Master's thesis

2019

Cuthbert Shang Wui Ng

Master's thesis

NTNU
Norwegian University of
Science and Technology
Faculty of Engineering
Department of Geoscience and Petroleum

Cuthbert Shang Wui Ng

Using the Least-Squares Monte Carlo Algorithm to Optimize IOR Initiation Time

June 2019

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of
Science and Technology

Using the Least-Squares Monte Carlo Algorithm to Optimize IOR Initiation Time

Cuthbert Shang Wui Ng

Petroleum Engineering
Submission date: June 2019
Supervisor: Dr. Reidar Brumer Bratvold, IGP
Co-supervisor: Dr. Aojie Hong, UiS

Norwegian University of Science and Technology
Department of Geoscience and Petroleum

Abstract

The application of the Least-Squares Monte Carlo (LSM) algorithm in the oil and gas industry is increasing. Its use with a production model has been demonstrated to be insightful by Hong et al. (2018) in terms of optimizing the initiation time of an Improved Oil Recovery (IOR) process. This demonstration also reflects the application of decision analysis (DA) in solving the IOR initiation time problem, which is a sequential decision problem in reservoir management. In this context, DA provides a framework that can systematically address the sequential decision problem in reservoir management and generate insights in reservoir decision making.

The production model used in Hong et al. (2018) was the two-factor production model, which is developed by Parra-Sanchez (2010). This production model is a decline curve-based model and thus, it is computationally attractive. Additionally, it is formulated in terms of the recovery factor of a recovery phase. In this context, for each phase, this model depends on two parameters, namely theoretical ultimate recovery factor and time constant (Parra-Sanchez, 2010). Aside from this, pertaining to the use of LSM algorithm, the state variables used are generally modeled as Markovian processes (Longstaff and Schwartz, 2001; Smith, 2005; Willigers and Bratvold, 2009). However, in Hong et al. (2018), the state variable cannot be modeled as a Markovian process (the measured oil production rate is used as a state variable in this case and the details will follow later). Therefore, Hong et al. (2018) have slightly modified the LSM algorithm to handle non-Markovian processes.

The modified LSM algorithm used by Hong et al. (2018) as well as in this work is an approximate dynamic programming (ADP) approach that can provide a near-optimal solution to the IOR initiation time problem. The need for approximation stems from the fact that dynamic programming (DP) suffers from the curse of dimensionality when the state space grows. We call this ADP as a Sequential Reservoir Decision Making (SRDM) approach. SRDM is also referred to as a method in which future learning is considered in reservoir decision making. Besides that, for any sequential decision problem, taking uncertainty and information into account is important to support a person's decision making. Regarding this, Closed-Loop Reservoir Management (CLRM) has been a state-of-the-art method to solve the IOR initiation time problem. However, in this context, CLRM yields a suboptimal solution as compared to SRDM. This is because CLRM only considers outcomes and decisions based on current information whereas SRDM considers outcomes and decisions based on current and future information (Hong et al., 2018). In other words, as compared to CLRM, SRDM captures the additional value of learning. In this aspect, the value of learning can be estimated by the Value-Of-Information (VOI) framework, which is a robust tool used for decision analysis (Hong et al., 2018).

There are several works done and presented for discussion. These include the replication of the use of the two-factor production model with the modified LSM algorithm as shown in Hong et al. (2018). Its purpose was to develop the present author's understanding and validate the

implementation of the modified LSM algorithm. A sensitivity analysis was conducted to verify if using a richer (more terms as well as nonlinear terms) regression function in the modified LSM algorithm would provide significant improvements to the results. Then, sensitivity analysis on certain parameters related to the algorithm was also performed to generate some useful insights regarding the IOR initiation time problem. This work was also adding to the work by Hong et al. (2018) by including the uncertainties in economic parameters in the modified LSM algorithm that was not done in Hong et al. (2018). Additionally, the application of this modified algorithm with a reservoir simulation model was illustrated. This brief illustration aims at showing the applicability of the modified LSM method with a different type of production model and providing a foundation on which further works can be developed upon.

Preface

This thesis is written to fulfill the partial requirements of the course of TPG 4920 – Petroleum Engineering, Master's Thesis offered by the Department of Geoscience and Petroleum, Norwegian University of Science and Technology (NTNU). The research work presented in this report is the continuation of the previous works done for TPG 4560 – Petroleum Engineering, Specialization Project (Ng, 2018a) and TPG 4565 – Petroleum Engineering, Specialization Course (Ng, 2018b). Although I have spent about 6 months to complete this thesis (from January 2019 to June 2019), it is in fact the collective effort of mine for about 10 months as I started working on the two abovementioned courses in August 2018.

The research topic of this thesis was proposed to me by Dr. Reidar Brumer Bratvold and Dr. Aojie Hong due to my interest in the field of the Improved-Oil-Recovery (IOR). Regarding this, they have been conducting research on the application of the Least-Squares Monte Carlo (LSM) algorithm in solving for the IOR initiation time. Consult Hong et al. (2018) for the respective details. However, there are still some further works that can be done on this topic and I am fortunate to be given the opportunity to perform these works. The petroleum industry has begun to pay more attention to LSM, but it has yet been widely discussed in the oil and gas related papers. This could be proven when I made a recent search for "Least-Squares Monte Carlo" at the online library of the Society of Petroleum Engineers (SPE), OnePetro, the search returned only about 30 results. Thus, through this research work, some useful insights would hopefully be generated to the further development of the application of LSM in the petroleum industry. For those who are reading this thesis, I hope that it can serve as an eye-opener for you regardless of your academic majors.

Trondheim, 11 June 2019

Cuthbert Shang Wui Ng

Acknowledgement

This research work would not be able to be successfully completed without the helps from some individuals. Thus, I would like reserve this page in this report as my token of appreciation for them. First of all, I would like to express my greatest gratitude to my supervisors, Dr. Reidar Brumer Bratvold and Dr. Aojie Hong for their guidance, suggestion, care, support, and inspiration. They have spent a lot of times to answer my inquiries, discuss the progress of my works, and provide insightful comments to improve my work. Without them, I would not be able to overcome the challenges arisen during the period I worked on this thesis. It has truly been a fabulous learning experience to have worked under their supervision.

Certainly, I must dedicate my special thanks to my parents, Dominic and Anna, and my siblings, Clarence and Cornelius for their unconditional love and support for me. Besides that, I have also been helped and motivated by many friends and colleagues. The list of names is too long to be mentioned here. However, you know who you are. Every word of encouragement from you has undoubtedly helped me to move a step forward in the odyssey of completing this thesis. Most importantly, I would also like to thank Mr. Ahgheelan Sella Thurai for his time and effort in proofreading my thesis. His comments and suggestions are highly appreciated. In addition, I would like to acknowledge the support provided by the staffs at the Department of Geoscience and Petroleum at NTNU. Last but not least, I wholeheartedly thank PETRONAS because without PETRONAS, I would not be able to indulge in any petroleum engineering related work and achieve this milestone today.

Dedication

"In nomine Patri et Filli et Spritus Sancti, Amen.
Thank you, Lord for Your blessings on me throughout this challenging period.
Without You, it is impossible for me to complete this thesis.
I, hereby, dedicate this work to You.
Jesus, I trust in You, Amen."

Table of Contents

List of Figures

List of Tables

Nomenclature

A	= random variables
a	= slope for a regression function (calibration of OU parameters)
α_0	= y-intercept for a regression function
α_1	= slope for a regression function
B	= random variables
b	= y-intercept for a regression function (calibration of OU parameters)
β	= regression coefficients
$\boldsymbol{\beta}$	= vector of regression coefficients
$\tilde{\beta}$	= regression coefficients
$\breve{\beta}$	= regression coefficients
C	= chance / uncertainty
CAPEX	= capital expenditure
CF	= cumulative cashflow
D	= decision
δ	= approximation error in the least-squares regression
dz_t	= increment of standard Brownian motion (lognormal distribution)
ε	= random error term
E_R	= recovery factor
ΔE_R	= increment of recovery efficiency
ENPV	= expected net present value
EV	= expected value
$f(\cdot)$	= function (the sum of the squared vertical deviations)
i	= index of data point (for regression analysis) / also as index of time step in Chapter 6
j	= index of time step
k	= index of time step in Chapter 6
κ	= speed of mean reversion (lognormal distribution)
$L(\cdot)$	= Laguerre polynomials
m	= number of data points (for regression analysis)
μ	= long-term mean
N	= number of realizations for Monte Carlo Simulation
n	= number of years
N_o	= original oil in place, OOIP
NPV	= net present value
n_t	= number of recovery phases
OPEX	= operating cost
P	= generic price
P_o	= oil price
$\overline{P}$	= equilibrium oil price
π	= logarithm of a modeled parameter (stochastic process)
$\overline{\pi}$	= long-term mean for logarithm of a modeled parameter (stochastic process)
ψ	= correlation coefficient

q	= measured oil production rates
$\mathbf{q}$	= vector of measured rates
q_o	= oil production rate
R1	= subscript representing primary recovery
R2	= subscript representing secondary recovery
RF	= regression function
r	= discount rate
SD	= standard deviation
σ	= measure of process volatility
σ_π	= volatility for logarithm of a modeled parameter (stochastic process)
S	= a stochastic process
t	= time
Δt	= length of time step
τ	= time constant
θ	= speed of mean reversion
W_t	= Brownian motion, $dW_t \sim N(0,\sqrt{dt})$
x	= dependent variable (for regression analysis)
y	= independent variable (for deterministic linear regression)
$\tilde{y}$	= expected net present value of an alternative (regression analysis)
Y	= independent variable (for probabilistic linear regression)

Chapter 1

Introduction

Good reservoir management is essential for creating value from oil and gas reservoirs. In general, reservoir management can be defined as the application of available technology, financial and labor resources to maximize the economic performance and the recovery of a reservoir (Wiggins and Startzman, 1990). In this aspect, Wiggins and Startzman (1990) has also stated that "reservoir management can be understood as a series of operations and decisions from the initial stage of discovery of a reservoir to the final stage of abandonment of that reservoir". Oil and gas companies are maximizing the value creation from the reservoirs by making a good decision. A lot of decision problems in reservoir management are sequential[1]. This sequential nature enables the companies to learn from the information gathered, support their future decision making, and maximize the value creation from the reservoirs. Hence, reservoir management is about making decisions. To successfully conduct a reservoir management, decision analysis (DA) has been recommended due to its numerous advantages (Evans, 2000). Fundamentally, Howard (1988) has opined that "DA involves a systematic methodology to convert opaque[2] decision problems into transparent[3] decision problems through a series of transparent procedures". In terms reservoir management, DA is illustrated as a systematic means of evaluating different alternatives and finding the optimal one with the goal of maximizing the net present value (NPV) of a project (Evans, 2000). Pertaining to this, the application of DA in reservoir management (Hong et al., 2018) is discussed and extended in this work.

Improved Oil Recovery (IOR) is an important phase in reservoir management. It is a method applied in the oil and gas industry to recover or produce additional amount of hydrocarbon beyond primary recovery (Schlumberger, 2019b). The mechanism of primary recovery includes natural drive, gas cap drive, gravitational drainage, and so forth (Lake, 1989). Besides that, IOR methods include waterflooding, gas-flooding, polymer flooding, surfactant flooding, CO_2 flooding, thermal flooding, and so on. IOR is implemented when the primary recovery mechanism is insufficient to produce the hydrocarbon from the remaining reserve. However, IOR process might not be conducted if it is not economical. Therefore, there is a very important and practical decision regarding the implementation of IOR, namely when is the best time to initiate an IOR process? Scheduling an IOR process is essential because it pertains to the development plan of a field (Hong et al., 2018). For example, by having determined the optimal time to initiate an IOR, the oil company can estimate the period of the license for production that it should get in order to maximize the NPV of the related project.

---

[1] A decision problem is sequential if there is a series of decisions to be made as time proceeds. "Sequential decision making" describes the situation of solving a sequential decision problem.
[2] "Opaque" means difficult to understand in this context (Howard, 1988).
[3] "Transparent" means easier to understand or clear in this context (Howard, 1988).

Hong et al. (2018) built a useful and tractable decision model for analyzing an IOR initiation problem. In this context, they applied the two-factor production model proposed by Parra-Sanchez (2010). Hong et al. (2018) explained that "useful" implied that the model was relevant to provide insightful results upon the resolution of the decision whereas "tractable" denoted that the relevant analysis was computationally attractive. The details of the formulation of the production model will follow later. Besides that, regarding the production model parameters, there is a lack of knowledge of the reservoir properties and the impacts of recovery mechanism (Parra-Sanchez, 2010; Hong et al., 2018). Probability distributions are then used to capture the uncertainty and have been assigned to each of uncertain production model parameters (Hong et al., 2018).

Apart from these, Hong et al. (2018) explained that the application of two-factor production model did not consider the effect of future information. This effect can be understood as the impact of learning over time (Hong et al., 2018). They further mentioned that as production began, more data would be available and used to update the knowledge to support the decision (Hong et al., 2018). In the oil and gas industry, there are two different approaches used to include the impact of information: Closed-Loop Reservoir Management (CLRM) and Sequential Reservoir Decision Making (SRDM). These approaches serve as a priori analysis in which each of them is implemented before additional information is collected. Thus, whenever additional data is collected, both CLRM and SRDM can be readily applied to make use of these data. Although both approaches involve the process of data assimilation[4], the optimization aspect differs. The CLRM (Chen et al., 2009; Jansen et al., 2009; Wang et al., 2009) has been the state-of-the-art method used to integrate the effect of information on decision problems related to reservoir management (Hong et al., 2018). The CLRM approach includes a cycle of consistently updating a relevant model via history matching[5] to achieve optimization of production when there is additional data (Hong et al., 2018). By having matched the historical data, decision making is involved to determine the optimal production strategy. This will be explained more comprehensively later along with the theoretical and implementation details of the CLRM approach. Additionally, solving a CLRM problem[6] requires forward calculation and it will be discussed later as well.

SRDM is an approach that explicitly exploits the full structure[7] of a sequential decision problem (Hong et al., 2018; Alyaev et al., 2018). Therefore, as discussed in Hong et al. (2018), it would lead to additional value of learning as compared to CLRM. The CLRM is suboptimal

[4] Aanonsen et al. (2009) has counseled that a person aims to integrate empirical information into a numerical model in the work of data assimilation. Additionally, Hermant and Oilver (2011) has stated that if this integration of information is carried out sequentially in time, the process is known as "data assimilation".

[5] History matching involves the adjustment of a reservoir model to replicate the historical behavior of a reservoir (Schlumberger, 2019a). It is equivalently data assimilation as it is done sequentially in time (Hermant and Oilver, 2011). With respect to this, the ensemble Kalman filter (EnKF) is used for history matching. Refer to Aanonsen et al. (2009) and Hermant and Oilver (2011) for the details regarding the use of EnKF in history matching.

[6] A simple CLRM problem can be represented by a decision tree. It is illustrated later in Fig. 1.

[7] The full structure of a decision tree represents all the sequences of the uncertainties and decisions of the tree (Bratvold and Begg, 2010; Hong et al. 2018). A decision problem that can be represented by this full structure of decision tree is addressed as the SRDM problem in this work.

because it only takes the effect of current (at the time when the current decision is being made) information into account when making a decision. It does not consider the effect of information that will be available for future decisions. This explanation highlights the difference between the optimization aspect of SRDM and that of CLRM. The SRDM approach can be structured as a decision tree[8] and solved by using the standard decision tree roll back approach. However, for a complicated decision problem with a lot of uncertain outcomes, alternatives, and decision points, the decision tree would grow exponentially and not be very useful for communication. Furthermore, for such rich decision contexts, the decision tree roll-back procedure is not very efficient. This phenomenon is known as the "curse of dimensionality" (Powell, 2011). More details on this will follow later. To mitigate this curse, a method of approximate dynamic programming (ADP), specifically the Least-Squares Monte Carlo (LSM) algorithm, which was proposed by Longstaff and Schwartz (2001), is used. Due to approximation, LSM may generate a near-optimal solution[9]. Its main drawback is that it still suffers the curse of dimensionality in the action space in which the computation effort of LSM would increase exponentially with the number of both alternatives and decision points[10] (Powell, 2011; Hong et al. 2018).

Following this introduction, the theoretical basis regarding CLRM and SRDM [11] is discussed. After that, the background and the general framework of the LSM algorithm is explained. Thereafter, the author presents the formulation of the two-factor production model, the economic model used in Hong et al. (2018), and the steps of implementing the modified LSM algorithm. The results of the replication of the implementation of the modified LSM algorithm with the two-factor production model are then illustrated and compared to those shown in Hong et al. (2018). This replication contributes to a verification of the implementation of the modified LSM algorithm[12]. Next, the author conducts sensitivity analysis on different aspects, such as the choice of regression function, the production model parameters, the economic parameters, the number of decision points, the number of data points, and the standard deviation (SD) of measurement error, and discusses the results obtained from the corresponding sensitivity analysis. The results of these sensitivity analyses provide additional insights for the decisions of when to initiate IOR methods. Then, the author demonstrates an extension of the modified LSM algorithm by including the uncertainties in economic parameters and discusses the respective results. This extension is also implemented for the CLRM approach. Again,

[8] A decision tree is an approach used to visualize and solve a sequential decision problem (Bratvold and Begg, 2010; Hong et al. 2018). Refer to Bratvold and Begg (2010) for the detailed use of decision tree for calculation and communication.

[9] It is very computationally prohibitive to directly solve for the true optimum of a complex decision problem. The best approach to solve for the true optimal solution is hitherto by approximation in which the accuracy can be improved to enable the estimated solution to be as close as possible to the true solution. This improvement process is addressed as the convergence to the global optimum. In this work, "near-optimal" is sometimes interchangeably referred to as "optimal".

[10] Alternatives are the available options to be chosen in a decision making. Decision point means the point of time when a person is making a decision. For example, when the decision point is placed at Year 2, it means the person has to make a decision at Year 2.

[11] This is to reiterate that in Hong et al. (2018), the modified LSM algorithm is applied to solve the SRDM problem through approximation with the help of regression analysis. For CLRM, it involves the use of regression analysis in the form of forward calculation. The details of the implementation of each approach follow later.

[12] The author also explains and replicates the implementation of CLRM as presented in Hong et al. (2018). However, emphasis is placed on the implementation of the modified LSM in this work.

including economic uncertainties, the difference between SRDM and CLRM solutions are also illustrated and discussed. The author then uses a different problem setting to further illustrate the suboptimality of CLRM as compared to SRDM. The author also provides a brief demonstration on how the modified LSM algorithm can be applied in tandem with a simple reservoir simulation model to provide useful insights. The author delivers an overall discussion about the works presented and some suggestions for further works. Finally, conclusions follow.

Chapter 2

Optimization Methods for Data Assimilation Application

Uncertainty is inseparable from all significant decisions (Bratvold and Begg, 2010). In general, when we are investigating a problem with uncertain outcomes, we can reduce uncertainty by gathering more information (Bratvold and Begg, 2010). In reservoir management, there are many decision problems which involve a lot of uncertainties. This is also why Thakur (1996) opined that the data collection and management were vital to ensure the success of reservoir management. Therefore, one of the important questions regarding the use of information in reservoir management is how the data collected can be integrated to optimize decision policies[13]?

