View metadata, citation and similar papers at core.ac.uk

brought to you by CO

provided by King Fahd University of Petroleum and Min

A GLOBAL OPTIMIZATION APPROACH TO THE GRADUAL DEFORMATION METHOD OF HISTORY MATCHING

BY

ADEWALE WASIU ADENIJI

A Thesis Presented to the
DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

PETROLEUM ENGINEERING

MAY, 2015

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
DHAHRAN 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This thesis, written by **ADEWALE WASIU ADENIJI** under the direction of his thesis adviser and approved by his thesis committee, has been presented to and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE IN DEPARTMENT OF PETROLEUM ENGINEERING**.

Thesis Committee

Dr. Abeeb Adebowale Awotunde (Adviser)

Dr. Abdulaziz Albdulraheem (Member)

Dr. Abdullah Abdulaziz Al-Shuhail (Member)

Dr. Abdullah Sultan
Department Chairman

Dr. Salam A. Zummo
Dean of Graduate Studies

25/6/15
Date

©Adewale Wasiu Adeniji
2015

This research work is dedicated to Allah and the human race.

ACKNOWLEDGMENTS

My utmost gratitude goes to Allah for sparing my life and making my masters program a success. I appreciate the support of my wife, Zainab Cynthia Adeniji and other family members. I say a big thank you to my advisor, Dr. Abeeb Awotunde for believing in me even when I did not believe in myself. I also appreciate my thesis committee members, Dr. Abdulaziz Abdulraheem and Dr. Abdullah Al-Shuhail for their technical supports. Thumps up to Madhar Sahib for his candid advice about pursuing my research in the area of reservoir engineering. My appreciation is incomplete without thanking Lateef Kareem a.k.a Laton Spicy and Najmudeen Sibaweihi for their immense contributions towards the success of this study. Kudos to all friends that have contributed positively in one way or the other to make my life before and during the master program easy.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS **v**

TABLE OF CONTENTS **vii**

LIST OF TABLES **x**

LIST OF FIGURES **xi**

LIST OF ABBREVIATIONS **xiv**

ABSTRACT (ENGLISH) **xix**

CHAPTER 1 INTRODUCTION **1**

1.1 General Overview of History Matching 1

1.2 Problem Statement 4

1.3 Significance of the Study 5

1.4 Research Objectives 5

1.5 Thesis Organization 6

CHAPTER 2 LITERATURE REVIEW **7**

2.1 Gradual Deformation Method 7

2.1.1 Fundamental Principles of GDM 8

2.1.2 Forming Chains of Realizations 9

2.1.3 Speed of Convergence 13

2.2 Applications of GDM in History Matching 14

2.3 Gradient-based Algorithms 15
  2.3.1 Steepest Descent Algorithm 17
  2.3.2 Gauss-Newton Algorithm 20
  2.3.3 Levenberg-Marquardt Algorithm 23
2.4 Global Optimization Algorithms 26
  2.4.1 Differential Evolution Algorithm 27
  2.4.2 Particle Swarm Optimization Algorithm 33
  2.4.3 Simulated Annealing Algorithm 35

CHAPTER 3 METHODOLOGY **38**
3.1 Geostatistical Modeling 38
  3.1.1 Data Analysis 38
  3.1.2 Kriging 44
  3.1.3 Stochastic Simulation 45
3.2 Reservoir Modeling 46
  3.2.1 Reservoir Discretization 46
  3.2.2 Reservoir Rock Properties 47
  3.2.3 Reservoir Fluid Properties 47
  3.2.4 Reservoir Equilibration 52
  3.2.5 Well and Completion Specifications 53
  3.2.6 Well Constraints for Producers 55
  3.2.7 Well Constraints for Injectors 55
  3.2.8 Time Step and Report Time 55
  3.2.9 History Production Data 55
3.3 Implementation of History Matching 56
  3.3.1 Realizations 57
  3.3.2 Sample Application of GDM in History Matching 57
  3.3.3 Deformation Parameters 58
  3.3.4 Objective Function 59
  3.3.5 Sample Application of LM in GDM 59

3.3.6 Sample Application of DE in GDM 63
3.3.7 Sample Application of PSO in GDM 66
3.3.8 Sample Application of SA in GDM 68

CHAPTER 4 RESULTS AND DISCUSSION 72
4.1 Kriging 72
4.2 Stochastic Simulation 75
4.3 Reservoir Modeling 83
4.3.1 Well Locations 83
4.3.2 Reservoir Permeability and Porosity 84
4.4 History Matching 86
4.4.1 Objective Function Decay 86
4.4.2 Water Cut Match 88
4.4.3 Estimated Reservoir Permeability 108
4.5 Evaluation Criteria 113

CHAPTER 5 CONCLUSION AND RECOMMENDATION 117
5.1 Conclusion 117
5.2 Recommendation 118

REFERENCES 119

VITAE 126

LIST OF TABLES

3.1 Permeability and porosity measured from wells. 39
3.2 Variogram parameters. 44
3.3 Rock compressibility. 47
3.4 PVT properties of live oil. 47
3.5 Fluid density. 49
3.6 PVT properties of dry gas. 50
3.7 PVT properties of water. 50
3.8 Oil-water saturation. 51
3.9 Oil-gas saturation. 52
3.10 Reservoir initial state. 53
3.11 Variation of initial solution gas-oil ratio with depth. 53
3.12 Well information. 54

LIST OF FIGURES

2.1 Flow diagram of GDM. 9

2.2 Flow diagram of SD algorithm. 20

2.3 Flow diagram of GN algorithm. 22

2.4 Flow diagram of LM algorithm. 25

2.5 Flow diagram of DE algorithm. 29

2.6 Flow diagram of PSO algorithm. 35

2.7 Flow diagram of SA algorithm. 37

4.1 Porosity kriging. 73

4.2 Porosity kriging variance. 73

4.3 Permeability kriging. 74

4.4 Permeability kriging variance. 74

4.5 Permeability realization 1. 75

4.6 Permeability realization 2. 76

4.7 Permeability realization 3. 76

4.8 Permeability realization 4. 77

4.9 Permeability realization 5. 77

4.10 Permeability realization 6. 78

4.11 Permeability realization 7. 78

4.12 Permeability realization 8. 79

4.13 Permeability realization 9. 79

4.14 Permeability realization 10. 80

4.15 Permeability realization 11. 80

4.16 Permeability realization 12. 81
4.17 Permeability realization 13. 81
4.18 Permeability realization 14. 82
4.19 Permeability realization 15. 82
4.20 Planar view of wells locations and porosity distribution. 83
4.21 Planar view of wells locations and permeability distribution. . . . 84
4.22 Reservoir true permeability distribution. 85
4.23 Reservoir true porosity distribution. 85
4.24 Best OF decay. 87
4.25 Median OF decay. 87
4.26 Worst OF decay. 88
4.27 Best match of water cut from Producer 1. 89
4.28 Median match of water cut from Producer 1. 89
4.29 Worst match of water cut from Producer 1. 90
4.30 Best match of water cut from Producer 2. 91
4.31 Median match of water cut from Producer 2. 91
4.32 Worst match of water cut from Producer 2. 92
4.33 Best match of water cut from Producer 3. 93
4.34 Median match of water cut from Producer 3. 93
4.35 Worst match of water cut from Producer 3. 94
4.36 Best match of water cut from Producer 4. 95
4.37 Median match of water cut from Producer 4. 95
4.38 Worst match of water cut from Producer 4. 96
4.39 Best match of water cut from Producer 5. 97
4.40 Median match of water cut from Producer 5. 97
4.41 Worst match of water cut from Producer 5. 98
4.42 Best match of water cut from Producer 6. 99
4.43 Median match of water cut from Producer 6. 99
4.44 Worst match of water cut from Producer 6. 100
4.45 Best match of water cut from Producer 7. 101

4.46 Median match of water cut from Producer 7. 101
4.47 Worst match of water cut from Producer 7. 102
4.48 Best match of water cut from Producer 8. 103
4.49 Median match of water cut from Producer 8. 103
4.50 Worst match of water cut from Producer 8. 104
4.51 Best match of water cut from Producer 9. 105
4.52 Median match of water cut from Producer 9. 105
4.53 Worst match of water cut from Producer 9. 106
4.54 Best match of water cut from Producer 10. 107
4.55 Median match of water cut from Producer 10. 107
4.56 Worst match of water cut from Producer 10. 108
4.57 L1 norm of estimated reservoir permeability distribution. 109
4.58 Optimum estimated permeability field from GDM + LM. 110
4.59 Optimum estimated permeability field from GDM + DE. 111
4.60 Optimum estimated permeability field from GDM + PSO. 112
4.61 Optimum estimated permeability field from GDM + SA. 113
4.62 Measure of effectiveness of the optimizers. 114
4.63 Measure of efficiency of the optimizers. 115
4.64 Measure of reliability of the optimizers. 116

LIST OF ABBREVIATIONS

b_1 and b_2	Wolfe condition constants
bbl/stb	Reservoir barrel per stock tank barrel
B_g	Gas formation volume factor
B_o	Oil formation volume factor
B_w	Water formation volume factor
c_1	Cognitive parameter
c_2	Social parameter
cp	Centipoise
C_r	Rock compressibility
C_R	Crossover probability
C_w	Water compressibility
$defs$	Deformation stage number
dx	Constant change in $\vec{x}$
$\vec{d}_{cal}$	Simulated data
$\vec{d}_{meas}$	Measured or observed data (Noised simulated water cut)

$\vec{d}_{ycal}$	Simulated data from perturbed deformation parameters
D	Number of unknown parameters
f	Forward model function
F	Differential weight
g	Gradient of objective function
H	Newton Hessian
H_{SD}	Steepest descent Hessian
H_{GN}	Gauss-Newton Hessian
H_{LM}	Levenberg-Marquardt Hessian
i	Counter from 1 to N_p
I	Identity matrix
Ib/ft^3	Pound per cubic feet
$iter$	Iteration number
j	Counter from 1 to D
$jrand$	Randomly generated number from 1 to D
$\vec{k}$	Gaussian realization
$\vec{K}$	Standardized Gaussian realization
K_{rg}	Gas relative permeability
K_{rog}	Oil-gas relative permeability
K_{rw}	Water relative permeability
K_{row}	Oil-water relative permeability
k_x	Permeability in x-direction

$log_{10}(k_x)$	Logarithm to base 10 of k_x
$\vec{m}$	Mean of Gaussian realization
M	Number of grid cells in the reservoir
$Mscf/stb$	Million standard cubic feet per stock tan barrel
n	Number of Gaussian realizations of prior model
N	Number of data points in the match parameter (water cut)
Np	Number of population members
p	Directional derivative of gradient
$\vec{p}$	Particle's best position
P	Pressure
P_{cog}	oil-gas capillary pressure
P_{cow}	oil-water capillary pressure
P_{ref}	Pressure at reference depth
$psia$	Pound per squared inch
$\vec{q}$	Swarm's best position
$r1, r2, r3, r4,, r5$	Mutually exclusive integers randomly chosen from 1 to N_p
$\vec{r}_1$, $\vec{r}_2$	Randomly-generated numbers between 0 and 1
R_{so}	Solution gas-oil ratio
$\vec{s}$	Step size
$\vec{S}$	Sensitivity coefficients
S_g	Gas saturation
S_m	Sensitivity matrix

stb/day	Stock tank barrel per day
S_w	Water saturation
$\vec{v}$	Particle's velocity
w	Weight
$\vec{x}$	Deformation parameters in gradual deformation method
$\vec{x}_0$	Initial guess of deformation parameters
$\vec{x}_{opt}$	Optimum deformation parameters
x_{best}	The candidate solution with the least objective function
X	X-coordinates of sampled wells
$\vec{y}$	Trial vector
Y	Y-coordinates of sampled wells
Z	Z-coordinates of sampled wells
$\delta\vec{x}$	Descent direction
γ	Annealing parameter
$\Delta\Phi$	Difference between the new and old Objective function
μ_g	Gas viscosity
μ_o	Oil viscosity
ρ_o	Oil density
ρ_w	Water density
Φ	Objective function
$\vec{\nu}$	Mutant vector
ρ_g	Gas density

$\alpha_1,\alpha_2\ ,\alpha_3$	Coefficients in gradual deformation method
$\vec{\sigma}$	Standard deviation of Gaussian realization
ν_w	Water viscosibility

THESIS ABSTRACT

NAME:	Adewale Wasiu Adeniji
TITLE OF STUDY:	A Global Optimization Approach to the Gradual Deformation Method of History Matching
MAJOR FIELD:	Department of Petroleum Engineering
DATE OF DEGREE:	May, 2015

Due to large uncertainty and scarcity of hard data used to build a reservoir simulation model, geostatistically simulated estimates of reservoir parameters do not usually produce data that match the observed data. Such model cannot be used for forecasting since the model cannot present the actual behavior of the reservoir. The model parameters need to be modified until the model can at least reproduce the measured data through a process termed history matching. History matching may be done manually or automated (using optimization algorithms). Stochastic or gradient-based algorithms may be used. Recently, gradual deformation method (GDM) was proposed to constrain history matching to simple statistics so that important geologic information can be preserved. GDM estimates high number of reservoir parameters using few deformation parameters. An optimiza-

tion algorithm is required to optimize these deformation parameters. Traditionally, gradient-based algorithms are used for this purpose but they have challenges of local minimum entrapment. This study proposes the use of global optimization algorithms to estimate the deformation parameters in GDM. In doing this, three global algorithms namely differential evolution, particle swarm optimization, and simulated annealing algorithms were used to estimate the deformation parameters. For fair comparison with the gradient-based algorithm, Levenberg-Marquardt algorithm was also used to estimate the deformation parameters. The use of global optimization algorithms is aimed at minimizing the challenges of the gradient-based algorithms. A 3D synthetic reservoir comprising sixty-thousand grid cells was used in this study. Measured water cut was matched to estimate the reservoir permeability distribution. The reservoir was divided into sixty sub-regions with each region assigned a deformation parameter. Hence, sixty-thousand unknowns were reduced to sixty. From the solutions of the seven different realizations considered in this study, the performances of the algorithms were evaluated. In all the seven realizations, the global optimization algorithms outperformed the gradient-based algorithm with particle swarm optimization algorithm emerging the most effective, most efficient and most reliable amongst all the algorithms considered in this study.

مستخلص الرسالة

الإسم: أديوالي واسيو أدينيجي
عنوان الرسالة: النهج الأمثل العالمي إلى أسلوب تشويه التدريجي من التاريخ مطابقة.
التخصص: قسم هندسة البترول
التاريخ: مايو ، 2015

ونظرا لعدم اليقين الكبير وندرة البيانات الصلبة المستخدمة لبناء نموذج محاكاة المكامن، تقديرات محاكاة من المعلمات الخزان عادة لا تنتج البيانات التي تتطابق مع البيانات المرصودة. هذا النموذج لا يمكن استخدامها للتنبؤ منذ نموذج لا يمكن أن تقدم على السلوك الفعلي للخزان. تحتاج المعلمات نموذج إلى تعديل حتى يمكن للنموذج على الأقل استخراج البيانات المقاسة من خلال عملية تسمى التاريخ مطابقة. ويمكن أن يتم مطابقة التاريخ يدويا أو آليا (باستخدام خوارزميات الأمثل). مؤشر ستوكاستيك أو خوارزميات القائم على التدرج ويمكن استخدام. في الآونة الأخيرة، واقترح طريقة تشوه التدريجي (GDM) لتقييد التاريخ مطابقة لإحصاءات بسيطة بحيث يمكن الحفاظ المعلومات الجيولوجية الهامة. وتقدر GDM عدد كبير من المعلمات الخزان باستخدام عدد قليل من المعالم تشوه. مطلوب خوارزمية الأمثل لتحسين هذه المعايير تشوه. تقليديا، يتم استخدام الخوارزميات القائم على التدرج لهذا الغرض ولكن لديهم تحديات الحد الأدنى انحباس المحلي.

