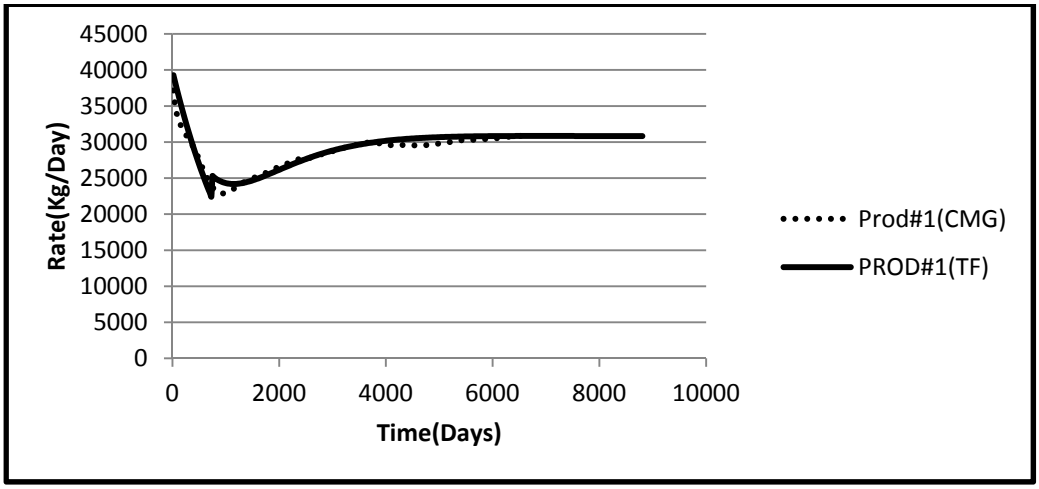


Figure 8.37 Comparison of transfer function simulation and grid-based simulation for case 6

Table 8.18 FF model parameters for case 6

Well	a	b
Prod#1	1.240231043	0.999999868

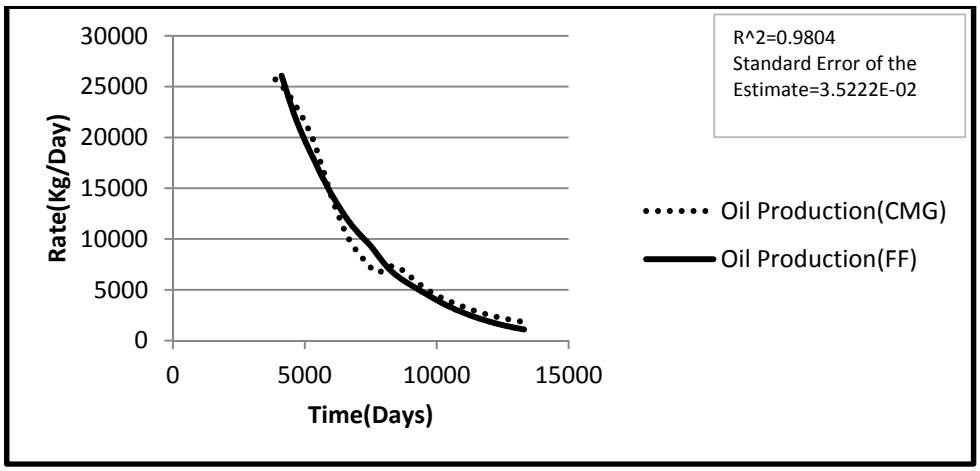


Figure 8.38 Comparison of fractional flow model combined with TF and grid-based simulation for FF case study

To sum up, in the six case studies, it was shown that the proposed method (TF and the FF) is capable of delivering acceptable results in forecasting the performance of oil wells during gas-flooding (immiscible and miscible). Due to the analytical solutions of this method, this approach can be utilised to provide a sensitivity analysis on the effect of gas-rate on ultimate oil recovery and consequently, it can supply a rough estimation about the optimal gas-injection rate quickly. Also, the obtained parameters for each case study give us some information about the reservoir specifications, for instance, the well connectivities allow us to know about the transmissibility.

8.4. Discussion

In this chapter, the development of a method based on Transfer Functions (TF) was presented to predict the performance of water-flooding and gas-flooding in the oil fields. The developed combination of TF can be extended to any reservoirs with the mentioned conditions. The only required data for this method are production and injection rates. By interpreting the history data, it is possible to estimate the matching parameters of the developed combination. After estimating these parameters, the forecasting can be made for various scenarios. This method can be categorised in the fast simulator group, as result of its analytical solution. The developed method should be applied in conjunction with a fractional flow method. For water-flooding problems, Gentil model was applied, as it does

not need much data, and for gas-flooding problems, a method which was developed by regression and the investigation of several case studies was used.

Different cases were employed to validate the derived equations. The results demonstrated a good agreement with those obtained from the common grid-based simulators. In addition, it is figured out that the TF parameters depend on the characteristics and the pattern of different sections of the reservoir and give us useful information, such as well connectivity and pore volume.

It is probably fair to state that the method has some limitations as well. This method should be utilised when enough data are not accessible or a quick overview of injection performance is required. When there is sufficient data or accurate estimation is sought, grid-based simulations would be more appropriate. In order to derive the equations, a series of assumptions were taken for granted, 1- bottomhole pressures are fixed, 2- the production drive is due to primary pressure and injection drive, there is not any aquifer or gas cap drive, and 3- in fractional flow development, the breakthrough time of gas flooding is assumed predictable prior to the implementation of the method. Therefore, reservoir with aquifer or gas cap, fractured reservoirs, or problems with variable bottomhole pressures might be modelled with another arrangement and orders of transfer functions. This method provides rough estimations and also it is in the primary phase of development and it needs further study.

The conclusive remarks of this chapter are as following:

- Transfer function approach is capable of delivering reliable and quick estimation about the future performance of oil wells, and the derived equations is extendable for other oil reservoirs.
- As result of analytical solution of the method and reliability of the outcomes, they can be used for sensitivity analysis on the effect of injection rate on ultimate oil

recovery, therefore, they can provide a rough estimation about the optimal injection rates quickly

- They also provide some key parameters which give us some information about the reservoirs specifications, including, transmissibility and well connectivities, pore volume, and fluid and rock compressibility.
- The only required data are the history of production and injection rates. By carrying out an inexpensive history matching, the unknown parameters are estimated.

Chapter 9

9. Conclusions and recommendations

Obtaining high-quality history matched models (reducing uncertainties in initially constructed reservoir model by inverse modelling of dynamic data) was the core goal of this thesis. Different new methods were introduced to overcome some of the challenges in history matching. This chapter summarises the key achievements of this dissertation and presents some recommendations for the future works.

9.1. Conclusive remarks

1- A software framework was developed in MATLAB which allowed us to carry out history matching automatically.

2- The application of artificial bee colony algorithm was investigated in history matching using a synthetic case which was reparameterised by a pilot point technique. In comparison with three different optimisation algorithms (genetic algorithm, simulated annealing and Levenberg-Marquette), the artificial bee colony algorithm outperformed. Forecasting using the obtained model had only one percent error. The case study was also reparameterised with three other techniques, zonation, spectral decomposition and bicubic spline to create new problems with different landscape shapes. The analyses indicated that the shape of landscape was changed by altering the reparameterisation operator. The four optimisation

algorithms were compared with each other in these problems. The Levenberg-Marquette algorithm was trapped into a local minimum in all, due to its gradient-based search. The genetic algorithm delivered good results, but in none of them, it was the best. The simulated annealing delivered poor results where spectral and pilot point technique were used. The artificial bee colony algorithm was the best method where pilot point and spectral decomposition were used, and in the other two, it delivered satisfactory results. Therefore, it is fair to state that the performance of artificial bee colony algorithm is not significantly influenced by the shape of landscape. The reason of ABC's success may lay on having a balance between search abilities, exploration and exploitation, in contrast to SA (which is more exploitative) and GA (which is more explorative). The artificial bee colony, similar to almost all stochastic optimisers, has two major drawbacks: a- it is computationally intensive, and b- it should be applied in conjunction with a reparameterisation. Thus, it is better to be applied in the problems in which the computational cost is not an issue and the number of variables is not large (or the reparameterisation can be implemented accurately).

3- To reduce the computational costs consistently in stochastic history matching algorithms, in general, it was proposed to including an evolution-control technique to the conventional (uncontrolled) approach of proxy-modelling. Three types of model management, population-based, individual-based and adaptive were implemented and assessed. The adaptive method was customised for the problem. History matching was executed using each of them in an oil reservoir (PUNQ-S3 model), and their outcomes were compared with an uncontrolled approach. The uncontrolled approach could not deliver acceptable results, which could be because of having offline training, an inadequate number of samples or inappropriate proxy fitting. The other three methods which made use of a model management technique

and online learning with the same amount of computation time articulated significantly superior results. The success of these approaches in comparison with the uncontrolled method was because of their emphasis on the approximation of the global optimum point rather than the approximation of the entire landscape shape. Among these three approaches, the adaptive approach performed better, due to having an updatable probability. In order to show the computational cost reduction, history matching was also performed by a regular approach (without proxy-modelling) for the same problem. The same result as the adaptive approach was achieved by the regular approach, but it was needed to call the original fitness function four times more than the adaptive approach. Hence, it is fair to state that the computational cost can be reduced by the controlled fitness approximation reliably. It is worth mentioning that the fitness approximation methods should be used when there are computational limitations and the number of variables is not large (or the reparameterisation can be implemented accurately).

4- A history matching algorithm was developed by combing two different concepts, wavelet image-fusion and evolutionary computation. By applying this algorithm, reparameterisation was unnecessary and history matching could be carried out directly. The outcomes of history matching using this algorithm were compared with the results of six methods, indirect and direct, in a synthetic and also PUNQ-S3. The developed algorithm outperformed. It had an acceptable speed of convergence. The final history matched models were realistic, and overshooting did not occur. Its success is because of its stochastic search and intelligent mating operator which prevent getting stuck in a local minimum and also keep the directional information of high-quality individuals. This algorithm can also enable 1- consideration of discrete variables by multibranched chromosomes, 2- execution on supercomputers, because of the ability of parallel processing in this algorithm

(the individuals in each generation does not have interactions with each other), and 3- construction of a robust hybrid method by combining it with a classical optimiser. The three indirect methods (GA-pilot point, LBFGS-spectral decomposition, LBFGS-zonation) converged to high-quality reparameterised models, but when the models were transformed back to the main domain, a considerable difference was seen between them and the reference model. The differences were mainly in consequence of reparameterisation errors. Overall, the methods of the indirect approach suffer from a number of disadvantages: 1- a high-quality prior model or a set of high-quality prior models is usually required to reparameterise the model properly, 2- overshooting may occur, and 3- unrealistic results may be achieved, due to converting an underdetermined to an overdetermined problem. On the other hand, the three algorithms (GA, ABC and LBFGS) used as a direct calibrator also did not deliver high-quality models, because of optimisation errors. Due to the nonlinearity of the inverse problem, and the high-dimensionality of the system, the optimisation algorithms were not very efficient. Among these three algorithms, LBFGS was performed slightly better, which expressed the drawbacks of the regular stochastic algorithms in high-dimensional history matching problems. None of the six algorithms was as good as the proposed algorithm. Therefore, it is fair to mention that the proposed method is capable of dealing with large history matching problems.

5- It was demonstrated that in those problems where the ratio of covariance matrixes (weighting factors) in the objective function is uncertain, a set of solutions should be provided to deliver an uncertainty analysis. To achieve a set of solutions, the random selection (conventional) and proposed (Pareto) approach were taken and compared with each other using a numerical example and a reservoir model. By the conventional approach, it was required to solve the inverse problem with many

different arrangements of weighting factors (covariance matrixes) at random. This approach was time-consuming, and it was shown that if a large number of arrangements are not picked, there will be high risks in decision making. In the proposed method, the likelihood and prior function were assumed as two different objective functions, and the corresponding Pareto front was approximated using a single run of a multi-objective genetic algorithm. By the designed post optimisation trade-off rules, a number of solutions were taken out from the Pareto front. The outcomes of these approaches expressed that the proposed approach was more efficient in terms of the quality of solutions and also computation. Thus, it is fair to state that when the weighting matrixes are uncertain, it is better to make use of the Pareto approach.

6- A method based on Transfer Functions (TF) was introduced to predict quickly the performance of oil reservoirs during water-flooding and gas-flooding with the minimum data. The developed method was combined to a fractional flow method. By investigating several case studies, a combination of TFs was found which is physically justifiable and also mathematically fitted to the results of grid-based simulator. Different cases were employed to validate the derived equations. The results demonstrated a good agreement with those obtained from the common grid-based simulators. The only required data for this method were production and injection rates. The TF parameters depend on the characteristics and the pattern of different sections of the reservoir and give us useful information, such as well connectivity, pore volume and fluid and rock compressibility. When there is sufficient data or accurate estimation is sought, grid-based simulations would be more appropriate, otherwise, the TF approach can be a suitable candidate.

7- An additional piece of work was done in this thesis in which the optimal infill drilling plane was estimated for a coal seam gas reservoir (semi-synthetic model

constructed based on the Tiffany unit coal bed data in the San Juan basin) by the developed framework in which the objective function was the net present value and the variables was the coordination of infill wells. It was sought to maximise the net present value for a period of ten years by locating the optimal number of wells in the best locations. To find the optimal number of infill wells, the optimisation was carried out several times by increasing sequentially the number of infill wells in each stage. The best scenarios were found for different water treatment expenses. When cost of water treatment was low, infill wells were mostly located in virgin section of the reservoir where reservoir pressure was high. When cost of water treatment was high, infill wells were mostly located on the transition section between virgin and depleted sections of the reservoir to minimise water production. It can be stated that the ratio of the cost of water treatment and disposal to gas price is a key economical parameter (with the assumption of insignificant spatial difference in the initial permeability distribution). Another study was also made by using a multi-objective optimisation in which the objectives were minimising the water treatment costs and maximising the gas production income by finding the optimal scenario for 20 infill wells into the same reservoir. Almost similar observations were seen. Thus, it is fair to state that the developed framework can be utilised suitably for infill drilling optimisation problems as well.

9.2. Future works

Research in the following directions can be extended.

1- In the image-fusion algorithm, the mutation is carried out by changing the value of a random zone of the reservoir (in which the number of gridblocks is predefined) with a random number. However, this operator provides good exploration and avoids premature convergence, in the late generations in which exploitation is more

important, it is not very efficient. An adaptive rule or an unfixed value for the number of the gridblocks may expedite the process.

2- A fixed homogenous value is allocated in the mutation operator of the image-fusion algorithm. Instead of using a homogenous zone, it is possible to use random realisations generated by geostatistics or fractal.

3- The image-fusion is capable of keeping directional information, thus, the author is seeking to apply this algorithm for channel modelling which can be a further step in the use of this algorithm.

4- In the investigation of image-fusion algorithm, only geomodel parameters were adjusted. It is possible to add any other variables to the algorithm by assuming multibranched chromosomes.

5- Comparing the image-fusion algorithm with a number of indirect methods on reservoir models in which the reparameterisation can be implemented accurately can be a further study.

6- The multi-objective algorithm can be combined by the image-fusion algorithm to study regularisation in those problems in which the reparameterisation has considerable error. In order to blend these algorithms, the selection of the image-fusion should be replaced by a rank scheme algorithm.

7- The multi-objective algorithm can be combined with the controlled fitness approximation method to reduce the computational costs.

8- In the approximation of the Pareto front, a multi-objective genetic algorithm with a NSGA-II ranking algorithm was used. This optimiser can be replaced by any other multi-objective optimiser. Thus, the improvement of the optimisation step in this part can be considered as a promising future work.