Production optimization is an essential part of reservoir management. A lot of production optimization problems involve sequential decision making because there is a series of decisions to be made as the production time advances (Hong et al., 2018). When production continues, more information, such as production data, are gathered. Thereafter, the data can be applied to communicate the uncertainties via history matching (Hong et al., 2018). In other words, as information is gathered, we learn what we did not know before having the information. As we learn more, our uncertainty is reduced. Thus, the effect of information should be included, and it can be done through the respective implementation of CLRM and SRDM to support the decision making in production optimization. In addition, each of these approaches is in fact the representation of a type of decision policy. The following briefly explains these two approaches and their corresponding types of decision policies.

2.1 Closed-Loop Reservoir Management

CLRM approach has been a cutting-edge method that is used to solve a decision problem in reservoir management. This approach applies the mechanism of model-based optimization in tandem with the assimilation of data to maximize the value creation from a reservoir over the lifetime of production (Jansen et al., 2009). There are two fundamental steps pertaining to this approach. The first step is history matching (computer-assisted) in which data is assimilated and the second step is adjusting some control parameters based upon the history-matched model (updated model) to optimize production (Wang et al., 2009). Moreover, this approach gets its name "closed-loop" because it is done to complete a loop: whenever new data is collected, a reservoir or production model would be updated and reservoir performance optimization would be carried out (Jansen et al., 2009; Wang et al., 2009; Hong et al., 2018). For as long as new data is available, this loop would continue to optimize production settings.

[13] Decision policies can be understood as the solutions to a decision problem.

Hong et al. (2018) discussed the general framework of the CLRM approach by using production optimization as an example. Basically, optimizing the expected value (EV) over the realizations of a production model is done to determine the corresponding production strategy (Hong et al., 2018). The essence of this approach is that different sets of realizations[14] of the model are used based upon where the decision stage is. For instance, at the initial stage (no data is available yet), prior distributions of relevant reservoir properties is applied to develop the initial realizations (Hong et al., 2018). The initial production strategy can then be determined based on those realizations. At later stages where new data is available, the realizations of the model are updated and used to determine the new production strategy for the remaining production time (Jansen et al., 2009; Wang et al., 2009; Hong et al., 2018). From this framework, we can understand that the CLRM approach includes the impact of information, and learning from that information, by utilizing the additional data to update the model to produce an optimal policy based on the knowledge the decision makers have at any point in time.

Hong et al. (2018) argued that the CLRM approach is a naïve decision policy. A decision policy is naïve when a sequential decision making problem is solved based on a priori knowledge without accounting for the sequential information gathered (Martinelli et al. 2013; Alyaev et al., 2018). The CLRM approach only takes future decisions into account but does not consider any learning over time[15] (Alyaev et al., 2018). To include future learning, uncertainties have to be solved based on both current and future data. In CLRM, the uncertainties are only associated with the currently available data and there is no continuous resolution of future uncertainties. This formulation is clearly shown by using a schematic decision tree in Fig. 1 as shown below.

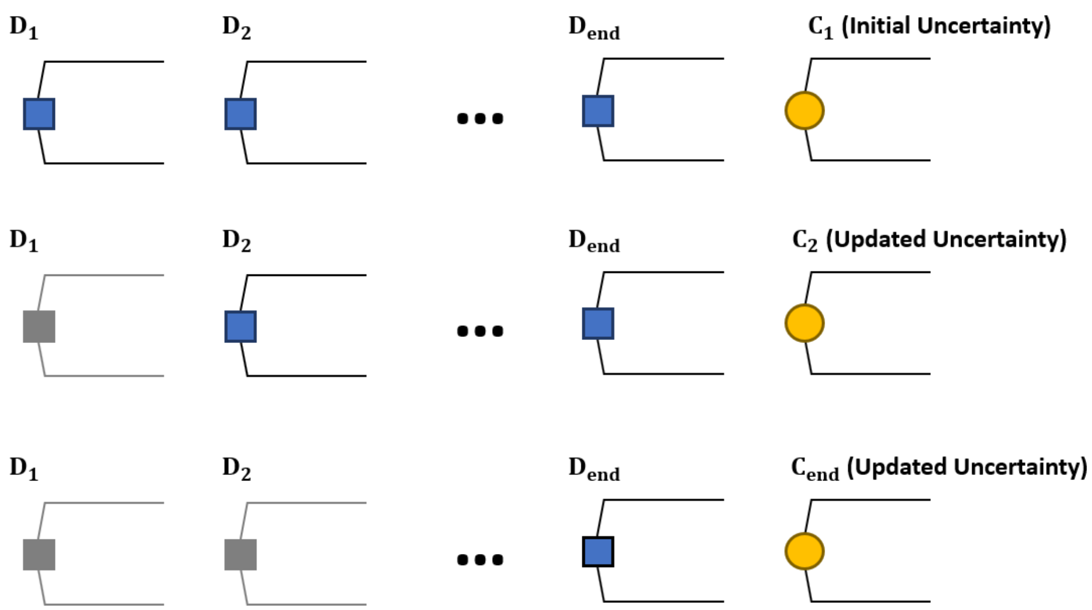


Fig. 1 – Schematic Decision Tree of the CLRM approach (Hong et al., 2018).

[14] Realization means a set of random variables or data. It is explained in detail later.
[15] It does not continuously resolve the uncertainties in future (Alyaev et al., 2018).

In Fig. 1, the square (denoted by the letter D) is the decision node whereas the circle (denoted by the letter C) is the chance node (also known as uncertainty node). As it can be observed, when a person makes a decision at the initial stage (indicated as D_1), he or she would use the initial uncertainty (indicated as C_1), which represents the currently available information. Only after he or she has made the decision at the initial stage and proceeded to next stage, the uncertainty would be updated accordingly as shown in Fig. 1. Hence, Fig. 1 illustrates that this approach only considers the effect of the information obtained before a decision is made but it does not take into account of the impact of the information that will be available on the future decisions. Thus, the remaining decisions are optimized by using the current information or data (Alyaev et al., 2018). This is why such approach is naïve. At each decision node, we naively assume that the information required for future decisions cannot be considered (Alyaev et al., 2018). In other words, we do not account for the fact that we learn more about the reservoir over time.

Through this decision tree illustration, it can be deduced that the CLRM approach does not represent the full decision structure in reservoir management and omits the learning over time. In this context, Hong et al. (2018) demonstrated that CLRM might lead to the suboptimal decision policy as compared to SRDM. This demonstration will be discussed later.

2.2 Sequential Reservoir Decision Making

Being different from CLRM, SRDM is an approach that represents the full decision structure of reservoir management. It is a farsighted policy because it solves a sequential decision making problem based upon both past and future information (Alyaev et al., 2018). Therefore, both uncertainties corresponding to the currently available data and future data are considered in SRDM. In this context, the future uncertainties are continuously resolved and conditioned on the information collected beforehand. To illustratively show the framework of SRDM, the decision tree for the approach is shown in Fig. 2. As it can be observed, it explicitly considers all possible information or data to yield an optimal policy. In this aspect, the optimal decision policy corresponding to SRDM can be solved by rolling back the decision tree.

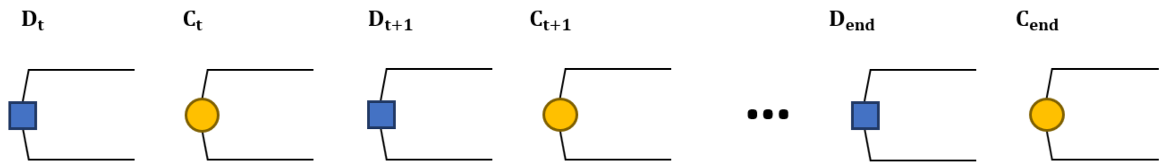


Fig. 2 – Schematic Decision Tree of the SRDM approach (Hong et al., 2018).

Based on the decision tree above, SRDM takes both the uncertainties corresponding to current and future data into account. Thus, at the decision stage of time step t, the decision maker makes a decision that is influenced by the uncertainty that has been resolved at time step t and the uncertainties which would be resolved in future (time step t +1 onwards until the end). Besides that, the responses corresponding to the decision nodes at different points of time are also integrated. With respect to this, SRDM exploits the full structure of a decision problem and thus, solving it would lead to the globally optimal decision policy (Hong et al., 2018). However, its drawback is its relatively more intensive computation as compared to CLRM (Hong et al., 2018).

2.3 Background of Least-Squares Monte Carlo (LSM)

In Dynamic Programming (DP)[16], the main problem is divided into smaller sub-problems which are solved backwards (Howard, 1960; Jafarizadeh and Bratvold, 2009). Therefore, a decision tree approach is a method of DP. In decision making, a decision tree is a useful tool used to solve a sequential decision problem (Bratvold and Begg, 2010). As discussed earlier, a SRDM problem can be expressed as a decision tree and solved accordingly. However, such an approach only works well with a limited number of possible outcomes, alternatives, and decision points. This is because the curse of dimensionality[17] would cause the decision problem to be computationally too expensive to be solved by using the decision tree approach.

LSM algorithm, which is an ADP, is constructed to overcome the curse of dimensionality (but not the curse of action space). The SRDM approach is utilizing the LSM algorithm to solve the decision problem. However, the LSM approach is still only useful in optimizing decision problems with limited number of alternatives and decision points. As the curse of action space occurs (increase in the number of alternatives or the number of decision points), the computational time of the LSM method would increase exponentially (Hong et al., 2018). This method was initially used to value American options[18] in the financial markets. It is essentially a combination of two steps, namely a Monte Carlo Simulation (MCS) step[19] and a Least-Squares step. Thus, it is important to understand the fundamentals of these two steps in order to appreciate the implementation of the LSM algorithm

2.3.1 Monte Carlo Simulation

Bratvold and Begg (2010) provides a good introduction of MCS with emphasis on its application in oil and gas industry. MCS is a very prevalent and robust method used in uncertainty analysis (Bratvold and Begg, 2010). In this context, it plays a pivotal role in decision analysis. This is due to the challenge of decision analysis in terms of the assessment of the uncertainty in the attributes[20] used to compute the values of decision options (Bratvold and Begg, 2010). By implementing MCS, the values of these attributes can be calculated from a model, which can relate the input variables to the attributes of interest (Bratvold and Begg, 2010).

In this context, MCS is termed to be the propagation of uncertainty from variables that can be assessed to variables used to make decisions (Bratvold and Begg, 2010). The "propagation of uncertainty" can be perceived by briefly explaining the example of production project in Smith

[16] Refer to Howard (1960) for the details of Dynamic Programming.

[17] The curse of dimensionality is made up of the curse of state space, the curse of outcome space, and the curse of action space (Powell, 2011). In the use of decision tree, when the numbers of uncertain outcomes, alternatives, and decision nodes increase, the curse of dimensionality would happen. Thus, the size of a decision tree would increase, and it would be computationally difficult to solve the tree.

[18] An option that can be exercised any time before expiry.

[19] The use of Monte Carlo Simulation is not only limited to the LSM method. It can be used in any ensemble-based method, including CLRM approach.

[20] Measurement scale used to quantify how well a decision option meets the given objective (Bratvold and Begg, 2010).

(2005). In the project, he aimed to maximize the economic performance of the project by selecting one of the three available options namely, to buy out the project (to fully own it), to continue with his current share of the project or to divest (to sell his share). Thus, he chose to use the net present value (NPV) as the attribute to determine the values of available options. However, the NPV could not be assessed directly. Thus, he implemented MCS to generate many samples of oil price and costs, which were the variables he used to calculate the NPV and determine the values of options.

Using the decision tree approach to solve a sequential decision problem is not viable because of the curse of dimensionality. The curse is indeed commonly encountered in the use of decision tree when it comes to the real option valuation (Smith, 2005). Therefore, the method of simulation has been proposed to mitigate it. Boyle (1977) suggested the use of MCS to solve the option valuation problem. It has been implemented to solve European options[21] for years (Stentoft, 2004). The curse of dimensionality is not incurred in valuing the European option despite using the uncertainty space with high order of magnitude (Stentoft, 2004). In this case, MCS can be aptly used to value options, which are path-dependent or have a lot of underlying uncertainties (Willigers and Bratvold, 2009). However, there is difficulty when MCS is applied to value American option. This is because while using the MCS to value American option, there is a simultaneous need to find out the optimal exercise policy (Stentoft, 2004). Thus, the LSM algorithm, which makes use of the forward modeling[22] of MCS, is suggested.

In general, in order to conduct MCS, a relevant model is required for the problem being investigated beforehand (Bratvold and Begg, 2010). Then, the uncertainty in the input variables is described in terms of the probability distributions (Bratvold and Begg, 2010). In this case, MCS randomly retrieves samples from the distributions and uses them to calculate the output variables based on the chosen model. Then, these output variables are stored for further analysis. The histogram of the output variables can be built and normalized to yield a probability distribution from which the statistical parameters, including means, variances, and so on, are computed (Bratvold and Begg, 2010). A set of input and output variables is known as realization and a loop of getting a realization is called as iteration (Bratvold and Begg, 2010).

[21] An option that can only be exercised at a predetermined time.

[22] Simulation of variables forward in time.

2.3.2 Principle of Least-Squares

Regression analysis is a statistical method that is utilized to investigate the relationship between two or more variables in a nondeterministic manner (Devore, 2010). These variables are categorized into two groups, namely the dependent variable and the independent variable. In general, the independent variable is denoted by x whereas the dependent variable is denoted by y. In this aspect, principle of least-squares is used to conduct the regression analysis. For LSM method, the linear regression analysis is most commonly done where the dependent variable is assumed to have a linear relationship with the independent variable (Longstaff and Schwartz, 2001; Smith, 2005; Jafarizadeh and Bratvold, 2009). In a deterministic model, when the relationship between the independent variable and the dependent variable is linear, such relationship is mathematically presented as shown below.

$$y = \alpha_0 + \alpha_1 x \tag{1}$$

where α_0 is the y-intercept and α_1 is the slope. This means that fixing the independent variable at a certain value will result in an observed value of the dependent variable (Devore, 2010). Thus, the dependent variable is associated with the independent variable. However, if this model is generalized to a probabilistic model, the EV of y (denoted as Y) is assumed to be a linear function of x (Devore, 2010). For a fixed value of x, its dependent variable is different from its EV by a random amount. Such random amount is represented by ε and known as random error term. Therefore, in a probabilistic linear regression model, there are parameters α_0, α_1, and ε in the regression function of relating y to x as described.

$$y = \alpha_0 + \alpha_1 x + \varepsilon \tag{2}$$

To relate Y to x, the equation is expressed as

$$Y = \alpha_0 + \alpha_1 x \tag{3}$$

The line corresponding to the equation above without the random error term is known as true regression line of Y (Devore, 2010). With the random error term, the points of (x, y) would scatter about the line. So, the principle of least squares is used to estimate the values of parameters α_0 and α_1 in Equation (3). Based on this principle, a regression line provides a good fit to the data if there is are small vertical deviations between the points and the line (Devore, 2010). Mathematically, the vertical deviation of a point (x_i, y_i) from the regression line is:

$$\text{height of point} - \text{height of line} = y_i - (\alpha_0 + \alpha_1 x_i) \tag{4}$$

The sum of the squared vertical deviations from the points (x_1, y_1) to (x_i, y_i) to the regression line is

$$f(\alpha_0, \alpha_1) = \sum_{i=1}^{m} [y_i - (\alpha_0 + \alpha_1 x_i)]^2 \tag{5}$$

The regression coefficients of α_0 and α_1 (that minimize the Equation (5)) are determined by finding the partial derivatives of $f(\alpha_0, \alpha_1)$ with respect to both α_0 and α_1 (Devore, 2010). Then, both of the partial derivatives are equated to be zero and these yield a set of normal equations (Devore, 2010). Solving the normal equations would result in the values of α_0 and α_1. Refer to Devore (2010) for the comprehensive explanation of this step.

However, the regression analysis can also be a bit more complicated when it comes to non-linear regression and multiple regression. For multiple regression, it involves more than one dependent variable and the interaction term between these dependent variables can be included into the regression function. Nonlinear regression is simply performing the same analysis by using higher order of polynomial functions or more complicated basis function, such as Hermite, Legendre, Chebyshev, Jacobi polynomials, and so forth (Longstaff and Schwartz, 2001). Finding the regression coefficient of those slightly sophisticated regression function is also done by using the same step as explained for the simple linear regression model. However, the sum of the squared vertical deviations from the points to the regression line is different depending on the type of regression function. Refer to Devore (2010) for the details of this step for more complicated regression function.

This is a brief explanation of how the least-squares method is applied in regression analysis. Although finding the regression coefficients can be more demanding as the type of regression function becomes more complicated, the whole regression analysis can be easily conducted with the help of a software package. For instance, in MATLAB R2019a (2019), regression analysis can be done by using the function `regress()`.

2.3.3 General Workflow of the LSM method

Jafarizadeh and Bratvold (2009) and Alkhatib et al. (2013) provided a meticulous step-by-step procedure of the LSM method. Fundamentally, the LSM method is approximating the conditional EV by using regression analysis. Its general workflow begins by performing the MCS to sample many possible outcomes[23] (can be known as realizations) and compute the objective function[24]. Then, in order to compute the optimal exercise policy at a decision node, the expected future values for each alternative (conditional on the resolution of all uncertainties up to that time) are determined by using the least squares regression (Jafarizadeh and Bratvold, 2009). The expected future values are also known as the continuation[25] values. After that, beginning recursively from the last decision node, the optimal policy is attained by choosing the alternative that produces the highest continuation value given known information.

[23] Sets of data which are usually used to compute the state variables which are used in regression analysis if these variables are not readily available (or are not sampled directly).
[24] Function or function value in which maximizing it is the purpose of the optimization. Objective function would be regressed on state variables during the regression analysis.
[25] It is essential to understand that the option of continuation here only applies to the option of continuing with current state of decision.

Chapter 3

Least-Squares Monte Carlo Algorithm for the IOR Initiation Time Problem

The application of the LSM method in the petroleum industry has been increasing for the past decade. Several studies have been conducted to demonstrate its robust application in the oil and gas industry (Smith, 2005; Willigers and Bratvold, 2009; Jafarizadeh and Bratvold, 2009; Jafarizadeh and Bratvold, 2012; Jafarizadeh and Bratvold, 2013; Alkhatib et al., 2013; Thomas and Bratvold, 2015; Hong et al., 2018). In reservoir management, IOR is implemented to increase hydrocarbon production over the lifetime of a field. In order to optimize an IOR process, one of the important criteria to be considered is the optimal time to start the IOR process. In this context, Hong et al. (2018) showed that optimizing the initiation time of an IOR method was a sequential decision making problem. The generalization of this decision problem is that a decision maker must decide if he or she should either keep the current phase of recovery or shift to the next phase of recovery (Hong et al., 2018).

With respect to this, Hong et al. (2018) demonstrated how the LSM algorithm could be used to solve this sequential decision problem. They applied this algorithm along with a simple production model and economic model to illustrate how the optimal IOR initiation time could be determined[26]. The production model used is the two-factor production model which was developed by Parra-Sanchez (2010). For simplicity, the use of either the modified LSM method (SRDM approach) or CLRM approach along with two-factor production model and economic model as shown in Hong et al. (2018) would be referred to as "HBL's model" in this work. This chapter mainly discusses the two-factor production model and the economic model implemented in the HBL's model and explains the problem setting of the case study in Hong et al. (2018) to illustrate how this model is replicated for further analysis.