وتقترح هذه الدراسة استخدام خوارزميات التحسين العالمية لتقدير المعلمات تشوه في GDM. في القيام بذلك، ثلاثة خوارزميات العالمية وهي الفرق التطور، سرب الجسيمات الأمثل، وكانت تستخدم خوارزميات الصلب محاكاة لتقدير المعلمات تشوه. وعلى سبيل المقارنة عادلة مع الخوارزمية القائم على التدرج، وكان يستخدم ليبرغ-ماركوارت خوارزمية أيضا لتقدير المعلمات تشوه. ويهدف استخدام خوارزميات التحسين العالمية في التقليل من تحديات الخوارزميات القائم على التدرج. تم استخدام خزان صناعي 3D تضم والستين ألف خلايا الشبكة في هذه الدراسة. وواكب انقطاع المياه قياسها لتقدير التوزيع نفاذية الخزان. تم تقسيم الخزان إلى مناطق فرعية والستين مع كل منطقة تعيين المعلمة تشوه. وبالتالي، خفضت والستين ألف المجهولة إلى ستين. من الحلول من سبعة انجازاتهم مختلفة تعتبر في هذه الدراسة تم تقييم أداء الخوارزميات. في جميع انجازاتهم سبعة، تفوقت خوارزميات الأمثل العالمية الخوارزمية القائم على التدرج مع خوارزمية الأمثل سرب الجسيمات الناشئة الأكثر فعالية، الأكثر كفاءة والأكثر موثوقية بين جميع خوارزميات بحثها في هذه الدراسة.

CHAPTER 1

INTRODUCTION

1.1 General Overview of History Matching

Large portions of hydrocarbon reservoirs are inaccessible. Hence, hard data are acquired with the use of sophisticated equipment to build a reservoir model that can simulate the reservoir behavior. The model is built by integrating data from well logs, well test analysis, core samples, seismic surveys, pressure-volume-temperature (PVT) analysis etc. Also, during the time of producing from the reservoir, data such as water, oil and gas rates, bottom-hole flowing pressure, water cut are measured and archived as production data. Seismic responses are also obtained at exactly the same location at two different times with usually the same equipment to generate reliable time-lapse seismic data. The time-lapse seismic data are then interpreted as pressure and saturation changes in the reservoir between the two different measurement times. Both the production and time-lapse seismic data constitute what is generally known as historic data. In reality, when the reservoir model is used to generate production data, the historic data is not reproduced. This implies that the model does not present the actual behavior of the reservoir (that is, the measured data are not the actual properties of the reservoir). This is not unconnected to the fact that there could be noise in the data. Also, interpolation errors are unavoidable if the data are not representative of the reservoir as they are usually taken at few locations in the reservoir. The few data are then populated over the entire reservoir using geostatistical algorithms.

In order to have a model that can reproduce the historic data, the model parameters need to be adjusted. The process of adjusting the latter to honor the historic data is regarded as history matching. It is a type of ill-posed inverse problems because the problem is strongly under-determined due to large number of the unknown parameters compared to few measured data. It also has an infinite number of solutions which all honor the measured data equally well. The parameters (for instance, permeability and porosity) of the model that reproduces the historic data are not necessarily the actual properties of the reservoir. But, this model is more reliable than the one that do not reproduce the historic data and have better predictive abilities since it can at least behave like the reservoir. The essence of history matching is to have a valid model that can be used to forecast future productions. This is necessary for efficient and effective reservoir management and development.

History matching was originally done manually by experienced reservoir engineers for relatively small reservoirs. They adjust some of sensitive properties of the reservoirs by trial and error until an acceptable history match is achieved. For large reservoirs, it is done automated using optimization algorithms which could either be stochastic or gradient-based. Stochastic algorithms includes particle swarm optimization (PSO) algorithm, differential evolution (DE) algorithm, covariance matrix adaptive evolution strategy (CMA-ES), genetic algorithm (GA), simulated annealing (SA) algorithm, ant colony, neighborhood algorithm, etc. Gradient-based algorithms include Levenberg-Marquardt (LM) algorithm, conjugate gradient algorithm, steepest descent (SD) algorithm, Gauss-Newton (GN) algorithm, Newton's method, etc.

The solutions from all the mentioned algorithms may not preserve the geological structure of the reservoir. In the late 90s, a parameterization technique known as gradual deformation method was proposed. This technique helps to constrain the reservoir model to hard data in order to preserve the geological structure of the reservoir while reproducing the historic data. Gradual deformation method (GDM) preserves the geological structure of the reservoir by ensuring that the spatial variability (mean, variance and histogram) of the prior and posterior models are the same. Another pro of GDM is that GDM reduces the dimensionality of the inverse problem to few deformation parameters. These deformation parameters are conventionally estimated with gradient-based algorithms. Although, gradient-based algorithms give fast convergence, they are local and can thus find only a limited region of the problem space.

In this study, we proposed global optimization algorithms as alternatives to gradient-based algorithms in estimating deformation parameters in GDM for better history matching performance. Water cut data from all the producers in the 3D synthetic reservoir model were matched to estimate the reservoir permeability distribution. With the use of GDM, the dimensionality of our inverse problem was reduced from six thousand to sixty. This concept of dimensionality reduction can also be applied to real reservoir. The performances of the algorithms were evaluated to determine the most effective, the most efficient and the most reliable algorithm.

1.2 Problem Statement

Uncertainties in reservoir parameters estimation are unavoidable. One of which is why a newly built reservoir model does not honor the historic data of the same reservoir. Nonetheless, forecast and vital decisions need to be made to efficiently and effectively manage and develop the reservoir. History matching is aimed at providing some level of confidence in the reservoir model by iteratively adjusting the model until the simulated production data matches the historic data. The deformation parameters in GDM are conventionally estimated with gradient-based algorithms. But, gradient-based algorithms have inherent shortcoming of being trapped in the local minima. Hence, global optimization algorithms are proposed as alternatives to gradient-based algorithms. In this study, LM algorithm was selected as the gradient-based algorithm. The choice of LM algorithm was based on the fact that it is one of the most widely used gradient-based algorithms because of its rapid convergence and suitability for estimating few unknown parameters. Three global optimization algorithms namely DE, PSO and SA algorithms were used in this research.

1.3 Significance of the Study

The use of global optimization algorithms in GDM is aimed at minimizing the challenge of local minimum entrapment peculiar to the gradient-based algorithms. As the history matching performance gets better, the reliability and robustness of reservoir models get improved. Once a reliable and robust model is developed, production forecasts can be performed and consequently, critical decisions about infill drilling, work-over, production/injection rate, surface facilities procurement etc. can be rightly made. The proper execution of these decisions will help minimize expenses and maximize profits.

1.4 Research Objectives

The objectives of this study are to:

1. Optimize GDM parameters using a gradient-based algorithm (LM).
2. Optimize GDM parameters using three different global optimization algorithms (DE, PSO and SA) separately.
3. To replace the gradient-based algorithm in GDM with a global optimization algorithm.

4. To evaluate the performances of the three global optimization algorithms embedded in GDM against the performance of the LM used in GDM.

1.5 Thesis Organization

This thesis comprises five chapters. Chapter 1 gives the general overview of history matching, problem statement, significance of study and research objectives. Chapter 2 presents the review of existing literatures on GDM, its applications in history matching, and the concepts of gradient-based and global optimization algorithms. Chapter 3 presents how DE, PSO, SA and LM were applied in GDM to history match the historic water cut data to estimate the synthetic reservoir permeability distribution. Chapter 4 presents the results of how the objective function (OF) decayed as the historic water cut was matched by the simulated water cut from all the algorithms in the seven different realizations considered in this study. Chapter 5 gives the conclusion of this study as well as recommendations for future research studies.

CHAPTER 2

LITERATURE REVIEW

In this chapter, existing literatures related to GDM, history matching and optimization algorithms were reviewed. The different studies on GDM and its applications in history matching are presented. Also, the concepts of gradient-based and global optimization algorithms are discussed.

2.1 Gradual Deformation Method

The gradual deformation method (GDM) was proposed in 1998 [1]. It was initially developed for gradually changing Gaussian-related stochastic reservoirs models while preserving their spatial variability. Then, GDM was extended to non-Gaussian reservoir models simulated from sequential indicator and Boolean algorithms [2]. GDM is a method for gradually deforming continuous geostatistical models to generate reservoir models which honor historic production data. Starting from an initial geological model that does not match history, GDM allows to gradually change the initial model without compromising the geological continuity of the reservoir model, until a history match is achieved [3].

2.1.1 Fundamental Principles of GDM

GDM is a parameterization technique that aims at generating continuous perturbations of a prior model, so that the resulting posterior model can match the history data better. It also ensures that the statistical variability of the prior and posterior models are the same. GDM significantly reduces the dimensionality of unknown parameter space of stochastic models to few deformation parameters. Irrespective of the number of grid cells in the reservoir model, it is modified by varying the parameters. This method preserves the properties of the stochastic model due to the fact that linear combinations of multi-Gaussian random functions are still multi-Gaussian random functions [4].

The concept of GDM is described in Figure 2.1.

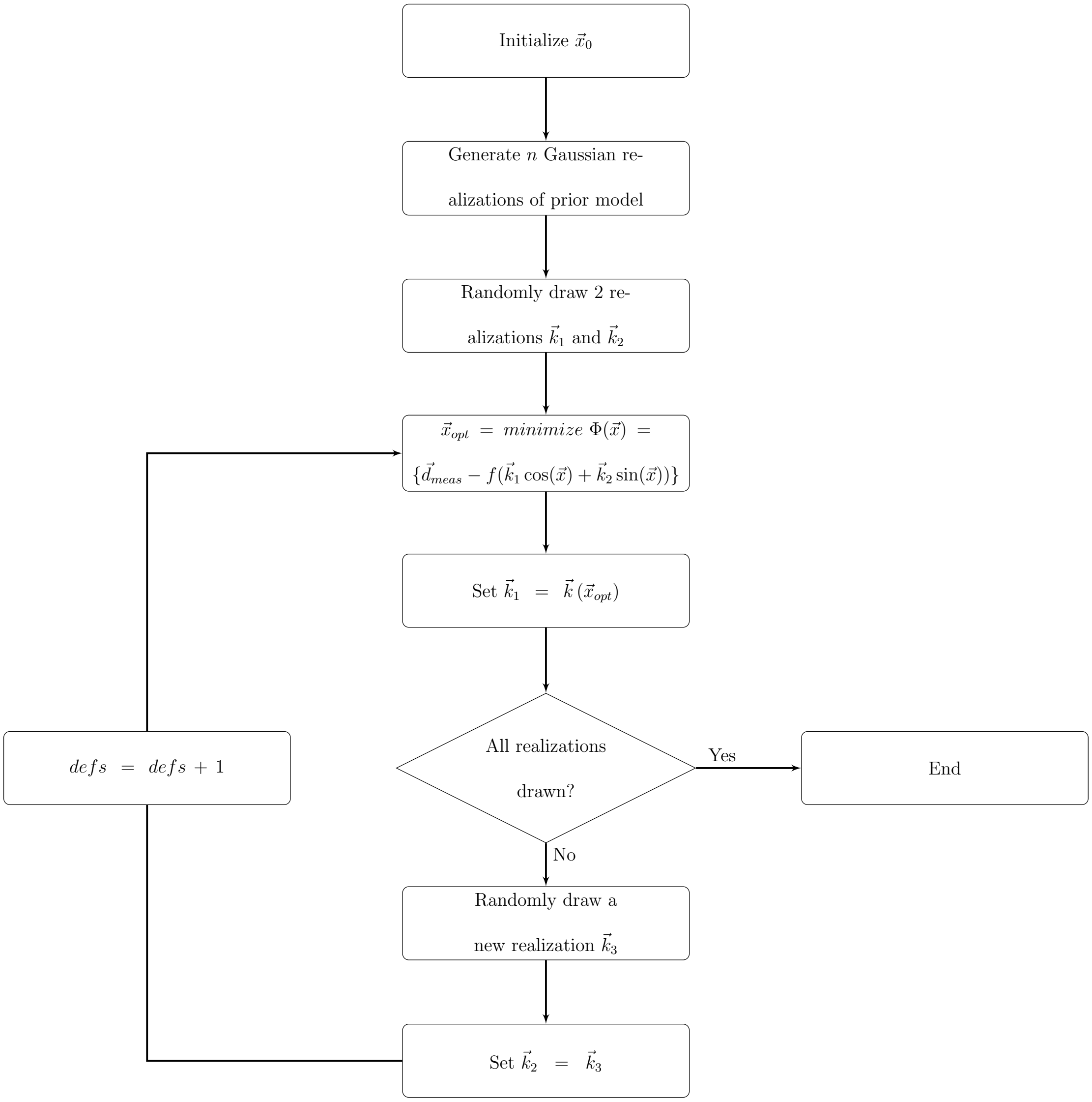

Initialize $\vec{x}_0$
Generate n Gaussian realizations of prior model
Randomly draw 2 realizations $\vec{k}_1$ and $\vec{k}_2$
$\vec{x}_{opt} = minimize\ \Phi(\vec{x}) = \{\vec{d}_{meas} - f(\vec{k}_1 \cos(\vec{x}) + \vec{k}_2 \sin(\vec{x}))\}$
Set $\vec{k}_1 = \vec{k}(\vec{x}_{opt})$
All realizations drawn?
Yes
End
No
Randomly draw a new realization $\vec{k}_3$
Set $\vec{k}_2 = \vec{k}_3$
$defs = defs + 1$

Figure 2.1: Flow diagram of GDM.

2.1.2 Forming Chains of Realizations

Realizations of posterior model are obtained by perturbing those of the prior model. The perturbation is achieved by the linear combination of the prior model realizations. After several linear combinations, chains of realizations are formed.

This helps to achieve faster convergence and better history matching performance. Any of the three methods below can be used to linearly combine realizations:

1. Unconditional method

   Two or more independent standard Gaussian realizations, $\vec{k}_1$ and $\vec{k}_2$ may be unconditionally combined linearly as follows:

   $$\vec{k} = \alpha_1 \vec{k}_1 + \alpha_2 \vec{k}_2 \tag{2.1}$$

   Posterior model, $\vec{k}$ has the same statistical properties (mean, variance, variogram, histogram etc.) as the realizations of the prior models, $\vec{k}_1$ and $\vec{k}_2$. The statistical preservation is achieved by ensuring that:

   $$\alpha_1{}^2 + \alpha_2{}^2 = 1 \tag{2.2}$$

2. Conditional method

   In reality, the linear combination in Equation 2.1 does not preserve the statistical properties of the prior model in the posterior model when there is hard conditioning data (for instance, production data). Therefore, prior geological knowledge about the reservoir structure is expressed as a geostatistical constraint that must be integrated into the conditioning process [2].

To solve this problem, Ying and Gomez (2000) proposed to linearly combine three independent realizations as shown below [5]:

$$\vec{k} = \alpha_1 \vec{k}_1 + \alpha_2 \vec{k}_2 + \alpha_3 \vec{k}_3 \tag{2.3}$$

Under the joint conditions that:

$$\alpha_1 + \alpha_2 + \alpha_3 = 1 \tag{2.4}$$

$$\alpha_1{}^2 + \alpha_2{}^2 + \alpha_3{}^2 = 1 \tag{2.5}$$

This guarantees that the posterior model, $\vec{k}$ honors the hard data as well as the variogram. The coefficients can easily be re-parameterized to a single parameter, x with the following equations [4]:

$$\alpha_1 = \frac{1}{3} + \frac{2}{3} \cos(x) \tag{2.6}$$

$$\alpha_2 = \frac{1}{3} + \frac{2}{3}\sin\left(\frac{-\pi}{6} + x\right) \tag{2.7}$$

$$\alpha_3 = \frac{1}{3} + \frac{2}{3}\sin\left(\frac{-\pi}{6} - x\right) \tag{2.8}$$

3. Dependent realizations method

Most studies on GDM combine independent realizations. However, Hu (2002) investigated combining dependent realizations to improve the numerical stability of GDM and derived a new formulation that can combine dependent realizations. Posterior models only preserve the statistical properties (mean, variance, covariance etc.) of the stochastic (prior) model theoretically. They are never perfect due to numerical fluctuations; they never have the exact mean, variance and covariance of the prior model, and are never completely independent of each other. To eliminate this challenge, their standardized realizations (prior models) were proposed to be combined instead of directly combining the realizations [6].

$$\vec{k} = \vec{k}_1 \cos(x) + \vec{k}_2 \sin(x) \tag{2.9}$$

Their standardized realizations should be combined as thus:

$$\vec{K} = \frac{\vec{k}_1 - \vec{m}}{\vec{\sigma}} \cos(x) + \frac{\vec{k}_2 - \vec{m}}{\vec{\sigma}} \sin(x) \tag{2.10}$$

By this, the mean of the realization is then exactly zero and its variance is 1.

2.1.3 Speed of Convergence

GDM is known for slow convergence because it reduces large dimensionality of a stochastic model to few parameters. The convergence of the OF when using GDM was investigated by Le Ravalec *et al.* (2000) [4]. They mentioned that the optimization process with GDM uses a hybrid of a random search scheme and gradient-based computations. They proposed that OF exhibits an exponential convergence rate shown in Equation 2.11.

$$\Phi(x) \;\propto\; \exp\left(\frac{-n}{M}\right) \tag{2.11}$$

Where $\Phi(x)$ is the OF, n is number of realization chain (number of times, realizations have been linearly combined), M is the number of grid cells in the reservoir model.