9- In the applied ABC, the individuals are altered one-dimensionally either by onlookers or employed bees. This one-dimensional search, however brings a

comprehensive search, it causes the low speed of convergence. The search can be multi-dimensional. The best way is to develop an adaptive change for the dimension in which the optimum can be approximated quicker.

10- In this thesis, the main objective in each of the problems was to estimate the most probable model (MAP). In order to have a risk analysis, a set of solutions with their reliability is required. It is suggested to perform one of the uncertainty analysis methods for risk analysis.

11- In the proxy-modelling approaches, artificial neural network was used as the proxy model. The proxy model can be replaced by any other mathematical functions, such as kriging, support vector machine, polynomial and even multi-layered neural network. This substitution may increase the quality of the controlled fitness approximation.

12- A genetic algorithm with a customised crossover was implemented in the proxy-modelling chapter. The optimiser can be replaced by any other methods. Implementation of another optimiser may increase the quality of the controlled fitness approximation.

13- In the transfer function approach, a number of assumptions were made to develop the mathematical formula. In order to remove the restrictions, it is possible to change the set of transfer function.

14- The parameters calculated from the transfer function approach can be used for various studies such as fracture detection.

15- The automatic history matching framework can be utilised for further studies in different directions, for instance, it can be used for simulation of reservoirs that have stress-dependent permeability; a geomechanical simulator should be included into the framework.

Appendix

a.1. MATLAB Coding

In order to execute history matching automatically, a software framework is required. This framework should be capable of running the reservoir simulator (ECLIPSE), reading the outputs of the executed simulation, analysing the output data based on an objective function, and modifying the input data according to a specific rule based on the analyses. This framework is developed into MATLAB programming language, and has two phases. In the first phase, the code developments of an interface between MATLAB and ECLIPSE, the objective function and reparameterisation coding are presented. In the second phase, of this chapter, the codes related to the optimisation methods are presented. In order to explain the process, history matching for PUNQ-S3 is coded is explained.

The developed framework not only can be used for automatic history matching problems, but also it can be utilised for economical analyses and production optimisation problems. In the third section of this appendix, the framework is changed in a way to be utilised for a production optimisation problem (well placement optimisation).

Infill drilling (well placement) in coalbed methane reservoirs has not been studied broadly previously. The developed framework was a good excuse to propose an infill drilling project. The main goal of the project was finding the best locations for the infill wells in order to maximise the net present value. The net present value (NPV) of a well placement scenario in coal bed methane (CBM) reservoirs is sensitive to several terms, such as water treatment price, gas price, reservoir heterogeneities, water saturation map, pressure etc.

Thus, the manual optimisation of well placement in these reservoirs becomes challenging. The author altered the framework to fit in the project's requirements, and also designed a specific optimisation method for the problem. The MATLAB codes and results for a field case study (San Juan basin) are presented in the third of this chapter. Interesting outcomes were seen by analysing the optimal well locations. The results were published into two peer-reviewed journal papers which are attached to the appendix.

a.1.1. Interface development

The development of the interface consists of three different parts:

1- Coupling of MATLAB with ECLIPSE,

2- Objective function coding, and

3- Reparameterisation coding.

Each part is explained individually.

a.1.1.1. Coupling of MATLAB with ECLIPSE

The programming of this section is divided itself into three steps:

i- Providing (modifying) an input data for ECLIPSE by MATLAB,

ii- Executing ECLIPSE from MATLAB, and

iii- Extracting output of ECLIPSE output file into MATLAB.

Each steps of this procedure is described in the following sections.

a.1.1.1.1. ECLIPSE input data generation

The first step is the construction of an input file for ECLIPSE. This input file gets the decision variables (for example, porosity and permeabilities in history matching problems, or well locations in infill drill optimisation), and the rest of input files which are known

like PVT, well location and geomodel should be provided manually and copied into an address (C:\ECLIPSERUNS\Model#1\) before running any jobs from MATLAB[11]. The MATLAB function takes the decision variables as a vector and generates a data file into the given address.

In the case study (PUNQ-S3), there are 7980 variables which are porosity and permeabilities in each gridblock. The other ECLIPSE files for the case study which are fixed can be downloaded from the following URL http://www3.imperial.ac.uk/earthscienceandengineering/research/perm/punq-s3model/onlinedataset. These files (31445697.data and 31443696.geo) should be renamed as PUNQS3.data and PUNQS3.geo and copied into the mentioned address prior to history matching. Also, into the downloaded data file (PUNQS3.data), a keyword ("EXCEL") should be included into the file before the "SCHEDULE" keyword. The MATLAB code for generating the porosity and permeability input file (PUNQS3.prp) is as follows:

```
function PoroPerm(m)
%  All rights reserved. 2010-2013, Mohammad Sayyafzadeh
%
%  Redistribution and use in source and binary forms, with or without
%  modification, are permitted provided that the following conditions are
%  met:
%
%     * Citing the thesis or one of my history matching papers.
%     * For only academic purposes
%----------------------------
Poro=m(1:2660);poro1=Poro(1:532);poro2=Poro(533:1064);
poro3=Poro(1065:1596);poro4=Poro(1597:2128);poro5=Poro(2129:2660);
Permx=m(2661:5320);permx1=Permx(1:532);permx2=Permx(533:1064);
permx3=Permx(1065:1596);permx4=Permx(1597:2128);permx5=Permx(2129:2660);
Permz=m(5321:7980);permz1=Permz(1:532);permz2=Permz(533:1064);
permz3=Permz(1065:1596);permz4=Permz(1597:2128);permz5=Permz(2129:2660);
%----------------------------
Folder=strcat('c:\ECLIPSERUNS\MODEL#1');Folder1=strcat(Folder,'\PUNQS3');
Folder2=strcat(Folder1,'.PRP');fid=fopen(Folder2,'wt');
%-------------------------------------------
%PermX
k={
'PERMX'};
[rows,~]=size(k);
for i=1:rows
    fprintf(fid,'%s\n',k{i,1});
end

% First Layer
for i=1:(length(permx1))/4
    for j=1:4
    fprintf(fid,'%5.3f ',permx1(j+(4*(i-1))));
    end
```

[11] The rest of files do not need the modifications, since they do not include any decision variables. Hence, it does not affect the process of automatic history matching. For instance, PVT data, SCAL data, well data are known in PUNQ-S3 model, therefore, these files are copied into the address. In one of the studies in this thesis, SCAL data was assumed uncertain as well, hence, its corresponding data file was generated by the MATLAB.

```
    fprintf(fid,'\n');
end

% Second Layer
for i=1:(length(permx2))/4
    for j=1:4
    fprintf(fid,'%5.3f ',permx2(j+(4*(i-1))));
    end
    fprintf(fid,'\n');
end

% Third Layer
for i=1:(length(permx3))/4
    for j=1:4
    fprintf(fid,'%5.3f ',permx3(j+(4*(i-1))));
    end
    fprintf(fid,'\n');
end

% Forth Layer
for i=1:(length(permx4))/4
    for j=1:4
    fprintf(fid,'%5.3f ',permx4(j+(4*(i-1))));
    end
    fprintf(fid,'\n');
end

% Fifth Layer

for i=1:(length(permx5))/4
    for j=1:4
    fprintf(fid,'%5.3f ',permx5(j+(4*(i-1))));
    end
    fprintf(fid,'\n');
end
 %---------------------------------
 %PermZ
k3={
'/'

'PERMZ'};
[rows,~]=size(k3);

for i=1:rows
fprintf(fid,'%s\n',k3{i,1});
end

% First Layer
for i=1:(length(permz1))/4
    for j=1:4
    fprintf(fid,'%5.3f ',permz1(j+(4*(i-1))));
    end
    fprintf(fid,'\n');
end

% Second Layer
for i=1:(length(permz2))/4
    for j=1:4
    fprintf(fid,'%5.3f ',permz2(j+(4*(i-1))));
    end
    fprintf(fid,'\n');
end

% Third Layer
for i=1:(length(permz3))/4
    for j=1:4
    fprintf(fid,'%5.3f ',permz3(j+(4*(i-1))));
    end
    fprintf(fid,'\n');
end

% Forth Layer
for i=1:(length(permz4))/4
    for j=1:4
    fprintf(fid,'%5.3f ',permz4(j+(4*(i-1))));
    end
    fprintf(fid,'\n');
```

```
end

% Fifth Layer
for i=1:(length(permz5))/4
    for j=1:4
    fprintf(fid,'%5.3f ',permz5(j+(4*(i-1))));
    end
    fprintf(fid,'\n');
end

 %--------------------------
% Porosity
 k4={
'/'
'PORO'};
[rows,~]=size(k4);
for i=1:rows
    fprintf(fid,'%s\n',k4{i,1});
end

% First Layer
for i=1:(length(poro1))/4
    for j=1:4
    fprintf(fid,'%5.3f ',poro1(j+(4*(i-1))));
    end
    fprintf(fid,'\n');
end

% Second Layer
for i=1:(length(poro2))/4
    for j=1:4
    fprintf(fid,'%5.3f ',poro2(j+(4*(i-1))));
    end
    fprintf(fid,'\n');
end

% Third Layer
for i=1:(length(poro3))/4
    for j=1:4
    fprintf(fid,'%5.3f ',poro3(j+(4*(i-1))));
    end
    fprintf(fid,'\n');
end

% Forth Layer
for i=1:(length(poro4))/4
    for j=1:4
    fprintf(fid,'%5.3f ',poro4(j+(4*(i-1))));
    end
    fprintf(fid,'\n');
end
% Fifth Layer
for i=1:(length(poro5))/4
    for j=1:4
    fprintf(fid,'%5.3f ',poro5(j+(4*(i-1))));
    end
    fprintf(fid,'\n');
end
k5={
'/'
};
[rows,~]=size(k5);
for i=1:rows
    fprintf(fid,'%s\n',k5{i,1});
end
fclose(fid);
end
```

a.1.1.1.2. ECLIPSE execution

In order to run ECLIPSE, the following comments can execute ECLIPSE from MATLAB.

These comments are included into a function. This function does not need any input, and

the output is the execution of the ECLIPSE. In this part, the main file of ECLIPSE (*.data) should be addressed.

```
function ECLIPSE
%  All rights reserved. 2010-2013, Mohammad Sayyafzadeh
%
%  Redistribution and use in source and binary forms, with or without
%  modification, are permitted provided that the following conditions are
%  met:
%
%     * Citing the thesis or one of my history matching papers.
%     * For only academic purposes
    Folder=strcat('eclrun eclipse c:\ECLIPSERUNS\MODEL#1\PUNQS3.data');
    system(Folder);
end
```

If the ECLIPSE license is on a server, you may need to make sure that the ECLIPSE is executed without any license failure. To check the license failures, it is recommended to read the PRT file of ECLIPSE, and if there is a relevant string about license failure, the ECLIPSE should be executed one more time. This procedure needs a “while” keyword in which the condition is “if license failure exists, repeat the execution”. The required code for this checking can be provided upon the request by the author.

a.1.1.1.3. Output file extraction

The third step is the extraction of the data from generated RMS file. We need to read the digits from the generated RMS file which is a text file (consists of characters), and import them into MATLAB. This part is coded with the following function (RMSReader). In this function, it is essential to provide the address of file which is required the data file to be extracted. This function automatically extracts all the numbers, such as WOPR (well oil production rates), WGPR (Well gas production rate) in each time steps. Then, it puts the extracted data into a Matrix (Data). The code for extracting the digits from the output file for PUNQ-S3 model is as follows:

```
function Data = RSMReader
%  All rights reserved. 2010-2013, Mohammad Sayyafzadeh
%
%  Redistribution and use in source and binary forms, with or without
%  modification, are permitted provided that the following conditions are
%  met:
%
%     * Citing the thesis or one of my history matching papers.
%     * For only academic purposes
Folder=strcat('c:\ECLIPSERUNS\Model#1\PUNQS3.rsm');
```

```
fid = fopen(Folder, 'r');
a=textscan(fid,'%10s');
a=a{1,1}';i=1;aa(1,1)=100;u=1;
for j=1:length(a)
    b=str2num(a{1,j});
    if ~length(b)==0
        aa(1,i)=b;
        i=i+1;
        u=1;
    else
        if u==1
            u=2;
            aa(1,i)=(1/0);
            i=i+1;
        end
    end
end
[~, b]=find(aa==inf);
NoRows=(b(2)-b(1)-1)/10;
kk=1;
bsize=size(b);
for j=1:bsize(2)-1
    for i=(j-1)*NoRows+1:j*NoRows
        dd1(kk,10*j-9:10*j)=aa(1,10*i-9+j:10*i+j);
        kk=kk+1;
    end
    kk=1;
end
for i=1:NoRows
    dd2(i,1:2)=aa(1,2*i-1+b(end):2*i+b(end));
end
Data=[dd1 dd2];
fclose('all');
end
```

Using these three functions, it is possible to write, run and read ECLIPSE by MATLAB in each fitness function call. Now, an objective function should be coded to analysis each scenario based on a specific objective function formulation.

a.1.1.2. Objective function coding

This section is coded in MATLAB as a function (Evaluation) which takes decision variables and provides a fitness value for the given variables. All the three functions are used in the code. In this function, the history, covariance matrixes and the corresponding times steps should be provided. In the following code, a Bayesian framework (it has only likelihood term)[12] is used for objective function computations. The code is as follows:

```
function [S] = Evaluation(r)
%  All rights reserved. 2010-2013, Mohammad Sayyafzadeh
%
%  Redistribution and use in source and binary forms, with or without
%  modification, are permitted provided that the following conditions are
%  met:
%
%     * Citing the thesis or one of my history matching papers.
%     * For only academic purposes
```

[12] If the objective function has a prior knowledge term (regularisation term), a prior model must be provided into the MATLAB function, and the corresponding value for the prior knowledge should be calculated.

```
Timesteps=[..];
Times=size(Timesteps);
% History
VectorOBS=[..];
% Covariance Matrix
OBSCOVD=[..];
m=r;
PoroPerm(m);
Eclipse;
Results=ExcelReader;
ND=size(Results);
for i=1:ND(1)
    for j=1:Times(1)
        if Results(i,1)==Timesteps(j)
            Results(i,1)=0;
        end

    end
end
Results(1,:)=[];
ND(1)=ND(1)-1;
i=0;
while i<(ND(1));
    i=i+1;
    if Results(i,1)~=0
        Results(i,:)=[];
        ND(1)=ND(1)-1;
        i=i-1;
    end
end
Results(:,1)=[];Results(:,1)=[];Results(:,9)=[];Results(:,18)=[];Results(:,27)=[];
Data=Results;
Dim=size(Data);
VectorCal=reshape(Data,[Dim(1)*Dim(2),1]);
%---------------------
    Li=(VectorCal(i)-VectorOBS(i))*OBSCOV^-1*(VectorCal(i)-VectorOBS(i))';
S=Li/2;
end
```

a.1.1.3. Reparameterisation coding

The history matching procedure may have a reparameterisation. If it has a reparameterisation method, the programming in the objective function code needs an additional step before generating the input file. This function should take the individual vector (decision variables) (r) and provide the m. Depend on the reparameterisation technique, the procedure will be different. Four reparameterisation techniques are coded in here, 1- Zonation, 2- Pilot point, 3- Spectral decomposition and 4-Bicubic spline. It should be mentioned that if a reparameterisation is used, the input variable of the objective function in MATLAB will be the corresponding elements of the reparameterised model, and these elements will be transformed into m using one of the following functions[13]; as follows:

[13] When full-parameterisation is used, r is equal to m.