3.1 Two-Factor Production Model

Parra-Sanchez (2010) developed this production model based on the exponential declines. This model is powerful because it can be used to include multiple phases of recovery to achieve the production optimization over the lifetime of the production (Parra-Sanchez, 2010; Hong et al., 2018). The general formulation of the recovery factor of this model is shown below.

$$E_R(t) = E_R^0 + (E_R^\infty - E_R^0) \times \left(1 - e^{\frac{-t}{\tau}}\right) \quad (6)$$

[26] To reiterate, Hong et al. (2018) also implemented the CLRM approach to solve the IOR initiation time problem. The implementation of CLRM in Hong et al. (2018) would be replicated as well and the corresponding details will follow later.

The descriptions of the parameters of the production model are shown in Table 1. The production model parameters are also addressed as the petrophysical parameters in this work.

Parameters	**Description**
$E_R(t)$	Recovery Factor until time t
E_R^0	Recovery Factor until time 0 (t = 0).
E_R^∞	Theoretical ultimate recovery factor
τ	Time constant

Table 1 – Explanation of Production Model Parameters (Parra-Sanchez, 2010).

Recovery factor, E_R is the fraction of Original Oil In-Place (OOIP) that has been produced. Besides that, this model gets its name to be "two-factor" because its formulation relies upon two parameters, namely E_R^∞ and τ. Both parameters depend on the properties of reservoir and recovery mechanism (Parra-Sanchez, 2010; Hong et al., 2018). In this context, E_R^∞ is the theoretical recovery factor that a recovery mechanism can ultimately attain whereas τ is the description of how fast the increment of the recovery factor is for a recovery mechanism (Parra-Sanchez, 2010; Hong et al., 2018). For simplicity, these two parameters are normally assumed to be time-invariant for a certain recovery phase. However, the values of these two parameters can be time-variant for different recovery phases or even within the same phase.

Besides that, Hong et al. (2018) considered only two recovery phases for the IOR initiation problem, namely primary recovery and secondary recovery. Therefore, based on two-factor production model, the primary recovery can be mathematically described as shown below.

$$E_{R1}(t) = E_{R1}^0 + (E_{R1}^\infty - E_{R1}^0) \times \left(1 - e^{\frac{-t}{\tau_1}}\right) \tag{7}$$

The subscript R1 indicates the primary recovery in the equation above. The primary recovery factor at time 0 (t = 0), E_{R1}^0 is zero. Then, the primary recovery phase would be switched to the secondary phase at time t_{R1}. So, the period of the primary recovery phase is t_{R1} and the period of the secondary recovery phase is t_{R2} (The subscript R2 indicates secondary recovery). This results in the total lifetime of both recovery phases to be $t = t_{R1} + t_{R2}$. The secondary recovery factor can then be computed using the formula below.

$$E_{R2}(t) = E_{R1}(t_{R1}) + \Delta E_{R2}^\infty \times \left(1 - e^{\frac{-(t-t_{R1})}{\tau_2}}\right) \tag{8}$$

where $E_{R1}(t_{R1})$ is the primary recovery factor at the end of primary recovery and ΔE_{R2}^∞ is the theoretical ultimate increment of secondary recovery factor when there is a shift from primary recovery phase to secondary recovery phase. Having the recovery factor calculated, the oil production rate is calculated to be

$$q_{o,j} = \frac{N_o \times \left[E_R(t_j) - E_R(t_{j-1})\right]}{\Delta t_j} \tag{9}$$

where $q_{o,j}$ is the oil production rate at the time step j, N_o is the estimated OOIP, and Δt_j is the period between two time steps j and j – 1 in which $\Delta t_j = t_j - t_{j-1}$.

3.2 Economic Model

In Hong et al. (2018), the objective function used is the NPV. In order to compute the NPV, an economic model is required. The economic model is developed based upon the other essential parameters, such as the oil price per barrel, P_o, the oil production rate, q_o, the capital expenditure, CAPEX, and the operating cost, OPEX. The total cashflow from time steps j-1 to j, CF_j, is mathematically expressed as

$$CF_j = q_{o,j} P_o \Delta t_j - \left(CAPEX_j + OPEX_j\right) \tag{10}$$

where the subscript j indicates the time step. Then, the NPV is calculated by using the equation as shown below.

$$NPV = \sum_{j=0}^{n_t} \frac{CF_j}{(1+r)^{t_j}} \tag{11}$$

where n_t indicates the total number of time steps and r is the discount rate. Based on this objective function, the NPV is a function of the lifetime (duration or period) of a recovery method (Hong et al., 2018). So, the main goal of optimizing the production is attained by determining the best time to switch from the current recovery method to another one to maximize the NPV (Hong et al., 2018). In other words, the lifetimes of both primary and secondary recovery have to be adjusted to yield the maximum NPV. It is important to emphasize that in Hong et al. (2018), only the effect of uncertainties pertaining to the reservoir properties are considered. This means that the economic parameters are assumed to be known and invariant over time. This assumption is not realistic because the economic parameters vary over time in real life and are material[27] to the decisions. Therefore, the inclusion of economic uncertainties has been done in this work and would be discussed later. Besides that, it can be noticed that taxation, which is also material, is not included in this economic model. Thus, this can be another drawback of this model.

[27] According to Bratvold and Begg (2010), there are four criteria which makes the information gathered to be worthwhile. These criteria include observable, relevant, material, and economic. The information is material if considering it would change the decision made. Refer to Bratvold and Begg (2010) for the details of the other three criteria.

3.3 Problem Setting of the Case Study in Hong et al. (2018)

This section mainly discusses the problem setting of the example used to be solved by HBL's model. In this example, Hong et al. (2018) considered two recovery phases and aimed at finding the optimal time to switch from primary recovery to secondary recovery in an oil field. The lifetime of the production is assumed to be 50 years. There are three available alternatives in this example, which are continuing with the current recovery phase, switching to secondary recovery phase, and terminating the production. The switch of recovery phases can occur any time during the lifetime of production but only for once. The same situation applies to option of terminating the production. Additionally, termination can occur either before or after the switch and once it happens, the oil field is entirely abandoned, and production would not be reinitiated.

3.3.1 Measured Oil Production Rates (State Variables)

Two-factor production model is used to model the oil production. In this case, since only 2 recovery phases are considered, there are five production model parameters to be used in this example and each of the parameters is assigned with its corresponding probability distribution (Hong et al., 2018). Hong et al. (2018) used the truncated normal distributions for all the production model parameters. This type of distribution can avoid some unrealistic values of production model parameters to be sampled and used for further analysis. To truncate the normal distribution, the maximum and minimum boundaries have to be defined. Then, if a sample of parameter is randomly retrieved from the normal distribution and its value exceeds the maximum of the boundaries (is less than the minimum of the boundaries), the sampled value is then changed to be the maximum (minimum) (Hong et al., 2018). The respective mean, SD and boundaries for each parameter are shown in Table 2.

Parameter	N_o (MMbbl)	E_{R1}^{∞} (fraction)	τ_1 (years)	ΔE_{R2}^{∞} (fraction)	τ_2 (years)
Mean	240	0.2	16	0.15	7
SD	35	0.05	2	0.05	1.5
Minimum	10	0.05	1	0.01	1
Maximum	1000	0.5	30	0.31	13

Table 2 – Means, SDs, and boundaries for the truncated normal distribution of each petrophysical parameters (Hong et al., 2018).

Besides that, the production model parameters are also correlated to each other. Therefore, the correlation coefficients between the parameters, which are used to develop the multi-variate normal distribution, are built and shown in Table 3.

	N_o	E_{R1}^{∞}	τ_1	ΔE_{R2}^{∞}	τ_2
N_o	1.00	-0.80	0.16	0.56	-0.08
E_{R1}^{∞}	-0.80	1.00	0.20	-0.70	0.10
τ_1	0.16	0.20	1.00	-0.30	-0.20
ΔE_{R2}^{∞}	0.56	-0.70	-0.30	1.00	-0.30
τ_2	-0.08	0.10	-0.20	-0.30	1.00

Table 3 – Correlation coefficients matrix of the petrophysical parameters (Hong et al., 2018).

After sampling the production model parameters, they are used to calculate the oil production rates. However, these oil rates are the modeled rates. In this case, the measured oil rates are used as state variables in the LSM algorithm. Thus, the measurement errors are sampled from the normal distribution with the mean of zero and the SD of 10% of the modeled rates. Then, the sampled errors are added accordingly to the modeled rates to yield the measured oil rates.

3.3.2 Economic Parameters for NPV Calculation (Objective Function)

The economic parameters used to calculate the NPV in HBL's model are listed in the table below.

Economic Parameters	Values	Units
Oil Price	50	$/bbl
CAPEX (Primary)	50	$ million
CAPEX_2After1 (Secondary)	40	$ million
CAPEX_2No1 (Secondary)	75	$ million
OPEX (Primary)	20	$ million/ year
OPEX (Secondary)	30	$ million/ year
Discount Rate	12%	per year

Table 4 – Values of the economic parameters[28] (Hong et al., 2018).

In Hong et al. (2018), for secondary recovery phase, there are two types of capital expenditure, namely, the capital cost of initiating secondary recovery without having primary recovery (indicated as CAPEX_2No1) and the capital cost of initiating secondary recovery after having primary recovery (indicated as CAPEX_2After1). The CAPEX is only deducted at the year when the recovery phase is started whereas the OPEX is deducted every year depending upon which recovery mechanism is being used.

[28] The dollar sign "$" means US dollar (USD).

3.4 Implementation of the HBL's Model

To apply the HBL's model, MCS is first used to generate the N samples of the five production model parameters (Hong et al., 2018). The forward modeling is conducted to generate the modeled data (modeled oil rates) from Year 1 to Year 50. Then, the measurement errors are generated and added to the modeled rates to yield the measured rates. This sample set of measured data (measured oil rates) is known as a path of measured data because it has a series of data point in time (For each path, there are 50 measured rates being generated). In order to have N paths of measured data, the same procedure is repeated for each of the N sample sets of the production model parameters. After the MCS step, Hong et al. (2018) proceed to the least-squares step. For each path of measured oil rates, the NPV of each alternative is also calculated. Since risk neutrality[29] is assumed in this example, the goal is to optimize the Expected NPV (ENPV) over the uncertain parameters (Hong et al., 2018). To approximate the ENPV of each alternative (conditioned on the measured data), for every path, the NPV of each alternative is regressed on the measured data accordingly.

In addition, Hong et al. (2018) have made a slight modification in the least-squares step. When the LSM algorithm is first introduced to value American option, the decision maker only needs to decide if he or she should instantly exercise the option at current stock price or hold the option and exercise it at a future stock price (Longstaff and Schwartz, 2001; Hong et al., 2018). In this aspect, the stock price, which is the uncertainty, is modeled as a Markovian process[30]. However, uncertainties regarding the reservoir properties and recovery mechanism are essentially not Markovian processes as they are influenced by both most recent and previous values (Hong et al. 2018). Thus, Hong et al. (2018) did a modification in which the NPV are regressed on a path of measured data (previous and current data) to approximate the ENPV.

3.4.1 Use of Value-Of-Information (VOI) Framework

As it has been mentioned earlier, the SRDM approach would induce additional value of learning as compared to the CLRM approach because it considers learning over time to produce an optimal decision policy. In this case, how can this value of learning be quantified? Pertaining to this question, Hong et al. (2018) used the VOI framework[31] to estimate the value of learning.

[29] There are three types of risk attitudes, namely risk-neutral, risk-averse, and risk-seeking (Hillson and Murray-Webster, 2005). Consult Hillson and Murray-Webster (2005) for a more comprehensive explanation of risk attitudes. For a risk-neutral person, only the expect values are used in decision making (Hong et al., 2018). This means when there are two options with equal expected value, he or she would not have any preference on any of these options.

[30] Markov process is a stochastic model used to describe a series of events whose respective probability only relies upon the state in the previous event (Gagniuc, 2017). Refer to Gagniuc (2017) for the details of Markov process.

[31] The Value-Of-Information (VOI) is a well-known concept in decision making. It is used to indicate the maximum buying price that is spent to acquire information or data. VOI is generally Value of Imperfect Information. This means that the information gathered does not reveal the truth (or can be known as state of nature). Refer to Bratvold and Begg (2010) for details of imperfect information. Besides that, VOI is referred to VOI corresponding to the globally optimal decision policy that is SRDM in this example (Hong et al., 2018). However, Hong et al. (2018) use this term loosely by referring to the VOI corresponding to CLRM as "special VOI". In this work, special VOI is referred as to VOI of CLRM. Consult Howard (1966) and Bratvold et al. (2009) for more comprehensive discussion about VOI framework.

To implement the VOI framework in HBL's model, the Decision Without Information (DWOI) of the IOR initiation problem must be determined first and the corresponding ENPV of DWOI is known as EV Without Information (EVWOI). The ENPV of every alternative is calculated by averaging the NPV corresponding to each alternative over all the paths based on prior realizations. Then, the alternative with the highest ENPV is the DWOI and the respective ENPV is the EVWOI.

After calculating the EVWOI, the EV with Imperfect Information (EVWII) must solved to estimate the VOI. This is where CLRM and SRDM are respectively implemented. By using any of these two approaches, the optimal decision would be identified for every information path. Then, all the optimal NPVs (corresponding to this optimal decision on path-by-path basis) are averaged over all the paths and this would result in the EVWII. The decision corresponding to EVWII is Decision With Imperfect Information (DWII or DWI). Then, the VOI is simply computed by finding the difference between EVWII and EVWOI. The difference between the VOI estimated by using SRDM and the VOI estimated by using CLRM can be thought as the value of learning.

Apart from these, the Decision With Perfect Information (DWPI) can also be determined in this sequential decision problem. In the context of reservoir engineering, perfect information is the information that reveals the true reservoir properties and impacts of recovery mechanism (Hong et al., 2018). In this IOR initiation problem, the EV With Perfect Information (EVWPI) can be determined by first identifying the maximum NPV for every path based on prior realizations or distributions. Then, averaging these NPVs over the paths would result in the EVWPI. In this aspect, every path would have its optimal decision with perfect information. The difference between EVWPI and EVWOI is the value of perfect information (VOPI)[32].

3.4.2 Integrating the Effect of Information in Hong et al. (2018)

As it has been explained earlier, both CLRM and SRDM approaches are optimization methods for data assimilation application in reservoir management. In this aspect, how are these approaches practically implemented in the case study presented in Hong et al. (2018)? Basically, the implementation of these approaches depends on how the measured oil production rates are used in the regression analysis. During the regression analysis, the CLRM approach is applied based on the forward calculation whereas the SRDM approach is included by using the backward calculation (similar to the rolling-back procedure used to solve for a decision tree). The detailed explanation is as follow.

[32] Perfect information is the information that is always correct as it reflects the state of nature or the truth. Thus, VOPI is the upper limit of VOI. In this case, any VOI estimated in decision making should never exceed VOPI.

3.4.2.1 CLRM Approach

In the IOR initiation problem, Hong et al. (2018) determined both the switch time and the corresponding stopping time of a recovery phase (conditioned on the switch time). This attempt is logical because after the switch from the primary recovery to secondary recovery has been found out, the corresponding stopping time of secondary recovery needs to be determined as well. Since CLRM is applying forward calculation, the algorithm starts at Year 1. At Year 1, Hong et al. (2018) calculate the ENPVs of all 1326 alternatives[33] based on the prior information (averaging the NPV corresponding to every alternative over all the paths). Then, the largest ENPV is selected (Regression analysis is done to calculate the conditional EV given additional information. At Year 1, there is no additional information. Thus, the EV is estimated based on prior distribution). In this case, there would be three possibilities.

1. If the largest ENPV corresponds to the option of "stopping at Year 1," the optimal decision is to terminate production immediately.
2. If the largest ENPV corresponds to any option that suggests to "switch at Year 1" (switch at Y1 and stop at any time between Year 1 and Year 50), the optimal decision is to switch immediately and proceed to Year 2 for further analysis.
3. If the largest ENPV corresponds to any option that suggests to "switch at any time apart from Year 1", the optimal decision is to not switch at Y1 and proceed to Year 2 for further analysis.

At Year 2, there are two possibilities:
1. If the optimal decision is to switch at Year 1 (the second situation at the previous step), then the decision maker only needs to decide if he or she should stop instantly. There are now only 50 options left (stop at Year 2, Year 3, and so forth). Then, the NPVs of these 50 options are regressed accordingly on the data at Year 1 to calculate the respective ENPVs. For each path, the largest ENPV is chosen. After that, if the optimal decision is to stop at Year 2, then the production should be terminated instantly. If not, the regression analysis is performed again, but only 49 options are considered and the data used are at Year 1 and Year 2. This loop is repeated until the optimal decision is determined.
2. If the optimal decision is not to switch at Year 1 (the third situation at the previous step), there are 1275 options left. The NPVs of these 1275 options are regressed on the data at Year 1 to calculate the respective ENPVs. For each path, the largest ENPV is chosen. In this case, there are three possibilities follow: to stop at Year 2, to switch at Year 2 and proceed to Year 3, or to not switch at Year 2 and proceed to Year 3.

[33] There are 1326 alternatives available in this example (with all possible combinations of the lifetime of primary recovery and the lifetime of secondary recovery) because the lifetime of the oil field is assumed to be 50 years. Then, when the switch time is at Year 1, this indicates the lifetime of primary recovery is zero and there are 51 corresponding choices of stopping time (to stop the secondary recovery at Year 1, Year 2 and until Year 51). In this case, to stop the secondary recovery at Year 51 means that the secondary recovery is not terminated until the end of production. By using this logic, there would be 52-n options for the switch time at Year n. This totals up to 1326 alternatives (knowing that the switch time at Year 51 is to continue with primary recovery for the lifetime of production).

At Year 3, the similar procedure done at Year 2 needs to be performed again, but the data used is at Year 1 and Year 2. As the time moves forward by a year, the number of data points used in regression analysis increases by one. This is the essence of forward calculation.

3.4.2.2 SRDM Approach

The implementation of SRDM is the demonstration of backward calculation. However, when SRDM is being applied, the stopping time given a particular switch time also has to be determined. Since it is a backward calculation, the algorithm begins at Year 50. At Year 50, the decision maker needs to determine the stopping time by assuming that he or she has switched at this year. In this case, there are only two available options, namely “stop at Year 50” and “continue with secondary recovery at Year 50”. Thus, the NPVs corresponding to these 2 options are regressed on the data from Year 1 to Year 49 given there is a switch at Year 50 to estimate the ENPVs. After that, the highest ENPV is chosen for each path. The same analysis is done for other previous years. As the time moves backward by a year, the number of data points used in regression analysis decreases by one and the number of available alternatives increases by one. This step mainly determines the best stopping time for each year based on the assumption that the switch is done at that year.

After this, the decision maker determines whether he or she should continue with primary recovery or switch to secondary recovery with its corresponding optimal stopping time (determined at the previous step). Thus, at Year 50, the NPVs of “switching to secondary recovery at Year 50 with its respective optimal stopping time” and “only having primary continue until Year 50” are regressed correspondingly on the measured data from Year 1 to Year 50 (conditioned on having only primary recovery) to estimate the ENPVs. Then, the largest ENPV is chosen for each path. At Year 49, the NPVs of these optimal decisions at Year 50 and the alternative of switching to secondary recovery at Year 49 with its respective optimal stopping time are regressed correspondingly on the data from Year 1 to Year 49 (conditioned on having only primary recovery). The same process is repeated until the time reaches Year 1. As the time moves backward by a year, the number of data points used in regression analysis also reduces by one. The number of available alternatives also increases by one.