The convergence rate was experimentally found to increase when the reservoir is divided into sub-regions [2]. A deformation parameter is attributed to every sub-region, this helps to reduce components associated to a single deformation parameter. If the reservoir model is divided into S sub-regions with the same number of grid blocks, the convergence rate becomes:

$$\Phi(x) \;\propto\; \exp\left(\frac{-nS}{M}\right) \tag{2.12}$$

The greater the number of combined realizations, the smaller the OF. Furthermore, the convergence rate is strongly influenced by the number of combined realizations. If $(S+1)$ realizations are combined to form a chain, that is, the number of deformation parameters is S, then, the convergence rate is given by:

$$\Phi(x) \;\propto\; \exp\left(\frac{-nS}{M}\right) \tag{2.13}$$

Apparently, combining $(S+1)$ realizations or dividing the reservoir into S sub-regions is equivalent in terms of convergence. When n is larger than M, the OF becomes negligible with respect to M and decreases exponentially.

2.2 Applications of GDM in History Matching

GDM was combined with multiple-point geostatistics and a fast streamline simulator to achieve acceptable history matching with limited amount of flow simulations [3]. Mezghani and Roggero (2001) applied GDM to directly update fine-scale geostatistical reservoir models during history matching process. The deformation parameters were estimated using an undisclosed gradient-based algorithm [9]. History matching was achieved by combining streamline-based approach with GDM. Gauss-Newton algorithm was used in GDM to estimate the deformation parameters [10]. Busby *et al.* (2009) combined GDM with adaptive response surface methodology to history match production data [11]. They used an undisclosed optimization algorithm used to estimate the deformation parameters. Reis *et al.* (2000) applied GDM to history match the production data from PBR oil field (Offshore Brazil) by building fine scale geostatistical lithofacies model using the non-stationary truncated Gaussian approach. The deformation parameters were manually selected [12] without using any optimization algorithm. Le Gallo *et al.* (2000) proposed a new history matching methodology based on GDM to constrain 3D geostatistical reservoir models to well and production data [13]. Gervais *et al.* (2007) applied local GDM to history production data from the PBR oil field (Offshore Brazil). They used an undisclosed gradient-based optimization algorithm in GDM to estimate the deformation parameters [14].

2.3 Gradient-based Algorithms

Gradient-based algorithms are methods of solving optimization problems where the search directions are dependent on the gradient of the OF at a particular point. Unlike the global methods, they have fast convergence but are trapped in local minimum of the OF. Hence, they are poor in wide navigation of the search space. For problems with non-differentiable OF, gradient-based algorithms cannot be used. These methods include GN, SD, LM, conjugate gradient algorithms etc. For fast convergence, gradient-based methods computes sensitivity coefficients which are used to define the changes in the simulated data with respect to small variation in the model parameter being perturbed. However, some of these methods such as conjugate gradient and GN algorithms do not use sensitivity coefficients. As such, they converge slowly. The coefficients may be estimated in any of these ways: substitution, adjoint and forward sensitivity methods. Substitution method involves perturbing each element of the vector of the unknown parameter at a time and estimating its effect (sensitivity coefficients) on the simulated data. After all the elements in vector has been perturbed, all the coefficients are put together to obtain a sensitivity matrix. This method is simple to use but it is time consuming most especially when the vector of unknown parameters has many elements. Substitution method requires the OF to be computed $M + 1$ number of times. The adjoint and forward sensitivity methods require the estimation of the OF M and N number of times, respectively. Hence, the forward sensitivity method does not depend on the number of design parameters but rather on the number of match parameters [15]. The forward sensitivity method has been applied in history matching [16], [17], and [15]. The adjoint method was first used by Jacquard (1965) to estimate fourteen design parameters of a 2D single-phase transient flow model [18]. The adjoint method has also been used by researchers to compute sensitivity coefficients [19] and [20].

2.3.1 Steepest Descent Algorithm

Steepest descent (SD) algorithm also known as gradient descent algorithm was derived from the expansion of Taylor series truncated at first order. As such, SD algorithm is a first order optimization algorithm and simple to use because it does not require second-order derivatives. In this method, the search direction to minimize the OF is the opposite of the gradient of the OF at a particular point. Consider the expansion of a multi-dimensional optimization problem truncated at second-order approximation:

$$f(\vec{x} + \delta\vec{x}) \approx f(\vec{x}) + \delta\vec{x} g(\vec{x}) + \delta\vec{x}^T H \delta\vec{x} \tag{2.14}$$

For an OF given as:

$$\Phi(x) = \left(\vec{d}_{cal} - \vec{d}_{meas}\right)^T \left(\vec{d}_{cal} - \vec{d}_{meas}\right) \tag{2.15}$$

The gradient of the OF is:

$$g(\vec{x}) = {S_m}^T \left(\vec{d}_{cal} - \vec{d}_{meas}\right) \tag{2.16}$$

Sensitivity matrix is:

$$S_m = \frac{\partial \vec{d}_{cal}}{\partial \vec{x}} \tag{2.17}$$

By differentiating Equation 2.14 with respect to $\delta\vec{x}$, we obtain:

$$g(\vec{x}) + H\delta\vec{x} = \vec{0} \tag{2.18}$$

Equation 2.14 is the basis of Newton's method and the H is an exact Hessian given by the differentiation of the gradient of the OF with respect to $\vec{x}$. However, the Hessian used by SD algorithm is an approximated one and it is given by:

$$H_{SD} = ||g(\vec{x})||_2.I \tag{2.19}$$

By substituting H_{SD} in Equation 2.18, we obtain:

$$\vec{g} + (||g(\vec{x})||_2.I)\delta\vec{x} = \vec{0} \tag{2.20}$$

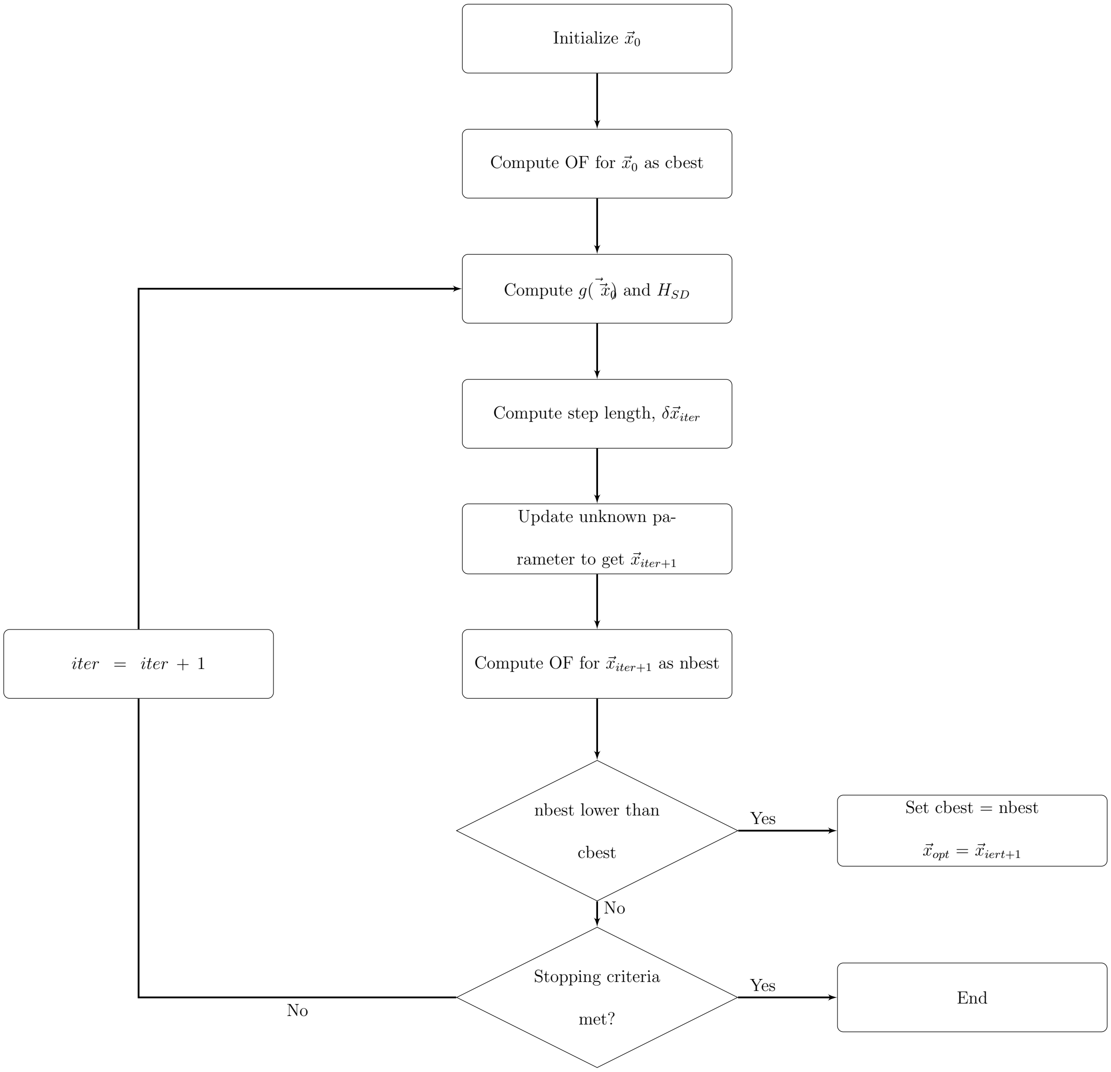
By solving the linear equation for $\delta\vec{x}$, we obtain:

$$\delta\vec{x} = \frac{g(\vec{x})}{||g(\vec{x})||_2} \tag{2.21}$$

The vector of unknown parameters is updated as:

$$\vec{x}_{iter+1} = \vec{x}_{iter} + \vec{\delta x}_{iter} \tag{2.22}$$

Procedural steps of SD algorithm are described in Figure 2.2.



Initialize $\vec{x}_0$
Compute OF for $\vec{x}_0$ as cbest
Compute $g(\vec{x}_0)$ and H_{SD}
Compute step length, $\delta\vec{x}_{iter}$
Update unknown parameter to get $\vec{x}_{iter+1}$
Compute OF for $\vec{x}_{iter+1}$ as nbest
$iter = iter + 1$
nbest lower than cbest
Yes
Set cbest = nbest
$\vec{x}_{opt} = \vec{x}_{iert+1}$
No
Stopping criteria met?
Yes
End
No

Figure 2.2: Flow diagram of SD algorithm.

2.3.2 Gauss-Newton Algorithm

Gauss-Newton (GN) algorithm is a modification of Newton's method used for solving non-linear least square optimization problems. Unlike the Newton's method, GN algorithm does not require the second derivative of OF. The modified method was named after the Mathematicians Friedrich Gauss and Isaac Newton who proposed it. Another drawback of Newton's method is the singularity of Hessian obtained from the second derivative of OF. To eliminate this, pre-conditioning is often done. By this, the GN algorithm is more efficient in terms of convergence than Newton's method. The Hessian of GN algorithm can be prevented from being ill-posed by having a good initial guess so that the eigen values can span a wide range of acceptable orders of magnitude [21]. The GN algorithm has a disadvantage of convergence control difficulty. Sometimes, the Hessian diagonal elements may be too small or big and would not give a good step size for the next iteration. This problem is usually eliminated by modifying the algorithm [22].

Procedural steps of GN algorithm are described in Figure 2.3.

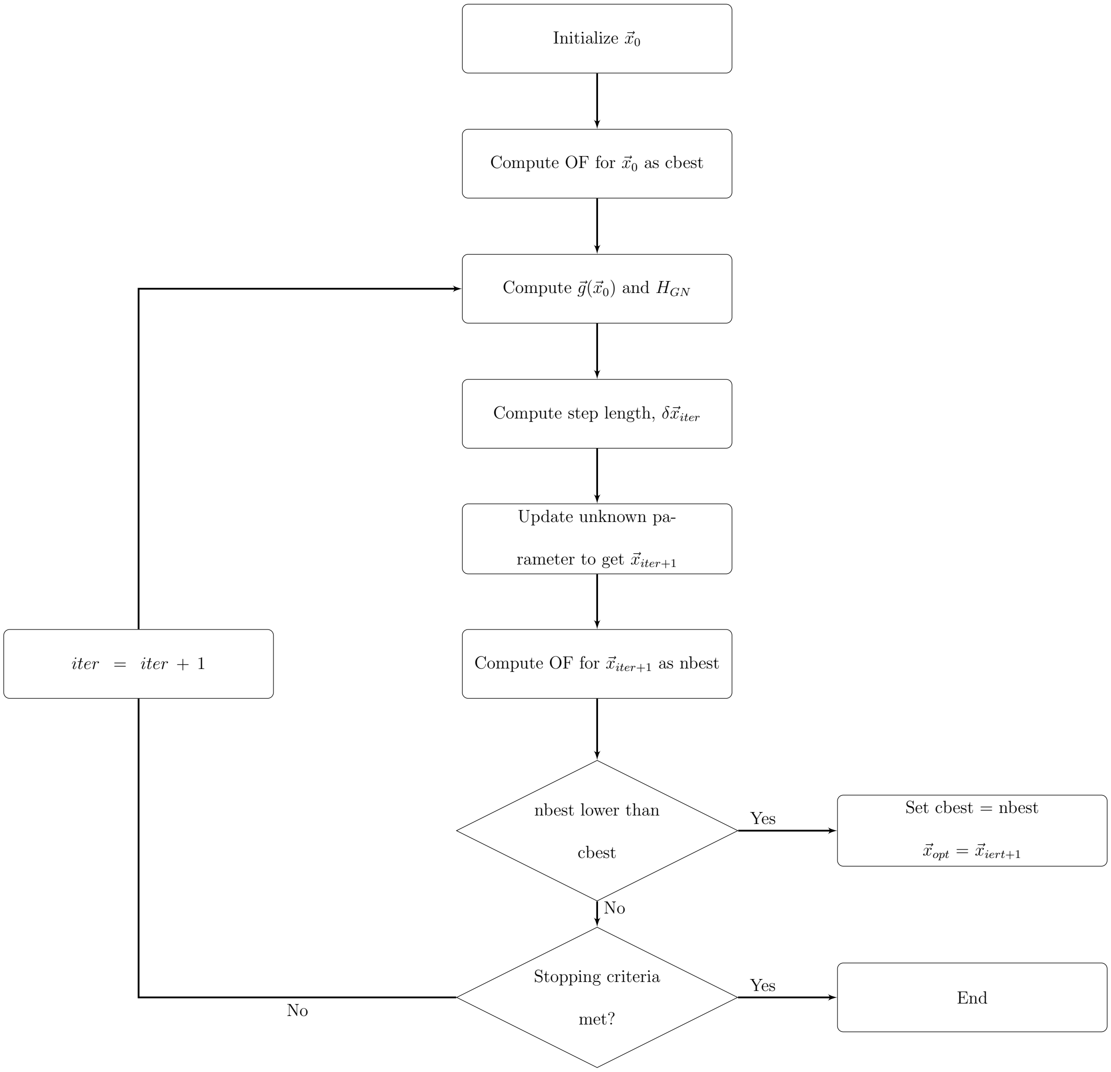

Initialize $\vec{x}_0$
Compute OF for $\vec{x}_0$ as cbest
Compute $\vec{g}(\vec{x}_0)$ and H_{GN}
Compute step length, $\delta\vec{x}_{iter}$
Update unknown parameter to get $\vec{x}_{iter+1}$
$iter = iter + 1$
Compute OF for $\vec{x}_{iter+1}$ as nbest
nbest lower than cbest
Yes
Set cbest = nbest
$\vec{x}_{opt} = \vec{x}_{iert+1}$
No
Stopping criteria met?
Yes
End
No

Figure 2.3: Flow diagram of GN algorithm.

In GN algorithm, an approximate Hessian is also used and is given by:

$$H_{GN} = S_m^T S_m \tag{2.23}$$

If the Hessian is positive definite, GN algorithm gives a quadratic convergence in the neighborhood of the true solution.