a.1.1.3.1. Zonation

The MATLAB code is as follows (it is used for PUNQ-S3 and each layer of the reservoir model for each property is divided into 4 zones[14])

```
function m=Zonation(r)
%  All rights reserved. 2010-2013, Mohammad Sayyafzadeh
%
%  Redistribution and use in source and binary forms, with or without
%  modification, are permitted provided that the following conditions are
%  met:
%
%     * Citing the thesis or one of my history matching papers.
%     * For only academic purposes

var=r;
%----------------------- 1st Layer
m1(1:10,1:14)=var(1);m1(11:19,1:14)=var(2);m1(1:10,15:28)=var(3);m1(11:19,15:28)=var(4);
VectorPoro(1,1:19*28)=reshape(m1,[1 19*28]);
% ----------------------- 2nd Layer
m1(1:10,1:14)=var(5);m1(11:19,1:14)=var(6);m1(1:10,15:28)=var(7);m1(11:19,15:28)=var(8);
VectorPoro(1,1+19*28:2*19*28)=reshape(m1,[1 19*28]);
%-------------- 3rd Layer
m1(1:10,1:14)=var(9);m1(11:19,1:14)=var(10);m1(1:10,15:28)=var(11);m1(11:19,15:28)=var(12);
VectorPoro(1,1+2*19*28:3*19*28)=reshape(m1,[1 19*28]);
%----------- 4th Layer
m1(1:10,1:14)=var(13);m1(11:19,1:14)=var(14);m1(1:10,15:28)=var(15);m1(11:19,15:28)=var(16)
;VectorPoro(1,1+3*19*28:4*19*28)=reshape(m1,[1 19*28]);
%---------- 5th Lyaer
m1(1:10,1:14)=var(17);m1(11:19,1:14)=var(18);m1(1:10,15:28)=var(19);m1(11:19,15:28)=var(20)
;VectorPoro(1,1+4*19*28:5*19*28)=reshape(m1,[1 19*28]);
%-------------------------- PermX
%----------------------- 1st Layer
m1(1:10,1:14)=var(21);m1(11:19,1:14)=var(22);m1(1:10,15:28)=var(23);m1(11:19,15:28)=var(24)
;VectorPoro(1,1+5*19*28:6*19*28)=reshape(m1*30*100,[1 19*28]);
% ----------------------- 2nd Layer
m1(1:10,1:14)=var(25);m1(11:19,1:14)=var(26);m1(1:10,15:28)=var(27);m1(11:19,15:28)=var(28)
;VectorPoro(1,1+6*19*28:7*19*28)=reshape(m1*30*100,[1 19*28]);
%-------------- 3rd Layer
m1(1:10,1:14)=var(29);m1(11:19,1:14)=var(30);m1(1:10,15:28)=var(31);m1(11:19,15:28)=var(32)
;VectorPoro(1,1+7*19*28:8*19*28)=reshape(m1*30*100,[1 19*28]);
%----------- 4th Layer
m1(1:10,1:14)=var(33);m1(11:19,1:14)=var(34);m1(1:10,15:28)=var(35);m1(11:19,15:28)=var(36)
;VectorPoro(1,1+8*19*28:9*19*28)=reshape(m1*30*100,[1 19*28]);
%---------- 5th Layer
m1(1:10,1:14)=var(37);m1(11:19,1:14)=var(38);m1(1:10,15:28)=var(39);m1(11:19,15:28)=var(40)
;VectorPoro(1,1+9*19*28:10*19*28)=reshape(m1*30*100,[1 19*28]);
%----------------PermZ
%----------------------- 1st Layer
m1(1:10,1:14)=var(41);m1(11:19,1:14)=var(42);m1(1:10,15:28)=var(43);m1(11:19,15:28)=var(44)
;VectorPoro(1,1+10*19*28:11*19*28)=reshape(m1*30*100,[1 19*28]);
% ----------------------- 2nd Layer
m1(1:10,1:14)=var(45);m1(11:19,1:14)=var(46);m1(1:10,15:28)=var(47);m1(11:19,15:28)=var(48)
;VectorPoro(1,1+11*19*28:12*19*28)=reshape(m1*30*100,[1 19*28]);
%-------------- 3rd Layer
m1(1:10,1:14)=var(49);m1(11:19,1:14)=var(50);m1(1:10,15:28)=var(51);m1(11:19,15:28)=var(52)
;VectorPoro(1,1+12*19*28:13*19*28)=reshape(m1*30*100,[1 19*28]);
%----------- 4th Layer
m1(1:10,1:14)=var(53);m1(11:19,1:14)=var(54);m1(1:10,15:28)=var(55);m1(11:19,15:28)=var(56)
;VectorPoro(1,1+13*19*28:14*19*28)=reshape(m1*30*100,[1 19*28]);
%---------- 5th Lyaer
m1(1:10,1:14)=var(57);m1(11:19,1:14)=var(58);m1(1:10,15:28)=var(59);m1(11:19,15:28)=var(60)
;VectorPoro(1,1+14*19*28:15*19*28)=reshape(m1*30*100,[1 19*28]);
%-------------------
VectorPoro=abs(VectorPoro);
m=VectorPoro;
end
```

[14] If the number of zones is different, the codes will be required to be modified manually. It is not a general form for it.

a.1.1.3.2. Pilot point

In this approach, we need to make use of geostatistical software to interpolate properties between the gridblocks. SGeMS was coupled with MATLAB before, thus the provided interface by Thomas Mejer Hansen (http://mgstat.sourceforge.net/) is used. In this approach, the well data may be added to the code. For each layer and each property, a SGeMS should be executed. In follows, only coding for a single layer and single property is shown in which the variogram parameters are unknown and it is assumed there are 10 pilot points with variable locations.

```
function [ m ] = PilotPoint(r)
%  All rights reserved. 2010-2013, Mohammad Sayyafzadeh
%
%  Redistribution and use in source and binary forms, with or without
%  modification, are permitted provided that the following conditions are
%  met:
%
%     * Citing the thesis or one of my history matching papers.
%     * For only academic purposes
Var=abs(r);
NoPP=[10];
MODE='Gaussian'
NoX=19;
NoY=28;
%---------- Well Data
X=[..];
Y=[..];
V1=[..];
%-------------
X(size(X)+1:size(X)+NoPP)=Var(6:6+NoPP);
Y(size(Y)+1:size(Y)+NoPP)=Var(6+NoPP+1:6+2*NoPP+1);
V1(size(V1)+1:size(V1)+NoPP)=Var(6+2*NoPP+2:6+3*NoPP+2);
V1=V1';
Z=ones(NoPP+size(X),1);
S=sgems_get_par('sgems3');
S.d_obs=[X Y Z V1];
S.dim.nx=NoX;
S.dim.ny=NoY;
S.dim.nz=1;
S.dim.x0=1;
S.dim.y0=1;
S.dim.z0=1;
MediumVar=Var(3);
Angelx=Var(4);
Angely=Var(5);
Contrib=Var(2);
S.XML.parameters.Nb_Realizations.value=40;
S.XML.parameters.Seed.value=3412;
S.XML.parameters.Max_Conditioning_Data.value=20;
S.XML.parameters.Search_Ellipsoid.value=[10 10 10 0 0 0];
S.XML.parameters.Variogram.nugget=Var(1);
S.XML.parameters.Variogram.structure_1.contribution=Contrib;
S.XML.parameters.Variogram.structure_1.type=MODE;
S.XML.parameters.Variogram.structure_1.ranges.max=1.5*MediumVar;
S.XML.parameters.Variogram.structure_1.ranges.medium=MediumVar;
S.XML.parameters.Variogram.structure_1.ranges.min=0.25*MediumVar;
S.XML.parameters.Variogram.structure_1.angles.x=Angelx;
S.XML.parameters.Variogram.structure_1.angles.y=Angely;
S.XML.parameters.Variogram.structure_1.angles.z=0;
S=sgems_grid(S);
[m1,v1]=etype(S.D);
VectorPoro(1,:)=reshape(m1,[1 532]);
m=VectorPoro;
```

```
end
```

a.1.1.3.3. Spectral decomposition

In this approach, a series of computations should be done prior to a reparameterisation. The W_P and D matrix should be calculated before a reparameterisation. In order to compute these two matrixes, a prior covariance matrix is required (CB). After calculating these two terms, the matrixes are required to be saved or copied into Spectral Decomposition function. Then, the computed elements are utilised for the reparameterisation.

```
CB=[..];
NoElements=[..];
DL=diag(diag(CB)).^0.5;
[UP,GamaP]=eigs(CB,NoElements);
[UP,GamaP] = cdf2rdf(UP,GamaP);
D=DL;
WP=UP;
```

These elements can be used for reparameterisation.

```
function [ m ] = SpectralD(r)
%  All rights reserved. 2010-2013, Mohammad Sayyafzadeh
%
%  Redistribution and use in source and binary forms, with or without
%  modification, are permitted provided that the following conditions are
%  met:
%
%     * Citing the thesis or one of my history matching papers.
%     * For only academic purposes
D=[..];
WP=[..];
m=D*WP*r;
end
```

a.1.1.3.4. Bicubic spline

This reparameterisation is similar to pilot point technique, but spline interpolation is used. In the example, it is assumed the locations of matching gridblocks are known. For this method, it is required to provide the well data into the function. This function gets the decision variables and provides the spatial properties (m). The coding for only one layer and one property of PUNQS3 is shown.

```
function [ m ] = Spline( r )
%  All rights reserved. 2010-2013, Mohammad Sayyafzadeh
%
%  Redistribution and use in source and binary forms, with or without
%  modification, are permitted provided that the following conditions are
%  met:
%
%     * Citing the thesis or one of my history matching papers.
%     * For only academic purposes
```

```
var=r;
%-------------Well data
x=[..];
y=[..];
z=[..];
%-----------------------
%----location of gridblocks
x(7:11)=[..];
y(7:11)=[..];
ti1 = 1:1:19;
ti2 = 1:1:28;
[xi,yi] = meshgrid(ti1,ti2);
%----------------------- 1st Layer Porosity
z(7:11)=var(1:5)/100;
m1 = griddata(x,y,z,xi,yi,'cubic');
VectorPoro(1,1:19*28)=reshape(m1',[1 19*28]);
m=VectorPoro
end
```

Up to this point, a MATLAB function (Evaluation) is formulated which gets the decision variables (m or r) and provides the fitness value based on a formulated objective function, and it may have a reparameterisation operator. Now, a code is required to carry a rule for updating the decision variables, according to the calculated fitness values to find the best history matched model.

a.1.2. Optimisation development

The updates can be conducted by a GA, Steepest Descent, Simulated Annealing, BFGS, ABC or any other optimisation method. In this study, a number of these algorithms are utilised, hence, the corresponding codes are developed or used. ABC was coded in MATLAB by Karaboga and can be found in the following URL: http://mf.erciyes.edu.tr/abc/. For Simulated Annealing, GA, multi-objective GA and BGFS, MATLAB toolboxes typically are used. In this study, we have developed some optimisation codes which are explained in the following sections.

a.1.2.1. GA with a customised crossover

The coding consists of four main parts, 1- GA body, 2- initial population generator, 3- crossover and mutation function, and 4- tournament function.

a.1.2.1.1. GA body

In this part, the variables, objective function, recombination and evaluations are being done. The coding is as follows:

```
% Required Information
%  All rights reserved. 2010-2013, Mohammad Sayyafzadeh
%
%  Redistribution and use in source and binary forms, with or without
%  modification, are permitted provided that the following conditions are
%  met:
%
%     * Citing the thesis or one of my history matching papers.
%     * For only academic purposes
NoVar=[..];
NoPop=[..];
pCross=[..];
pTour=[..];
pMut=[..];
NoElit=[..];
NoGen=[..];
UB=[..];
LB=[..];
Range=UB-LB;
objfun='[..]'; %cost function to be optimized (@Evaluation)
objective=str2func(objfun);
%---------------
rand('seed',1);
% Generating intial population
Pop=popgen(NoPop,NoVar,LB,Range);inpop=Pop;
%---------------
IterationNo=0;
%------------
for i=1:NoPop
    Fitness(i)= objective(Pop(i,:));
end
%-------------------
% %----------------
mine=min(Fitness)
while (IterationNo<NoGen)
    % Mutation and Crossover
    IterationNo=IterationNo+1;
    popmut=OurCross(Pop,Fitness,pTour,pCross,pMut,LB,Range,IterationNo,NoGen);
    %-----------------------
        for i=1:NoPop
            Fitness2(i)= objective(popmut(i,:));
        end
        % %-------------------
    [ke2 ki2]= sort(Fitness2);
    %Elite
        newpop((1:NoPop),:)=popmut((ki2(1:NoPop)),:);
        [ke1 ki1]=sort(Fitness);
        popelit=Pop(ki1(1:NoElit),:);
        FitnessElit=Fitness(ki1(1:NoElit));
        Pop=newpop((1:(NoPop-NoElit)),:);
        Pop((NoPop-NoElit+1):NoPop,:)=popelit;
        Fitness(1:NoPop-NoElit)=Fitness2(ki2(1:(NoPop-NoElit)));
        Fitness((NoPop-NoElit+1):NoPop)=FitnessElit;
        %---------------------
    SetAns(IterationNo)=min(Fitness);
    FitnessGlobal=min(Fitness);
    [r Ind]=min(Fitness);
    Global=Pop(Ind,:);
    GlobalSet(IterationNo,:)=Global;
    semilogy(SetAns);
end
```

This code needs a number of functions:

a.1.2.1.2. Initial population generator

popgen is the function which creates the initial population randomly over the given bounds;

```
function [ Pop ] = popgen( NoPop,NoVar,LB,Range )
%POPGEN Summary of this function goes here
%   Detailed explanation goes here
Pop=LB+Range*rand([NoPop NoVar]);
end
```

a.1.2.1.3. Crossover and mutation function

In this section, the customised crossover and mutation function is coded. In this function, the current population and the corresponding fitness values are given. Its output is a new population. In this function, we need to provide the mutation and crossover probabilities.

```
function popmut=OurCross(Pop,Fitness,pTour,pCross,pMut,LB,Range,ItterationNo,NoGen)
%  All rights reserved. 2010-2013, Mohammad Sayyafzadeh
%
%  Redistribution and use in source and binary forms, with or without
%  modification, are permitted provided that the following conditions are
%  met:
%
%     * Citing the thesis or one of my history matching papers.
%     * For only academic purposes
aa=size(Pop);
CM=randperm(aa(1));
for j=1:ceil(pCross*aa(1)/5)
    % Tournoment
    NoSelectedPair=2;
    [ SelectedPop FitnessSelected ] = Tournoment( Fitness,Pop,pTour,NoSelectedPair );
    %-----------------
    fFitness=zeros([1 (NoSelectedPair*2)]);
    for i=1:(NoSelectedPair*2)
        if (FitnessSelected(i)>=0)
            fFitness(i)=1./(FitnessSelected(i)+1);
        else
            fFitness(i)=1+abs(FitnessSelected(i));
        end
    end
    SumFitnessSelected=sum(fFitness);
    for k=1:NoSelectedPair*2
        w(k)=fFitness(k)/SumFitnessSelected;
    end

    popcross(5*j-4,:)=zeros([1 aa(2)]);
    for k=1:NoSelectedPair*2
        popcross(5*j-4,:)=popcross(5*j-4,:)+w(k)*SelectedPop(k,:);
    end

    noise=(LB+(Range*5*exp(-(ItterationNo+1)/300)))*rand([4 aa(2)]);

    for k=1:1
        UUU=randperm(aa(2));
        popcross(5*j-4+k,:)=popcross(5*j-4,:);
        popcross(5*j-4+k,UUU)=noise(k,UUU)+popcross(5*j-4,UUU);
    end
    %-----------------
    M=SelectedPop(3,:);
    F=SelectedPop(4,:);
    pCrossChor=0.5;
    popcross(5*j-2,:)=F;
```

```
    pp2=ceil(rand([1 ceil(pCrossChor*aa(2))])*aa(2));
    popcross(5*j-2,pp2)=M(pp2);

    noise=(LB+(Range*5*exp(-(ItterationNo+1)/300)))*rand([4 aa(2)]);

    for k=1:1
        UUU=randperm(aa(2));
        popcross(5*j-2+k,:)=popcross(5*j-2,:);
        popcross(5*j-2+k,UUU)=noise(k,UUU)+popcross(5*j-2,UUU);
    end


    M=SelectedPop(1,:);
    F=SelectedPop(3,:);
    score1=FitnessSelected(1);
    score2=FitnessSelected(3);
    ratio=1.2;
    if(score1 < score2) % parent1 is the better of the pair
        popcross(5*j,:) = F + ratio .* (M - F);
    else % parent2 is the better one
        popcross(5*j,:) = M + ratio .* (F - M);
    end
end
%----------------
%Mutation
popmut=Pop;
VV=randperm(aa(1));
OO=size(popcross);
popmut(VV(1:OO(1)),:)=popcross;
UU=randperm(aa(1));
for i=1:ceil(aa(1))
    RandMu=rand;
    if RandMu<pMut
        muGen=ceil(aa(2)*rand([1 1]));
        popmut(UU(i),muGen)=LB+Range*rand([1 1]);
    end
end
%-----------------
UB=LB+Range;
for i=1:aa(1)
    for j=1:aa(2)
        if popmut(i,j)<LB
            popmut(i,j)=LB;
        end
        if popmut(i,j)>UB
            popmut(i,j)=UB;
        end
    end
end
end
```