3.5 Results of Replication of the HBL's Model

Upon having explained the application of the LSM algorithm in Hong et al (2018), the HBL's model (considering CLRM and SRDM) is first implemented (or replicated) in this work. To validate if this replicated model is correct, the EVWIIs (considering both CLRM and SRDM), EVWOI, EVWPI, and VOIs (considering both CLRM and SRDM) estimated by this replicated model are compared to those approximated in Hong et al. (2018). Prior to conducting the comparison, 100000 samples of the five petrophysical parameters are first retrieved by using the statistical parameters listed in Table 2 and the correlation coefficients shown in Table 3 in the replicated model. After that, these sampled sets of petrophysical parameters are used along with the economic parameters described in Table 4 to estimate the abovementioned values in the replicated HBL's model. The comparison of these results with those estimated in Hong et al. (2018) is shown below.

Results	**(million USD)**	
	Hong et al. (2018)	**Replicated Model**
EVWOI	756.20	756.19
EVWPI	909.00	909.01
$EVWII_{SRDM}$	805.90	805.90
$EVWII_{CLRM}$	804.10	804.09
VOI_{SRDM}	49.70	49.71
VOI_{CLRM}	47.90	47.90

Table 5 – Comparison of EVWOI, EVWPI, EVWIIs, and VOIs estimated in Hong et al. (2018) and by the replicated model in this work.

Based on the results of comparison, it can be deduced that the replicated HBL's model has the correct implementation of the modified LSM algorithm (and CLRM) as presented in Hong et al. (2018). In addition, the negligible difference between the results from the replicated model and those presented in Hong et al. (2018) is caused by the Monte-Carlo sampling error (as different sets of samples are used). It can also be noticed that SRDM induces an additional value of learning of $1.8 million as compared to CLRM. This illustrates the suboptimality[34] of the CLRM solution. After validating the correctness of this replicated model, there is another analysis being done on it. In this aspect, Jafarizadeh and Bratvold (2009) explained that using larger number of paths enabled the convergence of the result to the global optimum in LSM algorithm. In other words, it denotes that using higher number of paths can improve the accuracy of the VOI estimate (whether it be CLRM or SRDM). To illustrate the trend of this improvement of the VOI estimate, the sensitivity analysis of the number of paths on the VOI estimate is conducted by using 100, 500, 1000, 5000, and 10000 paths. Each number of paths

[34] Both SRDM and CLRM provide approximate solutions, but the one estimated by SRDM is theoretically closer to the true optimum due to the additional value of learning. Albeit the value of learning induced by SRDM might not be significant in this case, for certain problem setting, this value can be significant. Refer to Hong et al. (2018) for details. Additionally, in this work, the author uses a different problem setting to show the significance of the value of learning in later chapter.

is run 100 times and the respective means and standard deviation (SD) of the estimated VOI are calculated. This analysis is done for both SRDM and CLRM approaches. The result of this sensitivity analysis for CLRM is shown in Fig. 3 whereas that of SRDM is shown in Fig. 4.

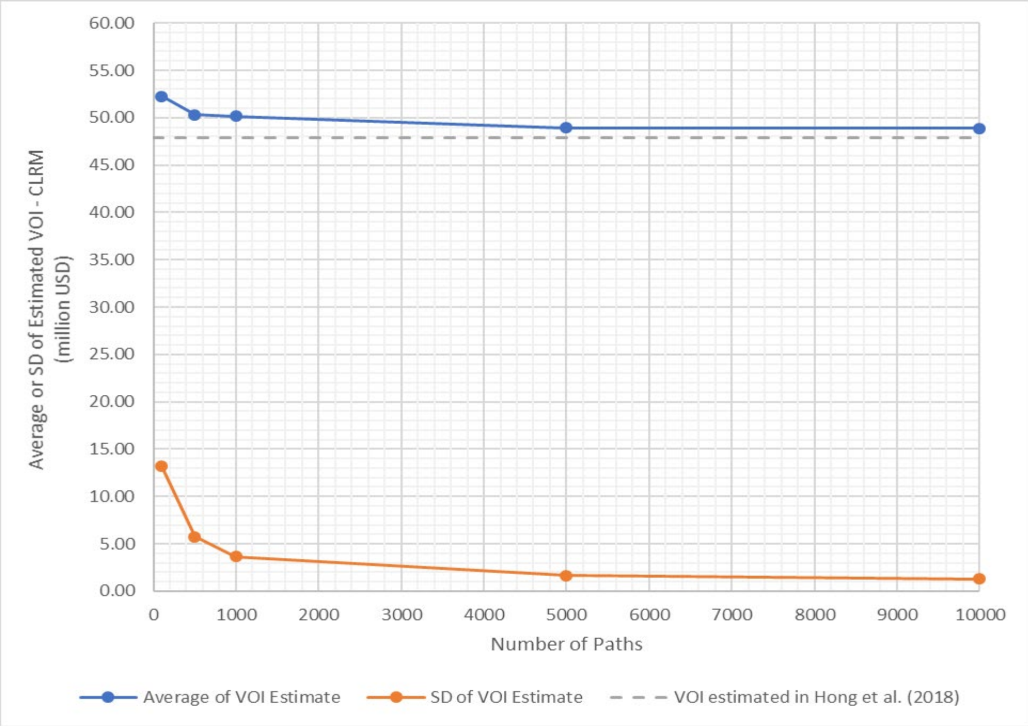


Fig. 3 – Average and SD of 100 approximated VOIs for different number of paths (CLRM approach).

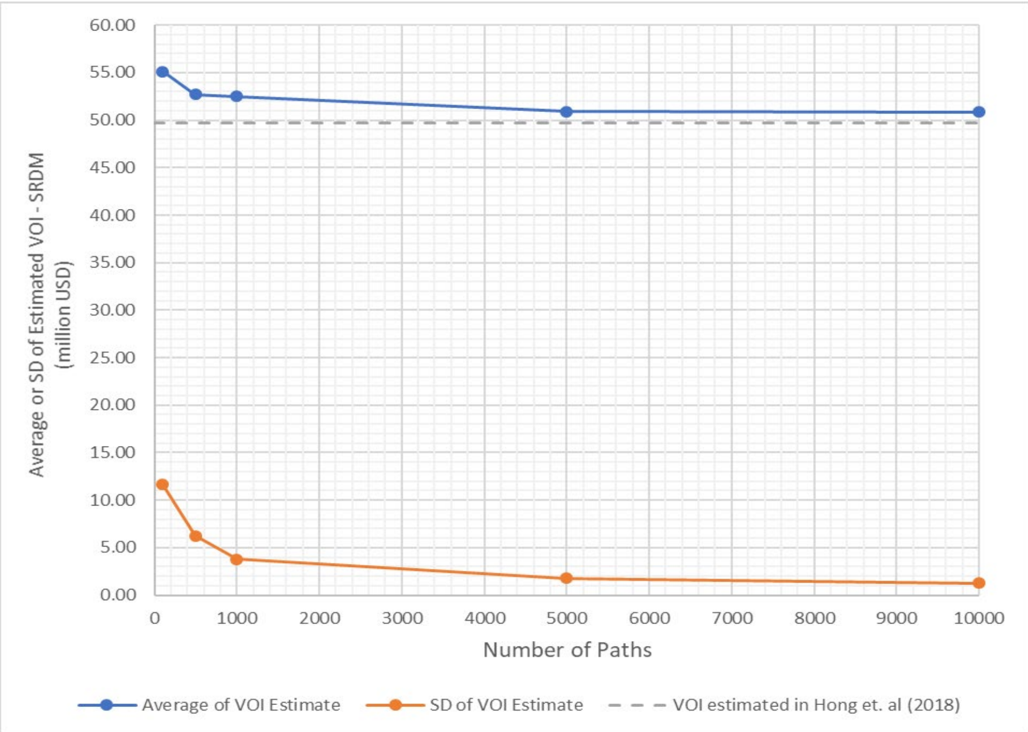


Fig. 4 – Average and SD of 100 approximated VOIs for different number of paths (SRDM approach).

For both CLRM and SRDM approaches, it can be seen that the estimation of VOI is becoming unbiased as the number of paths is increasing. This means that the average of the estimated VOIs over 100 runs for different number of paths is converging to the VOI estimated in Hong et al. (2018), which is more accurate[35] (closer to the exact VOI). Furthermore, the SD of the estimated VOIs over 100 runs for different number of paths also reduces when the number of paths increases. When 10000 paths are run for 100 times, the SD for CLRM approach is $1.29 million whereas that of SRDM is $1.24 million. As compared to the VOIs (of both CLRM and SRDM) estimated using 100000 paths, these SDs are considered to be not significant. Thus, this shows that using 10000 paths provides a sufficiently good estimation of VOI. However, using number of paths more than 10000 as done in Hong et al. (2018) provides even higher accuracy of estimated VOI as discussed.

[35] VOI estimated in Hong et al. (2018) has higher accuracy because it is approximated using higher number of MCS paths, which is 100000. Thus, this VOI is closer to the global optimum that is the exact VOI.

Chapter 4

Sensitivity Analysis on the Choice of Regression Function

In the least-squares approach (regression analysis), an approximate value function is determined to relate the expected future value of an option to the uncertain parameters. Most of the studies showed that in LSM, the use of simple linear regression as the approximate value function was sufficient to provide a good estimation due to the linear relationship between the objective function and the state variables (Smith, 2005; Hong et al., 2018). Additionally, some studies also illustrated that the use of higher order functions yielded a similar result as the simplified linear function does (Moreno and Navas, 2003). In this case, Moreno and Navas (2003) discussed that in real option valuation, the choice of basis function did not affect the results significantly for simple decision problems, such as American option and European option. However, for more complicated decision problems, such as American-Bermuda-Asian option[36], the choice of basis function would significantly affect the results. Therefore, Alkhatib et al. (2013) opined that it was important to conduct the sensitivity analysis to the choice of regression function to evaluate if there was any significant impact on the results obtained.

The sensitivity analysis to the choice of regression function has been done with the purpose of generating insights about the use of regression function[37] to solve the IOR initiation time problem. For this sensitivity analysis, more sophisticated regression functions are established by including different types of dependency term among the production rates and an exponential term with the use of Laguerre polynomials. A detailed analysis on an exception case is also done and will be discussed later.

4.1 Inclusion of Dependency Among Production Rates

For the case study problem in Hong et al. (2018), the relationship between the NPV (objective function) and the measured oil production rates (uncertain parameter or state variables) is linear. Thus, it can be deduced that using higher orders regression function would not enhance the results significantly. In Hong et al. (2018), multiple linear regression function is used to approximate ENPV of an alternative which $\tilde{y} = \mathrm{ENPV}(\text{"alternative"}) \mid \mathbf{q}$ (in which $\mathbf{q}$ is a

[36] Bermudan option is in between American and European options which it can be exercised either at the predetermined date or any time before expiry. Asian option is an exotic option whose payoff is determined by averaging the underlying price over a period of time (Martinkutė-Kaulienė, 2012).

[37] A good regression function should correctly approximate the true conditional EV. However, the true conditional EV is unknown in the real world. So, another criterion used to check if a regression function is good is how close the function can lead to a near-optimal decision policy. For example, given two regression functions, RF1 and RF2, if RF2 yields a higher EVWII than RF1 does, then RF2 is generally better (provides more accurate estimation).

vector that consists of a series of measured oil production rates in time) due to the modeling of state variables as non-Markovian processes.

$$\tilde{y} = \beta_0 + \beta_1 q_1 + \beta_2 q_2 + \beta_3 q_3 + \cdots + \beta_t q_t \tag{12}$$

where β_t is the regression coefficient and q_t is the measured oil production rate at time t. Albeit the dependency of the expected future value on current and previous data has been included by conducting multiple linear regression (Hong et al., 2018), there is undeniably a dependency (interaction) among the measured oil rates. Such interaction reflects the occurrence of a change in NPV when there is a change in the measured oil production rate at time t+1, which relies on the change in the measured oil production rate at time t. Would a great improvement of results be achieved if the interaction among the production rates is included in regression analysis? In addition to this, there are not many works done in the literature to substantiate if in a non-Markovian process, integrating dependency terms would provide much better results, especially in the context of the IOR initiation time problem. In order to conduct this analysis, six different types of dependency terms have been included. Each type of dependency terms is included accordingly to yield the corresponding regression function for the analysis in this section as shown.

$$\tilde{y} = \beta_0 + \beta_1 q_1 + \beta_2 q_2 + \beta_3 q_3 + \cdots + \beta_t q_t + \tilde{\beta}_1 \sqrt{q_1 q_2} + \tilde{\beta}_2 \sqrt{q_2 q_3} + \cdots + \tilde{\beta}_{t-1} \sqrt{q_{t-1} q_t} \tag{13}$$

$$\tilde{y} = \beta_0 + \beta_1 q_1 + \beta_2 q_2 + \cdots + \beta_t q_t + \breve{\beta}_1 \sqrt{q_1 q_2 q_3} + \breve{\beta}_2 \sqrt{q_2 q_3 q_4} + \cdots + \breve{\beta}_{t-2} \sqrt{q_{t-2} q_{t-1} q_t} \tag{14}$$

$$\tilde{y} = \beta_0 + \beta_1 q_1 + \cdots + \beta_t q_t + \tilde{\beta}_1 \sqrt{q_1 q_2} + \cdots + \tilde{\beta}_{t-1} \sqrt{q_{t-1} q_t} + \breve{\beta}_1 \sqrt{q_1 q_2 q_3} + \cdots + \breve{\beta}_{t-2} \sqrt{q_{t-2} q_{t-1} q_t} \tag{15}$$

$$\tilde{y} = \beta_0 + \beta_1 q_1 + \beta_2 q_2 + \cdots + \beta_t q_t + \tilde{\beta}_1 q_1 q_2 + \tilde{\beta}_2 q_2 q_3 + \cdots + \tilde{\beta}_{t-1} q_{t-1} q_t \tag{16}$$

$$\tilde{y} = \beta_0 + \beta_1 q_1 + \beta_2 q_2 + \cdots + \beta_t q_t + \breve{\beta}_1 q_1 q_2 q_3 + \breve{\beta}_2 q_2 q_3 q_4 + \cdots + \breve{\beta}_{t-2} q_{t-2} q_{t-1} q_t \tag{17}$$

$$\tilde{y} = \beta_0 + \beta_1 q_1 + \beta_2 q_2 + \cdots + \beta_t q_t + \tilde{\beta}_1 q_1 q_2 + \cdots + \tilde{\beta}_{t-1} q_{t-1} q_t + \breve{\beta}_1 q_1 q_2 q_3 + \cdots + \breve{\beta}_{t-2} q_{t-2} q_{t-1} q_t \tag{18}$$

4.1.1 Results

To analyze the impact of having dependency terms (with uncertainties being modeled as non-Markovian process), the sampling is first done with 10000 paths by using the same problem setting stated in Table 2, Table 3, and Table 4. Prior to including these dependency terms, a simple linear regression without any dependency terms is used to estimate the EVWIIs by applying SRDM and CLRM approach (these are EVWIIs of the base case in this subsection). EVWII estimated using SRDM is $807.49 million whereas EVWII approximated using CLRM is $805.24 million. EVWIIs estimated by the linear regression function with the corresponding dependency terms are tabulated as shown below.

Dependency Term	million USD	
	$EVWII_{SRDM}$	$EVWII_{CLRM}$
Equation (13)	807.74	805.46
Equation (14)	808.07	806.29
Equation (15)	808.42	806.93
Equation (16)	808.16	806.81
Equation (17)	808.07	806.52
Equation (18)	807.98	806.46

Table 6 – Results of Comparison of EVWIIs (considering both SRDM and CLRM) estimated by the linear regression function with 6 different dependency terms.

The percentage of value of improvement[38] is also calculated for each EVWII by computing the percentage of the fraction of the difference between the newly estimated EVWII and the EVWII of the base case to the EVWII of the base case. The results are presented in Fig. 5 and Fig. 6.

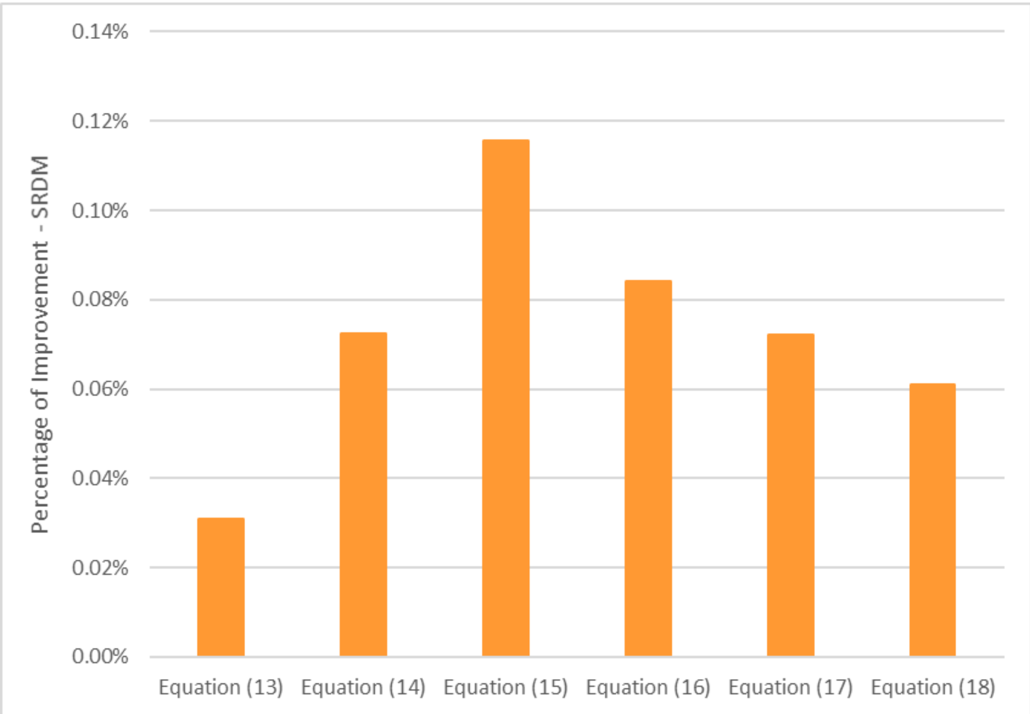


Fig. 5 – Plot of the Percentage of Improvement on EVWII estimate (SRDM).

[38] The author uses the word of "improvement" in this chapter to indicate the increase of the EVWIIs estimated by the more complicated regression functions from the EVWIIs of the base case as shown in Table 6.