The search direction is computed by solving:

$$\vec{g} + H_{GN}\vec{\delta x} = 0 \tag{2.24}$$

The vector of unknown parameters is updated as:

$$\vec{x}_{iter+1} = \vec{x}_{iter} + \vec{\delta x}_{iter} \tag{2.25}$$

2.3.3 Levenberg-Marquardt Algorithm

Levenberg-Marquardt (LM) algorithm is a local optimization strategy originally proposed by Levenberg in 1944 [23] and modified by Marquardt in 1963 [24]. LM algorithm requires the gradient of OF as well as Hessian to solve non-linear least squares problems. LM algorithm is also known as damped least-squares (DLS) method. Its high efficiency is due to its adoption of a sensitivity matrix that contains significant information about the curvature of the OF, thus resulting in a quadratic convergence rate when close to the local minimum. LM algorithm is actually a combination of two minimization methods: SD and GN algorithms. In SD algorithms, the OF is minimized by updating the unknown parameters in the negative direction of its gradient while in GN algorithm, OF is minimized by assuming it is locally quadratic, and finding its minimum [25]. When the current OF estimate is far from the local minimum, LM algorithm behaves like SD algorithm and when OF estimate is close to the local minimum, LM algorithm behaves like GN algorithm. Levenberg modified GN algorithm by adding a product of damping factor and an identity matrix to perturb the Hessian diagonal elements. This modification enables the Hessian to be always positive definite and diagonally dominant. The magnitude of the diagonal elements has a great influence on the eigen values which are responsible for the positive definiteness of the Hessian. Matsui and Tanaka (1994) discovered that the damping factor should be the median of the eigen values of the Hessian [26]. The damping factor is a variable and usually adjusted from one iteration to another. When the damping factor is small, LM algorithm behaves like GN algorithm and takes nearly zero step length. When the damping factor is large, LM algorithm behaves like SD algorithm and takes a large step length in the descent direction. The choice of damping factor has a huge impact on the performance of LM algorithm. Marquardt(1963) suggested that the damping factor should be decreased by a factor if there is a sufficient reduction in the OF and vice-versa [24]. Many authors have applied LM algorithm in history matching problems [27],[28] and [29]. Xian (2015) applied LM algorithm to reservoir parameter estimation and estimated the damping factor using DE algorithm [30].

While LM algorithm can be very fast when applied to solve small to medium sized inverse problems, it is susceptible to being trapped in a local optimum in the neighborhood of the starting guess. This drawback makes it unsuitable for solving non-convex or multi-modal optimization problems. Furthermore, the gradient and Hessian of the OF that needs to be computed make LM algorithm unsuitable for optimization problems with discontinuities.

Procedural steps of LM algorithm are described in Figure 2.4.

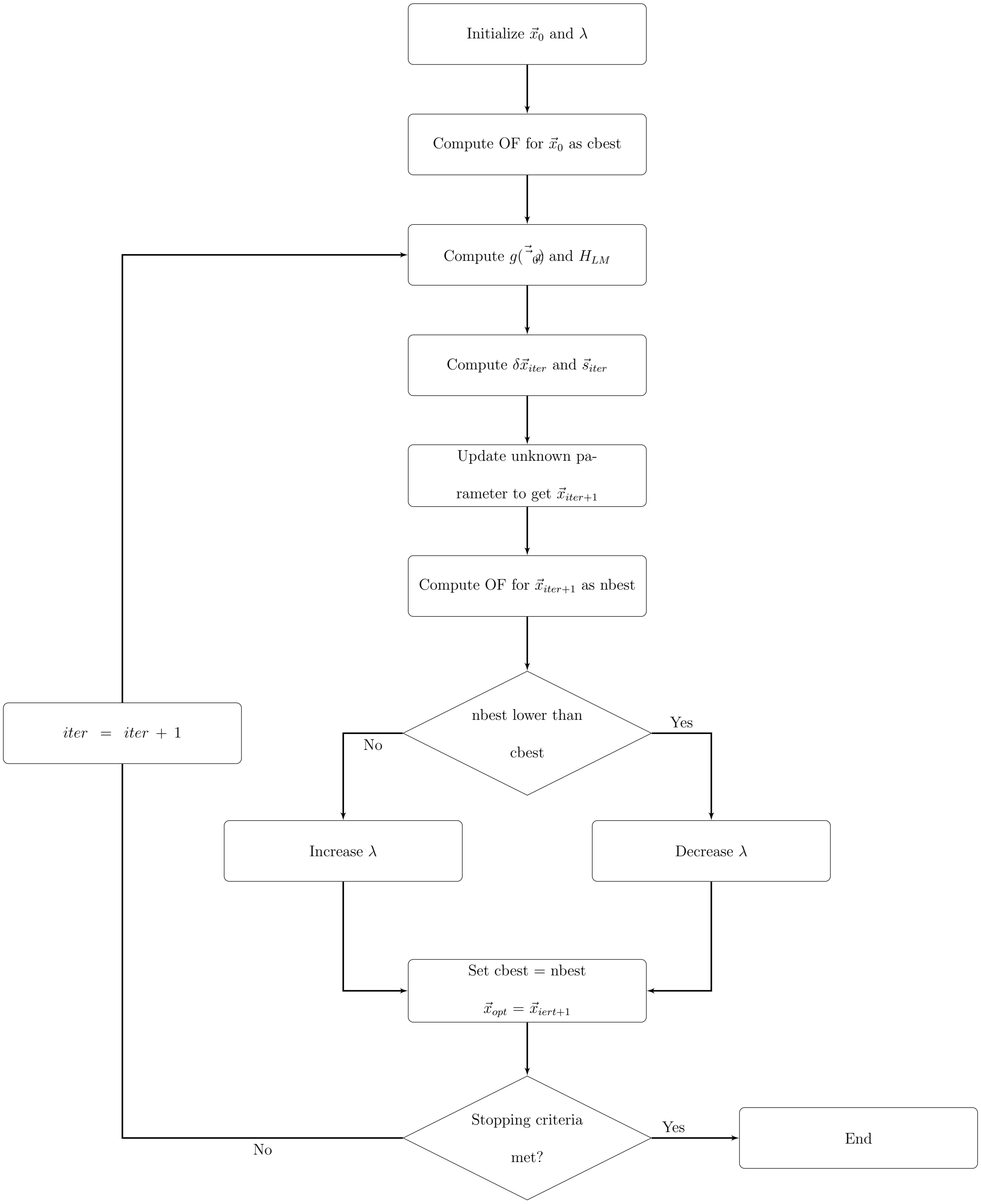

Initialize $\vec{x}_0$ and λ
Compute OF for $\vec{x}_0$ as cbest
Compute $g(\vec{\theta})$ and H_{LM}
Compute $\delta\vec{x}_{iter}$ and $\vec{s}_{iter}$
Update unknown parameter to get $\vec{x}_{iter+1}$
Compute OF for $\vec{x}_{iter+1}$ as nbest
nbest lower than cbest
Yes
No
Increase λ
Decrease λ
Set cbest = nbest
$\vec{x}_{opt} = \vec{x}_{iert+1}$
Stopping criteria met?
Yes
End
No
$iter = iter + 1$

Figure 2.4: Flow diagram of LM algorithm.

The LM algorithm Hessian is:

$$H_{LM} = H_{GN} + \lambda I \tag{2.26}$$

Solve for descent direction as:

$$H_{LM}\delta\vec{x} = -\vec{g}(\vec{x}) \tag{2.27}$$

Estimate step size using line search and update the vector of unknown parameters as:

$$\vec{x}_{iter+1} = \vec{x}_{iter} + \vec{s}_{iter}\delta\vec{x}_{iter} \tag{2.28}$$

2.4 Global Optimization Algorithms

Unlike the gradient-based algorithms, randomized approaches also known as global optimization algorithms are theoretically capable to reach the global optimum of the optimization problem, although they converge slowly [2]. Global optimization algorithms possess the ability to escape local minima and navigate complex, non-convex and even non-smooth OF landscape due to the adoption of random numbers. Global algorithms do not require the first or second derivative of the OF. Example of global optimization algorithms include SA, DE, GA, PSO, CMA-ES algorithms etc. These algorithms are based on certain characteristic features of molecular, biological, insects and neurological systems. Global algorithms have wide applications in history matching [31], [32] and [33]. Awotunde (2015) estimated parameters of a radial composite reservoir for well test analysis using DE, LM, CMA-ES and PSO algorithms [29].

2.4.1 Differential Evolution Algorithm

Differential evolution (DE) algorithm was proposed by Rainer Storn and Kenneth Price in 1997 [34]. As a global method, DE algorithm navigates farther from the initial guess to seek a candidate solution in the search space that will give the least OF. This suffices to say that this candidate solution is a global minimum. Lee and El-Sharkawi (2008) opined that DE algorithm is the simplest algorithm for solving multi-modal optimization problems [35]. DE algorithm was designed to fulfill the following users requirements:

1. Capability to handle non-differentiable, non-linear and multi-modal OF.
2. Parallel computing of intensive OF.
3. Ease of use.
4. Good convergence properties [34]

Many authors have worked on DE algorithm to improve its performance [36], [37],[38], [39], and [40].

Fundamental Principles of DE Algorithm

DE algorithm optimizes model parameters by iteratively improving candidate solutions (agents) through a cycle of population initialization, mutation, crossover and selection.

The concept of DE algorithm is described in Figure 2.5.

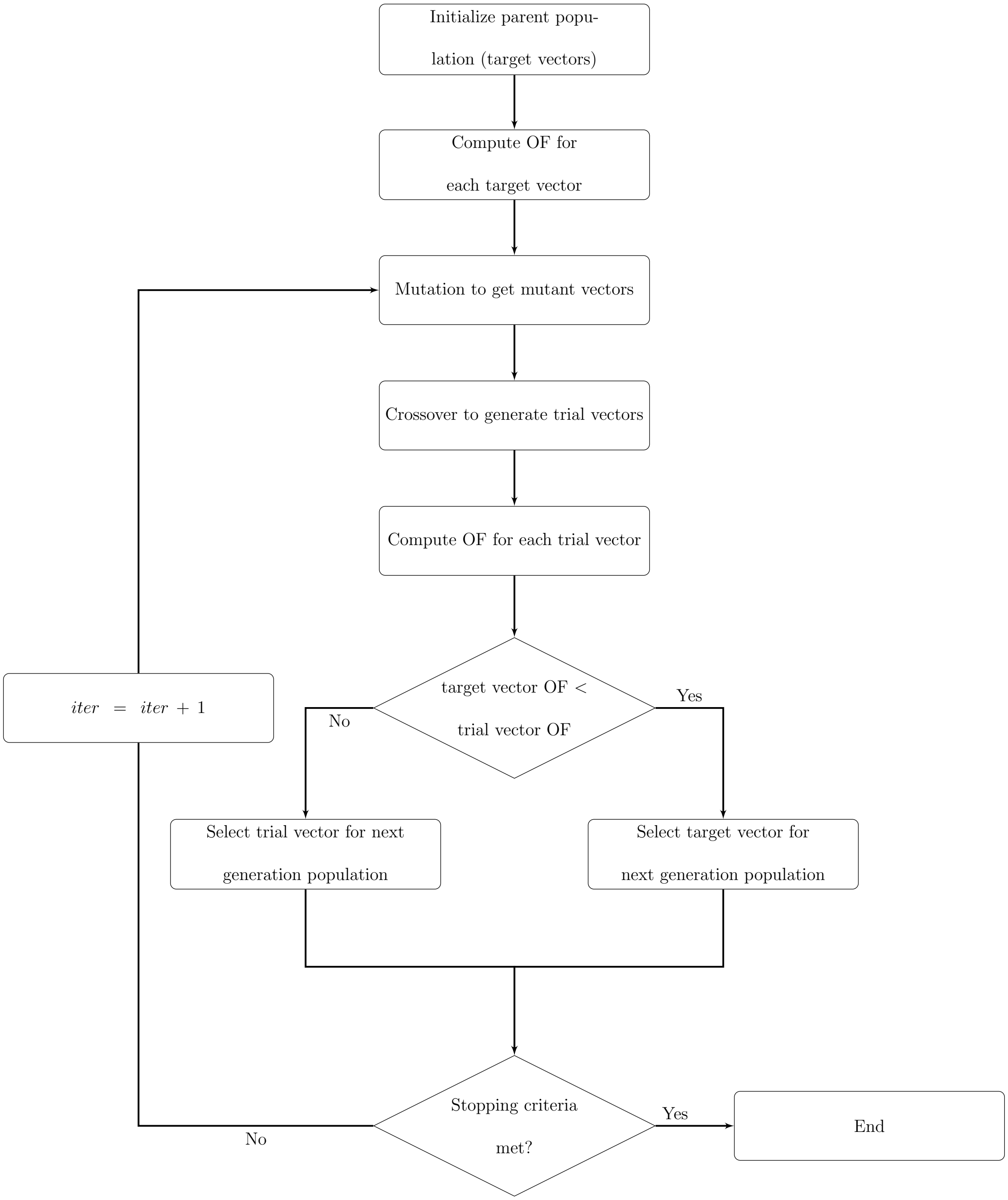

Initialize parent population (target vectors)
Compute OF for each target vector
Mutation to get mutant vectors
Crossover to generate trial vectors
Compute OF for each trial vector
target vector OF < trial vector OF
Yes
No
iter = iter + 1
Select trial vector for next generation population
Select target vector for next generation population
Stopping criteria met?
Yes
No
End

Figure 2.5: Flow diagram of DE algorithm.

Population Initialization

A population, P of dimension D x Np is initialized. Each vector of the population is regarded as genome/chromosome, target vector or candidate solution. Each of these solutions is perturbed to minimize the OF. Lower and upper bounds are specified for the unknown parameter to estimate the genomes. The initial population is computed as:

$$\vec{x}_0 = \vec{l}_b + rand(D, 1) \ \in \ [0, 1](\vec{u}_b - \vec{l}_b) \tag{2.29}$$

Where $rand_{i,j}$ [0, 1] is a uniformly distributed random number ranging from 0 to 1 (that is, $0 \leq rand_{i,j}$ [0, 1] ≤ 1) and is instantiated independently for each component of the ith vector. D is the dimension of the optimization problem (that is, number of unknown parameters),$\vec{l}_b$ and $\vec{u}_b$ are vectors of the lower and upper bounds of the unknown parameters.

Mutation

After the population has been initialized, they are mutated. Mutation involves linearly combining the best or randomly selected candidate solutions for each target vector, $\vec{x}_i$ to come up with a mutant/donor vector,$\vec{\nu}_i$. The six different mutation variants are given below:

"DE/rand/1"

$$\vec{\nu}_i = \vec{x}_{r1} + F(\vec{x}_{r2} - \vec{x}_{r3}) \tag{2.30}$$

"DE/best/1"

$$\vec{\nu}_i = \vec{x}_{best} + F(\vec{x}_{r1} - \vec{x}_{r2}) \tag{2.31}$$

"DE/current-to-best/1"

$$\vec{\nu}_i = \vec{x}_i + F(\vec{x}_{best} - \vec{x}_i) + F(\vec{x}_{r1} - \vec{x}_{r2}) \tag{2.32}$$

"DE/best/2"

$$\vec{\nu}_i = \vec{x}_{best} + F(\vec{x}_{r1} - \vec{x}_{r2}) + F(\vec{x}_{r3} - \vec{x}_{r4}) \tag{2.33}$$

"DE/rand/2"

$$\vec{\nu}_i = \vec{x}_{r1} + F(\vec{x}_{r2} - \vec{x}_{r3}) + F(\vec{x}_{r4} - \vec{x}_{r5}) \tag{2.34}$$

"DE/rand to best/1"

$$\vec{\nu}_i = \vec{x}_{r1} + F(\vec{x}_{r2} - \vec{x}_{r3}) + F(\vec{x}_{best} - \vec{x}_{r1}) \tag{2.35}$$

Crossover

After the target vectors have been mutated, crossover is done to determine which agent in the mutant and target vectors will cross over to a new population called the trial vector. This crossover process (also known as parameter mixing or recombination) is based on probability and it must be ensured that at least one agent in the mutant vector crosses over to the trial vector. The trial vector contains elements which are either from the target or donor vector. There are basically two crossover methods: exponential (or two-point modulo) and binomial (or uniform). The binomial crossover is performed on each pair of corresponding elements in the target and mutant vectors by randomly generating a number between 0 and 1. Its value is compared with crossover probability, (C_R) to determine which of the elements in the two vectors will crossover. Its process is given as:

If ($rand_{i,j} \in [0,\ 1] \leq\ C_R$ or $j = jrand$)

$$y_{j,i} = \nu_{j,i} \tag{2.36}$$

Otherwise,

$$y_{j,i} = x_{j,i} \tag{2.37}$$

Selection

Selection between the target and trial vectors is done to determine which of them would survive to the next generation (iteration) based on how well they minimize the OF. The vector that gives the least value of the OF is selected for the next generation (or iteration) and the process continues until the OF convergences. With these processes, the DE algorithm does not get trapped in the local minimum. The selection operation is described as:

If $f(\vec{y_i}) \leq f(\vec{x_i})$

$$\vec{y}_{i,new} = \vec{y_i} \tag{2.38}$$

Otherwise,

$$\vec{y}_{i,new} = \vec{x_i} \tag{2.39}$$

2.4.2 Particle Swarm Optimization Algorithm

Particle swarm optimization (PSO) algorithm was proposed in 1995 [41] based on swarm intelligence. PSO algorithm comes from the research on the bird flock or fish school movement behavior [42]. The algorithm iteratively improves a population (swarm) of candidate solutions (particles) by moving them around in the search space based on the particles' positions and velocities. The behavioral model each particle exhibits follows three rules [43]:

1. Separation: each particle tries to move away from its neighbors if they are too close.
2. Alignment: each particle steers towards the average heading of its neighbors.
3. Cohesion: each particle tries to go towards the average position of its neighbors.