a.1.2.1.4. Tournament function

A tournament operator is used for the selection. In this function, a tournament size is selected and in the tournament size, a pair of chromosomes is selected. The corresponding coding is as follows:

```
function [ SelectedPop FitnessSelected ] = Tournoment( Fitness,Pop,pTour,NoSelectedPair )
%TOURNOMENT Summary of this function goes here
%   Detailed explanation goes here
aa=size(Pop);
for i=1:NoSelectedPair
    a=ceil(aa(1)*(rand(pTour,1)));
    b=ceil(aa(1)*(rand(pTour,1)));
    [~, e1]=min(Fitness(a));
    [~, e2]=min(Fitness(b));
    while a(e1)==b(e2)
        a=ceil(aa(1)*(rand(pTour,1)));
        b=ceil(aa(1)*(rand(pTour,1)));
```

```
        [~, e1]=min(Fitness(a));
        [~, e2]=min(Fitness(b));
    end
    SelectedPop(2*i-1,:)=Pop(a(e1),:);
    SelectedPop(2*i,:)=Pop(b(e2),:);
    FitnessSelected(2*i-1)=Fitness(a(e1));
    FitnessSelected(2*i)=Fitness(b(e2));

end
end
```

In order to carry out history matching or optimisation using this algorithm, it is required to create the aforementioned MATLAB codes and copy them into a same directory.

a.1.2.2. EA with the image-fusion technique

Another optimisation algorithm developed in this study is an evolutionary algorithm in which crossover operator is the image-fusion technique and has a specific mutation. The body is similar to GA and the only difference is that the population is a 4D matrix, and the chromosomes are 3D. The body code is as follows:

```
% Required Information
%  All rights reserved. 2010-2013, Mohammad Sayyafzadeh
%
%  Redistribution and use in source and binary forms, with or without
%  modification, are permitted provided that the following conditions are
%  met:
%
%     * Citing the thesis or one of my history matching papers.
%     * For only academic purposes
clear
clc
NoProperties=3;
NoLayers=5;
NoX=19;
NoY=28;
Dim=NoProperties*NoLayers;
% based on Gibbs formula
NoPop=15;pCross=0.9;pTour=4;pMut=0.25;NoElit=1;NoGen=1400;
UB1=0.45;LB1=0.001;LB2=0.001;
UB2=1200;LB3=0.001;UB3=1200;
objfun='LikePOverDet'; %cost function to be optimized
objective=str2func(objfun);
%---------------
rand('seed',2312)
% Generating intial population
Pop=popgen(NoPop);
%---------------
ItterationNo=0;
%------------
for i=1:NoPop
    Individual(:,:,:)=Pop(i,:,:,:);
    Fitness(i)=objective(Individual);
end
FitnessGlobal=min(Fitness);
%-------------------
% %----------------
mine=min(Fitness)
while (ItterationNo<NoGen)
    % Mutation and Crossover
    ItterationNo=ItterationNo+1;
    popmut=IMFUCrossOver(Pop,Fitness,pTour,pCross,pMut,FitnessGlobal);
    %-----------------------
```

```
    for i=1:NoPop
        Individual(:,:,:)=popmut(i,:,:,:);
        Fitness2(i)=objective(Individual);
    end
    % %-------------------
    [ke2 ki2]= sort(Fitness2);
    %Elite
    newpop((1:NoPop),:,:,:)=popmut((ki2(1:NoPop)),:,:,:);
    [ke1 ki1]=sort(Fitness);
    popelit=Pop(ki1(1:NoElit),:,:,:);
    FitnessElit=Fitness(ki1(1:NoElit));
    Pop=newpop((1:(NoPop-NoElit)),:,:,:);
    Pop((NoPop-NoElit+1):NoPop,:,:,:)=popelit;
    %----------
    Fitness(1:NoPop-NoElit)=Fitness2(ki2(1:(NoPop-NoElit)));
    Fitness((NoPop-NoElit+1):NoPop)=FitnessElit;
    %---------------------
    SetAns(IterationNo)=min(Fitness);
    SetAns(end)
    FitnessGlobal=min(Fitness);
    [r Ind]=min(Fitness);
    Global=Pop(Ind,:,:,:);
    pause(0.00001)
    subplot(4,3,6)
    semilogy(SetAns);title(sprintf('Fitness'));
    x=[1:1:19];
    y=[1:1:28];
    subplot(4,3,1);m1(:,:)=Global(1,1,:,:);imagesc(x,y,m1',[0 0.3]);
    subplot(4,3,2);m2(:,:)=Global(1,2,:,:);imagesc(x,y,m2',[0 0.3]);
    subplot(4,3,3);m3(:,:)=Global(1,3,:,:);imagesc(x,y,m3',[0 0.30]);
    subplot(4,3,4);m4(:,:)=Global(1,4,:,:);imagesc(x,y,m4',[0 0.3]);
    subplot(4,3,5);m5(:,:)=Global(1,5,:,:);imagesc(x,y,m5',[0 0.3]);

   %------------------------
    [ Dis ] = DistanceC( Pop );
    Distance(IterationNo)=Dis;
    subplot(4,3,12);
    plot(Distance);
    pause(1)
end
```

The following three functions are used in the above algorithm.

a.1.2.2.1. Initial population generator

The initialisation is heuristic and to generate the initial population, geostatistical correlation is used. The code is as follows:

```
function [ Pop ] = popgen( NoPop )
%  All rights reserved. 2010-2013, Mohammad Sayyafzadeh
%
%  Redistribution and use in source and binary forms, with or without
%  modification, are permitted provided that the following conditions are
%  met:
%
%     * Citing the thesis or one of my history matching papers.
%     * For only academic purposes

%Perm X
P1L=[..];P2L=[..];P3L=[..];P4L=[..];P5L=[..];
Poro1L=P1L;Poro2L=P2L;Poro3L=P3L;Poro4L=P4L;Poro5L=P5L;
for i=1:6
    Poro1L(i,4)=P1L(i,4)/0.278;
end
for i=1:6
    Poro2L(i,4)=P2L(i,4)/0.109;
end

for i=1:6
    Poro3L(i,4)=P3L(i,4)/0.243;
```

```
end

for i=1:6
    Poro4L(i,4)=P4L(i,4)/0.2;
end

for i=1:6
    Poro5L(i,4)=P5L(i,4)/0.275;
end
for i=1:NoPop
    % 1st Layer Poro
    NoX=19;
    NoY=28;
    X=Poro1L(:,1);
    Y=Poro1L(:,2);
    Z=ones(6,1);
    V1=Poro1L(:,4);
    S=sgems_get_par('sgems1');
    S.d_obs=[X Y Z V1];
    S.dim.nx=NoX;
    S.dim.ny=NoY;
    S.dim.nz=1;
    S.dim.x0=1;
    S.dim.y0=1;
    S.dim.z0=1;
    R=200+rand*300;
    S.XML.parameters.Nb_Realizations.value=50;
    S.XML.parameters.Seed.value=1;
    S.XML.parameters.Max_Conditioning_Data.value=3;
    S.XML.parameters.Search_Ellipsoid.value=[10 10 10 0 0 0];
    S.XML.parameters.Variogram.nugget=0.5*rand;
    S.XML.parameters.Variogram.structure_1.contribution=rand*500;
    S.XML.parameters.Variogram.structure_1.type='Gaussian';
    S.XML.parameters.Variogram.structure_1.ranges.max=1.5*R;
    S.XML.parameters.Variogram.structure_1.ranges.medium=R;
    S.XML.parameters.Variogram.structure_1.ranges.min=0.25*R;
    S.XML.parameters.Variogram.structure_1.angles.x=180*rand;
    S.XML.parameters.Variogram.structure_1.angles.y=180*rand;
    S.XML.parameters.Variogram.structure_1.angles.z=0;
    S=sgems_grid(S);
    [m1,v1]=etype(S.D);
    m1=m1*0.278;
    % ----------------------- 2nd Layer
    NoX=19;
    NoY=28;
    X=Poro2L(:,1);
    Y=Poro2L(:,2);
    Z=ones(6,1);
    V1=Poro2L(:,4);
    S=sgems_get_par('sgems1');
    S.d_obs=[X Y Z V1];
    S.dim.nx=NoX;
    S.dim.ny=NoY;
    S.dim.nz=1;
    S.dim.x0=1;
    S.dim.y0=1;
    S.dim.z0=1;
    R=200+rand*300;
    S.XML.parameters.Nb_Realizations.value=50;
    S.XML.parameters.Seed.value=1;
    S.XML.parameters.Max_Conditioning_Data.value=3;
    S.XML.parameters.Search_Ellipsoid.value=[10 10 10 0 0 0];
    S.XML.parameters.Variogram.nugget=0.5*rand;
    S.XML.parameters.Variogram.structure_1.contribution=rand*500;
    S.XML.parameters.Variogram.structure_1.type='Gaussian';
    S.XML.parameters.Variogram.structure_1.ranges.max=1.5*R;
    S.XML.parameters.Variogram.structure_1.ranges.medium=R;
    S.XML.parameters.Variogram.structure_1.ranges.min=0.25*R;
    S.XML.parameters.Variogram.structure_1.angles.x=180*rand;
    S.XML.parameters.Variogram.structure_1.angles.y=180*rand;
    S.XML.parameters.Variogram.structure_1.angles.z=0;
    S=sgems_grid(S);
    [m2,v1]=etype(S.D);
    m2=m2*0.109;
    %----------------3rd Layer Poro
    NoX=19;
    NoY=28;
    X=Poro3L(:,1);
    Y=Poro3L(:,2);
```

```
Z=ones(6,1);
V1=Poro3L(:,4);
S=sgems_get_par('sgems1');
S.d_obs=[X Y Z V1];
S.dim.nx=NoX;
S.dim.ny=NoY;
S.dim.nz=1;
S.dim.x0=1;
S.dim.y0=1;
S.dim.z0=1;
R=200+rand*300;
S.XML.parameters.Nb_Realizations.value=50;
S.XML.parameters.Seed.value=1;
S.XML.parameters.Max_Conditioning_Data.value=3;
S.XML.parameters.Search_Ellipsoid.value=[10 10 10 0 0 0];
S.XML.parameters.Variogram.nugget=0.5*rand;
S.XML.parameters.Variogram.structure_1.contribution=rand*500;
S.XML.parameters.Variogram.structure_1.type='Gaussian';
S.XML.parameters.Variogram.structure_1.ranges.max=1.5*R;
S.XML.parameters.Variogram.structure_1.ranges.medium=R;
S.XML.parameters.Variogram.structure_1.ranges.min=0.25*R;
S.XML.parameters.Variogram.structure_1.angles.x=180*rand;
S.XML.parameters.Variogram.structure_1.angles.y=180*rand;
S.XML.parameters.Variogram.structure_1.angles.z=0;
S=sgems_grid(S);
[m3,v1]=etype(S.D);
m3=m3*0.242;
%--------- 4th Layer
NoX=19;
NoY=28;
X=Poro4L(:,1);
Y=Poro4L(:,2);
Z=ones(6,1);
V1=Poro4L(:,4);
S=sgems_get_par('sgems1');
S.d_obs=[X Y Z V1];
S.dim.nx=NoX;
S.dim.ny=NoY;
S.dim.nz=1;
S.dim.x0=1;
S.dim.y0=1;
S.dim.z0=1;
R=200+rand*300;
S.XML.parameters.Nb_Realizations.value=50;
S.XML.parameters.Seed.value=1;
S.XML.parameters.Max_Conditioning_Data.value=3;
S.XML.parameters.Search_Ellipsoid.value=[10 10 10 0 0 0];
S.XML.parameters.Variogram.nugget=0.5*rand;
S.XML.parameters.Variogram.structure_1.contribution=rand*500;
S.XML.parameters.Variogram.structure_1.type='Gaussian';
S.XML.parameters.Variogram.structure_1.ranges.max=1.5*R;
S.XML.parameters.Variogram.structure_1.ranges.medium=R;
S.XML.parameters.Variogram.structure_1.ranges.min=0.25*R;
S.XML.parameters.Variogram.structure_1.angles.x=180*rand;
S.XML.parameters.Variogram.structure_1.angles.y=180*rand;
S.XML.parameters.Variogram.structure_1.angles.z=0;
S=sgems_grid(S);
[m4,v1]=etype(S.D);
m4=m4*0.2;
%------------------- 5th Layer Poro
NoX=19;
NoY=28;
X=Poro5L(:,1);
Y=Poro5L(:,2);
Z=ones(6,1);
V1=Poro5L(:,4);
S=sgems_get_par('sgems1');
S.d_obs=[X Y Z V1];
S.dim.nx=NoX;
S.dim.ny=NoY;
S.dim.nz=1;
S.dim.x0=1;
S.dim.y0=1;
S.dim.z0=1;
R=200+rand*300;
S.XML.parameters.Nb_Realizations.value=50;
S.XML.parameters.Seed.value=1;
S.XML.parameters.Max_Conditioning_Data.value=3;
S.XML.parameters.Search_Ellipsoid.value=[10 10 10 0 0 0];
S.XML.parameters.Variogram.nugget=0.5*rand;
```

```
    S.XML.parameters.Variogram.structure_1.contribution=rand*500;
    S.XML.parameters.Variogram.structure_1.type='Gaussian';
    S.XML.parameters.Variogram.structure_1.ranges.max=1.5*R;
    S.XML.parameters.Variogram.structure_1.ranges.medium=R;
    S.XML.parameters.Variogram.structure_1.ranges.min=0.25*R;
    S.XML.parameters.Variogram.structure_1.angles.x=180*rand;
    S.XML.parameters.Variogram.structure_1.angles.y=180*rand;
    S.XML.parameters.Variogram.structure_1.angles.z=0;
    S=sgems_grid(S);
    [m5,v1]=etype(S.D);
    m5=m5*0.275;
    Pop(i,1,:,:)=m1;
    Pop(i,2,:,:)=m2;
    Pop(i,3,:,:)=m3;
    Pop(i,4,:,:)=m4;
    Pop(i,5,:,:)=m5;
 end
```