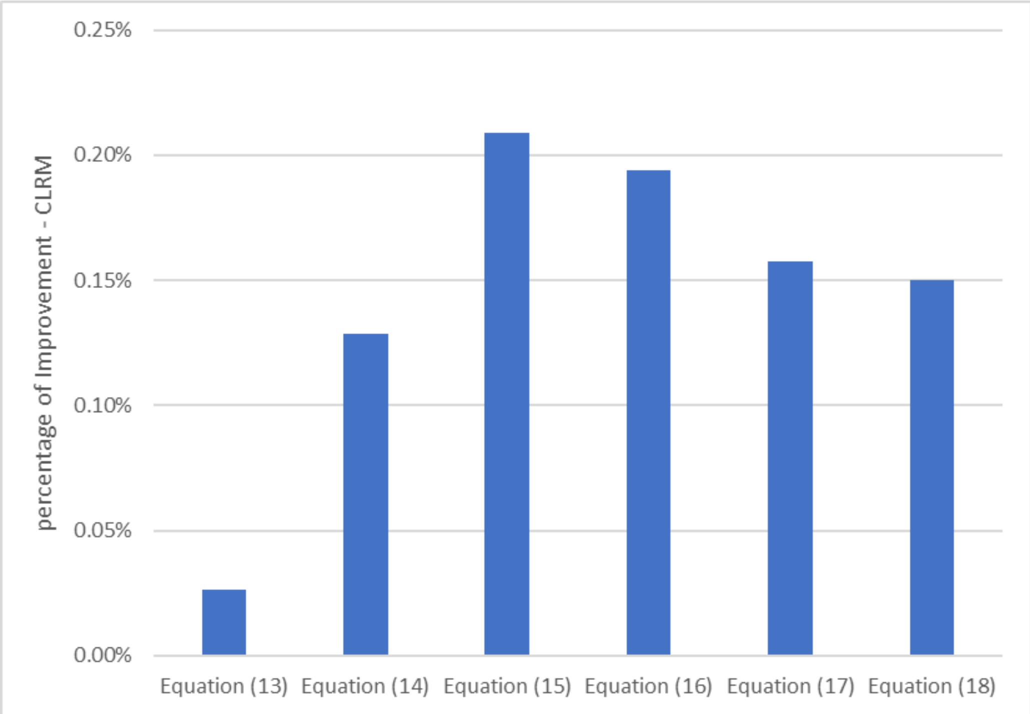


Fig. 6 – Plot of the Percentage of Improvement on EVWII estimate (CLRM).

From this analysis, it can be deduced that including the dependency terms would improve the estimation of the EVWII. However, such improvement is insignificant (not even up to 1% of improvement). Moreover, including these dependency terms would also induce higher computational time as the terms become more complicated. This shows that using the linear regression without the dependency terms already provides a sufficiently good approximation.

4.2 Inclusion of Exponential Term in the Regression Function

Albeit using a more complicated type of polynomial (by increasing the order) would not enhance the result significantly, it has not been demonstrated yet if having a more complex type of term, such as an exponential term, in the regression function would yield much better improvement of results. In this aspect, Laguerre polynomials, which consist of the exponential terms, are used as the basis function in regression analysis for the estimation of the ENPV of an alternative. For the purpose of illustration, only the first three Laguerre polynomials are used, and they are expressed as shown below:

$$\mathrm{L}_0(\mathbf{q}) = \mathrm{e}^{-(\mathbf{q}/2)} \tag{19}$$

$$\mathrm{L}_1(\mathbf{q}) = \mathrm{e}^{-(\mathbf{q}/2)} \times (1 - \mathbf{q}) \tag{20}$$

$$\mathrm{L}_2(\mathbf{q}) = \mathrm{e}^{-(\mathbf{q}/2)} \times [1 - 2\mathbf{q} + (\mathbf{q}^2/2)] \tag{21}$$

Then, the ENPV of the alternative can be expressed using the polynomials above.

$$\mathrm{ENPV}(\text{"alternative"}) \mid \mathbf{q} = \sum_{j=0}^{2} \boldsymbol{\beta}_j \mathrm{L}_j(\mathbf{q}) \tag{22}$$

4.2.1 Analysis of the Impact of Renormalization

Prior to estimating the EVWIIs by using Laguerre polynomials, there is an important issue to be considered regarding its application. Directly applying the Laguerre polynomials might cause the occurrence of either the computational underflows or overflows (during regression) because of the exponential terms (Longstaff and Schwartz, 2001). In Longstaff and Schwartz (2001), computational underflows would happen because the stock price ranged from 36 to 44 and directly substituting these values into the exponential terms would result in very small values. To mitigate this problem, they divided the cashflow and prices by the strike price. This recommended approach is termed as renormalization. Thus, for the example in Hong et al. (2018), how can the renormalization be done? As the measured oil production rates are represented as the exponential term, the goal of renormalization is to make the measured rates in the exponential term to be neither too big nor too small.

To achieve this goal, these rates can be divided by an arbitrary value. With respect to this, there is no standardized approach used to determine this arbitrary value. Thus, a few values can be used and one of them is the average of the measured rates corresponding to the DWOI. To obtain this value, the DWOI must be first determined and the measured rates corresponding to it are averaged over all the paths. Besides that, another arbitrary value suggested is the mean of the maximum measured production rate (over all the paths) and the minimum measured production rate (over all the paths). Then, any value in the range between average measured rate of DWOI and the mean measured rate of the maximum and minimum rates can be used as the arbitrary value as well.

4.2.2 Results

To analyze the impact of including a more complicated term, the sampling of realization is redone by using the same problem setting as shown in Table 2, Table 3, and Table 4 with 10000 MCS paths. In this context, EVWIIs are respectively estimated by applying SRDM and CLRM approach (these are the EVWIIs of the base case in this subsection). Thus, the EVWII estimated using SRDM is \$809.05 million whereas the EVWII approximated using CLRM is \$806.89 million[39]. Then, the DWOI corresponds to having 5 years of primary recovery and 15 years of secondary recovery. For renormalization, the average rate corresponding to the DWOI is found out to be 1.30 MMbbl/year. Besides that, the mean of the maximum and the minimum measured production rate (over all the paths) is computed to be 21.40 MMbbl/year. Two other values are arbitrarily chosen between 1.30 MMbbl/year and 21.40 MMbbl/year and they are 7 MMbbl/year and 14 MMbbl/year. EVWIIs estimated by using the arbitrary values for renormalization are shown below.

Laguerre Polynomials	**(million USD)**	
Without Renormalization	**$EVWII_{SRDM}$**	**$EVWII_{CLRM}$**
	810.24	810.07
Arbitrary Value (With Renormalization)	**(million USD)**	
	$EVWII_{SRDM}$	**$EVWII_{CLRM}$**
1.30 MMbbl/year	810.25	810.08
7.00 MMbbl/year	810.51	810.17
14.00 MMbbl/year	810.44	810.21
21.40 MMbbl/year	810.64	810.23

Table 7 – Results of Comparison of EVWIIs (considering both SRDM and CLRM) by using Laguerre Polynomials with and without renormalization.

Based on the result above, it can be deduced that for this case study, including the impact of renormalization would not improve the results significantly. This indicates that directly applying the Laguerre polynomials would neither induce computational underflows or overflows for the case study in Hong et al. (2018). This is because the exponential terms of the measured oil production rates without renormalization are already neither too big nor too small to cause either the overflows or underflows. In other words, the regression analysis with the direct use of Laguerre polynomials works well in this case study problem. However, renormalization is still recommended prior to using Laguerre polynomials in LSM, especially when the range between the variables is small and the exponential values of those variables would be either too large or too small.

[39] EVWIIs of SRDM and CLRM here are different from those in Section 4.1.1 due to sampling error.

In addition, the percentage of improvement on EVWIIs (as compared to EVWII of the base case here) are computed and plotted in Fig. 7 and Fig. 8. It can be observed that using the Laguerre polynomials (either without or without renormalization) also does not result in significant improvement of EVWIIs. This further substantiates that linear regression is a good approximation in this case study.

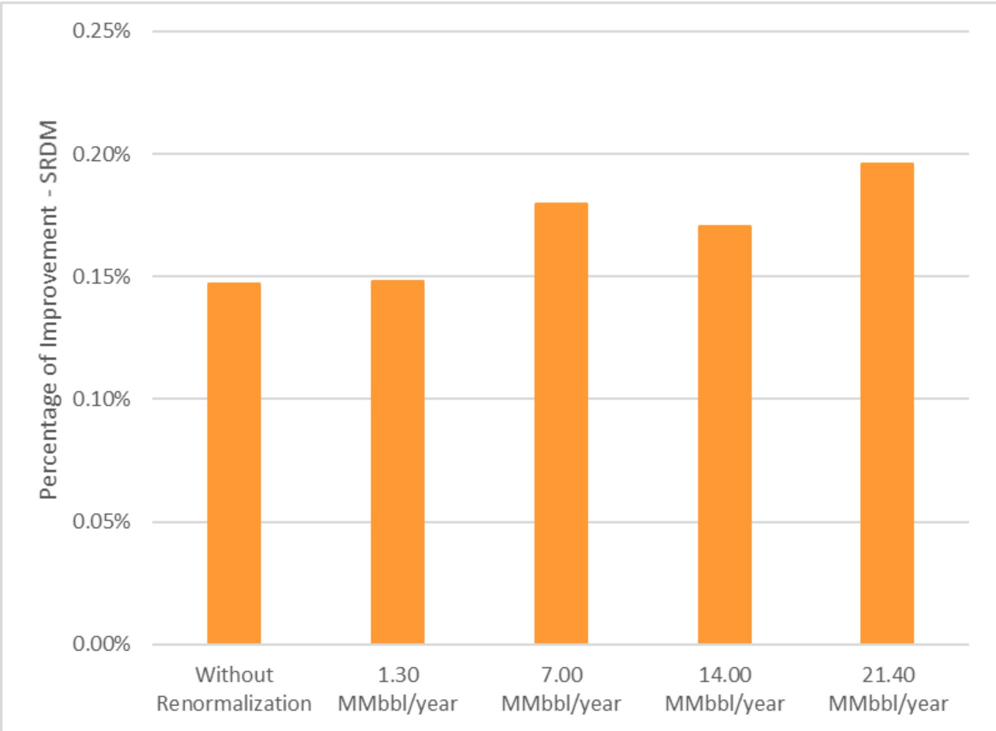


Fig. 7 – Plot of the Percentage of Improvement on EVWII estimate (SRDM) by using Laguerre Polynomials.

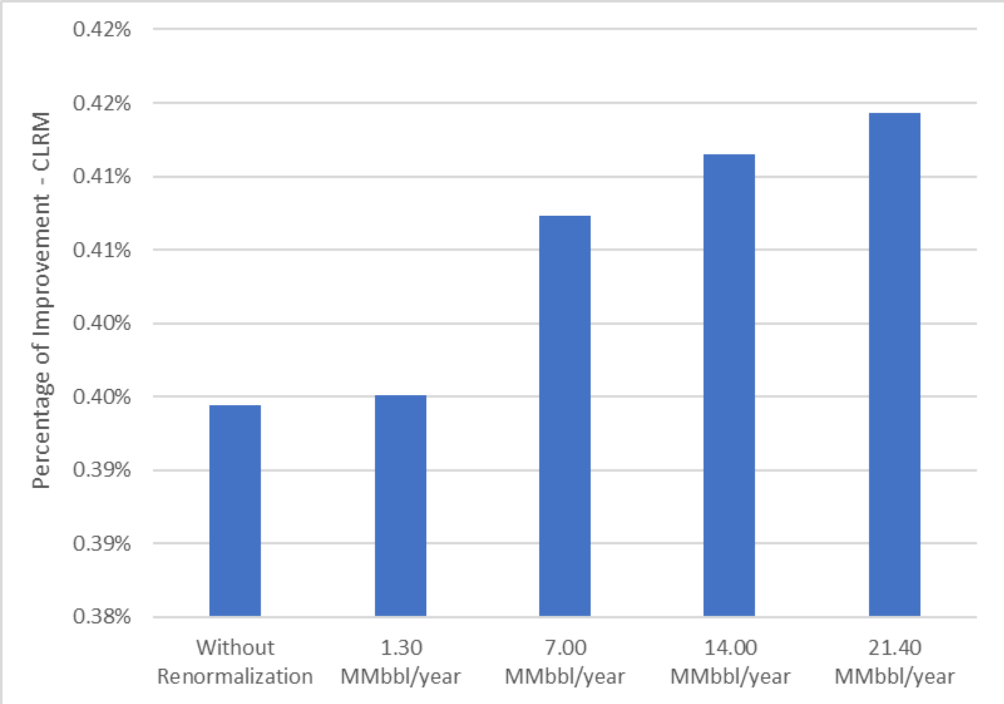


Fig. 8 – Plot of the Percentage of Improvement on EVWII estimate (CLRM) by using Laguerre Polynomials.

4.3 Analysis on the Exceptional Case

Hong et al. (2018) illustrated that the EVWII of SRDM would theoretically always be higher than that of CLRM. However, since approximate solutions to the EVWIIs are provided by using regression analysis, it is possible to have a case where the EVWII of SRDM is less than that of CLRM. In this context, another analysis has been conducted in this work to evaluate if adding dependency terms or including the exponential terms would improve the condition by making EVWII of SRDM to be higher than that of CLRM. In this context, an additional regression function (Laguerre polynomials with dependency terms shown in Equation (13)) is also used in this analysis.

In order to generate a case where the EVWII of SRDM is lower than that of CLRM, the following economic parameters are used as shown in Table 8. By using the same statistics of production model parameter as shown in Table 2 and Table 3, 10000 MCS paths is used to perform this analysis. Then, EVWIIs are respectively estimated by applying SRDM and CLRM approach (these are the EVWIIs of the base case in this subsection). The EVWII estimated using SRDM is \$329.58 million whereas the EVWII approximated using CLRM is \$330.17 million. The result of the analysis is shown in Table 9.

Economic Parameters	Values	Units
Oil Price	30	$/bbl
CAPEX (Primary)	70	$ million
CAPEX (Secondary)	70	$ million
OPEX (Primary)	20	$ million/ year
OPEX (Secondary)	40	$ million/ year
Discount Rate	12%	per year

Table 8 – Value of economic parameters used to generate the exceptional case.

Type of Equations	million USD	
	$EVWII_{SRDM}$	$EVWII_{CLRM}$
Equation (13)	329.97	330.33
Equation (14)	330.39	330.52
Equation (15)	330.39	330.51
Equation (16)	330.20	330.29
Equation (17)	330.28	330.66
Equation (18)	330.48	330.58
Laguerre w/o	330.46	330.66
Laguerre w	330.62	330.76

Table 9 – Results of Comparison of EVWIIs (considering both SRDM and CLRM) estimated by 8 different regression functions. “Laguerre w/o” means Laguerre polynomials without dependency terms whereas “Laguerre w” means the otherwise.

Based on the results, it can be noted that having dependency terms or the exponential terms would enhance both EVWIIs. For this exceptional case, using regression analysis would still induce approximation error which causes the EVWII of CLRM to be slightly higher than that of SRDM. This can be one of the drawbacks of the LSM method in which approximation error would produce a theoretically unrealistic result (depending on the problem setting).

Chapter 5

Sensitivity Analysis on Model Parameters and Other Variables

5.1 Petrophysical and Economic Parameters

In Hong et al. (2018), there are different model parameters used to yield the decision policy regarding the initiation time of IOR. As discussed, these model parameters are the petrophysical parameters from the two-factor production model and the economic parameters from the simple economic model. In this context, as the estimate of any of these parameters is changed, the results of the decision policy would also change. Therefore, it is vital to conduct a detailed analysis find out how sensitive the results are to the changes of the estimate of each of the parameters. By doing this sensitivity analysis, we can identify which parameter would produce a great impact on the decision policy. This in turn helps us to determine the uncertainties of the parameters which can materially affect our decision (Bratvold and Begg, 2010). Then, we can place more emphasis on collecting more information about these impactful parameters to reduce their uncertainties if having this additional information can add value (Bratvold and Begg, 2010).

5.1.1 Procedure of Sensitivity Analysis

The sensitivity analysis is done on the five petrophysical parameters, namely OOIP, E_{R1}^{∞}, ΔE_{R2}^{∞}, τ_1, and τ_2 by finding the changes of plus and minus 15% of the corresponding mean and SD of these parameters as listed in Table 2. In this case, when the value of a parameter changes, those of others remain constant. After that, EVWIIs corresponding to the change of these means and SDs of the parameters are estimated by using both SRDM and CLRM with 25000 paths and other details listed in Table 3 and Table 4. The tornado plots are then respectively made for the means and the SDs. To make the tornado plot, the EVWIIs corresponding to minus 15% of the estimate of each parameter is used as the lower limit whereas the EVWIIs corresponding to plus 15% of the estimate of each parameter is used as the upper limit. With respect to this, the EVWIIs corresponding to the original estimates is used as the base. For the base values, EVWII of SRDM is estimated to be \$808.24 million whereas EVWII of CLRM is approximated to be \$806.49 million. Furthermore, the difference between EVWII of SRDM and EVWII of CLRM for each parameter is computed and the respective tornado plot is also made by using the same approach as explained earlier. The base value for the tornado plot of the difference between the EVWIIs is \$808.24 million - \$806.49 million = \$1.75 million.

For the sensitivity analysis on the economic parameters, the changes of plus and minus 15% of the economic parameters listed in Table 4 are calculated. Nonetheless, in this case, the sampling of realizations is redone by applying 25000 paths and other details listed in Table 2 and Table

3 for the calculation of EVWIIs. However, the realizations are only sampled once. Then, the base value for the tornado plot is determined. With respect to this, EVWII of SRDM is estimated to be \$807.84 million whereas EVWII of CLRM is approximated to be \$805.78 million[40]. Thus, the base value for the difference between the EVWIIs is \$2.06 million. After that, the same approach as discussed earlier is implemented to make the respective tornado plots.

Apart from this, for both petrophysical and economic parameters, the EVWIIs of the estimates of these parameters corresponding to the changes of plus and minus 10% as well as plus and minus 20% are also calculated. Then, the graph of the EVWIIs against the percentage change of the respective parameters is plotted for further analysis. The same procedure is also done for the cases of EVWOI, difference between EVWII of SRDM and EVWII of CLRM, and VOIs (considering both SRDM and CLRM).

5.1.2 Results and Discussions

5.1.2.1 Mean of the Petrophysical Parameters

a. EVWII and EVWOI

The tornado sensitivity plots of the EVWIIs (considering SRDM and CLRM) corresponding to the means of the petrophysical parameters are respectively shown in Fig. 9 and Fig. 10. Both figures show that the mean of OOIP is the most impactful variable and is followed by those of ΔE_{R2}^{∞}, τ_1, E_{R1}^{∞}, and τ_2. This indicates that by changing the mean of OOIP, the EVWII would change more significantly in relative to the change of the mean of other parameters. Besides that, pertaining to the direction of the change of EVWII, it shows that when the mean of OOIP, ΔE_{R2}^{∞} or E_{R1}^{∞} is increased (decreased) by 15%, the resulting EVWII is larger (smaller) than the base EVWII. This opposite scenario is noted for the parameters of τ_1, and τ_2. This demonstration can be explained by using the formulation of two-factor production model. When the mean of any of these parameters increases, the value of the respective sample retrieved from the distribution would generally increase. Based on the production model and the equation of measured oil production rates, having higher (lower) OOIP, ΔE_{R2}^{∞} or E_{R1}^{∞} would produce higher (lower) measured oil production rates. Thus, it leads to higher (lower) cashflow and results in higher (lower) EVWII. For τ_1 and τ_2, they would have an opposite effect due to the exponential declines in the production model.

The graph of EVWIIs against the percentage change of the mean of the parameters is plotted in Fig. 11 for the case of SRDM and in Fig. 12 for the case of CLRM. Based on these two figures, it can be noted that as the means of OOIP, ΔE_{R2}^{∞} and $E_{R1}^{\infty}$are respectively increased, the EVWIIs would monotonically increase. However, for τ_1 and τ_2, as their means respectively increase, EVWIIs would monotonically decrease. This result is another representation of the mechanism of the two-factor production model as expounded.

[40] EVWIIs of SRDM and CLRM in the sensitivity analysis on economic parameters are different from those in the sensitivity analysis on petrophysical parameters due to sampling error.