The movement of each particle is affected by its local best known position and that of the swarm. This is aimed at moving the swarm towards the best solution and making the OF converges. PSO algorithm is simple and easy to implement and has been widely applied in the fields of neutral network training, function optimization, model classification, machine study, signal procession, vague system and automatic adaptation controls etc. [44].

The advantages of PSO algorithm are [42]:

1. It is based on swarm intelligence. As such, it can be applied to both scientific and engineering research.
2. Its speed of search is very fast as the particles are moved towards the best position of the swarm.
3. Its computations are very simple.

The PSO algorithm is described in Figure 2.6.

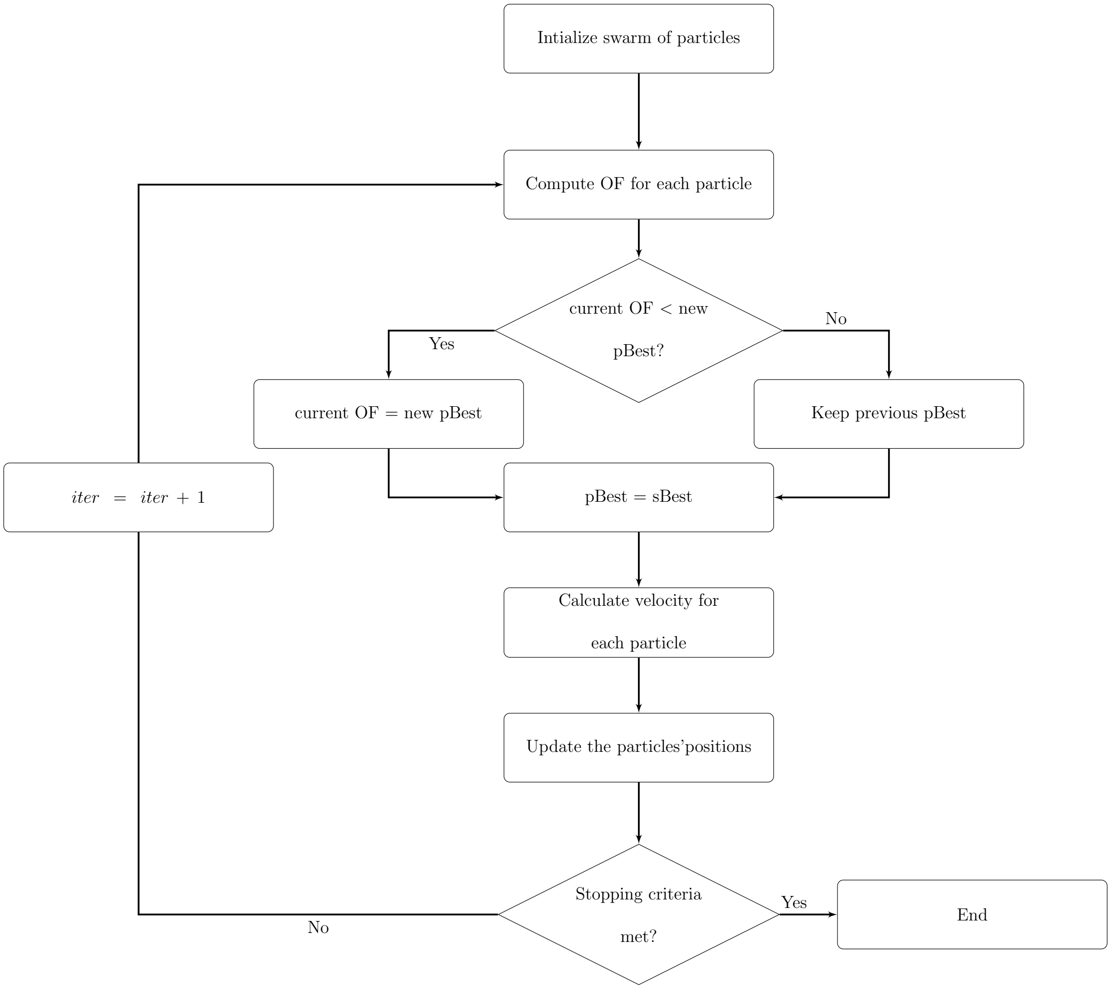


Figure 2.6: Flow diagram of PSO algorithm.

2.4.3 Simulated Annealing Algorithm

The concept of simulated annealing (SA) algorithm is based on the annealing process in metallurgy. The annealing process involves heating and controlled cooling of a material to enlarge its crystals and decrease their defects. These two effects depends on the thermodynamic free energy of the material. SA algorithm is a good choice if an acceptable solution is being sought within a short time rather than the best possible solution. The rate of cooling is inversely and directly proportional to the decrease in thermodynamic free energy and temperature respectively. Decreasing the energy is tantamount to minimizing the OF by seeking better solutions. Sometimes, SA algorithm accepts worse solutions to search more extensively for an optimal solution. The process of slow cooling is implemented as a slow decrease in the probability of accepting worse solutions as it explores the search space. SA algorithm was proposed by Scott Kirkpatrick *et al.* [45] and by Vlado Cerny in 1985 [46].

Procedural steps of SA algorithm are described in Figure 2.7.

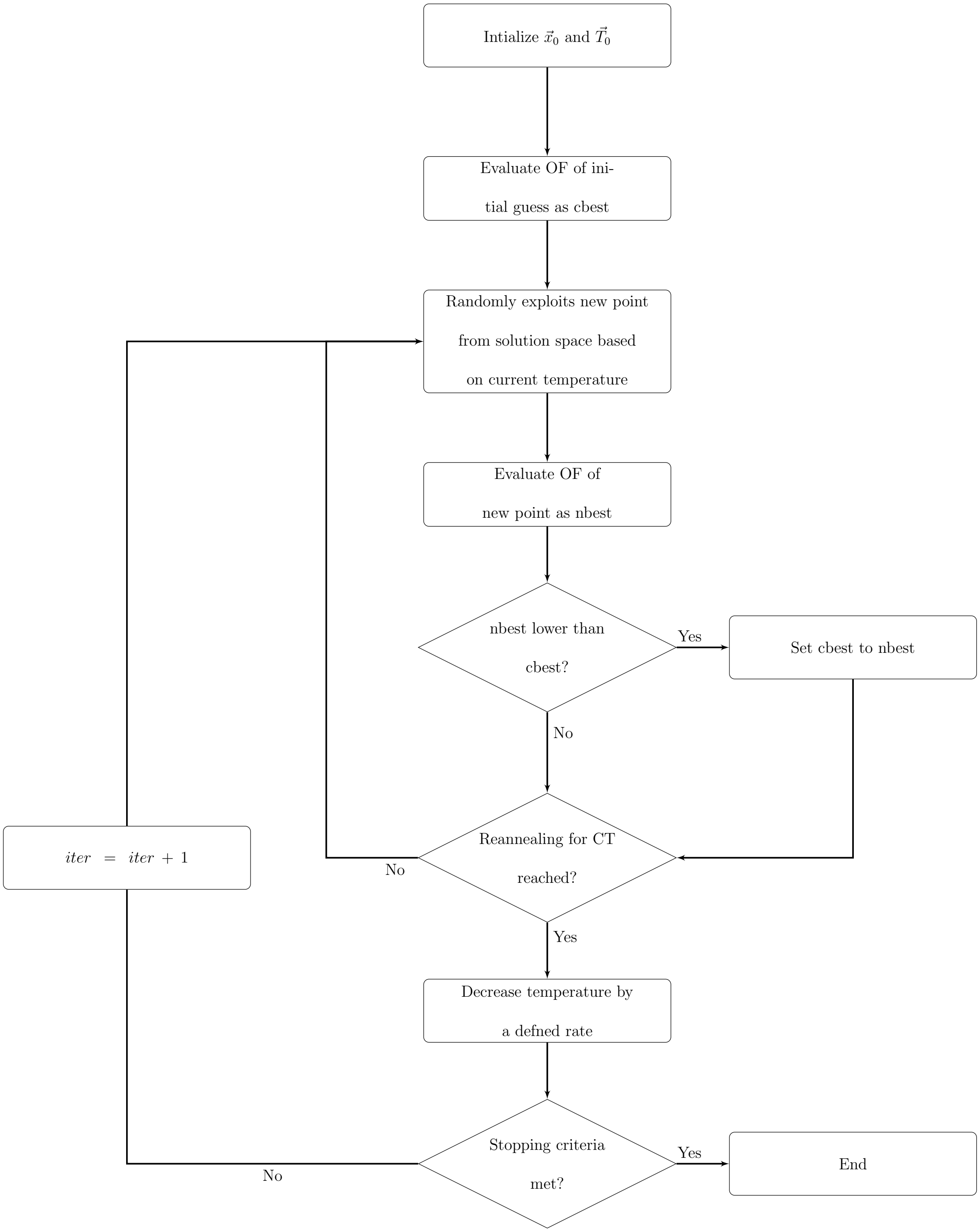

Intialize $\vec{x}_0$ and $\vec{T}_0$
Evaluate OF of initial guess as cbest
Randomly exploits new point from solution space based on current temperature
Evaluate OF of new point as nbest
nbest lower than cbest?
Yes
Set cbest to nbest
No
Reannealing for CT reached?
No
$iter = iter + 1$
Yes
Decrease temperature by a defned rate
Stopping criteria met?
Yes
End
No

Figure 2.7: Flow diagram of SA algorithm.

CHAPTER 3

METHODOLOGY

3.1 Geostatistical Modeling

In this section, the characterization of the 3D synthetic reservoir simulation model used in this study with a 2D dataset was presented. Geostatistical modeling involves the spatial interpolation of parameters measured at sparse locations in the domain of interest to the entire domain of interest and generating several equi-probable realizations of these parameters through stochastic simulation for the purpose of history matching. Porosity and permeability data obtained from eighty-five wells (well logs) at different locations in a reservoir were populated to the entire reservoir.

3.1.1 Data Analysis

The dataset used for geostatistical modeling comprises the X, Y and Z coordinates of the sampled locations (wells) in ft, porosity in percentage and permeability in the x-direction in md. The range of the Z coordinate is relatively small. As such, the dataset is assumed to be two-dimensional (2D). In order to make the permeability distribution Gaussian, its logarithm to base ten was estimated and used for parameterization. The dataset is given in Table 3.1.

Table 3.1: Permeability and porosity measured from wells.

$X(ft)$	$Y(ft)$	$Z(ft)$	$\phi(\%)$	$k_x(md)$	$log_{10}k_x$
6050	4150	7037.1531	14.652	421	2.624
2650	4350	7031.4993	14.509	453	2.656
1750	6950	7036.9185	14.064	528	2.723
2550	950	7024.0156	15.108	561	2.749
4950	6850	7035.0411	13.919	577	2.761
1450	450	7028.4249	13.130	604	2.781
3950	3350	7033.2458	14.572	893	2.951
8450	2450	7037.0216	15.081	968	2.986
9350	750	7040.0446	13.910	1010	3.004
1350	1050	7029.1565	13.402	1050	3.021
5350	2550	7035.4579	14.940	1070	3.029
3750	6450	7031.44	15.216	2020	3.305
2750	5550	7035.2379	14.578	19	1.279
4750	4550	7034.9151	14.248	26.6	1.425
7650	1550	7033.4161	14.428	30.5	1.484

Continued on next page.

Table 3.1 – *Continued from previous page.*

$X(ft)$	$Y(ft)$	$Z(ft)$	$\phi(\%)$	$k_x(md)$	$log_{10}(k_x)$
2350	4850	7032.3707	15.261	32.2	1.508
8350	7850	7043.8217	16.186	32.9	1.517
9750	4850	7037.3289	14.208	33.5	1.525
8450	6550	7038.0937	16.958	35.7	1.553
450	1850	7027.6737	13.835	38	1.580
250	5950	7032.3485	14.186	40.6	1.609
4550	650	7028.0296	14.038	90	1.954
4550	6850	7034.096	14.369	92	1.964
4950	6450	7036.3549	13.402	95.7	1.981
3150	50	7026.0119	15.895	97.1	1.987
1850	2550	7029.1077	12.867	97.6	1.989
8150	450	7036.6456	15.104	107	2.029
9150	6750	7038.7817	15.774	118	2.072
4750	3450	7031.9071	14.133	119.5	2.077
8950	1550	7037.5415	13.337	121.9	2.086
4950	7750	7043	15.136	233	2.367
3550	4450	7033.0404	15.085	258	2.412
9650	3550	7042.5253	14.250	285	2.455
1150	2850	7029.7727	12.681	337	2.528

Continued on next page.

Table 3.1 – *Continued from previous page.*

$X(ft)$	$Y(ft)$	$Z(ft)$	$\phi(\%)$	$k_x(md)$	$log_{10}(k_x)$
3650	4450	7032.2021	14.938	349	2.543
6950	1850	7034.9958	15.601	361	2.558
4250	5050	7032.5227	13.780	394	2.595
4050	4350	7031.6919	15.291	400	2.602
7350	5950	7039.784	15.688	42.3	1.626
3150	1150	7030.3988	15.368	43	1.633
5950	6450	7034.9988	14.328	44	1.643
9050	3550	7038.3948	14.737	44.9	1.652
5650	3550	7035.3294	15.055	45.3	1.656
6250	1550	7030.4755	14.889	46	1.663
1350	6350	7038.0117	14.436	47.1	1.673
1350	2150	7029.1415	12.149	48	1.681
4250	5650	7035.0381	13.624	48.7	1.688
750	450	7024.0766	14.188	52.8	1.723
3650	650	7032.3753	14.907	168	2.225
5350	2050	7032.9269	15.203	172	2.236
3550	950	7033.334	15.347	180	2.255
1950	4250	7031.0577	15.939	188	2.274
8550	3050	7034.2641	15.727	191.3	2.282

Continued on next page.

Table 3.1 – *Continued from previous page.*

$X(ft)$	$Y(ft)$	$Z(ft)$	$\phi(\%)$	$k_x(md)$	$log_{10}(k_x)$
7050	5050	7038.2068	15.324	194	2.288
5750	2450	7037.6076	14.045	199	2.299
6650	7850	7036.9157	14.403	216	2.334
950	6050	7038.5827	14.359	224	2.350
7550	1450	7033.966	14.601	74.3	1.871
3250	450	7029.2442	16.146	75.3	1.877
4450	3050	7032.2726	15.773	76	1.881
2250	1150	7027.8722	13.623	76.5	1.884
6450	5150	7033.7745	15.102	79.2	1.899
5450	2850	7033.9646	15.355	79.8	1.902
1750	350	7028.3525	13.843	80.5	1.906
8150	1850	7033.9217	14.943	82	1.914
450	2550	7030.3675	14.414	83.2	1.920
6450	6450	7038.3666	13.618	83.7	1.923
7650	4650	7038.2326	16.379	88.3	1.946
3650	3450	7032.2048	14.258	124.1	2.094
8150	6250	7037.3239	15.777	126.5	2.102
50	4450	7031.7183	14.655	130	2.114
850	5850	7037.7883	14.363	139	2.143

Continued on next page.

Table 3.1 – *Continued from previous page.*

$X(ft)$	$Y(ft)$	$Z(ft)$	$\phi(\%)$	$k_x(md)$	$log_{10}(k_x)$
8750	5550	7038.016	15.966	143	2.155
7450	4150	7038.374	16.010	151	2.179
4150	5450	7032.7182	13.964	158	2.199
2050	7250	7038.4526	14.265	162	2.210
5550	7650	7039.7254	15.768	165.8	2.220
250	2450	7029.5005	14.591	55	1.740
6550	750	7032.2007	15.138	56.2	1.750
9450	850	7040.5902	14.095	57.9	1.763
1750	3750	7033.2246	15.149	59.2	1.772
1850	3450	7030.6871	13.958	62	1.792
7250	6650	7039.6297	14.738	63.4	1.802
2450	4550	7031.2961	15.069	66	1.820
4850	2850	7032.8684	15.804	69	1.839

An essential part of geostatistical data analysis is building a variogram of the dataset to determine the degree of spatial correlation between any rock property at two different locations across the reservoir. Details of the parameters used to build the variogram are given in Table 3.2.