a.1.2.2.2. Crossover and mutation function

As mentioned above, the crossover operator is the wavelet transform image-fusion technique. The image-fusion is performed by Wavelet Toolbox of MATLAB. The mutation was explained in the chapter 5; this function gets the current population and provides the new population, the code for the mutation is as follows:

```
function popmut=IMFUCrossOver(Pop,Fitness,pTour,pCross,pMut,FitGlobal)
aa=size(Pop);
CM=randperm(aa(1));
for j=1:ceil(pCross*aa(1))
    % Tournoment
    NoSelectedPair=1;
    [ SelectedPop FitnessSelected ] = Tournoment( Fitness,Pop,pTour,NoSelectedPair );
    %-----------------
    F(:,:,:)=SelectedPop(1,:,:,:);
    M(:,:,:)=SelectedPop(2,:,:,:);
    Child(j,:,:,:)=M;
    for i=1:aa(2)
        R=rand;
        if R<0.1
            A='min';
        else
            A='rand';
        end
        FP(:,:)=F(i,:,:);
        MP(:,:)=M(i,:,:);
        Child(j,i,:,:) = IFU( FP,MP,'sym4',5,'rand',A );
    end
end
%----------------------
%Mutation
popmut=Pop;
VV=randperm(aa(1));
OO=size(Child);
popmut(VV(1:OO(1)),:,:,:)=Child;
for i=1:ceil(aa(1))
    RandMu=rand;
    if RandMu<pMut
        A1=3;
        IA=ceil(rand*(19-A1));
        B1=3;
        IB=ceil(rand*(28-B1));
        j=ceil(rand*5);
        if j==1
            Mean=0.1589;
            STD=0.0935;
            temp1=ones([A1 B1])*(random('normal',Mean,STD,[1 1]));
        end
```

```
        if j==2
            Mean=0.0736;
            STD=0.0367;
            temp1=ones([A1 B1])*(random('normal',Mean,STD,[1 1]));
        end
        if j==3
            Mean=0.1814;
            STD=0.0789;
            temp1=ones([A1 B1])*(random('normal',Mean,STD,[1 1]));
        end
        if j==4
            Mean=0.1080;
            STD=0.0531;
            temp1=ones([A1 B1])*(random('normal',Mean,STD,[1 1]));
        end
        if j==5
            Mean=0.1902;
            STD=0.0865;
            temp1=ones([A1 B1])*(random('normal',Mean,STD,[1 1]));
        end
        popmut(i,j,IA:IA+A1-1,IB:IB+B1-1)=temp1;
    end
end
for i=1:ceil(aa(1))
    RandMu=rand;
    if RandMu<pMut
        A1=2;
        IA=ceil(rand*(19-A1));
        B1=5;
        IB=ceil(rand*(28-B1));
        j=ceil(rand*5);
        if j==1
            Mean=0.1589;
            STD=0.0935;
            temp1=ones([A1 B1])*(random('normal',Mean,STD,[1 1]));
        end
        if j==2
            Mean=0.0736;
            STD=0.0367;
            temp1=ones([A1 B1])*(random('normal',Mean,STD,[1 1]));
        end
        if j==3
            Mean=0.1814;
            STD=0.0789;
            temp1=ones([A1 B1])*(random('normal',Mean,STD,[1 1]));
        end
        if j==4
            Mean=0.1080;
            STD=0.0531;
            temp1=ones([A1 B1])*(random('normal',Mean,STD,[1 1]));
        end
        if j==5
            Mean=0.1902;
            STD=0.0865;
            temp1=ones([A1 B1])*(random('normal',Mean,STD,[1 1]));
        end
        popmut(i,j,IA:IA+A1-1,IB:IB+B1-1)=temp1;
    end

end
for i=1:ceil(aa(1))
    RandMu=rand;
    if RandMu<pMut
        A1=5;
        IA=ceil(rand*(19-A1));
        B1=2;
        IB=ceil(rand*(28-B1));
        j=ceil(rand*5);
        if j==1
            Mean=0.1589;
            STD=0.0935;
            temp1=ones([A1 B1])*(random('normal',Mean,STD,[1 1]));
        end
        if j==2
            Mean=0.0736;
            STD=0.0367;
            temp1=ones([A1 B1])*(random('normal',Mean,STD,[1 1]));
        end
        if j==3
            Mean=0.1814;
```

```
            STD=0.0789;
            temp1=ones([A1 B1])*(random('normal',Mean,STD,[1 1]));
        end
        if j==4
            Mean=0.1080;
            STD=0.0531;
            temp1=ones([A1 B1])*(random('normal',Mean,STD,[1 1]));
        end
        if j==5
            Mean=0.1902;
            STD=0.0865;
            temp1=ones([A1 B1])*(random('normal',Mean,STD,[1 1]));
        end
        popmut(i,j,IA:IA+A1-1,IB:IB+B1-1)=temp1;
    end
end
popmut=ActiveCellChange(popmut);
end
```

a.1.2.2.3. Tournament function

The other function is tournament which is as follows:

```
function [SelectedPop FitnessSelected ] = Tournoment(Fitness,Pop,pTour,NoSelectedPair)
%TOURNOMENT Summary of this function goes here
%   Detailed explanation goes here
aa=size(Pop);
for i=1:NoSelectedPair
    a=ceil(aa(1)*(rand(pTour,1)));
    b=ceil(aa(1)*(rand(pTour,1)));
    [~, e1]=min(Fitness(a));
    [~, e2]=min(Fitness(b));
    while a(e1)==b(e2)
        a=ceil(aa(1)*(rand(pTour,1)));
        b=ceil(aa(1)*(rand(pTour,1)));
        [~, e1]=min(Fitness(a));
        [~, e2]=min(Fitness(b));
    end
    SelectedPop(2*i-1,:,:,:)=Pop(a(e1),:,:,:);
    SelectedPop(2*i,:,:,:)=Pop(b(e2),:,:,:);
    FitnessSelected(2*i-1)=Fitness(a(e1));
    FitnessSelected(2*i)=Fitness(b(e2));

end
end
```

a.1.3. Infill drilling optimisation

As mentioned before, the developed interface can also be used for production and field development optimisation. In order to employ it for these purposes, the following changes are required: i- the objective function, instead of being a misfit error, is a net present value (generally speaking, an economical objective function), ii- the decision variables, instead of being the uncertain parameters of the problems, are the parameters which are going to be optimised, for instance the coordinates for a new infill well. We applied the interface for a field development optimisation in a coalbed methane reservoir. In this problem, the

decision variables are well locations; hence they are discrete variables. The objective function is the net present value with 10 percent net cash flow. In this problem, the water treatment price and gas price are required.

In this problem, the interface should generate an input file for the ECLIPSE which should have the addresses of the infill wells and the properties of them, such as well skins, control term and completion. Afterward, ECLIPSE should be executed, and then the cumulative gas production and water production for the period of understudy (10 years) should be extracted from the output file of ECLIPSE.

a.1.3.1. Coding

a.1.3.1.1. Interface

The codes for these three steps are as follows:

a.1.3.1.2. Objective function (NPV)

The objective function code is as follows (for 60 infill wells):

```
function [ NPV ] = Evaluation( WL )
%EVALUATION Summary of this function goes here
%   Detailed explanation goes here

infill(WL);
ECLIPSE;
Results=ExcelReader;

Timesteps=[0,120,240,360,480,600,720,840,960,1080,1200,1320,1440,1560,1680,1800,1920,2040,2
160,2280,2400,2520,2640,2760,2880,3000,3120,3240,3360,3480,3600,3720,3840,3960,4080,4200,43
20,4440,4560,4680,4800,4920,5040,5160,5280,5400,5520,5640,5760,5880,6000,6120,6240,6360,648
0,6600,6720,6840,6960,7080,7200;]';
Times=size(Timesteps);
ND=size(Results);
    j=1;
    for i=1:ND(1)
        for j=1:Times(1)
            if Results(i,1)==Timesteps(j)
                Results(i,1)=0;
            end

        end
    end
    Results(1,:)=[];
    ND(1)=ND(1)-1;
    i=0;
    while i<(ND(1));
        i=i+1;
        if Results(i,1)~=0
```

```
            Results(i,:)=[];
            ND(1)=ND(1)-1;
            i=i-1;
        end
    end
GT10year=Results(30,6);
WT10year=Results(30,5);
iii=1;
jjj=33;
while iii<=10
    WT(iii)=Results(jjj,5);
    GT(iii)=Results(jjj,6);
    iii=iii+1;
    jjj=jjj+3;
end
GTeachyear(1)=GT(1)-GT10year;
WTeachyear(1)=WT(1)-WT10year;
for i=2:10
    GTeachyear(i)=GT(i)-GT(i-1);
    WTeachyear(i)=WT(i)-WT(i-1);
end
for i=1:10
    Net(i)=(2.5*GTeachyear(i)*10^3-1.2*WTeachyear(i)*10^3-500*125*12)*0.92/(1.1)^i;
end
NPV=sum(Net)-80*1000000;
NPV=-NPV
end
```

a.1.3.1.3. GA for well placement

The well placement optimisation is better to have a specific optimisation algorithm, since the variables are discrete. GA with following form is used as the optimiser.

```
% Required Information
clear
clc
NoVar=160;
NoPop=40;
pCross=0.9;
pTour=4;
pMut=0.2;
NoElit=1;
NoGen=1000;
UBX=73;
LBX=1;
UBY=37;
LBY=1;
objfun='Evaluation'; %cost function to be optimized
objective=str2func(objfun);
%---------------
rand('seed',7)
% Generating intial population
Pop=popgen(NoPop,NoVar,UBX,UBY);
inpop=Pop;
%---------------
IterationNo=0;
%------------
for i=1:NoPop
    Fitness(i)=objective(Pop(i,:));
end
%-------------------
% %----------------
mine=min(Fitness)
while (IterationNo<NoGen)
    % Mutation and Crossover
    IterationNo=IterationNo+1;
    popmut=Scattred(Pop,Fitness,pTour,pCross,pMut,UBX,UBY);
    %-----------------------
    for i=1:NoPop
        Fitness2(i)=objective(popmut(i,:));
    end
    % %-------------------
```

```
    [ke2 ki2]= sort(Fitness2);
    %Elite
    newpop((1:NoPop),:)=popmut((ki2(1:NoPop)),:);
    [ke1 ki1]=sort(Fitness);
    popelit=Pop(ki1(1:NoElit),:);
    FitnessElit=Fitness(ki1(1:NoElit));
    Pop=newpop((1:(NoPop-NoElit)),:);
    Pop((NoPop-NoElit+1):NoPop,:)=popelit;
    Fitness(1:NoPop-NoElit)=Fitness2(ki2(1:(NoPop-NoElit)));
    Fitness((NoPop-NoElit+1):NoPop)=FitnessElit;
    %----------
    %---------------------
    SetAns(ItterationNo)=min(Fitness);
    %     SetAns(end)
    FitnessGlobal=min(Fitness);
    [r Ind]=min(Fitness);
    Global=Pop(Ind,:);
    GlobalSet(ItterationNo,:)=Global;
    %%%%%SetAns(end)
    pause(0.00001)
    subplot(1,2,1)
    semilogy(SetAns);
    subplot(1,2,2)
    bar(Global)
end

function popmut=Scattred(Pop,Fitness,pTour,pCross,pMut,UBX,UBY)
aa=size(Pop);
CM=randperm(aa(1));
for j=1:ceil(pCross*aa(1))
    % Tournoment
    NoSelectedPair=1;
    [ SelectedPop FitnessSelected ] = Tournoment( Fitness,Pop,pTour,NoSelectedPair );
    %-----------------
    M=SelectedPop(1,:);
    F=SelectedPop(2,:);
    pCrossChor=0.5;
    popcross(j,:)=F;
    pp2=randperm(aa(2));
    popcross(j,pp2(1:pCrossChor*aa(2)))=M(pp2(1:pCrossChor*aa(2)));
end
%----------------
%Mutation
popmut=Pop;
VV=randperm(aa(1));
OO=size(popcross);
popmut(VV(1:OO(1)),:)=popcross;
UU=randperm(aa(1));
for i=1:aa(1)
    RandMu=rand;
    if RandMu<pMut
        muGen=randi(aa(2),[1 1]);
        if mod(muGen,2)==0
            popmut(i,muGen)=randi(UBY,[1 1]);
        else
            popmut(i,muGen)=randi(UBX,[1 1]);
        end
    end
end
     function [ Pop ] = popgen(NoPop,NoVar,UBX,UBY)
        for i=1:NoVar
            if mod(i,2)==0
                Pop(:,i)=randi(UBY,[NoPop 1]);
            else
                Pop(:,i)=randi(UBX,[NoPop 1]);
            end
        end

   function [ SelectedPop FitnessSelected ] = Tournoment(Fitness,Pop,pTour,NoSelectedPair)
            %TOURNOMENT Summary of this function goes here
            %   Detailed explanation goes here
            aa=size(Pop);
            for i=1:NoSelectedPair
                a=ceil(aa(1)*(rand(pTour,1)));
                b=ceil(aa(1)*(rand(pTour,1)));
                [~, e1]=min(Fitness(a));
                [~, e2]=min(Fitness(b));
                while a(e1)==b(e2)
                    a=ceil(aa(1)*(rand(pTour,1)));
                    b=ceil(aa(1)*(rand(pTour,1)));
```

```
                [~, e1]=min(Fitness(a));
                [~, e2]=min(Fitness(b));
            end
            SelectedPop(2*i-1,:)=Pop(a(e1),:);
            SelectedPop(2*i,:)=Pop(b(e2),:);
            FitnessSelected(2*i-1)=Fitness(a(e1));
            FitnessSelected(2*i)=Fitness(b(e2));

        end
    end
```

a.1.3.2. Results of Infill drilling

The results of the well placement optimisation can be found in the following two peer-reviewed journal papers. “Salmachi, Alireza, Mohammad Sayyafzadeh, and Manouchehr Haghighi "Infill well placement optimization in coal bed methane reservoirs using genetic algorithm" Fuel (2013)” and “Salmachi, Alireza, Mohammad Sayyafzadeh, and Manouchehr Haghighi "Optimisation and economical evaluation of infill drilling in CSG reservoirs using a multi-objective genetic algorithm" APPEA 2013.

In the first paper, the optimal number and the optimal locations of infill wells for San Juan basin is sought. In this paper, a sensitivity analysis is also made for the water treatment price. In the second paper, the multi-objective approach is used in which there are two objectives: 1- maximising the corresponding net present value for the gas production, and 2- minimising the corresponding net present value for water treatment costs.

a.1.3.2.1. First paper

STATEMENT OF AUTHORSHIP

Infill well placement optimization in coal bed methane reservoirs using genetic algorithm

Fuel, 2013

Alireza Salmachi

Statement of contribution (in terms of the conceptualization of the work, its realization and its documentation):

Performing the simulations and the comparative study, data interpretation and writing the manuscript

Certification that the statement of contribution is accurate

Signed..... .....
Date.....

Mohammad Sayyafzadeh (candidate)

Statement of contribution (in terms of the conceptualization of the work, its realization and its documentation):

Construction of the framework, development of the GA code for the framework, evaluating the manuscript, and participating in the manuscript writing

Certification that the statement of contribution is accurate and permission is given for the inclusion of the paper in the thesis

Signed.. .....
Date...

Manouchehr Haghighi

Statement of contribution (in terms of the conceptualization of the work, its realization and its documentation):

Supervision of the work, evaluating the manuscript, and participating in the manuscript writing

Certification that the statement of contribution is accurate and permission is given for the inclusion of the paper in the thesis

Signed.. .....
Date.....