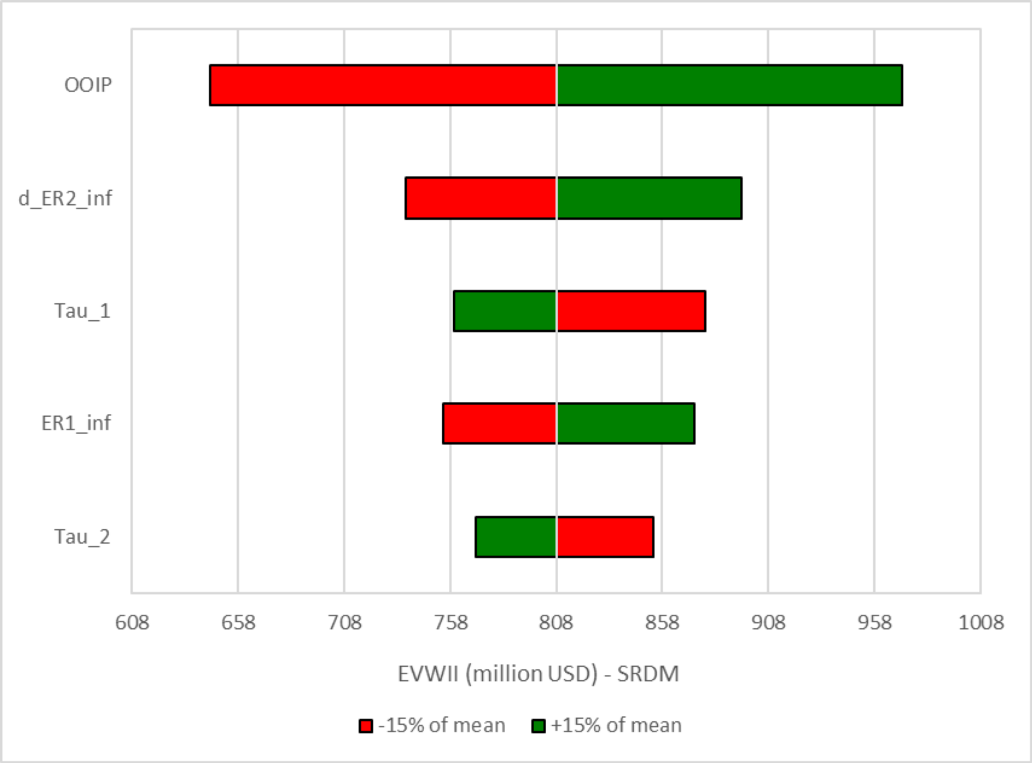


Fig. 9 – Sensitivity Tornado Plot for EVWIIs of the means of Petrophysical Parameters (SRDM approach).

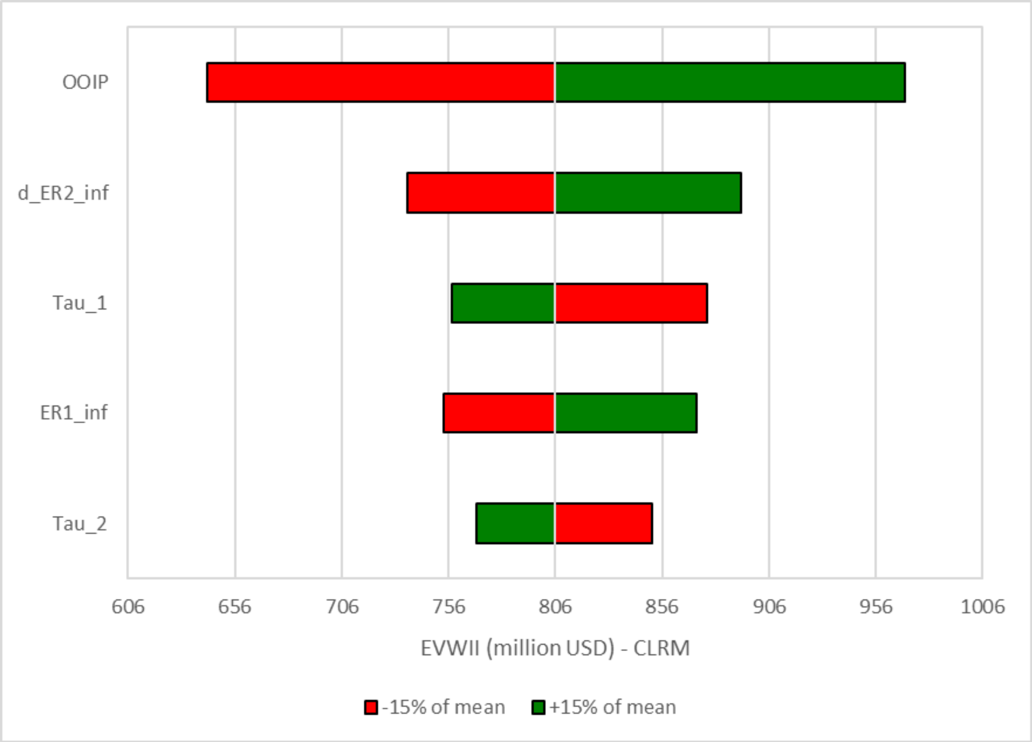


Fig. 10 – Sensitivity Tornado Plot for EVWIIs of the means of Petrophysical Parameters (CLRM approach).

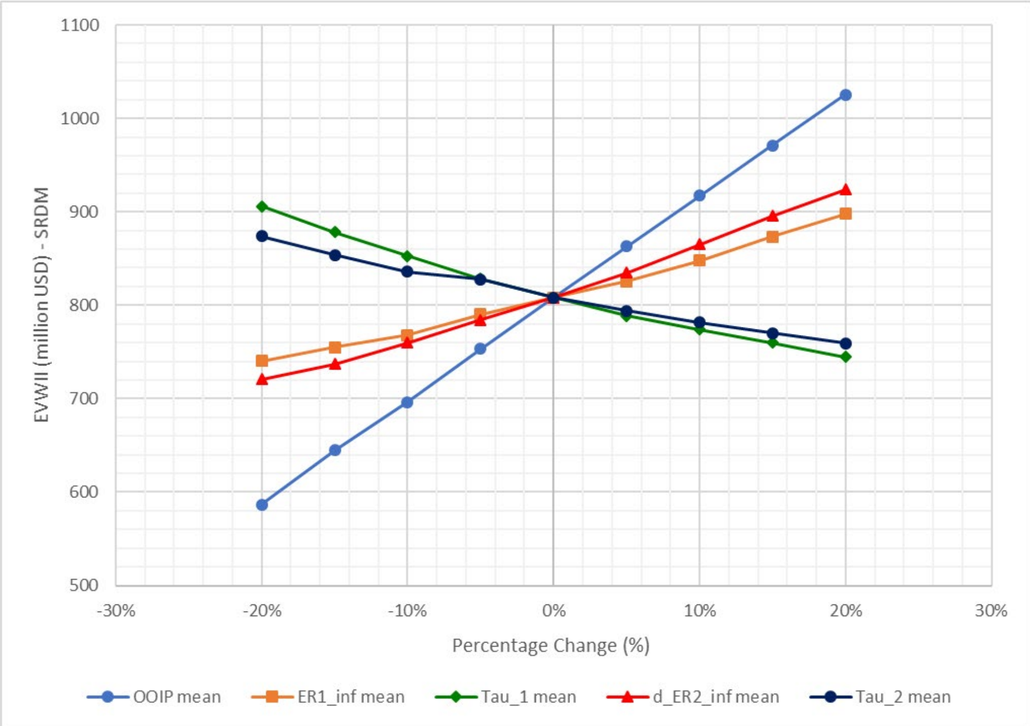


Fig. 11 – Graph of EVWIIs of the means of Petrophysical Parameters against the percentage change of the corresponding parameters (SRDM approach).

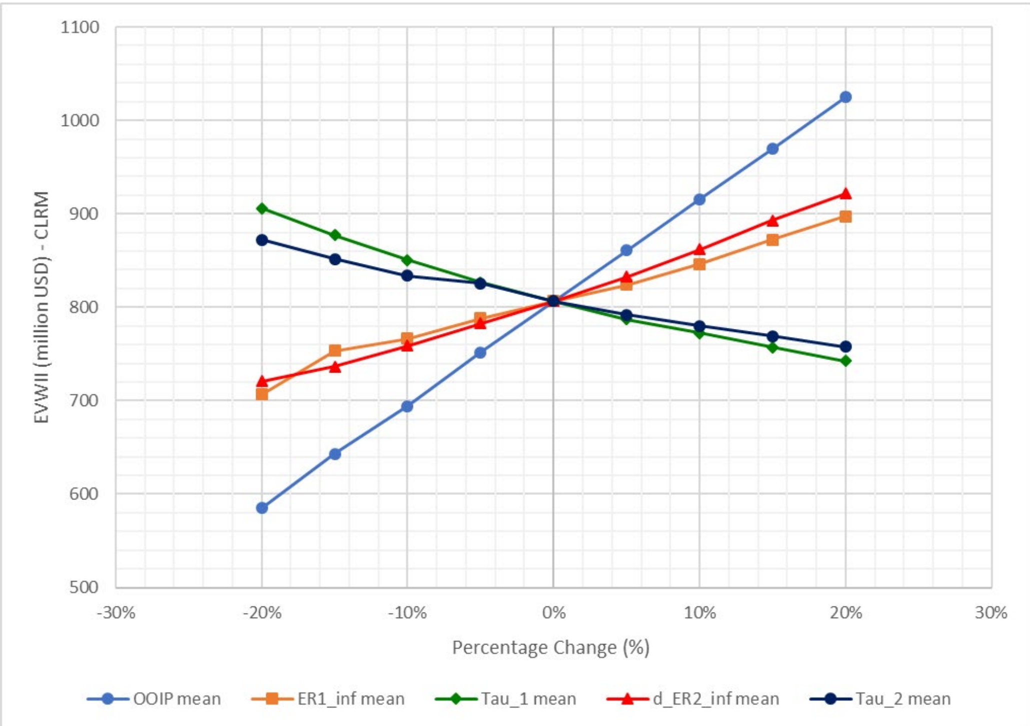


Fig. 12 – Graph of EVWIIs of the means of Petrophysical Parameters against the percentage change of the corresponding parameters (CLRM approach).

EVWOIs corresponding to the change of the mean of each parameter are determined and the respective result is shown in Fig. 13. Based on Fig. 13, the trend of monotonicity demonstrated by each parameter is similar to the results shown in Fig. 11 and Fig. 12. This is also because of the general trend of increase (decrease) in the values of samples as the mean of parameter increases (decreases) and the formulation of the two-factor production model as explained for the case of EVWIIs.

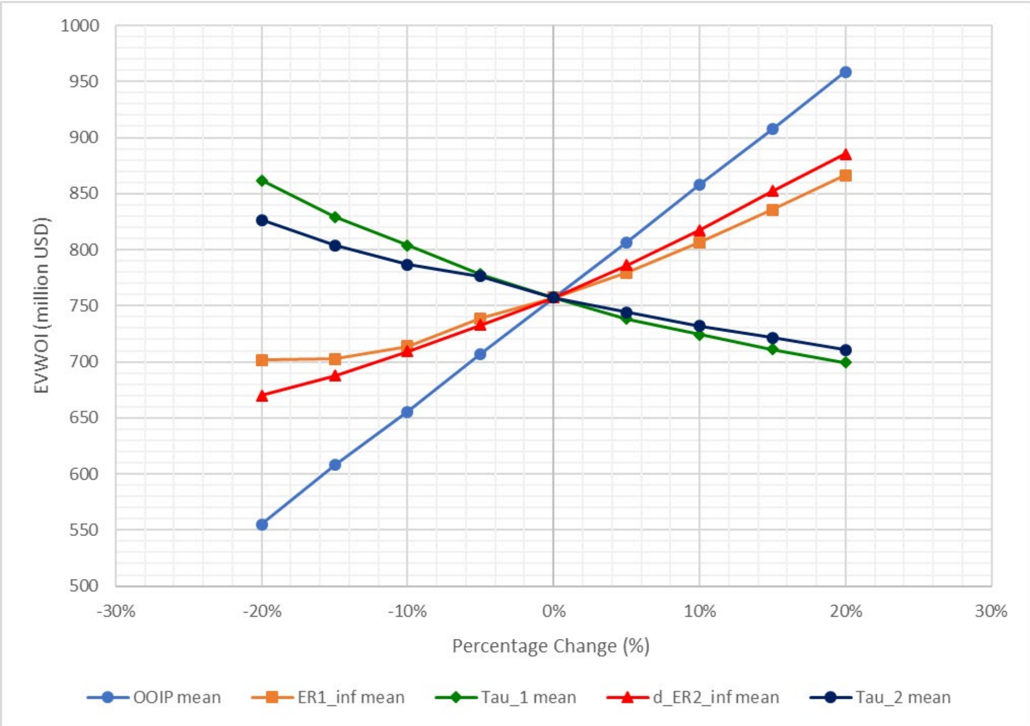


Fig. 13 – Graph of EVWOIs of the means of Petrophysical Parameters against the percentage change of the corresponding parameters.

b. Difference between EVWIIs and VOI

The sensitivity tornado plot regarding the difference between EVWII of SRDM and EVWII of CLRM (corresponding to the mean of parameters) is shown in Fig. 14. Based on this figure, it shows that the mean of ΔE_{R2}^{∞} is the most impactful variable and is followed by those of τ_2, τ_1, OOIP, and E_{R1}^{∞}. Therefore, the result demonstrates that having higher mean of ΔE_{R2}^{∞} would yield a larger difference between EVWII of SRDM and EVWII of CLRM. However, from the tornado plot, it can be noted that for E_{R1}^{∞}, the resulting difference between the EVWIIs shows that the non-monotonic behavior as the corresponding mean increases or decreases.

With respect to this, Fig. 15 illustrates the graph of the difference between EVWIIs against the percentage change of the mean of each parameter. It shows that in general, the difference between EVWIIs fluctuates in response to the changes of the mean of each parameter. Thus, the non-monotonic behavior is also shown by changing the means of other parameters. This is because when there is a change in the mean of any of these parameters, it would lead to the respective changes in DWII corresponding to SRDM and CLRM. Since the change in the difference between these two DWIIs is not easily traceable (as the mean of a parameter is changed, the sampling of other parameters would be influenced due to the assigned correlation coefficient and this would impact the resulting decision policy), the trend of the difference between EVWIIs with respect to changing the mean of any of these parameters is not easily defined.

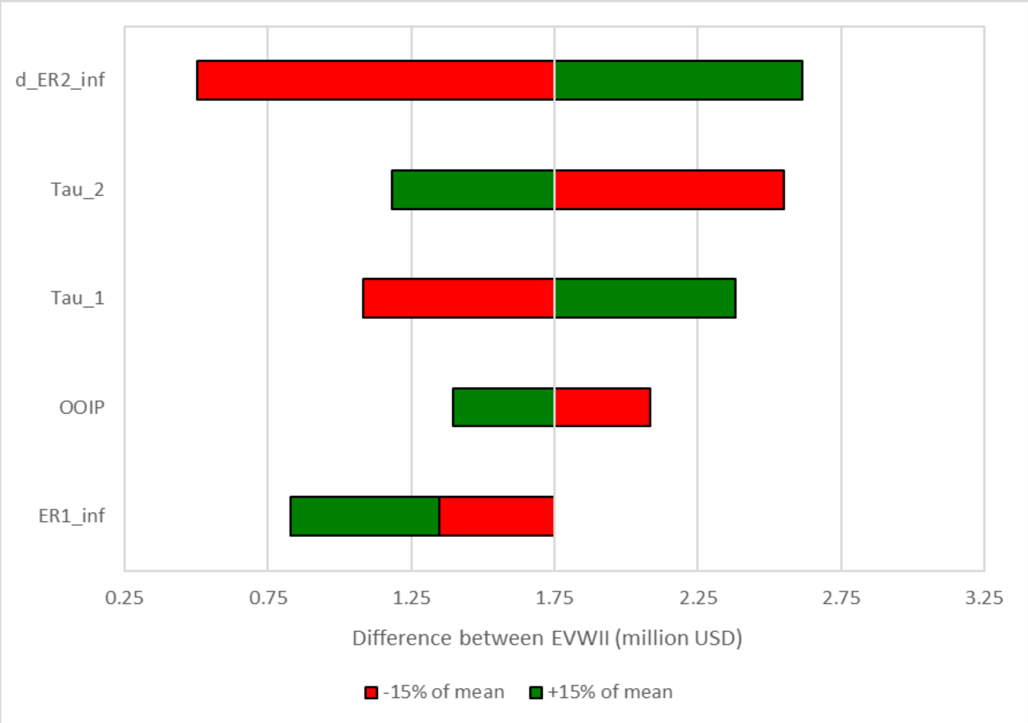


Fig. 14 – Sensitivity Tornado Plot for Difference between EVWIIs of SRDM and CLRM corresponding to the means of Petrophysical Parameters.

Furthermore, Fig. 15 shows that as the percentage change is -20%, the difference between EVWIIs of the mean of E_{R1}^{∞} is relatively much higher than those of others. This is because as the mean of E_{R1}^{∞} is reduced by -20%, the decision policy of CLRM only consists of the lifetime of primary recovery of 0 years[41]. Thus, the corresponding cashflow of this decision policy reduces and the resulting EVWII of CLRM becomes much lower. Thus, the difference between EVWII of SRDM and that of CLRM would be much higher. This also denotes that the corresponding VOI of CLRM would be also much lower as compared to the VOIs of CLRM of others.

[41] Refer to Appendix A: Supplementary Figures for the details of the illustration of the decision policies of SRDM and CLRM for this problem setting. The supplementary figures presented in Appendix A would demonstrate more vividly the difference between these two decision policies which leads to a large difference between both estimated EVWIIs.

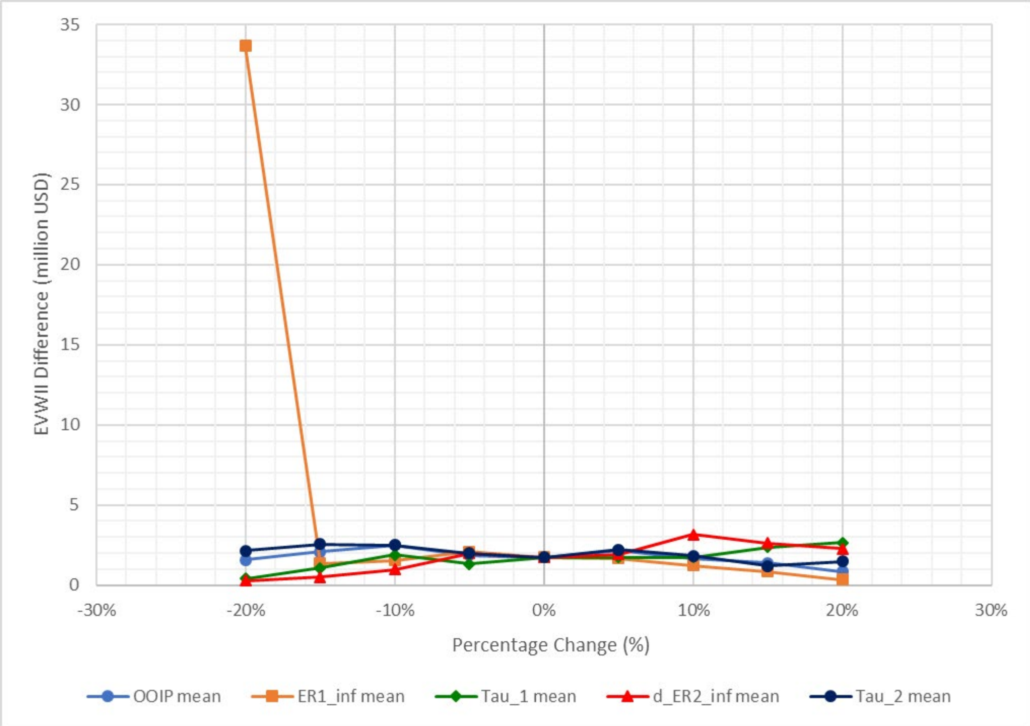


Fig. 15 – Graph of Difference between EVWIIs of the means of Petrophysical Parameters against the percentage change of the corresponding parameters.