Table 3.2: Variogram parameters.

Variogram parameter	**Value**
Lag distance, ft	1000
Lag tolerance, ft	500
Number of lags	15
Number of directions	5
Azimuth, $degree$	0, 0, 45, 90, and 135
Azimuth tolerance, $degree$	90, 22.5, 22.5, 22.5, and 22.5
Bandwidth, ft	10^8, 5000, 5000, 5000, and 5000
Nugget effect	0
Sill	1
Variogram type	Exponential
Maximum range, ft	5500
Medium range, ft	5500
Minimum range, ft	5500

3.1.2 Kriging

Kriging is used to predict the value of a parameter at a location by computing a weighted average of the known values of that parameter at the neighboring locations based on the variogram. Higher weights are given to neighbors with higher spatial correlation. Porosity and permeability from the eighty-five wells were populated to the entire 2D reservoir model.

The rectangular 2D model of size 10000 ft x 8000 ft x 100 ft was discretized into a grid of 50 x 40 cells to be assigned the kriged values. Each grid cells have dimension of 200 ft x 200 ft x 100 ft. Since the dataset is 2D, the kriged rock properties can be considered for only a layer in the 3D simulation model. As such, three realizations of each rock property would be required to characterize the three layers in the 3D model. Note that the discretization of the 2D model and that of each layer of the 3D model (discussed in Subsection 3.2.1) are the same.

kriged porosity and the kriging variance (porosity estimation error) were obtained from porosity kriging. Also, kriged permeability and the kriging variance (permeability estimation error) were obtained from permeability kriging. Ordinary kriging type was adopted.

3.1.3 Stochastic Simulation

For the characterization of the 3D model, three realizations of porosity as well as permeability are required. Stochastic simulations was used to generate three realizations of porosity using sequential Gaussian simulation (SGS) algorithm with simple kriging type. These porosity realizations were used for characterizing the 3D model and served as the true or reference porosity field of the model.

GDM of history matching involves linearly combining multiple realizations of the design parameter (which in this study is the reservoir permeability). Therefore, forty-eight equi-probable permeability realizations were generated stochastically using SGS algorithm with simple kriging type.

Three permeability realizations were used to characterize the 3D model. This served as the true or reference permeability field of the model. The remaining forty-five realizations were grouped to create fifty realizations for the 3D model to be used in the GDM for the purpose of history matching.

3.2 Reservoir Modeling

In this section, the 3D model geometry and discretization, rock and fluid properties, well specifications and completions data and other pertinent parameters used to build the synthetic model are presented. Although the model is synthetic but the values of the parameters are based on real field data. The reservoir modeling was done using Eclipse commercial simulator.

3.2.1 Reservoir Discretization

This part of the modeling goes to the "GRID" section of the Eclipse input file. The model has a rectangular geometry with the size 10000 ft x 8000 ft x 300 ft. The reservoir was discretized into 6000 (50 x 40 x 3) grid cells with each cell having a dimension of 200 ft x 200 ft x 100 ft. The method of solving the pressures and saturations in the grid cells is fully implicit. The top of the reservoir is at a depth of 7000 ft.

3.2.2 Reservoir Rock Properties

This part goes to the "PROPS" section of the Eclipse input file. The permeability and porosity of the reservoir rock were obtained by geostatistical modeling discussed above. The permeability is assumed anisotropic such that its value in the z-direction equals half of its value in the x-direction and the permeability in the x and y directions are equal. The rock compressibility is presented in Table 3.3.

Table 3.3: Rock compressibility.

$P_{ref}(psia)$	$C_r(psia^{-1})$
5801.5	2.8 x 10^{-6}

3.2.3 Reservoir Fluid Properties

This part also goes to the "PROPS" section. The reservoir fluids are oil, water, dissolved gas and dry gas (gas cap). The reservoir has no aquifer connected to it. The fluid properties are presented in Tables 3.4 to 3.9.

Table 3.4: PVT properties of live oil.

$R_{so}(Mscf/stb)$	$P(psia)$	$B_o(bbl/stb)$	$\mu_o(cp)$
0.137	250.0	1.274	0.107
0.368	500.0	1.459	0.094
0.433	570.3	1.510	0.091
0.449	587.8	1.523	0.090

Continued on next page.

Table 3.4 – *Continued from previous page.*

$R_{so}(Mscf/stb)$	$P(psia)$	$B_o(bbl/stb)$	$\mu_o(cp)$
0.466	605.4	1.536	0.089
0.500	640.5	1.562	0.088
	781.1	1.548	0.091
	1062.2	1.522	0.096
	1500.0	1.489	0.105
	1783.8	1.471	0.111
	2000.0	1.458	0.115
	2500.0	1.433	0.124
	3035.7	1.410	0.134
	3400.0	1.397	0.141
0.642	781.1	1.674	0.083
0.973	1062.2	1.934	0.073
	1500.0	1.871	0.081
	1783.8	1.838	0.085
	2000.0	1.816	0.089
	2500.0	1.773	0.097
	3035.7	1.735	0.105
	3400.0	1.714	0.111
1.686	1500.0	2.508	0.060

Continued on next page.

Table 3.4 – *Continued from previous page.*

$R_{so}(Mscf/stb)$	$P(psia)$	$B_o(bbl/stb)$	$\mu_o(cp)$
	1783.8	2.437	0.064
	2000.0	2.392	0.067
	2500.0	2.309	0.074
	3035.7	2.240	0.081
	3400.0	2.202	0.086
2.403	1783.8	3.107	0.052
	2000.0	3.025	0.055
	2500.0	2.881	0.061
	3035.7	2.770	0.067
	3400.0	2.709	0.071
5.000	3300.0	3.500	0.041
	3400.0	3.460	0.042

Table 3.5: Fluid density.

$\rho_o(Ib/ft^3)$	$\rho_w(Ib/ft^3)$	$\rho_g(Ib/ft^3)$
42.28	62.43	0.0971

Table 3.6: PVT properties of dry gas.

$P(psia)$	$B_g(Mscf/stb)$	$\mu_g(cp)$
250.0	12.651	0.012
500.0	6.076	0.013
570.3	5.662	0.013
587.8	5.558	0.013
605.4	5.455	0.013
640.5	5.248	0.013
781.1	4.420	0.013
1062.2	3.413	0.014
1500.0	2.037	0.016
1783.8	1.494	0.018
2000.0	1.326	0.019
2500.0	1.073	0.023
3035.7	0.916	0.026
3400.0	0.845	0.029

Table 3.7: PVT properties of water.

$P_{ref}(psia)$	$B_w(bbl/stb)$	$C_w(psia^{-1})$	$\mu_w(cp)$	ν_w
3118	1.013	2.7 x 10^{-6}	0.4	0.0

Table 3.8: Oil-water saturation.

S_w	K_{rw}	K_{row}	$P_{cow}(psia)$
0.20	0.000	0.900	50.0
0.22	0.000	0.803	45.0
0.30	0.001	0.487	25.0
0.40	0.009	0.221	12.5
0.50	0.045	0.078	6.3
0.60	0.154	0.014	2.5
0.70	0.387	0.001	1.3
0.73	0.480	0.000	1.1
0.80	0.800	0.000	0.8
1.00	1.000	0.000	0.0

Table 3.9: Oil-gas saturation.

S_g	K_{rg}	K_{rog}	$P_{cog}(psia)$
0.00	0.000	0.900	0.0
0.06	0.000	0.525	0.0
0.10	0.000	0.375	0.0
0.14	0.000	0.213	0.0
0.19	0.002	0.106	0.0
0.24	0.006	0.042	0.0
0.29	0.013	0.011	0.0
0.33	0.035	0.001	0.0
0.37	0.061	0.000	0.0
0.80	0.900	0.000	0.0

3.2.4 Reservoir Equilibration

This subsection goes to the "SOLUTION" section of Eclipse input file. The initial state of the reservoir is given in Tables 3.10 and 3.11.

Table 3.10: Reservoir initial state.

Equilibration parameter	**Value**
Datum depth, ft	7000
Datum pressure, $psia$	3400
Oil-water contact depth, ft	7285
Gas-oil contact depth, ft	7050
Oil-water capillary pressure, $psia$	0

Table 3.11: Variation of initial solution gas-oil ratio with depth.

Depth (ft)	$R_{so}(Mscf/stb)$
6000	0.77
8000	0.77

These parameters are used by Eclipse to estimate the initial pressures and saturations in all the grid cells.

3.2.5 Well and Completion Specifications

This subsection goes to the "SCHEDULE" section of the Eclipse input file. A total of fifteen vertical wells are strategically placed in the reservoir to give high sweep efficiency. There are ten producers and five injectors. The wells information are presented in Table 3.12.

Table 3.12: Well information.

Well name	**Well type**	**x-coordinate**	**y-coordinate**	**Completion layer**	**Phase**
P1	producer	3	37	2	oil
P2	producer	20	34	2	oil
P3	producer	35	36	2	oil
P4	producer	15	22	2	oil
P5	producer	30	25	2	oil
P6	producer	45	29	2	oil
P7	producer	5	7	2	oil
P8	producer	22	20	2	oil
P9	producer	27	5	2	oil
P10	producer	43	18	2	oil
I1	injector	11	32	3	water
I2	injector	24	30	3	water
I3	injector	38	27	3	water
I4	injector	12	15	3	water
I5	injector	35	16	3	water

The producers are completed in the second layer (oil zone) while the injectors are completed in the third layer (water zone). All wells have a diameter of 0.625 ft and zero skin.

3.2.6 Well Constraints for Producers

This part goes to the "SCHEDULE" section of the Eclipse input file. Each production well is open for production and primarily controlled by an oil rate target of 700 stb/day. Each producer is secondarily controlled by a minimum bottom-hole pressure of 2000 $psia$.

3.2.7 Well Constraints for Injectors

This part goes to the "SCHEDULE" section of the Eclipse input file. All injection wells are water injectors. Each of the injectors is primarily controlled by a bottom-hole pressure target whose maximum value is set at 5500 $psia$. The maximum surface flow rate for each injector is 4200 stb/day.

3.2.8 Time Step and Report Time

This subsection goes to the "SCHEDULE" section of the Eclipse input file. The minimum and maximum length of the next time step is 0.1 day and 1 day respectively. Maximum length of all time steps after the next is 5 days. The minimum and maximum number of Newton iterations in a time step are defaulted. Simulation is done for 10 years starting from November 01, 2011 to October 31, 2021.

3.2.9 History Production Data

The synthetic true reservoir model was used to generate production data. Water cut was reported from the ten production wells at the end every month for a period of ten years. Each well reports 121 water cut data points for the entire simulation period (making a total of 1210 water cut data points for all the production wells).

In real life, history production data are measured in the field and measurements are prone to noise. Therefore, Gaussian noise was added to the water cut to simulate field measurement as follows:

$$d_{meas} = W_c + W_c \left(\vec{R} \sqrt{N_{sr}} \right) \tag{3.1}$$

where W_c is the reported water cut; N_{sr} is the noise-to-sound ratio which equals 10^{-6}; R is the vector of N normally distributed random numbers and d_{meas} is the noisy water cut.

The noisy water cut is taken as the measured data to be matched in the history matching process.

3.3 Implementation of History Matching

In this section, how both the gradient-based and global optimization algorithms were applied in the GDM of history matching are presented. The true permeability distribution of the synthetic reservoir is assumed to be unknown and will be determined through GDM. The noisy water cut is also assumed to be the available history production data.

Thus, in the history matching process, the noisy water cut was matched to estimate the permeability distribution of the reservoir.

3.3.1 Realizations

In this study, seven different realizations were considered. Each realization uses a different random seed. The seed is used by the algorithm to generate random numbers which are used for computations while it is running. With the same seed, the same set of random numbers are generated every time the algorithm is run. Also, for each realization, LM, DE, PSO and SA algorithms were used in GDM to optimize the deformation parameters. The random seeds 0, 14807, 911, 111, 237, 4567, 2541 were used for Realizations 1 to 7, respectively.

3.3.2 Sample Application of GDM in History Matching

In this study, the fifteen permeability realizations (discussed in Chapter 3, Subsection 3.1.3) were transformed to standard normal score and linearly combined to generate new Gaussian realizations using Equation 2.10.

The vector of mean is the kriged permeability. The number of elements in the vectors of mean and the standard deviation is equal to the number of grid cells in a layer of the synthetic model (that is, each grid cell has its own mean and standard deviation). The same vectors of mean and standard deviation were used in all the layers of the synthetic model.

In this work, each linear combination of two Gaussian realizations is regarded as a deformation stage. In the first stage, $\vec{K_1}$ and $\vec{K_2}$ are Gaussian realizations selected from the fifteen Gaussian realizations. An optimization algorithm is used to estimate $\vec{x}$. The optimum $\vec{x}$ is that vector of deformation parameters that gave the minimum OF in that stage. This optimum $\vec{x}$ is used to estimate the optimum $\vec{K}$ which is used as $\vec{K}_1$ in the second deformation stage and a new $\vec{K}_2$ is selected from the pool based on the third random number. An optimization algorithm is again used to estimate $\vec{x}$. The optimum $\vec{x}$ is used to estimate the optimum $\vec{K}$ which is used as $\vec{K}_1$ in the next deformation stage if and only if the minimum OF in the second stage is less than that of the first stage, otherwise, the optimum $\vec{K}$ from the first deformation stage will be used as $\vec{K}_1$ in the current deformation stage. A new $\vec{K}_2$ is selected from the pool based on the next random number and the process continues in the cycle discussed above until all the fifteen Gaussian realizations have been drawn. There is a total of fourteen deformation stages.

At the end of the 14^{th} deformation stage, the $\vec{K}$ that gives the overall minimum OF is the optimum permeability distribution of the synthetic reservoir model. These procedures were repeated for each optimization algorithm used.

3.3.3 Deformation Parameters

The synthetic reservoir model was divided into sixty sub-regions with each sub-region having one hundred grid cells and a deformation parameter assigned to it. Hence, there are sixty parameters to be estimated by each of LM, DE, PSO and SA algorithms. The lower and upper bounds for all the deformation parameters are the natural logarithm of 10^{-3} and π in radians, respectively. The natural logarithm of these bounds have been used for effective optimization. Note that whenever the deformation parameters are to be used to estimate the reservoir permeability, the deformation parameters are first back-transformed to their real values by computing their exponents.

3.3.4 Objective Function

Objective function (OF) was estimated as the sum of squares of the difference between the simulated and measured water cut using Equation 2.15.

3.3.5 Sample Application of LM in GDM

LM algorithm is the gradient-based algorithm used in this study to estimate the deformation parameters in GDM. Initial guess of the deformation parameters was computed using their lower and upper bounds with the formulation below:

$$\vec{x}_0 = \vec{l}_b + rand(L_x, 1) \ \in \ [0,1](\vec{u}_b - \vec{l}_b) \tag{3.2}$$

The deformation parameters are used to estimate the reservoir permeability which is used generate simulated production data, d_{cal}.

Sensitivity Coefficients Estimation

Sensitivity coefficients represent how the change in the design parameter ($\vec{x}$) affects the reservoir response (for example, water cut). In this study, sensitivity coefficients were estimated using the substitution method where each element of

$\vec{x}_0$ is perturbed and replaced with the new value in the $\vec{x}_0$. The new $\vec{x}_0$ is used to estimate permeability. The estimated permeability is used to generate simulated production data and a new OF, $\vec{d}_{ycal}$ is estimated. Sensitivity coefficients are estimated with the formulation in Equation 3.3.

$$\vec{S} = \frac{\vec{d}_{ycal} - \vec{d}_{cal}}{dx}; \tag{3.3}$$

Where $dx = 5$ x 10^{-3} and $\vec{S}$ is a vector of sensitivity coefficients. This process is repeated for all the elements in $\vec{x}_0$ after which a sensitivity matrix is computed. The N x D matrix was computed by multiplying each $\vec{S}$ by its corresponding element in $\vec{x}_0$ before the perturbation. The new $\vec{S}$ for all the elements in the perturbed $\vec{x}_0$ are placed in the columns that correspond to the location of the elements to form the sensitivity matrix, S_m.