Salmachi, A., Sayyafzadeh, M. and Haghighi, M. (2013) Infill well placement optimization in coal bed methane reservoirs using genetic algorithm.
Fuel, v. 111, pp. 248–258, September 2013

a.1.3.2.2. Second paper

STATEMENT OF AUTHORSHIP

Optimisation and Economical Evaluation of Infill Drilling in Coal Seam Gas Reservoirs Using a Multi Objective Genetic Algorithm

Journal of APPEA, 2013

Alireza Salmachi

Statement of contribution (in terms of the conceptualization of the work, its realization and its documentation):

Performing the simulations and the comparative study, data interpretation and writing the manuscript

Certification that the statement of contribution is accurate

Signed.. .....
Date....

Mohammad Sayyafzadeh (candidate)

Statement of contribution (in terms of the conceptualization of the work, its realization and its documentation):

Construction of the framework, evaluating the manuscript, and participating in the manuscript writing

Certification that the statement of contribution is accurate and permission is given for the inclusion of the paper in the thesis

Signed.. .....
Date.....

Manouchehr Haghighi

Statement of contribution (in terms of the conceptualization of the work, its realization and its documentation):

Supervision of the work, evaluating the manuscript, and participating in the manuscript writing

Certification that the statement of contribution is accurate and permission is given for the inclusion of the paper in the thesis

Signed.. .....
Date.....

Salmachi, A., Sayyafzadeh, M. and Haghighi, M. (2013) Optimisation and economical evaluation of infill drilling in csg reservoirs using a multi-objective genetic algorithm.
Australian Petroleum Production and Exploration Association (APPEA) Journal, v. 53, pp. 381-390.

Bibliography

. Wikipedia. Available: (http://upload.wikimedia.org/wikipedia/commons/thumb/a/a9/Finite_element_solution.svg/800px-Finite_element_solution.svg.png) [Accessed].

AANONSEN, S. I., NÆVDAL, G., OLIVER, D. S., REYNOLDS, A. C. & VALLÈS, B. 2009. The Ensemble Kalman Filter in Reservoir Engineering--a Review. *SPE Journal,* 14**,** pp. 393-412.

AKAY, B. & KARABOGA, D. 2012. A modified Artificial Bee Colony algorithm for real-parameter optimization. *Information Sciences,* 192**,** 120-142.

AL-NAJEM, A., SIDDIQUI, S., SOLIMAN, M. & YUEN, B. Year. Streamline Simulation Technology: Evolution and Recent Trends. *In:* SPE Saudi Arabia Section Technical Symposium and Exhibition, 2012.

ARPS, J. J. Year. Analysis of decline curves. *In:* American Institute of Mining and Metallurgical Engineers -- Meeting, May 1944, 1944 New York, NY, United States. American Institute of Mining and Metallurgical Engineers, 20.

AZIZ, K. & SETTARI, A. 1979. *Petroleum reservoir simulation*, Chapman & Hall.

AZIZ, K. & SETTARI, A. 2002. *Petroleum reservoir simulation*, Calgary, K. Aziz & A. Settari.

BAKER, R. O., ANDERSON, T. & SANDHU, K. 2003. Using Decline Curves to Forecast Waterflooded Reservoirs: Fundamentals and Field Cases. *Canadian International Petroleum Conference.* Calgary, Alberta: Petroleum Society of Canada.

BALLESTER, P. J. & CARTER, J. N. 2007. A parallel real-coded genetic algorithm for history matching and its application to a real petroleum reservoir. *Journal of Petroleum Science and Engineering,* 59**,** 157-168.

BARKER, J., MAARTEN, C. & LARS, H. Year. Quantifying uncertainty in production forecasts: Another look at the PUNQ-S3 problem. *In:* SPE Annual Technical Conference and Exhibition, 2000.

BATYCKY, R. P., BLUNT, M. J. & THIELE, M. R. 1997. A 3D Field-Scale Streamline-Based Reservoir Simulator. *SPE Reservoir Engineering,* 12**,** 246-254.

BERTA, D., HARDY, H. H. & BEIER, R. A. 1994. Fractal Distributions of Reservoir Properties and Their Use in Reservoir Simulation. *International Petroleum Conference and Exhibition of Mexico.* Veracruz, Mexico: Society of Petroleum Engineers.

BISSELL, R. C., DUBRULE, O., LAMY, P., SWABY, P. & LEPINE, O. 1997. Combining Geostatistical Modelling With Gradient Information for History Matching: The Pilot Point Method. *SPE Annual Technical Conference and Exhibition.* San Antonio, Texas: 1997 Copyright 1997, Society of Petroleum Engineers, Inc.

BOUVIER, J., KAARS-SIJPESTEIJN, C., KLUESNER, D., ONYEJEKWE, C. & VAN DER PAL, R. 1989. Three-dimensional seismic interpretation and fault sealing investigations, Nun River Field, Nigeria. *AAPG Bulletin,* 73**,** 1397-1414.

BUCKLEY, P. S. 1964. *Techniques of process control*, Wiley.

CAERS, J. 2005. *Petroleum geostatistics*, Richardson, TX, Society of Petroleum Engineers.

CARTER, J. 2004. Using Bayesian Statistics to Capture the Effects of Modelling Errors in Inverse Problems. *Mathematical geology,* 36**,** 187-216.

CHAVENT, G., DUPUY, M. & LEMONNIER, P. 1975. History Matching by Use of Optimal Theory. 15**,** 74-86.

CHRISTIE, M., DEMYANOV, V. & ERBAS, D. 2006. Uncertainty quantification for porous media flows. *Journal of Computational Physics,* 217**,** 143-158.

COATS, K. H., DEMPSEY, J. R. & HENDERSON, J. H. 1970. *A New Technique for Determining Reservoir Description from Field Performance Data*.

COELLO, C. & TOSCANO PULIDO, G. 2001. A Micro-Genetic Algorithm for Multiobjective Optimization

Evolutionary Multi-Criterion Optimization. *In:* ZITZLER, E., THIELE, L., DEB, K., COELLO COELLO, C. & CORNE, D. (eds.). Springer Berlin / Heidelberg.

COELLO, C. A. 2000. An updated survey of GA-based multiobjective optimization techniques. *ACM Comput. Surv.*, 32**,** 109-143.

COELLO, C. A. C., LAMONT, G. B. & VELDHUIZEN, D. A. V. 2007. *Evolutionary Algorithms for Solving Multi-Objective Problems, Second Edition*, Springer US.

CORNE, D., KNOWLES, J. & OATES, M. 2000. The Pareto Envelope-Based Selection Algorithm for Multiobjective Optimization

Parallel Problem Solving from Nature PPSN VI. *In:* SCHOENAUER, M., DEB, K., RUDOLPH, G., YAO, X., LUTTON, E., MERELO, J. & SCHWEFEL, H.-P. (eds.). Springer Berlin / Heidelberg.

COUGHANOWR, D. R., KOPPEL & KOPPEL, L. B. 1965. *Process Systems Analysis and Control*, pp. xii. 491. McGraw-Hill Book Co.: New York.

CRAIG, P., GOLDSTEIN, M., SEHEULT, A. & SMITH, J. 1996. Bayes linear strategies for matching hydrocarbon reservoir history. *Bayesian statistics,* 5**,** 69-95.

CULLICK, A. S., JOHNSON, W. D. & SHI, G. 2006. Improved and More Rapid History Matching With a Nonlinear Proxy and Global Optimization. *SPE Annual Technical Conference and Exhibition.* San Antonio, Texas, USA: Society of Petroleum Engineers.

DAY, R. O. & LAMONT, G. B. Year. An effective explicit building block MOEA, the MOMGA-IIa. *In:* Evolutionary Computation, 2005. The 2005 IEEE Congress on, 5-5 Sept. 2005 2005. 17-24 Vol.1.

DE BOOR, C. 1978. *A practical guide to splines,* New York, Springer.

DE MARSILY, G., LAVEDAN, G., BOUCHER, M. & FASANINO, G. 1984. Interpretation of interference tests in a well field using geostatistical techniques to fit the permeability distribution in a reservoir model. *Geostatistics for Natural Resources Characterization, Part,* 2**,** 831-849.

DEB, K., PRATAP, A., AGARWAL, S. & MEYARIVAN, T. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *Evolutionary Computation, IEEE Transactions on,* 6**,** 182-197.

DELSHAD, M., BASTAMI, A. & POURAFSHARY, P. 2009. The Use of Capacitance-Resistive Model for Estimation of Fracture Distribution in the Hydrocarbon Reservoir. *SPE Saudi Arabia Section Technical Symposium.* AlKhobar, Saudi Arabia: Society of Petroleum Engineers.

DEUTSCH, C. V. 2002. *Geostatistical reservoir modeling,* Oxford ; New York, Oxford University Press.

DOHERTY, J. 2003. Ground Water Model Calibration Using Pilot Points and Regularization. *Ground Water,* 41**,** 170-177.

DONG, Y. & OLIVER, D. S. 2008. Reservoir Simulation Model Updates via Automatic History Matching with Integration of Seismic Impedance Change and Production Data. *International Petroleum Technology Conference.* Kuala Lumpur, Malaysia: International Petroleum Technology Conference.

DOS SANTOS, E., XAVIER, C., GOLDFELD, P., DICKSTEIN, F. & WEBER DOS SANTOS, R. 2009. Comparing Genetic Algorithms and Newton-Like Methods for the Solution of the History Matching Problem. *Computational Science – ICCS 2009,* 5544**,** 377-386.

EMERICK, A. & REYNOLDS, A. 2012. History matching time-lapse seismic data using the ensemble Kalman filter with multiple data assimilations. *Computational Geosciences,* 16**,** 639-659.

ERICKSON, M., MAYER, A. & HORN, J. 2001. The Niched Pareto Genetic Algorithm 2 Applied to the Design of Groundwater Remediation Systems

Evolutionary Multi-Criterion Optimization. *In:* ZITZLER, E., THIELE, L., DEB, K., COELLO COELLO, C. & CORNE, D. (eds.). Springer Berlin / Heidelberg.

ERTEKIN, T., ABOU-KASSEM, J. & KING, G. 2001a. Basic applied reservoir simulation. *Richardson, Texas: Society of Petroleum Engineers.*

ERTEKIN, T., ABOU-KASSEM, J. H. & KING, G. R. 2001b. *Basic applied reservoir simulation*, Richardson, Tex., Society of Petroleum Engineers.
EVENSEN, G., HOVE, J., MEISINGSET, H., REISO, E., SEIM, K. S. & ESPELID, Ø. Year. Using the EnKF for assisted history matching of a North Sea reservoir model. *In:* SPE Reservoir Simulation Symposium, 2007.
EYDINOV, D., GAO, G., LI, G. & REYNOLDS, A. C. 2009. Simultaneous Estimation of Relative Permeability and Porosity/Permeability Fields by History Matching Production Data. *Journal of Canadian Petroleum Technology,* 48**,** 13-25.
FANCHI, J. R. 2001. Principles of Applied Reservoir Simulation (Second Edition). Burlington: Gulf Professional Publishing.
FANCHI, J. R. 2005. *Principles of applied reservoir simulation*, Gulf Professional Publishing.
FARMER, C. 2005. Geological Modelling and Reservoir Simulation. *In:* ISKE, A. & RANDEN, T. (eds.) *Mathematical Methods and Modelling in Hydrocarbon Exploration and Production.* Springer Berlin Heidelberg.
FIENEN, M. N., MUFFELS, C. T. & HUNT, R. J. 2009. On Constraining Pilot Point Calibration with Regularization in PEST. *Ground Water,* 47**,** 835-844.
FLETCHER, R. 1987. Practical methods of optimization.
FONSECA, C. M. & FLEMING, P. J. Year. Genetic Algorithms for Multiobjective
Optimization: Formulation, Discussion and Generalization. *In:* Proceedings of the Fifth International Conference on Genetic Algorithms, 1993 California. 416-423.
FOURMAN, M. P. Year. Compaction of Symbolic Layout using Genetic Algorithms. *In:* Genetic Algorithms and their Applications: Proceedings of the First International Conference on Genetic Algorithms, 1985 New Jersey. 141-153.
GAO, G., LI, G. & REYNOLDS, A. C. 2007. A Stochastic Optimization Algorithm for Automatic History Matching. *SPE Journal,* 12**,** pp. 196-208.
GAO, R. X. & YAN, R. 2011. *Wavelets: Theory and applications for manufacturing*, Springer.
GARDNER, W. 1984. Learning characteristics of stochastic-gradient-descent algorithms: A general study, analysis, and critique. *Signal Processing,* 6**,** 113-133.
GAVALAS, G. R., SHAH, P. C. & SEINFELD, J. H. 1976. Reservoir History Matching by Bayesian Estimation. 16.
GEIR, N. V., JOHNSEN, L., AANONSEN, S. & VEFRING, E. Year. Reservoir monitoring and continuous model updating using ensemble Kalman filter. *In:* SPE Annual Technical Conference and Exhibition, 2003.
GLOVER, F. 1989. Tabu search—part I. *ORSA Journal on computing,* 1**,** 190-206.
GLOVER, F. 1990. Tabu search—part II. *ORSA Journal on computing,* 2**,** 4-32.
GOLUB, G. H., HEATH, M. & WAHBA, G. 1979. Generalized Cross-Validation as a Method for Choosing a Good Ridge Parameter. *Technometrics,* 21**,** 215-223.
GRIMSTAD, A.-A., MANNSETH, T., NÆVDAL, G. & URKEDAL, H. 2003. Adaptive Multiscale Permeability Estimation. *Computational Geosciences,* 7**,** 1-25.
GRIMSTAD, A. A., MANNSETH, T., NÆVDAL, G. & URKEDAL, H. 2001. Scale Splitting Approach to Reservoir Characterization. *SPE Reservoir Simulation Symposium.* Houston, Texas: Copyright 2001, Society of Petroleum Engineers Inc.
GU, Y. & OLIVER, D. 2005. History matching of the PUNQ-S3 reservoir model using the ensemble Kalman filter. *SPE Journal,* 10**,** 217-224.
GUO, B., LYONS, W. C. & GHALAMBOR, A. 2007. *Petroleum production engineering: a computer-assisted approach*, Gulf Professional Publishing.
GWIAZDA, T. D. 2007. *Genetic algorithms reference*, Polongne : Tomasz Gwiazda, 2006.
HAJELA, P. & LIN, C. Y. 1992. Genetic search strategies in multicriterion optimal design. *Structural and Multidisciplinary Optimization,* 4**,** 99-107.
HAJIZADEH, Y., CHRISTIE, M. & DEMYANOV, V. 2011a. Ant colony optimization for history matching and uncertainty quantification of reservoir models. *Journal of Petroleum Science and Engineering,* 77**,** 78-92.
HAJIZADEH, Y., CHRISTIE, M. A. & DEMYANOV, V. 2010. History matching with differential evolution approach; a look at new search strategies. *SPE EUROPEC/EAGE Annual Conference and Exhibition.* Barcelona, Spain: Society of Petroleum Engineers.

HAJIZADEH, Y., CHRISTIE, M. A. & DEMYANOV, V. 2011b. Towards Multiobjective History Matching: Faster Convergence and Uncertainty Quantification. *SPE Reservoir Simulation Symposium.* The Woodlands, Texas, USA.

HANSEN, P. 2001. The L-curve and its use in the numerical treatment of inverse problems. *Computational inverse problems in electrocadiology***,** 119-142.

HARVEY, A. C. 1991. *Forecasting, structural time series models and the Kalman filter,* Cambridge university press.

HAUPT, R. L. & HAUPT, S. E. 2004. *Practical genetic algorithms,* Hoboken, N.J., John Wiley.

HE, N., REYNOLDS, A. C. & OLIVER, D. S. 1997. Three-Dimensional Reservoir Description From Multiwell Pressure Data and Prior Information. *SPE Journal,* 2**,** 312-327.

HEGSTAD, B. K. & OMRE, H. 1997. Uncertainty assessment in history matching and forecasting. *Geostatistics Wollongong,* 96**,** 585-596.