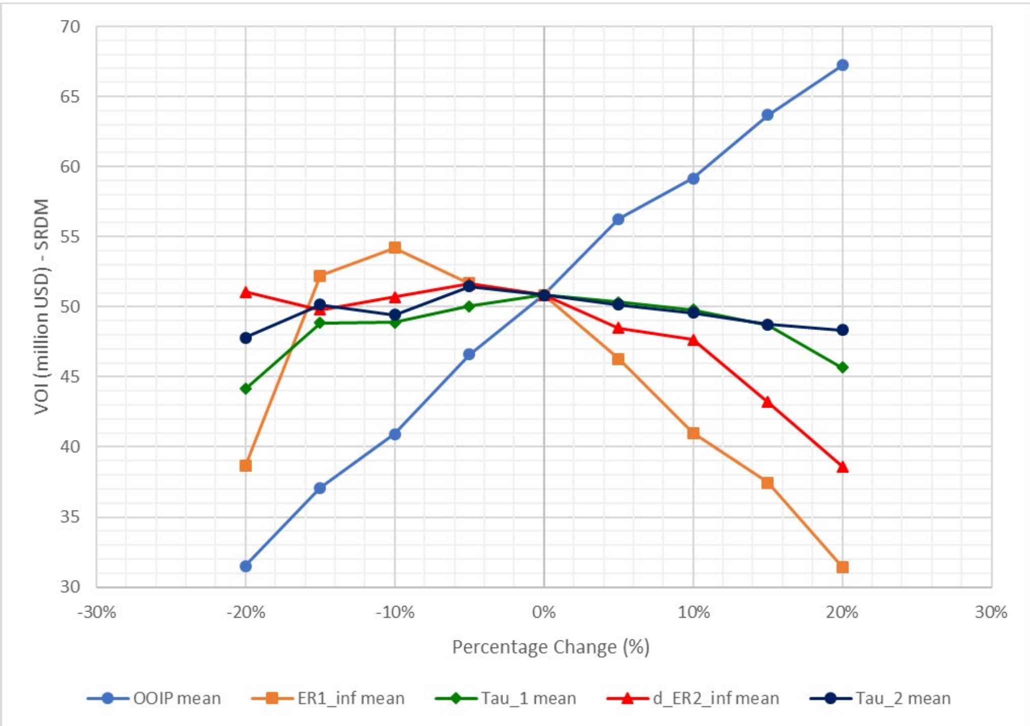


Fig. 16 – Graph of VOIs of the means of Petrophysical Parameters against the percentage change of the corresponding parameters (SRDM approach).

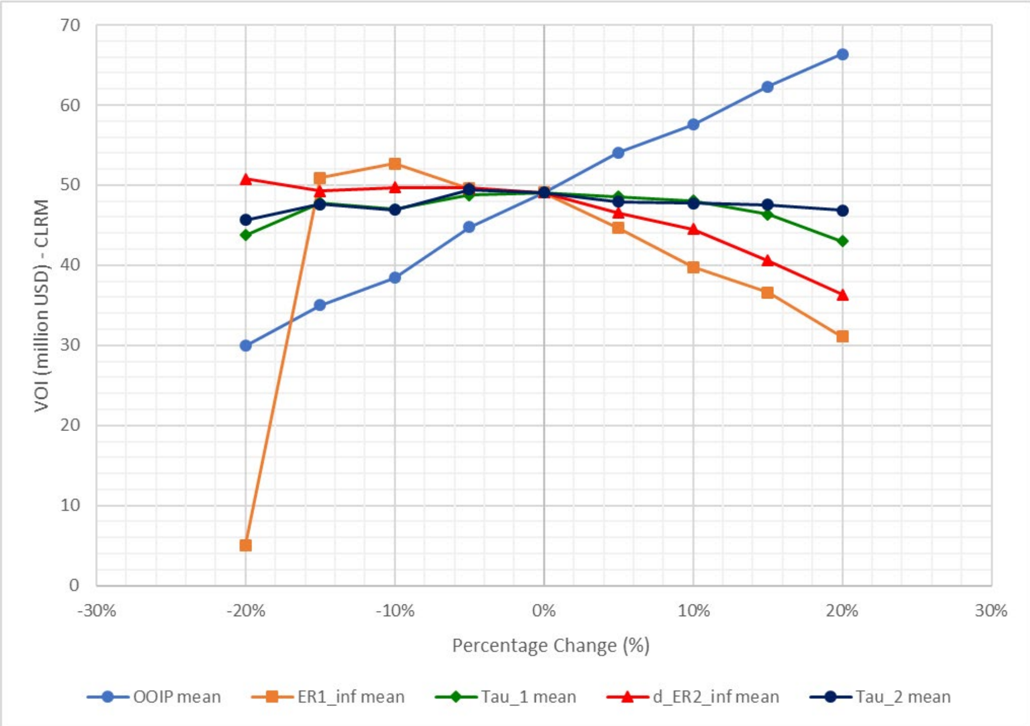


Fig. 17 – Graph of VOIs of the means of Petrophysical Parameters against the percentage change of the corresponding parameters (CLRM approach).

Besides that, the graph of VOIs against the change of the mean of parameters is respectively plotted in Fig. 16 for SRDM and in Fig. 17 for CLRM. Based on Fig. 16 and Fig. 17, VOIs are generally not monotonic with respect to the changes of the means of parameters, except for the mean of OOIP. As the mean of OOIP increases, there is a monotonic increase in the respective VOI. VOIs are highly dependent on DWII and DWOI. In general, as the means of parameters change, the corresponding DWII and DWOI would also change. With respect to this, the change in the difference between DWII and DWOI is also not straightforward. Thus, the trend of the change of VOIs with respect to the change in the mean of parameters can be either monotonic or non-monotonic for a particular parameter.

5.1.2.2 SD of the Petrophysical Parameters

a. EVWII and EVWOI

The tornado sensitivity plots of the EVWIIs (considering SRDM and CLRM) corresponding to the SDs of the petrophysical parameters are respectively illustrated in Fig. 18 and Fig. 19. Both figures portray that the SD of E_{R1}^{∞} is the most impactful variable and is followed by those of ΔE_{R2}^{∞}, OOIP, τ_2, and τ_1. So, by changing the SD of E_{R1}^{∞}, the EVWII would change more drastically in relative to the change of the SDs of other parameters. Regarding the direction of the change of EVWIIs in both tornado plots, it can be observed that as the SD of E_{R1}^{∞}, ΔE_{R2}^{∞} or τ_2 increases, the EVWII would increase. However, for OOIP and τ_2, as their SDs respectively reduce, the resulting EVWIIs increase. For SD, the direction of the change of EVWIIs is not as straightforward as that of in the case of mean. This is because as the SD increases (decreases) with the constant mean of distribution, it indicates that the samples retrieved from the distribution would be more (less) spread out from the mean. This does not mean that the values of the sampled variables would generally either increase or decrease.

Besides that, as compared to the change of the means of the parameters, the change of the SDs is having less impact to the values of EVWIIs. It is because the values of the samples retrieved would change more significantly (which results in much tremendous change in EVWII) as the mean of the parameter alters.

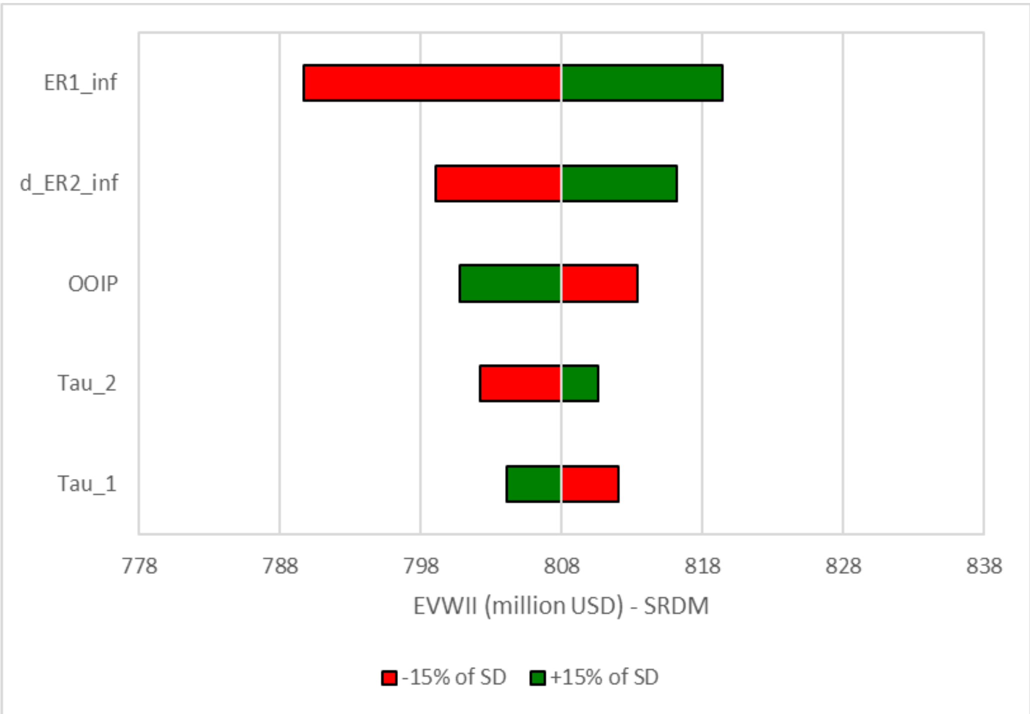


Fig. 18 – Sensitivity Tornado Plot for EVWIIs of the SDs of Petrophysical Parameters (SRDM approach).

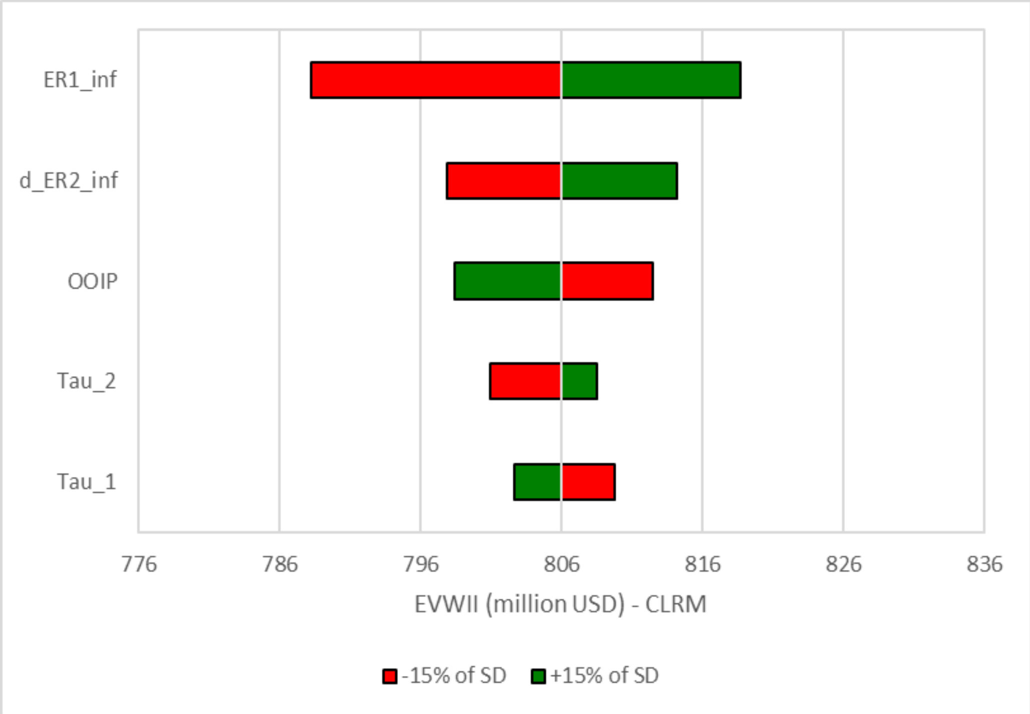


Fig. 19 – Sensitivity Tornado Plot for EVWIIs of the SDs of Petrophysical Parameters (CLRM approach).

The graph of EVWIIs against the percentage change is plotted in Fig. 20 for the case of SRDM and in Fig. 21 for the case of CLRM. These two figures show that only OOIP and E_{R1}^{∞} demonstrate the monotonic trend as the respective SD gradually increases. For the remaining parameters, the resulting EVWIIs fluctuate as the corresponding SD eventually increases. As discussed, the impact of changing the SDs of parameters on the decision policy and the corresponding EVWIIs cannot be easily predicted. Thus, the trend of the change of the EVWIIs with respect to the change of SD might not necessarily be monotonic.

EVWOIs corresponding to the change of the SD of each parameter are estimated and the respective result is shown in Fig. 22. Fig. 22 shows that EVWOIs are fluctuating as the SD of parameters increases. However, the EVWOIs do not fluctuate from each other by a very large amount. Knowing that changing the SDs of parameters does not indicate that the values of samples would increase or decrease in general, its effect on the EVWOIs is also not intuitively predictable. Therefore, a non-monotonic behavior of EVWOI as the SD changes might be observed.

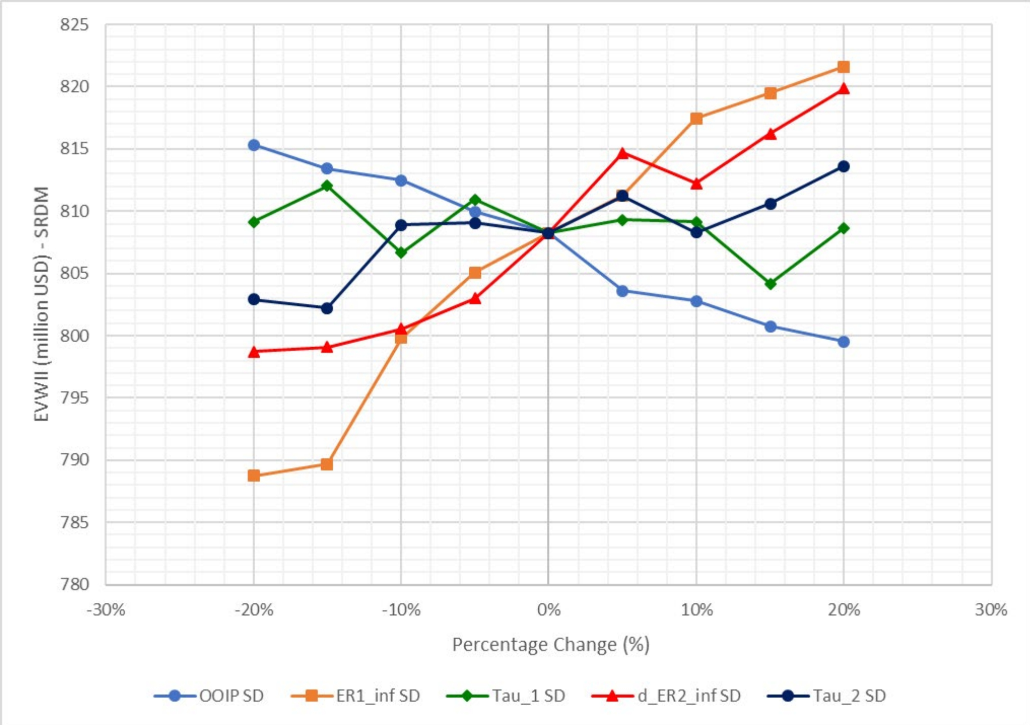


Fig. 20 – Graph of EVWIIs of the SDs of Petrophysical Parameters against the percentage change of the corresponding parameters (SRDM approach).

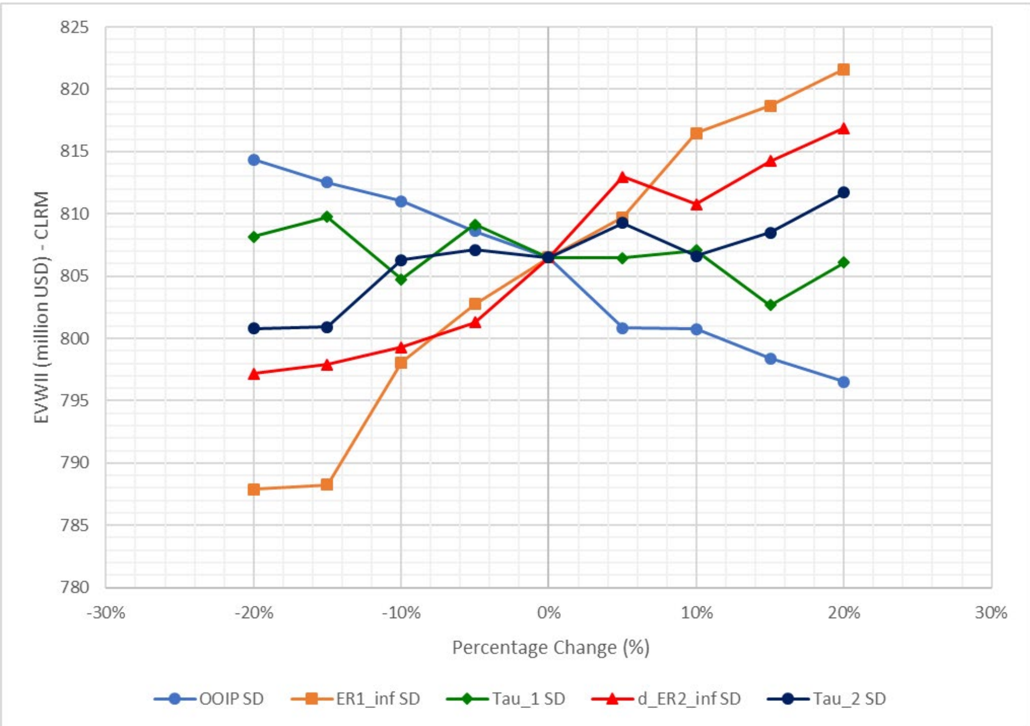


Fig. 21 – Graph of EVWIIs of the SDs of Petrophysical Parameters against the percentage change of the corresponding parameters (CLRM approach).