Gradient Estimation

The gradient of the OF was estimated using Equation 2.16. Its length is D.

Hessian Estimation

The LM algorithm Hessian is computed using Equation 2.26. Its dimension is D x D.

Line Search

The line search approach involves finding a descent direction along which the OF will be reduced and then estimates a step size that determines how far $\vec{x}$ should move along that direction.

Inexact line search can be performed by using backtracking line search or the Wolfe conditions. Strong Wolfe conditions given in Equations 3.4 and 3.5 were adopted in this study.

$$\left|\Phi(\vec{x}+\vec{s}\vec{\delta x})\right| \;\leq\; \left|\Phi(\vec{x})+b_1\vec{s}(\vec{g}(\vec{x})^T\vec{\delta x})\right| \tag{3.4}$$

$$\left|\vec{g}(\vec{x}+\vec{s}\vec{\delta x})^T\vec{\delta x}\right| \;\leq\; b_2\left|\vec{g}(\vec{x})^T\vec{\delta x}\right| \tag{3.5}$$

Where constants b_1 and b_2 are 10^{-5} and 0.9, respectively. Equations 3.4 and 3.5 are the sufficient decrease (or Armijo rule) and the curvature conditions, respectively which any descent direction and step size to be used must satisfy.

1. Descent direction

   The descent direction can be computed by various methods, such as gradient descent, Newton's method and Quasi-Newton method. Newton's method was adopted in this study. The resulting linear system of equations (see Equation 2.27) from Newton's method can be solved by various factorizations or iterative methods. In this work, it was solved using a MATLAB in-built function "linsolve" to estimate the descent direction, $\delta\vec{x}$. The function solves a linear system of equations using LU factorization with partial pivoting when H is square matrix and QR factorization with column pivoting otherwise. Since H is a square matrix, LU factorization with partial pivoting was used.

This search direction is expected to be a descent one but if not, it is forced to be as follows:

Firstly, the directional derivative, p of the gradient is estimated

$$p = \delta\vec{x}^T \vec{g}(\vec{x}) \tag{3.6}$$

If p is non-negative, then

$$\delta\vec{x} = -\delta\vec{x} \tag{3.7}$$

The estimated search direction was used to update the $\vec{x}_0$ to minimize the OF.

2. Step size

   Line search is done to determine the step size to move in the descent direction. Its lower and upper bounds are 10^{-15} and 10^{15} respectively. It is initialized as 1 and subsequently adjusted within its bounds until a step size that satisfies the strong Wolfe conditions is obtained.

The step size as well as the corresponding OF are updated as the line search approach is repeated in a loop until a maximum of five function evaluations is reached. The best step size is the step size that gives the least OF and also satisfy the curvature condition.

The best step size is used to update $\vec{x}_0$ using Equation 2.28

Damping Factor Estimation

After line search, the updated deformation parameters are used to estimate reservoir permeability which is then used to generate simulated production data and a new OF is computed. The value of the damping factor is maintained at its initial value of 10^{-2} if the OF decreases sufficiently. But if it reduces but not up to half of its penultimate value, the damping factor is divided by 10. If the function does not decrease at all, damping factor is updated by multiplying it by 10.

The algorithm runs for many iterations with the deformation parameters and damping factor being updated accordingly until the stopping criteria is met.

3.3.6 Sample Application of DE in GDM

The DE algorithm was used to estimate the deformation parameters in GDM. The concept of the algorithm is explained bellow.

Population Initialization

DE algorithm being a global algorithm requires a population of candidate solutions as initial guess. Since normally, a set of initial guess is a vector, therefore, a population of it would be a matrix with each row of the matrix being a population member. The number of population member, N_p which is 16 is computed using Equation 3.8.

$$N_p = 4 + floor(3\ln(D)) \tag{3.8}$$

Where L_x is the length of $\vec{x}$. The word "*floor*" is a command used by MATLAB to round-off the value in the bracket down to the nearest whole number.

The population with size N_p x D was computed using their lower and upper bounds as in Equation 2.29. Each population member (target vector) was then used to estimate permeability and OF is computed. The population member with the least OF is the current best member and the least OF is the current best OF.

Mutation

The mutation of each population member is computed using the mutation variant "*DE/best/1*" to generate a mutant vector. The mathematical formulation is given in Equation 2.31. Where the differential weight, $F = 0.85$.

Cross over

Crossover is done for each population member to generate a trial vector. The elements of this vector would either come from the target vector or mutant vector. To know which vector would donate to the trial vector, a vector of random numbers between 0 and 1 is generated. The size of the vector is the same as that of the trial and mutant vector. Each random number in the vector is compared with the cross-over probability, $C_R = 0.95$. If the first random number is less than C_R, first element in the mutant vector will cross-over to be the first element in the trial vector, but if otherwise, the target vector element will cross-over to the trial vector. The crossover process is repeated for all the elements in the target and mutant vectors and subsequently for all the population members. This variant of crossover is binomial.

Selection

In this part of the algorithm, the target or trial vector is selected to form the new population. Selection is also done for each target vector and its corresponding trial vector. They are used to estimate permeability and OF is computed. If the OF value of the trial vector is less than that of the target vector, the trial vector is selected, otherwise, the target vector is selected.

The mutation, crossover and selection processes are repeated for several iterations until the stopping criterion for each deformation stage is met. During all the computations, the overall best population member and OF are updated.

3.3.7 Sample Application of PSO in GDM

The PSO algorithm was used to optimize the deformation parameters in GDM. PSO algorithm also uses a population (swarm) of candidate solutions particles) as initial guess.

Population initialization

The initialization of population in PSO algorithm was done exactly as discussed in Subsection 3.3.6. Velocities of particles were initialized to zero.

Determine the Best Particle

Each particle of the population is used to estimate permeability and OF is estimated. The particle that gives the least OF is tagged the best particle in the population with other particles still retaining their positions and OFs as the current local best. The position and OF of the best particle are assigned the swarm's best position and OF.

Velocity Update

The particle's velocity is updated using Equation 3.9.

$$\vec{v} = w\vec{v} + c_1\vec{r}_1(\vec{p} - \vec{x}) + c_2\vec{r}_2(\vec{q} - \vec{x}) \tag{3.9}$$

Where $\vec{v}$ is particle's velocity, weight, $w = 0.7$, cognitive parameter, $c_1 = 1$, and social parameter, $c_2 = 1$. $\vec{p}$ is the particle's best position, $\vec{q}$ is the swarm's best position and $\vec{x}$ is the particle's current position. $\vec{r}_1$ and $\vec{r}_2$ are vectors of randomly-generated numbers between 0 and 1.

Position Update

The position of a particle is updated based on its local best position and the swarm's best position. This is done for every particle in the swarm using Equation 3.10.

$$\vec{x}_{iter+1} = \vec{x}_{iter} + \vec{v}_{iter} \tag{3.10}$$

It is ensured during the update that the new position does not fall below or above the lower and upper bounds respectively. The new swarm is evaluated by generating simulated production data and estimating the OF for all particles. If the new OF of a particle is less than its old one, the new OF is updated as its local best OF. The swarm's best position and OF are updated accordingly. The velocity and position updates are repeated for several iterations until the stopping criterion for each deformation stage is met.

3.3.8 Sample Application of SA in GDM

The SA algorithm was written in MATLAB programming language as an in-built function. This function was used to estimate the deformation parameters in GDM. Initial guess of the deformation parameters was computed using Equation 3.2.

SA Algorithm Options

These are some of the SA algorithm options (functions) adopted in this study. Other option parameters are defaulted. Note that the functions are invoked by putting "@" in front of their names.

1. Annealing function

   Annealing function is used to generate new $\vec{x}$ for the next iteration. The names of the available function are:

   (a) *annealingfast*: The step has length temperature, with direction uniformly at random. This is the default.

   (b) *annealingboltz*: The step has length square root of temperature, with direction uniformly at random.

   (c) *myfun*: This function allows the user to update $\vec{x}$ using a custom annealing algorithm, *myfun*.

   The default annealing function was used in this study.

2. Temperature function

Temperature function specify how the temperature will be lowered at each iteration while the algorithm is running. The names of the available temperature functions are:

(a) *temperatureexp*: This function updates the temperature schedule with the formulation below. This is the default function.

$$T = 0.95^{\gamma} T_0 \tag{3.11}$$

(b) *temperaturefast*: This function updates the temperature schedule with the formulation below.

$$T = \frac{T_0}{\gamma} \tag{3.12}$$

(c) *temperatureboltz*: This function updates the temperature schedule with the formulation below.

$$T = \frac{T_0}{\ln(\gamma)} \tag{3.13}$$

where T_0 and T are the initial and current temperatures, respectively and γ is the iteration number until re-annealing.

(d) *myfun*: This function allows the user to update the temperature schedule with a custom function, *myfun*.

In this study, the default temperature function was used.

3. Acceptance function

This function is used by the algorithm to determine if a new point is accepted or not. The names of the available acceptance functions are:

(a) *acceptancesa*: This is the simulated annealing acceptance function and also the default. It involves estimating an acceptance probability which recommends whether the new $\vec{x}$ should be accepted or not. For example, if the acceptance probability is

- $1:$ the new $\vec{x}$ must be accepted (the new solution is better).
- $0:$ the new $\vec{x}$ must be rejected (the new solution is infinitely worse).
- $0.5:$ the new $\vec{x}$ may or may not be accepted.

The acceptance probability is given as:

$$\frac{1}{1+\exp\left(\frac{\Delta\Phi}{T}\right)} \tag{3.14}$$

Where $\Delta\Phi$ is the difference between the new and old OF. Since T is always positive, larger $\Delta\Phi$ leads to smaller acceptance probability. If $\Delta\Phi$ is negative which signifies reduction in OF, smaller temperature leads to higher acceptance probability. Once the probability is estimated, it is compared to a randomly-generated number between 0 and 1.

   (b) *myfun*: This function allows the user to define an acceptance condition with a custom function, *myfun*.

   The simulated annealing acceptance function was used in this study.

4. Initial temperature

   Initial value of temperature, $T_0 = 300$. The algorithm expands the scalar initial temperature into a vector with same length as $\vec{x}$.

5. Re-anneal interval

   The re-anneal interval represents the number of $\vec{x}$ accepted before re-annealing. Re-annealing interval of 100 was used in this study and this is the default.

Several iterations were run until the stopping criterion for each deformation stage is met.

CHAPTER 4

RESULTS AND DISCUSSION

In this chapter, the results obtained from the geostatistical modeling, reservoir modeling and the history matching process are presented.

4.1 Kriging

Figures 4.1, 4.2, 4.3 and 4.4 are the kriged porosity, kriged porosity variance, kriged permeability and kriged permeability variance, respectively. The procedure of kriging has been discussed in Chapter 3, Subsection 3.1.2. In Figures 4.2 and 4.4, the blue-colored zones indicate areas of low estimation (interpolation) error. The low error is due to the fact that the eighty-five wells from which porosity and permeability were measured are clustered around that area thereby increasing the reliability of the interpolation.

Figure 4.1: Porosity kriging.

Figure 4.2: Porosity kriging variance.

Figure 4.3: Permeability kriging.

Figure 4.4: Permeability kriging variance.

4.2 Stochastic Simulation

Figures 4.5 to 4.19 are the fifteen permeability realizations to be linearly combined in the GDM of history matching. The process of obtaining these realizations has been discussed in Chapter 3, Subsection 3.1.3.

Figure 4.5: Permeability realization 1.

Figure 4.6: Permeability realization 2.

Figure 4.7: Permeability realization 3.

Figure 4.8: Permeability realization 4.

Figure 4.9: Permeability realization 5.

Figure 4.10: Permeability realization 6.

Figure 4.11: Permeability realization 7.

Figure 4.12: Permeability realization 8.

Figure 4.13: Permeability realization 9.

Figure 4.14: Permeability realization 10.

Figure 4.15: Permeability realization 11.

Figure 4.16: Permeability realization 12.

Figure 4.17: Permeability realization 13.

Figure 4.18: Permeability realization 14.

Figure 4.19: Permeability realization 15.

4.3 Reservoir Modeling

In this section, the properties of the 3D synthetic model used in this study are presented.

4.3.1 Well Locations

Figures 4.20 and 4.21 show the planar view of well locations with the true porosity and the permeability distributions of the reservoir's first layer, respectively.

Figure 4.20: Planar view of wells locations and porosity distribution.

Figure 4.21: Planar view of wells locations and permeability distribution.

4.3.2 Reservoir Permeability and Porosity

Figures 4.22 and 4.23 are the true permeability and porosity fields of the 3D synthetic reservoir model. In reality, the true permeability and porosity fields of a reservoir are unknown. But the true permeability field shall be used for comparison with the estimated permeability fields obtained from the algorithms in all the realizations considered in this study.

Figure 4.22: Reservoir true permeability distribution.

Figure 4.23: Reservoir true porosity distribution.

4.4 History Matching

In this section, the outcomes of the history matching process in terms of how the OF was reduced, how the measured water cut was matched and the optimized permeability are presented. Only the best, the median and the worst realizations for each algorithm out of the seven realizations are presented.

4.4.1 Objective Function Decay

Figures 4.24 to 4.26 show how the OF is reduced by the algorithms during optimization. The vertical axis represents the natural logarithm of the OF value. This was used to show the distinction in the variability of the OF decay. Figure 4.24, 4.25 and 4.26 are the best, the median and the worst reduction of the OF for all the algorithms, respectively. There was sharp decay at the beginning of the optimization, as expected. The rate of decay however reduced as the optimization approaches convergence. All the global algorithms reduced the OF lower than LM algorithm at the end of the optimization.

Figure 4.24: Best OF decay.

Figure 4.25: Median OF decay.

Figure 4.26: Worst OF decay.

4.4.2 Water Cut Match

Producer 1

Figures 4.27 to 4.29 show how the measured water cut in Producer 1 was matched by the simulated water cut obtained from the optimum solution of the algorithms. Figure 4.27, 4.28 and 4.29 are the best, the median and the worst match of the measured and simulated water cut from the optimum solution of the algorithms.

Figure 4.27: Best match of water cut from Producer 1.

Figure 4.28: Median match of water cut from Producer 1.

Figure 4.29: Worst match of water cut from Producer 1.

Producer 2

Figures 4.30 to 4.32 show how the measured water cut in Producer 2 was matched by the simulated water cut obtained from the optimum solution of the algorithms. Figure 4.30, 4.31 and 4.32 are the best, the median and the worst match of the measured and simulated water cut from the optimum solution of the algorithms.

Figure 4.30: Best match of water cut from Producer 2.

Figure 4.31: Median match of water cut from Producer 2.

Figure 4.32: Worst match of water cut from Producer 2.

Producer 3

Figures 4.33 to 4.35 show how the measured water cut in Producer 3 was matched by the simulated water cut obtained from the optimum solution of the algorithms. Figure 4.33, 4.34 and 4.35 are the best, the median and the worst match of the measured and simulated water cut from the optimum solution of the algorithms.

Figure 4.33: Best match of water cut from Producer 3.

Figure 4.34: Median match of water cut from Producer 3.

Figure 4.35: Worst match of water cut from Producer 3.

Producer 4

Figures 4.36 to 4.38 show the measured water cut in Producer 4 was matched by the simulated water cut obtained from the optimum solution of the algorithms. Figure 4.36, 4.37 and 4.38 are the best, the median and the worst match of the measured and simulated water cut from the optimum solution of the algorithms.

Figure 4.36: Best match of water cut from Producer 4.

Figure 4.37: Median match of water cut from Producer 4.

Figure 4.38: Worst match of water cut from Producer 4.

Producer 5

Figures 4.39 to 4.41 show how the measured water cut in Producer 5 was matched by the simulated water cut obtained from the optimum solution of the algorithms. Figure 4.39, 4.40 and 4.41 are the best, the median and the worst match of the measured and simulated water cut from the optimum solution of the algorithms.

Figure 4.39: Best match of water cut from Producer 5.

Figure 4.40: Median match of water cut from Producer 5.

Figure 4.41: Worst match of water cut from Producer 5.

Producer 6

Figures 4.42 to 4.44 show how the measured water cut in Producer 6 was matched by the simulated water cut obtained from the optimum solution of the algorithms. Figure 4.42, 4.43 and 4.44 are the best, the median and the worst match of the measured and simulated water cut from the optimum solution of the algorithms.

Figure 4.42: Best match of water cut from Producer 6.

Figure 4.43: Median match of water cut from Producer 6.

Figure 4.44: Worst match of water cut from Producer 6.