HOHN, M. E. 1999. *Geostatistics and petroleum geology,* Dordrecht, Kluwer.

HOLLAND, J. H. 1975. *Adaptation in Natural and Artificial Systems,* The University of Michigan Press, Ann Arbor.

HORN, J., NAFPLIOTIS, N. & GOLDBERG, D. E. Year. A niched Pareto genetic algorithm for multiobjective optimization. *In:* Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on, 27-29 Jun 1994 1994. 82-87 vol.1.

HUNT, R. J., DOHERTY, J. & TONKIN, M. J. 2007. Are models too simple? Arguments for increased parameterization. *Ground Water,* 45**,** 254-262.

JACQUARD, P. 1965. Permeability Distribution From Field Pressure Data. 5.

JAFARPOUR, B., GOYAL, V., MCLAUGHLIN, D. & FREEMAN, W. 2010. Compressed History Matching: Exploiting Transform-Domain Sparsity for Regularization of Nonlinear Dynamic Data Integration Problems. *Mathematical Geosciences,* 42**,** 1-27.

JAFARPOUR, B. & MCLAUGHLIN, D. 2007. History Matching with an Ensemble Kalman Filter and Discrete Cosine Parameterization. *SPE Annual Technical Conference and Exhibition.* Anaheim, California, U.S.A.: Society of Petroleum Engineers.

JAFARPOUR, B. & MCLAUGHLIN, D. B. 2009. Reservoir Characterization With the Discrete Cosine Transform. *SPE Journal,* 14**,** pp. 182-201.

JIN, Y. 2005. A comprehensive survey of fitness approximation in evolutionary computation. *Soft Computing - A Fusion of Foundations, Methodologies and Applications,* 9**,** 3-12.

JOURNEL, A. G. & HUIJBREGTS, C. 1978. *Mining geostatistics,* London ; New York, Academic Press.

JUNG, S. & CHOE, J. 2010. Stochastic Estimation of Oil Production by History Matching with Ensemble Kalman Filter. *Energy Sources, Part A: Recovery, Utilization, and Environmental Effects,* 32**,** 952 - 961.

KARABOGA, D. 2005. An idea based on honey bee swram for numerical optimization. Kayseri: Erciyes University.

KARABOGA, D. & AKAY, B. 2009. A comparative study of Artificial Bee Colony algorithm. *Applied Mathematics and Computation,* 214**,** 108-132.

KARABOGA, D. & BASTURK, B. 2008. On the performance of artificial bee colony (ABC) algorithm. *Applied Soft Computing,* 8**,** 687-697.

KELKAR, M. & PEREZ, G. 2002. *Applied geostatistics for reservoir characterization,* Richardson Tex., Society of Petroleum Engineers.

KITA, H., YABUMOTO, Y., MORI, N. & NISHIKAWA, Y. 1996. Multi-objective optimization by means of the thermodynamical genetic algorithm

Parallel Problem Solving from Nature — PPSN IV. *In:* VOIGT, H.-M., EBELING, W., RECHENBERG, I. & SCHWEFEL, H.-P. (eds.). Springer Berlin / Heidelberg.

KNOWLES, J. & CORNE, D. Year. The Pareto archived evolution strategy: a new baseline algorithm for Pareto multiobjective optimisation. *In:* Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on, 1999 1999. 105 Vol. 1.

KNOWLES, J. D. & CORNE, D. W. 2000. Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy. *Evolutionary Computation,* 8**,** 149-172.

KONAK, A., COIT, D. W. & SMITH, A. E. 2006. Multi-objective optimization using genetic algorithms: A tutorial. *Reliability Engineering & System Safety,* 91**,** 992-1007.

KULKARNI, K. & DATTA-GUPTA, A. 2000. Estimating relative permeability from production data: A streamline approach. *SPE Journal,* 5**,** 402-411.

KURSAWE, F. 1991. A variant of evolution strategies for vector optimization

Parallel Problem Solving from Nature. *In:* SCHWEFEL, H.-P. & MÄNNER, R. (eds.). Springer Berlin / Heidelberg.

LAKE, L. W., LIANG, X., EDGAR, T. F., AL-YOUSEF, A., SAYARPOUR, M. & WEBER, D. 2007. Optimization Of Oil Production Based On A Capacitance Model Of Production And Injection Rates. *Hydrocarbon Economics and Evaluation Symposium.* Dallas, Texas, U.S.A.: Society of Petroleum Engineers.

LANDA, J. L. 1997. *Reservoir parameter estimation constrained to pressure transients, performance history and distributed saturation data.* stanford university.

LAUMANNS, M. & OCENASEK, J. 2002. Bayesian Optimization Algorithms for Multi-objective Optimization

Parallel Problem Solving from Nature — PPSN VII. *In:* GUERVÓS, J., ADAMIDIS, P., BEYER, H.-G., SCHWEFEL, H.-P. & FERNÁNDEZ-VILLACAÑAS, J.-L. (eds.). Springer Berlin / Heidelberg.

LAVENUE, A. M. & PICKENS, J. F. 1992. Application of a coupled adjoint sensitivity and kriging approach to calibrate a groundwater flow model. *Water Resources Research,* 28**,** 1543-1569.

LEE, T.-Y. & SEINFELD, J. H. 1987. Estimation of two-phase petroleum reservoir properties by regularization. *Journal of Computational Physics,* 69**,** 397-419.

LEO, T.-Y., KRAVARIA, C. & SEINFELD, J. H. 1986. History Matching by Spline Approximation and Regularization in Single-Phase Areal Reservoirs. *SPE Reservoir Engineering*.

LI, B. & FRIEDMANN, F. 2005. Novel Multiple Resolutions Design of Experiment/Response Surface Methodology for Uncertainty Analysis of Reservoir Simulation Forecasts. *SPE Reservoir Simulation Symposium.* The Woodlands, Texas: 2005,. Society of Petroleum Engineers Inc.

LI, H., CHEN, S., YANG, D. & TONTIWACHWUTHIKUL, P. 2009. Estimation of Relative Permeability by Assisted History Matching Using the Ensemble Kalman Filter Method. *Canadian International Petroleum Conference.* Calgary, Alberta: Petroleum Society of Canada.

LI, H., MANJUNATH, B. S. & MITRA, S. K. 1995. Multisensor Image Fusion Using the Wavelet Transform. *Graphical Models and Image Processing,* 57**,** 235-245.

LI, L. & JAFARPOUR, B. 2010. A sparse Bayesian framework for conditioning uncertain geologic models to nonlinear flow measurements. *Advances in Water Resources,* 33**,** 1024-1042.

LI, R., REYNOLDS, A. C. & OLIVER, D. S. 2001. History Matching of Three-Phase Flow Production Data. *SPE Reservoir Simulation Symposium.* Houston, Texas: Copyright 2001, Society of Petroleum Engineers Inc.

LI, R., REYNOLDS, A. C. & OLIVER, D. S. 2003. Sensitivity Coefficients for Three-Phase Flow History Matching. 42.

LIANG, B., SEPEHRNOORI, K. & DELSHAD, M. 2009. A Weighted Ensemble Kalman Filter for Automatic History Matching. *Petroleum Science and Technology,* 27**,** 1062 - 1091.

LIU, N., BETANCOURT, S. & OLIVER, D. S. 2001. Assessment of Uncertainty Assessment Methods. *SPE Annual Technical Conference and Exhibition.* New Orleans, Louisiana: Copyright 2001, Society of Petroleum Engineers Inc.

LIU, N. & OLIVER, D. 2004. Automatic history matching of geologic facies. *SPE Journal,* 9**,** 429-436.

LIU, N. & OLIVER, D. S. 2005. Critical Evaluation of the Ensemble Kalman Filter on History Matching of Geologic Facies. *SPE Reservoir Evaluation & Engineering,* 8**,** pp. 470-477.

LODOEN, O. P. & OMRE, H. 2008. Scale-Corrected Ensemble Kalman Filtering Applied to Production-History Conditioning in Reservoir Evaluation. *SPE Journal,* 13**,** 177-194.

LOPHAVEN, S. N., NIELSEN, H. B. & SØNDERGAARD, J. 2002. DACE – A MATLAB KRIGING TOOLBOX. Lyngby – Denmark: Technical University of Denmark.

LORENTZEN, R., NAEVDAL, G., VALLES, B., BERG, A. & GRIMSTAD, A.-A. Year. Analysis of the ensemble Kalman filter for estimation of permeability and porosity in reservoir models. *In:* SPE Annual Technical Conference and Exhibition, 2005.

LORENTZEN, R. J., FLORNES, K. M. & NAEVDAL, G. 2009. History Matching Channelized Reservoirs Using the Ensemble Kalman Filter. *International Petroleum Technology Conference.* Doha, Qatar: 2009, International Petroleum Technology Conference.

LU, P. & HORNE, R. N. 2000. A Multiresolution Approach to Reservoir Parameter Estimation Using Wavelet Analysis. *SPE Annual Technical Conference and Exhibition.* Dallas, Texas: Copyright 2000, Society of Petroleum Engineers Inc.

MAKHLOUF, E., CHEN, W., WASSERMAN, M. & SEINFELD, J. 1993a. A general history matching algorithm for three-phase, three-dimensional petroleum reservoirs. *SPE Advanced Technology Series,* 1**,** 83-92.

MAKHLOUF, E. M., CHEN, W. H., WASSERMAN, M. L. & SEINFELD, J. H. 1993b. A General History Matching Algorithm for Three-Phase, Three-Dimensional Petroleum Reservoirs. *SPE Advanced Technology Series,* 1.

MANTICA, S., COMINELLI, A. & MANTICA, G. 2002. Combining Global and Local Optimization Techniques for Automatic History Matching Production and Seismic Data. *SPE Journal,* 7**,** 123-130.

MASCHIO, C., JOSÉ, D. & SCHIOZER 2005. Assisted History Matching Using Streamline Simulation. *Petroleum Science and Technology,* 23**,** 761 - 774.

MASCHIO, C., VIDAL, A. C. & SCHIOZER, D. J. 2008. A framework to integrate history matching and geostatistical modeling using genetic algorithm and direct search methods. *Journal of Petroleum Science and Engineering,* 63**,** 34-42.

MATHWORK, T. 2011a. gamultiobj.

MATHWORK, T. 2011b. MATLAB. 7.12.0 ed.

MATHWORK, T. 2011c. Neural network toolbox.

MATHWORK, T. 2011d. Wavelet Toolbox. Massachusetts.

MCKAY, M. D., BECKMAN, R. J. & CONOVER, W. J. 1979. Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics,* 21**,** 239-245.

MCLAUGHLIN, D. & TOWNLEY, L. R. 1996. A reassessment of the groundwater inverse problem. *Water Resources Research,* 32**,** 1131-1161.

MIKLES, J. & FIKAR, M. 2007. Process modelling, identification, and control. Berlin ; New York: Springer.

MJOLSNESS, E. & DECOSTE, D. 2001. Machine learning for science: state of the art and future prospects. *Science,* 293**,** 2051-2055.

MOHAMED, L., CHRISTIE, M. A. & DEMYANOV, V. 2010. Reservoir Model History Matching with Particle Swarms: Variants Study. *SPE Oil and Gas India Conference and Exhibition.* Mumbai, India: Society of Petroleum Engineers.

MOHAMED, L., CHRISTIE, M. A. & DEMYANOV, V. 2011. History Matching and Uncertainty Quantification: Multiobjective Particle Swarm Optimisation Approach. *SPE EUROPEC/EAGE Annual Conference and Exhibition.* Vienna, Austria: Society of Petroleum Engineers.

MOLGA, M. & SMUTNICKI, C. 2005. Test Functions for optimization needs.

MOU, C., PENG, L., YAO, D. & XIAO, D. 2005. Image Reconstruction Using a Genetic Algorithm for Electrical Capacitance Tomography. *Tsinghua Science & Technology,* 10**,** 587-592.

NÃ¦VDAL, G., JOHNSEN, L. M., AANONSEN, S. I. & VEFRING, E. H. 2003. Reservoir Monitoring and Continuous Model Updating Using Ensemble Kalman Filter. *SPE Annual Technical Conference and Exhibition.* Denver, Colorado: Society of Petroleum Engineers.

NIKOLOV, S., HILL, P., BULL, D. & CANAGARAJAH, N. 2001. Wavelets for image fusion. *In:* MEYER, A. P. A. F. (ed.) *Wavelets in signal and image analysis.* Dordrecht: Kluwer Academic Publishers.

NURHAN, K. 2009. A new design method based on artificial bee colony algorithm for digital IIR filters. *Journal of the Franklin Institute,* 346**,** 328-348.

OLEA, R. A. 1999. *Geostatistics for engineers and earth scientists,* Boston ; London, Kluwer Academic.

OLIVER, D. 1994. INCORPORATION OF TRANSIENT PRESSURE DATA INTO RESERVOIR. *IN SITU,* 18**,** 243-275.

OLIVER, D. & CHEN, Y. 2010. Recent progress on reservoir history matching: a review. *Computational Geosciences*, 1-37.

OLIVER, D. S., HE, N. & REYNOLDS, A. C. Year. Conditioning permeability fields to pressure data. *In:* 5th European conference on the mathematics of oil recovery, 1996.

OLIVER, D. S., REYNOLDS, A. C. & LIU, N. 2008. *Inverse theory for petroleum reservoir characterization and history matching*, Cambridge Univ Pr.

OLSSON, A. M. & SANDBERG, G. E. 2002. Latin hypercube sampling for stochastic finite element analysis. *Journal of Engineering Mechanics*, 128**,** 121-125.

OSTERLOH, W. T. 2008. Use of Multiple-Response Optimization to Assist Reservoir Simulation Probabilistic Forecasting and History Matching. *SPE Annual Technical Conference and Exhibition.* Denver, Colorado, USA: Society of Petroleum Engineers.

OSYCZKA, A. & KUNDU, S. 1995. A new method to solve generalized multicriteria optimization problems using the simple genetic algorithm. *Structural and Multidisciplinary Optimization*, 10**,** 94-99.

OUENES, A. & BHAGAVAN, S. 1994. Application of Simulated Annealing and Other Global Optimization Methods to Reservoir Description: Myths and Realities. *SPE Annual Technical Conference and Exhibition.* New Orleans, Louisiana: 1994 Society of Petroleum Engineers.

OUENES, A., BREFORT, B., MEUNIER, G. & DUPERE, S. 1993. A New Algorithm for Automatic History Matching: Application of Simulated Annealing Method (SAM) to Reservoir Inverse Modeling. Society of Petroleum Engineers.

PAJARES, G. & MANUEL DE LA CRUZ, J. 2004. A wavelet-based image fusion tutorial. *Pattern Recognition*, 37**,** 1855-1872.

PANIGRAHI, B. K., SHI, Y. & LIM, M.-H. 2011. *Handbook of swarm intelligence : concepts, principles and applications*, Berlin, Springer.

PEDDIBHOTLA, S., DATTA-GUPTA, A. & XUE, G. 1997. Multiphase Streamline Modeling in Three Dimensions: Further Generalizations and a Field Application. *SPE Reservoir Simulation Symposium.* Dallas, Texas: 1997 Copyright 1997, Society of Petroleum Engineers, Inc.

PELIKAN, M., SASTRY, K. & GOLDBERG, D. E. 2005. Multiobjective hBOA, clustering, and scalability. *Proceedings of the 2005 conference on Genetic and evolutionary computation.* Washington DC, USA: ACM.

PETROSIAN, A. A. & MEYER, F. O. G. 2001. *Wavelets in signal and image analysis : from theory to practice*, Dordrecht ; London, Kluwer Academic.