Producer 7

Figures 4.45 to 4.47 show how the measured water cut in Producer 7 was matched by the simulated water cut obtained from the optimum solution of the algorithms. Figure 4.45, 4.46 and 4.47 are the best, the median and the worst match of the measured and simulated water cut from the optimum solution of the algorithms.

Figure 4.45: Best match of water cut from Producer 7.

Figure 4.46: Median match of water cut from Producer 7.

Figure 4.47: Worst match of water cut from Producer 7.

Producer 8

Figures 4.48 to 4.50 show how the measured water cut in Producer 8 was matched by the simulated water cut obtained from the optimum solution of the algorithms. Figure 4.48, 4.49 and 4.50 are the best, the median and the worst match of the measured and simulated water cut from the optimum solution of the algorithms.

Figure 4.48: Best match of water cut from Producer 8.

Figure 4.49: Median match of water cut from Producer 8.

Figure 4.50: Worst match of water cut from Producer 8.

Producer 9

Figures 4.51 to 4.53 show how the measured water cut in Producer 9 was matched by the simulated water cut obtained from the optimum solution of the algorithms. Figure 4.51, 4.52 and 4.53 are the best, the median and the worst match of the measured and simulated water cut from the optimum solution of the algorithms.

Figure 4.51: Best match of water cut from Producer 9.

Figure 4.52: Median match of water cut from Producer 9.

Figure 4.53: Worst match of water cut from Producer 9.

Producer 10

Figures 4.54 to 4.56 show how the measured water cut in Producer 10 was matched by the simulated water cut obtained from the optimum solution of the algorithms. Figure 4.54, 4.55 and 4.56 are the best, the median and the worst match of the measured and simulated water cut from the optimum solution of the algorithms.

Figure 4.54: Best match of water cut from Producer 10.

Figure 4.55: Median match of water cut from Producer 10.

Figure 4.56: Worst match of water cut from Producer 10.

4.4.3 Estimated Reservoir Permeability

Figures 4.58 to 4.61 present the estimated reservoir permeability distribution obtained from the best realization of LM, DE, PSO and SA algorithms, respectively. To evaluate the performances of the algorithms in the estimation of the reservoir permeability, L1 norm of the estimate was computed. The norm is a measure of how close to the actual permeability is the estimated permeability. This implies that the algorithm with the lowest norm is the most accurate in optimizing the reservoir permeability. The L1 norm of permeability is:

$$k_{norm} = \frac{\Sigma \left| \ln\left(k_{meas}\right) - \ln\left(k_{est.}\right) \right|}{M} \tag{4.1}$$

Where k_{meas} and $k_{est.}$ are the true and estimated reservoir permeability distribution, respectively.

Figure 4.57: L1 norm of estimated reservoir permeability distribution.

Figure 4.57 presents the L1 norm of the estimated reservoir permeability distribution.

GDM + LM

In Figure 4.57, out of the seven realizations, GDM + LM gave the lowest norm of reservoir permeability estimate in only Realization 5. Therefore, for this realization, GDM + LM estimated the reservoir permeability most accurately.

Figure 4.58: Optimum estimated permeability field from GDM + LM.

GDM + DE

In Figure 4.57, out of the seven realizations, GDM + DE gave the lowest norm of permeability in only Realization 6. Therefore, for this realization, GDM + DE estimated the reservoir permeability most accurately.

Figure 4.59: Optimum estimated permeability field from GDM + DE.

GDM + PSO

In Figure 4.57, out of the seven realizations, GDM + PSO gave the lowest norm of permeability in Realizations 1 and 3. Therefore, for these realizations, GDM + PSO estimated the reservoir permeability most accurately.

Figure 4.60: Optimum estimated permeability field from GDM + PSO.

GDM + SA

In Figure 4.57, out of the seven realizations, GDM + SA gave the lowest norm of permeability in Realizations 2, 4 and 7. Therefore, for these realizations, GDM + SA estimated the reservoir permeability most accurately.

Figure 4.61: Optimum estimated permeability field from GDM + SA.

In summary, GDM + SA performed best in 3 realizations (that is, Realizations 2,4 and 7), GDM + PSO performed best in 2 realizations (that is, Realizations 1 and 3), GDM + DE and GDM + LM performed best in only Realizations 6 and 5, respectively. It suffices to say that GDM + SA performed best overall in estimating the reservoir permeability.

4.5 Evaluation Criteria

In this study, the performances of the optimization algorithms were evaluated based on three criteria which are:

1. Effectiveness: This is a measure of how close to the true value of a parameter is the estimated value. This implies that the algorithm whose estimated parameter (that is, simulated water cut) is closest to the true value (that is,

measured water cut) is the most effective and this is shown by the algorithm with the lowest L1 norm. The L1 norm was computed as:

$$d_{norm} = \frac{\Sigma \left| d_{meas} - d_{cal} \right|}{N} \tag{4.2}$$

Figure 4.62: Measure of effectiveness of the optimizers.

Figure 4.62 presents the L1 norm of the simulated water cut from all the producers. GDM + PSO gave the lowest L1 norm in all the realizations. Thus, in this study, GDM + PSO is the most effective algorithm for history matching purpose.

2. Efficiency: This refers to the number of function evaluations required to

reach a particular value of the OF. In this study, the value of the function used is 10% of its initial value. Thus, the algorithm that can reduce the OF to 10% of its initial value with the least number of function evaluations is the most efficient.

Figure 4.63: Measure of efficiency of the optimizers.

Figure 4.63 presents the number of time the OF was evaluated in order to reduce the function to 10% of its initial value. GDM + PSO was able to reduce the OF to 10% of its initial value with the lowest number of function evaluations in Realizations 2, 3, 6 and 7. GDM + LM algorithm was most efficient in Realization 1 while DE algorithm was most efficient in Realizations 4 and 5. Thus, in this study, GDM + PSO is the most efficient algorithm for history matching purpose.

3. Reliability: This criterion is a measure of how often an algorithm is able to reduce the OF to a preset value for different realizations considered in this study. In this study, the preset value is 0.5. Thus, the algorithm that most often reduce the OF to 0.5 in all the realizations is the most reliable.

Figure 4.64: Measure of reliability of the optimizers.

Figure 4.64 presents the OF value at the 4200^{th} function evaluation. All the global optimization algorithms were able to reduce the OF to 0.5 or below in the seven realizations. GDM + LM was able to reduce the OF below 0.5 in Realization 2 only. Thus, in this study, the global optimization algorithms are more reliable than the gradient-based algorithm.

CHAPTER 5

CONCLUSION AND RECOMMENDATION

In this chapter, relevant conclusions based on the outcome of this study were presented and possible recommendation for future research studies were given.

5.1 Conclusion

The following are the conclusions made from this study:

1. The GDM was adopted to history match the measured water cut in order to estimate the reservoir permeability.
2. A gradient-based algorithm (LM) was used to estimate the deformation parameters in GDM.
3. Three different global algorithms (DE, PSO, SA) were used separately to estimate the deformation parameters in GDM.

4. Seven realizations of the results were obtained.

5. In all the realizations, the global algorithms (DE, PSO and SA) were more effective, efficient and reliable than the gradient-based algorithm (LM).

6. PSO algorithm was the most effective, most efficient and most reliable amongst the global algorithms used in this study.

7. It has been established that using global optimization algorithms in GDM gave better history matching than using gradient-based algorithm.

5.2 Recommendation

The following recommendations are proposed for future studies:

1. 3D dataset should be used for geostatistical modeling.

2. More permeability realizations should be linearly combined in GDM to enhance its convergence.

3. Other global optimization algorithms should be used in GDM.

4. Sensitivity analysis should be conducted for each algorithm to determine the optimum parameter value to be used as well as the optimum number of sub-regions the reservoir should be divided into.

REFERENCES

[1] F. Roggero and L. Hu, “Gradual deformation of continuous geostatistical models for history matching,” in *SPE annual technical conference*, 1998, pp. 221–236.

[2] M. Le Ravalec-Dupin and B. Noetinger, “Optimization with the gradual deformation method,” *Mathematical Geology*, vol. 34, no. 2, pp. 125–142, 2002.

[3] J. Caers, “Efficient gradual deformation using a streamline-based proxy method,” *Journal of Petroleum Science and Engineering*, vol. 39, no. 1, pp. 57–83, 2003.

[4] L. Y. H. Ravalec-dupin, M. and B. Noetinger, “Sampling the conditional realization space using the gradual deformation method,” *Geostatistics*, p. 176186, 2000.

[5] Z. Ying and J. H. Gomez, “An improved deformation algorithm for automatic history matching,” *SCRF report 13*, 2000.

[6] L. Y. Hu, “Combination of dependent realizations within the gradual deformation method,” *Mathematical Geology*, vol. 34, no. 8, pp. 953–963, 2002.

[7] H. Mata-Lima, "Evaluation of the objective functions to improve production history matching performance based on fluid flow behaviour in reservoirs," *Journal of Petroleum Science and Engineering*, vol. 78, no. 1, pp. 42–53, 2011.

[8] D. S. Oliver and Y. Chen, "Recent progress on reservoir history matching: a review," *Computational Geosciences*, vol. 15, no. 1, pp. 185–221, 2011.

[9] M. Mezghani, F. Roggero *et al.*, "Combining gradual deformation and upscaling techniques for direct conditioning of fine scale reservoir models to dynamic data," *paper SPE*, vol. 71334, 2001.

[10] Y. Gautier, B. Nœtinger, F. Roggero *et al.*, "History matching using a streamline-based approach and gradual deformation," *SPE Journal*, vol. 9, no. 01, pp. 88–101, 2004.

[11] D. Busby, M. D. Feraille, V. Gervais-Couplet *et al.*, "Uncertainty reduction by production data assimilation combining gradual deformation with adaptive response surface methodology," in *EUROPEC/EAGE Conference and Exhibition.* Society of Petroleum Engineers, 2009.

[12] L. Reis, L. Hu, G. De Marsily, and R. Eschard, "Production data integration using a gradual deformation approach: Application to ar oil field (offshore brazil)," in *EUROPEC: European petroleum conference*, 2000, pp. 31–43.

[13] Y. Le Gallo and M. Le Ravalec-Dupin, "History matching geostatistical reservoir models with gradual deformation method," in *SPE annual technical conference*, 2000.

[14] V. Gervais-Couplet, Y. Gautier, L. Ravalec-Dupin, F. Roggero *et al.*, “History matching using local gradual deformation,” in *EUROPEC/EAGE Conference and Exhibition.* Society of Petroleum Engineers, 2007.

[15] A. A. Awotunde, “Relating time series in data to spatial variation in the reservoir using wavelets,” Ph.D. dissertation, Stanford University, 2010a.

[16] R. Bissell, “Calculating optimal parameters for history matching,” in *4th European Conference on the Mathematics of Oil Recovery*, 1994.

[17] J. R. P. Rodrigues, “Calculating derivatives for automatic history matching,” *Computational Geosciences*, vol. 10, no. 1, pp. 119–136, 2006a.

[18] P. Jacquard, “Permeability distribution from field pressure data,” *Society of Petroleum Engineers Journal*, vol. 5, no. 04, pp. 281–294, 1965.

[19] G. Chavent and M. Dupuy, “History matching by use of optimal theory,” 1975.

[20] B. Karcher and E. Aquitaine, “Use of parameter gradients for reservoir history matching,” 1989.

[21] M. K. Transtrum and J. P. Sethna, “Improvements to the levenberg-marquardt algorithm for nonlinear least-squares minimization,” *arXiv preprint arXiv:1201.5885*, 2012.

[22] T. Tan, "A computationally efficient gauss-newton method for automatic history matching," Society of Petroleum Engineers, Richardson, TX (United States), Tech. Rep., 1995.

[23] K. Levenberg, "A method for the solution of certain problems in least squares," *Quarterly of applied mathematics*, vol. 2, pp. 164–168, 1944.

[24] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *Journal of the Society for Industrial & Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.

[25] H. P. Gavin, "The levenberg-marquardt method for nonlinear least squares curve-fitting problems," pp. 1–17, 2013.

[26] H. Matsui and K. Tanaka, "Determination method of an initial damping factor in the damped-least-squares problem," *Applied optics*, vol. 33, no. 13, pp. 2411–2418, 1994.

[27] A. A. Awotunde, R. N. Horne *et al.*, "A multiresolution analysis of the relationship between spatial distribution of reservoir parameters and time distribution of well-test data," *SPE Reservoir Evaluation and Engineering*, vol. 14, no. 3, p. 345, 2011.

[28] A. A. Awotunde and R. N. Horne, "An improved adjoint-sensitivity computations for multiphase flow using wavelets," *SPE J*, vol. 17, pp. 402–417, 2012a.

[29] A. A. Awotunde, "Estimation of well test parameters using global optimization techniques," *Journal of Petroleum Science and Engineering*, vol. 125, pp. 269–277, 2015.

[30] X. Zhang, "Optimum damping factor for levenberg-marquardt algorithm with application to reservoir parameter estimation," *Master Thesis*, 2015.

[31] J. Zheng, Y. Liu, J. Li, D. Zhang, D. Chen *et al.*, "Application of permeability predictions in profile modification and water shutoff using genetic algorithms," in *SPE Asia Pacific Oil and Gas Conference and Exhibition.* Society of Petroleum Engineers, 2002.

[32] R. Portellaand and F. Prais, "Use of automatic history matching and geostatistical simulation to improve production forecast," in *Latin American and Caribbean Petroleum Engineering Conference.* Society of Petroleum Engineers, 1999.

[33] A. A. Awotunde, "Reservoir parameter estimation with improved particle swarm optimization," in *SPE Annual Technical Conference and Exhibition.* Society of Petroleum Engineers, 2012b.

[34] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *Journal of global optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[35] K. Y. Lee and M. A. El-Sharkawi, *Modern heuristic optimization techniques: theory and applications to power systems.* John Wiley & Sons, 2008, vol. 39.

[36] J. Zhang and A. C. Sanderson, "Da-aec: A differential evolution algorithm based on adaptive evolution control," *IEEE International Conference on Evolutionary Computation*, vol. 33, no. 13, pp. 3824–3830, 2007.

[37] C. C. Coello, G. B. Lamont, and D. A. Van Veldhuizen, *Evolutionary algorithms for solving multi-objective problems.* Springer Science & Business Media, 2007.

[38] J. Brest, A. Zamuda, B. Boskovic, M. S. Maucec, and V. Zumer, "High-dimensional real-parameter optimization using self-adaptive differential evolution algorithm with population size reduction," in *Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress on.* IEEE, 2008, pp. 2032–2039.

[39] W. Chen, Y.-j. Shi, and H.-f. Teng, "An improved differential evolution with local search for constrained layout optimization of satellite module," in *Advanced intelligent computing theories and applications. With aspects of artificial intelligence.* Springer, 2008, pp. 742–749.

[40] F. Neri and V. Tirronen, "Recent advances in differential evolution: a survey and experimental analysis," *Artificial Intelligence Review*, vol. 33, no. 1-2, pp. 61–106, 2010.

[41] J. Kennedy and R. . Eberhart, "Particle swarm optimization," in *IEEE International Conference on Neural Networks IV*, 1995, p. 19421948.

[42] Q. Bai, "Analysis of particle swarm optimization algorithm," *Computer and information science*, vol. 3, no. 1, p. p180, 2010.

[43] M. Dorigo, M. A. M. de Oca, and A. Engelbrecht, "Particle swarm optimization," *Scholarpedia*, vol. 3, no. 11, p. 1486, 2008.

[44] J.-C. ZENG and Z.-H. CUI, "A guaranteed global convergence particle swarm optimizer [j]," *Journal of computer research and development*, vol. 8, pp. 1333–1338, 2004.

[45] S. Kirkpatrick, M. Vecchi *et al.*, "Optimization by simmulated annealing," *science*, vol. 220, no. 4598, pp. 671–680, 1983.

[46] V. Černỳ, "Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm," *Journal of optimization theory and applications*, vol. 45, no. 1, pp. 41–51, 1985.

Vitae

- Name: Adewale Wasiu Adeniji
- Nationality: Nigerian
- Date of Birth: March 06, 1985
- Email: *tubabamne@yahoo.com*
- Permenant Address: 19, Community Close, Ikorodu, Lagos, Nigeria.
- Educational Background:

  King Fahd University of Petroleum & Minerals, Saudi Arabia (2012 - 2015)
  Master of Science in Petroleum Engineering
  Thesis Title: A Global Optimization Approach to the Gradual Deformation Method of History Matching.

  The Federal University of Technology, Akure, Nigeria (2005 - 2010)
  Bachelor of Engineering in Mining Engineering
  Project Title: Review of Nigerian Environmental Laws as Related to Mining from 1914 to Date.