PORTELLAAND, R. C. M. & PRAIS, F. 1999. Use of Automatic History Matching and Geostatistical Simulation to Improve Production Forecast. *Latin American and Caribbean Petroleum Engineering Conference.* Caracas, Venezuela: Society of Petroleum Engineers.

RAMGULAM, A., ERTEKIN, T. & FLEMINGS, P. B. 2007. Utilization of Artificial Neural Networks in the Optimization of History Matching. *Latin American & Caribbean Petroleum Engineering Conference.* Buenos Aires, Argentina: Society of Petroleum Engineers.

RANGAIAH, G. P. 2001. Evaluation of genetic algorithms and simulated annealing for phase equilibrium and stability problems. *Fluid Phase Equilibria*, 187–188**,** 83-109.

REICHLE, R. H., MCLAUGHLIN, D. B. & ENTEKHABI, D. 2002. Hydrologic data assimilation with the ensemble Kalman filter. *Monthly Weather Review*, 130**,** 103-114.

REMY, N., BOUCHER, A. & WU, J. 2009. *Applied geostatistics with SGeMS : a user's guide*, Cambridge, Cambridge University Press.

REYNOLDS, A. C., HE, N., CHU, L. & OLIVER, D. S. 1996. Reparameterization Techniques for Generating Reservoir Descriptions Conditioned to Variograms and Well-Test Pressure Data. *SPE Journal*, 1.

RODRIGUES, J. 2006. Calculating derivatives for automatic history matching. *Computational Geosciences*, 10**,** 119-136.

ROMERO, C. E. & CARTER, J. N. 2001. Using genetic algorithms for reservoir characterisation. *Journal of Petroleum Science and Engineering*, 31**,** 113-123.

ROMERO, C. E., CARTER, J. N., GRINGARTEN, A. C. & ZIMMERMAN, R. W. 2000. A Modified Genetic Algorithm for Reservoir Characterisation. *International Oil and Gas*

Conference and Exhibition in China. Beijing, China: Copyright 2000, Society of Petroleum Engineers Inc.

ROTONDI, M., NICOTRA, G., GODI, A., CONTENTO, F. M., BLUNT, M. J. & CHRISTIE, M. 2006. Hydrocarbon Production Forecast and Uncertainty Quantification: A Field Application. *SPE Annual Technical Conference and Exhibition.* San Antonio, Texas, USA: Society of Petroleum Engineers.

RUST, C. B. & CAUDLE, B. H. 1972. A STREAMLINE SIMULATOR FOR OIL BANK BUILD-UP IN A WATERFLOOD. Society of Petroleum Engineers.

SAEMI, M., AHMADI, M. & VARJANI, A. Y. 2007. Design of neural networks using genetic algorithm for the permeability estimation of the reservoir. *Journal of Petroleum Science and Engineering,* 59**,** 97-105.

SAHNI, I. & HORNE, R. N. 2005. Multiresolution Wavelet Analysis for Improved Reservoir Description. *SPE Reservoir Evaluation & Engineering,* 8**,** pp. 53-69.

SAHNI, I. & HORNE, R. N. 2006a. Generating Multiple History-Matched Reservoir-Model Realizations Using Wavelets. *SPE Reservoir Evaluation & Engineering,* 9**,** pp. 217-226.

SAHNI, I. & HORNE, R. N. 2006b. Stochastic History Matching and Data Integration for Complex Reservoirs Using a Wavelet-Based Algorithm. *SPE Annual Technical Conference and Exhibition.* San Antonio, Texas, USA: Society of Petroleum Engineers.

SAMPAIO, T. P., FILHO, V. J. M. F. & NETO, A. D. S. 2009. An Application of Feed Forward Neural Network as Nonlinear Proxies for Use During the History Matching Phase. *Latin American and Caribbean Petroleum Engineering Conference.* Cartagena de Indias, Colombia: Society of Petroleum Engineers.

SARMA, P., DURLOFSKY, L., AZIZ, K. & CHEN, W. 2006. Efficient real-time reservoir management using adjoint-based optimal control and model updating. *Computational Geosciences,* 10**,** 3-36.

SARMA, P., DURLOFSKY, L. J., AZIZ, K. & CHEN, W. H. 2007. A New Approach to Automatic History Matching Using Kernel PCA. *SPE Reservoir Simulation Symposium.* Houston, Texas, U.S.A.: Society of Petroleum Engineers.

SAYARPOUR, M., KABIR, C. S. & LAKE, L. W. 2009a. Field Applications of Capacitance-Resistance Models in Waterfloods. *SPE Reservoir Evaluation & Engineering,* 12**,** pp. 853-864.

SAYARPOUR, M., KABIR, S., SEPEHRNOORI, K. & LAKE, L. W. 2010. Probabilistic History Matching With the Capacitance-Resistance Model in Waterfloods: A Precursor to Numerical Modeling. *SPE Improved Oil Recovery Symposium.* Tulsa, Oklahoma, USA: Society of Petroleum Engineers.

SAYARPOUR, M., ZULUAGA, E., KABIR, C. S. & LAKE, L. W. 2007. The Use of Capacitance-Resistive Models for Rapid Estimation of Waterflood Performance. *SPE Annual Technical Conference and Exhibition.* Anaheim, California, U.S.A.: Society of Petroleum Engineers.

SAYARPOUR, M., ZULUAGA, E., KABIR, C. S. & LAKE, L. W. 2009b. The use of capacitance-resistance models for rapid estimation of waterflood performance and optimization. *Journal of Petroleum Science and Engineering,* 69**,** 227-238.

SCHAFFER, J. D. 1985. Multiple Objective Optimization with Vector Evaluated Genetic Algorithms. *Proceedings of the 1st International Conference on Genetic Algorithms.* L. Erlbaum Associates Inc.

SCHLUMBERGER 2010. ECLIPSE. 2010.2 ed.

SCHULZE-RIEGERT, R. W., HAASE, O. & NEKRASSOV, A. 2003. Combined Global and Local Optimization Techniques Applied to History Matching. *SPE Reservoir Simulation Symposium.* Houston, Texas.

SCHULZE-RIEGERT, R. W., KROSCHE, M., FAHIMUDDIN, A. & GHEDAN, S. G. 2007. Multiobjective Optimization With Application to Model Validation and Uncertainty Quantification. *SPE Middle East Oil and Gas Show and Conference.* Kingdom of Bahrain: Society of Petroleum Engineers.

SCHULZE-RIEGERT, R. W., KROSCHE, M., PAJONK, O. & MUSTAFA, H. 2009. Data Assimilation Coupled to Evolutionary Algorithms A Case Example in History Matching. *SPE/EAGE Reservoir Characterization and Simulation Conference.* Abu Dhabi, UAE: Society of Petroleum Engineers.

SHAH, S., GAVALAS, G. R. & SEINFELD, J. H. 1978. Error Analysis in History Matching: The Optimum Level of Parameterization.
SHELDON, J. W., HARRIS, C. D. & BAVLY, D. 1960. A Method for General Reservoir Behavior Simulation on Digital Computers.
SILVA, P. C., MASCHIO, C. & SCHIOZER, D. J. 2006. Applications of the Soft Computing in the Automated History Matching. *Canadian International Petroleum Conference.* Calgary, Alberta: Petroleum Society of Canada.
SILVA, P. C., MASCHIO, C. & SCHIOZER, D. J. Year. Application of neural network and global optimization in history matching. *In,* 2008 500-5th Avenue SW, Suite 425, Calgary, Alberta, AB T2P 3L5, Canada. Petroleum Society, 22-25.
SIVIA, D. S. & SKILLING, J. 2006. *Data analysis : a Bayesian tutorial,* Oxford, Oxford University Press.
SKJERVHEIM, J.-A., EVENSEN, G., AANONSEN, S., RUUD, B. O. & JOHANSEN, T.-A. Year. Incorporating 4D seismic data in reservoir simulation models using ensemble Kalman filter. *In:* SPE Annual Technical Conference and Exhibition, 2005.
SPALL, J. C. 1992. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *Automatic Control, IEEE Transactions on,* 37**,** 332-341.
SRINIVAS, N. & DEB, K. 1994. Muiltiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. *Evolutionary Computation,* 2**,** 221-248.
STARK, H. 1987. *Image recovery--theory and application,* Orlando, Academic Press.
STATHAKI, T. 2008. *Image fusion : algorithms and applications,* Amsterdam ; London, Academic Press.
STEWART, G. W. 1973. *Introduction to matrix computations,* Orlando ; London, Academic Press.
SUN, N.-Z. 1994. *Inverse problems in groundwater modeling,* Dordrecht ; Boston, Kluwer Academic.
SZETO, W. Y., WU, Y. & HO, S. C. 2011. An artificial bee colony algorithm for the capacitated vehicle routing problem. *European Journal of Operational Research,* 215**,** 126-135.
TAN, T. B. & KALOGERAKIS, N. 1992. A Three - Dimensional Three - Phase Automatic History Matching Model: Reliability Of Parameter Estimates. 31.
TARANTOLA, A. 1987. *Inverse problem theory : methods for data fitting and model parameter estimation,* Amsterdam ; Oxford, Elsevier.
TARANTOLA, A. 2005. *Inverse problem theory and methods for model parameter estimation,* Philadelphia, PA, Society for Industrial and Applied Mathematics.
TAUTENHAHN, U. & QI-NIAN, J. 2003. Tikhonov regularization and a posteriori rules for solving nonlinear ill posed problems. *Inverse Problems,* 19**,** 1.
TAVASSOLI, Z., CARTER, J. & KING, P. 2004. Errors in history matching. *SPE Journal,* 9**,** 352-361.
THOMAS, L. K., HELLUMS, L. J. & REHEIS, G. M. 1972. A Nonlinear Automatic History Matching Technique for Reservoir Simulation Models. 12**,** 508-514.
TIKHONOV, A. N. & ARSENIN, V. I. 1977. *Solutions of ill-posed problems,* Washington, D.C., V.H.Winston.
TOSCANO PULIDO, G. & COELLO COELLO, C. 2003. The Micro Genetic Algorithm 2: Towards Online Adaptation in Evolutionary Multiobjective Optimization
Evolutionary Multi-Criterion Optimization. *In:* FONSECA, C., FLEMING, P., ZITZLER, E., THIELE, L. & DEB, K. (eds.). Springer Berlin / Heidelberg.
TOWILL, D. R. 1970. *Transfer function techniques for control engineers,* London, Iliffe Books.
VAN LAARHOVEN, P. J. & AARTS, E. H. 1987. *Simulated annealing: theory and applications,* Springer.
VEFRING, E. H., NYGAARD, G. H., LORENTZEN, R. J., NAEVDAL, G. & FJELDE, K. K. 2006. Reservoir Characterization During Underbalanced Drilling (UBD): Methodology and Active Tests. *SPE Journal,* 11**,** pp. 181-192.
VELDHUIZEN, D. A. V. & LAMONT, G. B. 1999. Genetic Algorithms, Building Blocks, and Multiobjective Optimization. *Proceedings*
of the 1999 Genetic and Evolutionary Computation Conference. Florida.
VELEZ-LANGS, O. 2005. Genetic algorithms in oil industry: An overview. *Journal of Petroleum Science and Engineering,* 47**,** 15-22.

WALD, L. 1999. Some terms of reference in data fusion. *Geoscience and Remote Sensing, IEEE Transactions on,* 37**,** 1190-1193.
WELCH, G. & BISHOP, G. 1995. An introduction to the Kalman filter.
WEN, X.-H., DEUTSCH, C. V. & CULLICK, A. S. 1998. High-Resolution Reservoir Models Integrating Multiple-Well Production Data. *SPE Journal,* 3**,** 344-355.
WEN, X. H., LEE, S. & YU, T. 2006. Simultaneous integration of pressure, water cut, 1 and 4-D seismic data in geostatistical reservoir modeling. *Mathematical geology,* 38**,** 301-325.
WIKIPEDIA. *Conjugate gradient method* [Online]. Available: http://en.wikipedia.org/wiki/Conjugate_gradient_method#The_conjugate_gradient_method_as_a_direct_method [Accessed].
WONNACOTT, T. H. & WONNACOTT, R. J. 1981. *Regression : a second course in statistics,* New York, Wiley.
WU, Z., REYNOLDS, A. C. & OLIVER, D. S. 1999. Conditioning Geostatistical Models to Two-Phase Production Data. *SPE Journal,* 4**,** 142-155.
YANG, C., NGHIEM, L. X., CARD, C. & BREMEIER, M. 2007. Reservoir Model Uncertainty Quantification Through Computer-Assisted History Matching. *SPE Annual Technical Conference and Exhibition.* Anaheim, California, U.S.A.: Society of Petroleum Engineers.
YANG, P. H. & WATSON, A. T. 1988. Automatic History Matching With Variable-Metric Methods. *SPE Reservoir Engineering,* 3**,** 995-1001.
YAOCHU, J., OLHOFER, M. & SENDHOFF, B. Year. Managing approximate models in evolutionary aerodynamic design optimization. *In:* Evolutionary Computation, 2001. Proceedings of the 2001 Congress on, 2001 2001. 592-599 vol. 1.
YOUSEF, A. A., GENTIL, P. H., JENSEN, J. L. & LAKE, L. W. 2006. A Capacitance Model To Infer Interwell Connectivity From Production and Injection Rate Fluctuations. *SPE Reservoir Evaluation & Engineering,* 9**,** pp. 630-646.
ZHANG, D., LU, Z. & CHEN, Y. 2007. Dynamic Reservoir Data Assimilation With an Efficient, Dimension-Reduced Kalman Filter. *SPE Journal.*
ZHANG, F., REYNOLDS, A. C. & OLIVER, D. S. 2002. Evaluation of the reduction in uncertainty obtained by conditioning a 3D stochastic channel to multiwell pressure data. *Mathematical geology,* 34**,** 715-742.
ZHANG, F., REYNOLDS, A. C. & OLIVER, D. S. 2003a. An Initial Guess for the Levenberg–Marquardt Algorithm for Conditioning a Stochastic Channel to Pressure Data. *Mathematical geology,* 35**,** 67-88.
ZHANG, F., SKJERVHEIM, J.-A., REYNOLDS, A. C. & OLIVER, D. S. 2005a. Automatic History Matching in a Bayesian Framework, Example Applications. *SPE Reservoir Evaluation & Engineering,* 8.
ZHANG, F., SKJERVHEIM, J., REYNOLDS, A. & OLIVER, D. 2005b. Automatic history matching in a Bayesian framework, example applications. *SPE Reservoir Evaluation & Engineering,* 8**,** 214-223.
ZHANG, F., SKJERVHEIM, J. A., REYNOLDS, A. C. & OLIVER, D. S. 2003b. Automatic History Matching in a Bayesian Framework, Example Applications. *SPE Annual Technical Conference and Exhibition.* Denver, Colorado: Society of Petroleum Engineers.
ZIELONKA, A., HETMANIOK, E. & SŁOTA, D. 2011. Using the Artificial Bee Colony Algorithm for Determining the Heat Transfer Coefficient. Springer Berlin / Heidelberg.
ZITZLER, E. & THIELE, L. 1999. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *Evolutionary Computation, IEEE Transactions on,* 3**,** 257-271.
ZUBAREV, D. I. 2009. Pros and Cons of Applying Proxy-Models as a Substitute for Full Reservoir Simulations. *SPE Annual Technical Conference and Exhibition.* New Orleans, Louisiana: Society of Petroleum Engineers.
ZYDALLIS, J., VAN VELDHUIZEN, D. & LAMONT, G. 2001. A Statistical Comparison of Multiobjective Evolutionary Algorithms Including the MOMGA-II
Evolutionary Multi-Criterion Optimization. *In:* ZITZLER, E., THIELE, L., DEB, K., COELLO COELLO, C. & CORNE, D. (eds.). Springer Berlin / Heidelberg.