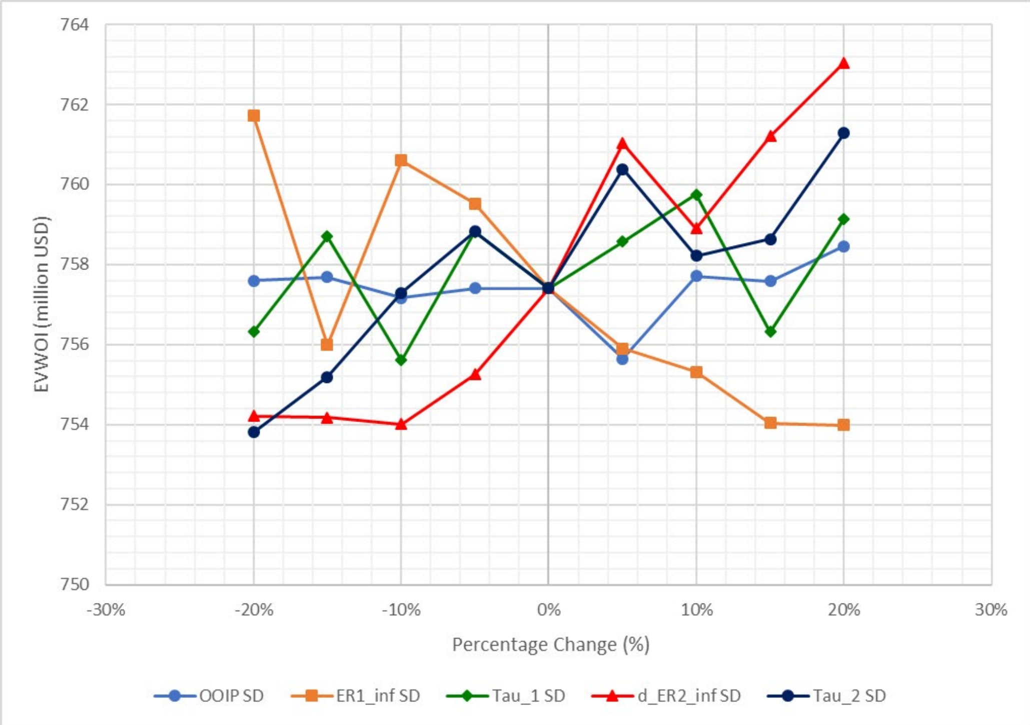


Fig. 22 – Graph of EVWOIs of the SDs of Petrophysical Parameters against the percentage change of the corresponding parameters.

b. Difference between the EVWIIs and VOI

The sensitivity tornado plot regarding the difference between EVWII of SRDM and EVWII of CLRM (corresponding to the SD of parameters) is illustrated in Fig. 23. Based on Fig. 23, it shows that the SD of OOIP is the most impactful variable and is followed by those of ΔE_{R2}^{∞}, τ_2, τ_1, and E_{R1}^{∞}. Therefore, the result demonstrates that having higher SD of OOIP would produce a larger difference between EVWII of SRDM and EVWII of CLRM. As compared to the change of the means of the parameters, the change of the SDs of the parameters is generally less impactful to the difference between EVWIIs. The reason is also due to the more drastic change in the values of samples retrieved as the means of distribution changes. Besides that, from the tornado plot, it can also be noted that for E_{R1}^{∞}, the resulting difference between the EVWIIs shows the non-monotonic trend as the corresponding SD increases.

With respect to this, the graph of the difference between the EVWIIs against the percentage change is plotted in Fig. 24. In general, the difference between the EVWIIs fluctuate with respect to the changes of the SD of each parameter. There is not a clear trend regarding the change of difference between the EVWIIs in response to the change of SD of the parameters due to the complexity of the change in terms of the difference between DWIIs of SRDM and CLRM.

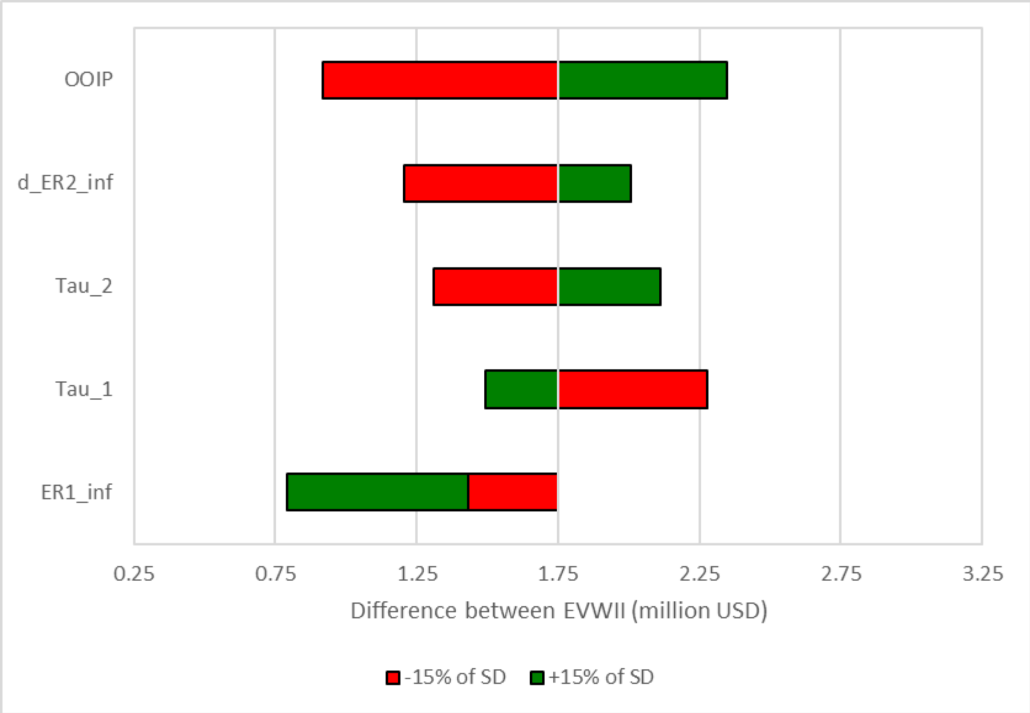


Fig. 23 – Sensitivity Tornado Plot for Difference between EVWIIs corresponding to the SDs of Petrophysical Parameters.

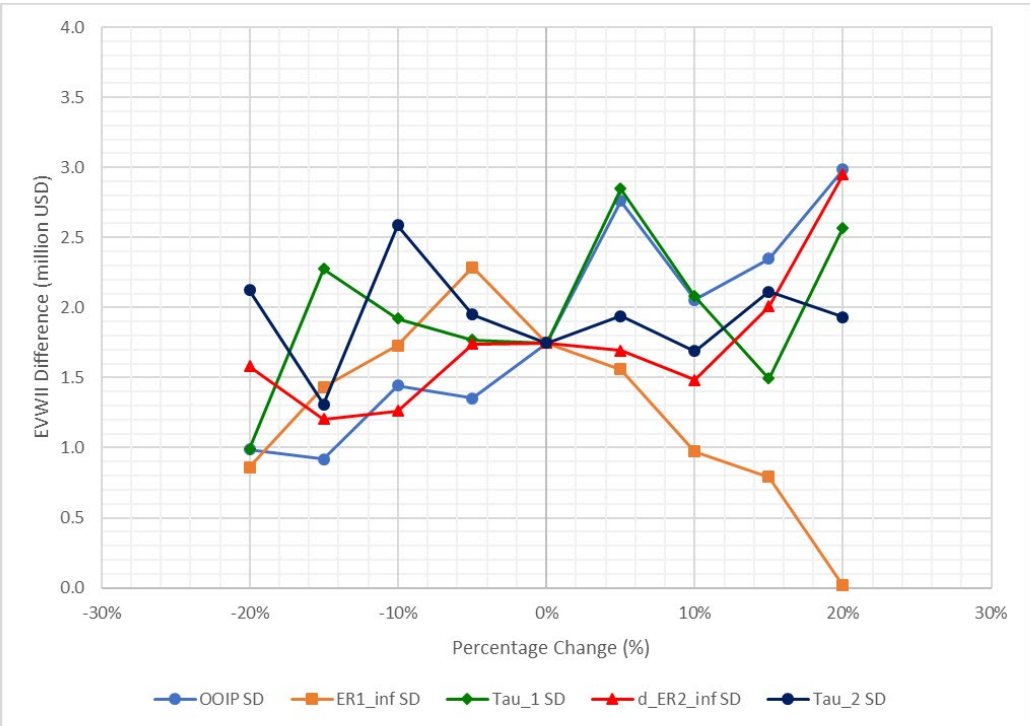


Fig. 24 – Graph of Difference between EVWIIs of the SDs of Petrophysical Parameters against the percentage change of the corresponding parameters.

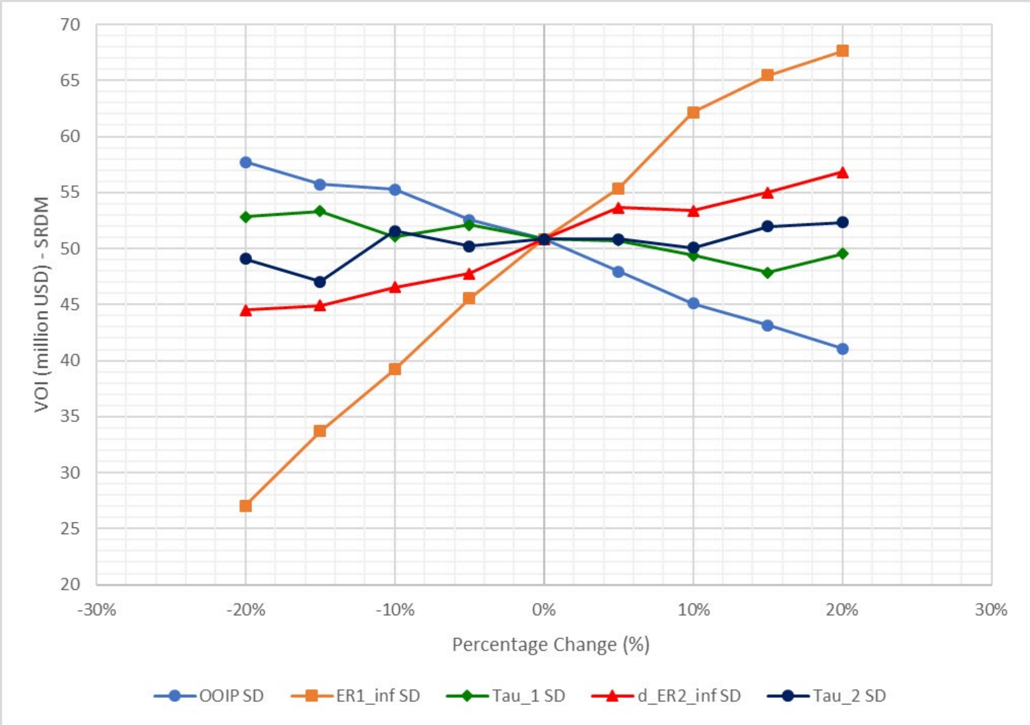


Fig. 25 – Graph of VOIs of the SDs of Petrophysical Parameters against the percentage change of the corresponding parameters (SRDM approach).

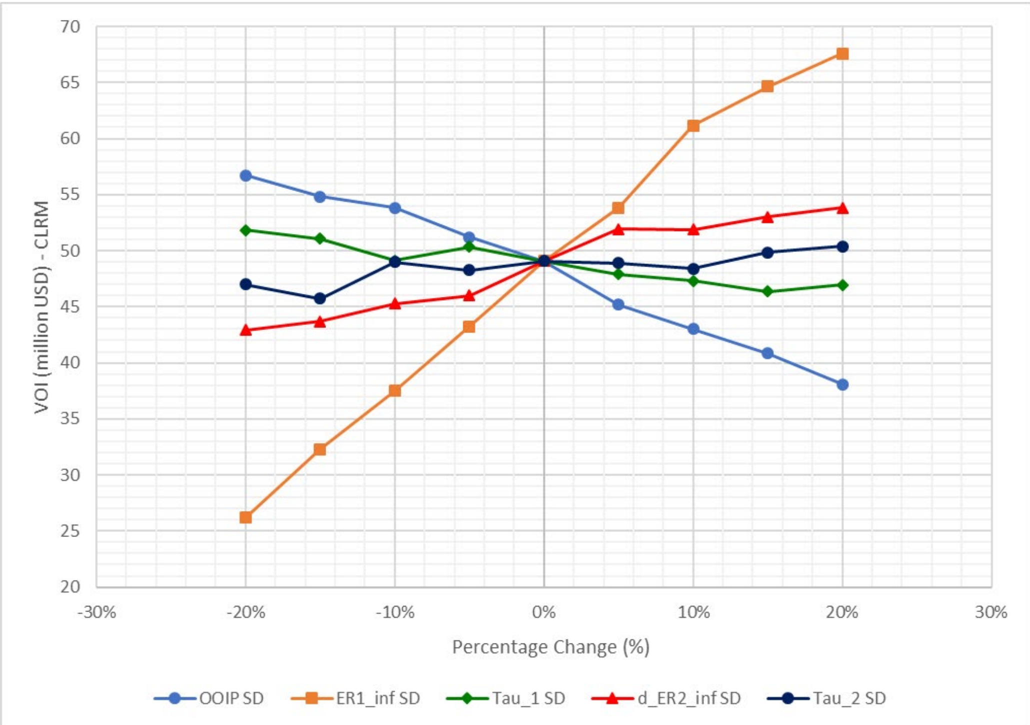


Fig. 26 – Graph of VOIs of the SDs of Petrophysical Parameters against the percentage change of the corresponding parameters (CLRM approach).

The graph of VOI against the change of the SD of parameters is respectively plotted in Fig. 25 for SRDM and in Fig. 26 for CLRM. Based on Fig. 25 and Fig. 26, VOIs fluctuate slightly with respect to the changes of the SD of parameters, except for the SDs of OOIP, E_{R1}^{∞}, and ΔE_{R2}^{∞}. As the SD of E_{R1}^{∞} increases, the corresponding VOI increases monotonically. Besides that, for OOIP, as the SD increases, the respective VOI decreases monotonically. For ΔE_{R2}^{∞}, as the SD becomes higher, the respective VOIs generally increases until the percentage change is 5% at which a decrease in value is seen and followed by increase in value. However, the trend of increase is generally shown for ΔE_{R2}^{∞}. When there is a monotonic increase (decrease) in VOIs, this indicates that the difference between EVWII and EVWOI is generally getting larger (smaller). Such distinct trend is observed in the cases of the SDs of OOIP, E_{R1}^{∞}, and ΔE_{R2}^{∞}. As the SD of parameters change, the corresponding DWIIs would be altered. Thus, the change in the difference between DWII and DWOI is also not traceable as discussed. Thus, the trend of the change of VOIs with respect to the change in the SD of parameters might not be distinct depending on the result of the sampling of the respective parameter as its SD is altered.

5.1.2.3 Economic Parameters

a. EVWII and EVWOI

The tornado sensitivity plots of the EVWIIs (considering SRDM and CLRM) corresponding to the economic parameters are respectively shown in Fig. 27 and Fig. 28. Both figures show that the oil price (or as the initial oil price as listed in Table 4) is the most impactful variable and is followed by discount rate, OPEX of secondary recovery, OPEX of primary recovery, CAPEX of primary recovery, CAPEX_2After1, and CAPEX_2No1.

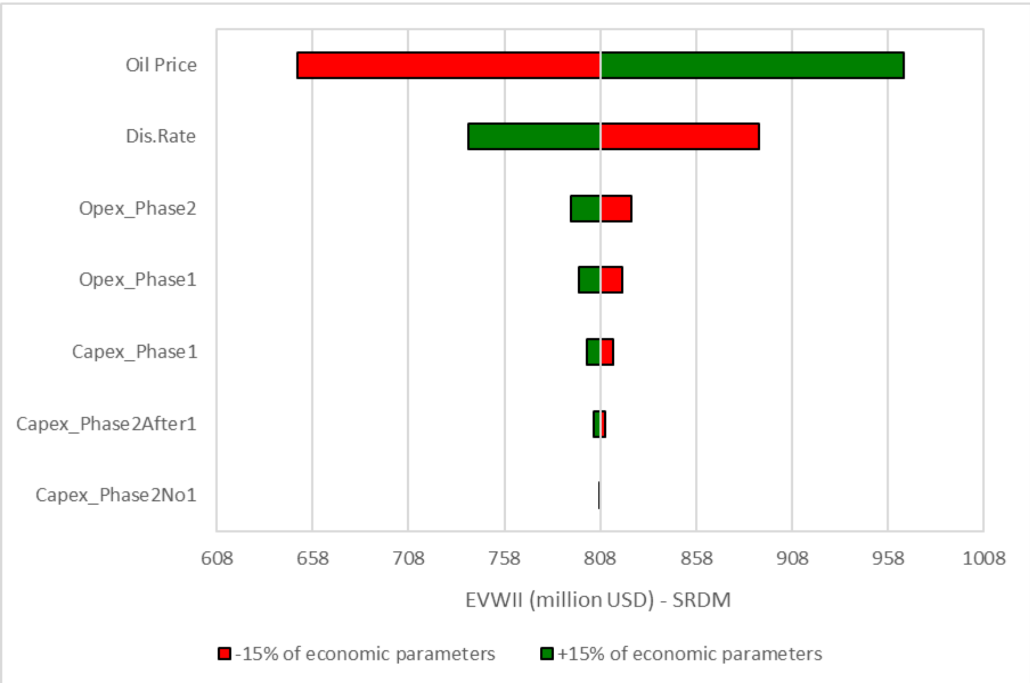


Fig. 27 – Sensitivity Tornado Plot for EVWIIs of the Economic Parameters (SRDM approach).

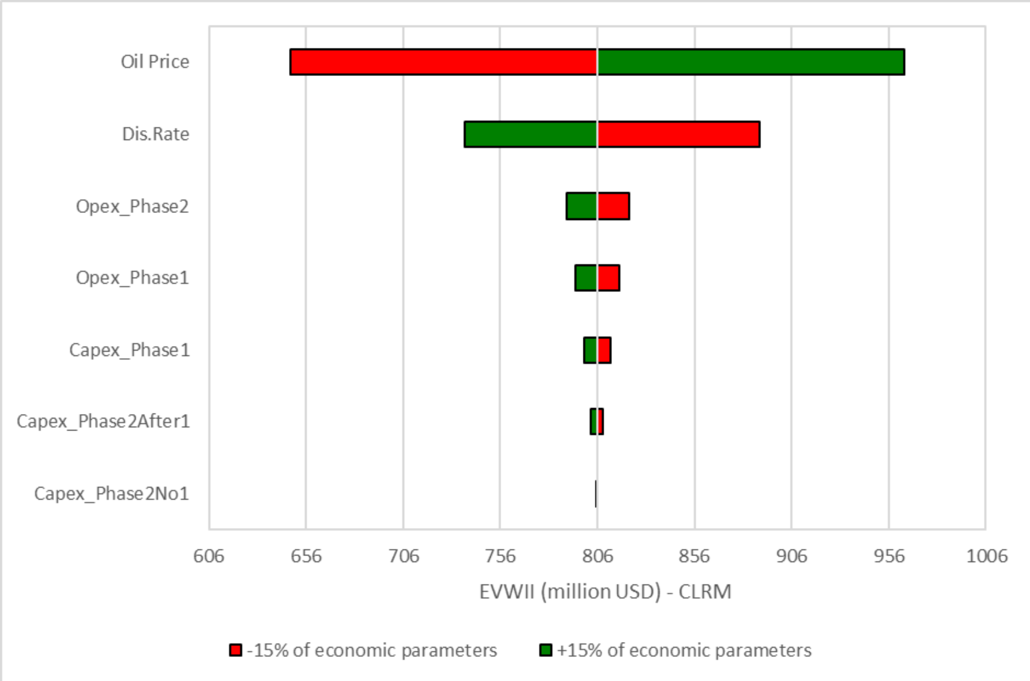


Fig. 28 – Sensitivity Tornado Plot for EVWIIs of the Economic Parameters (CLRM approach).

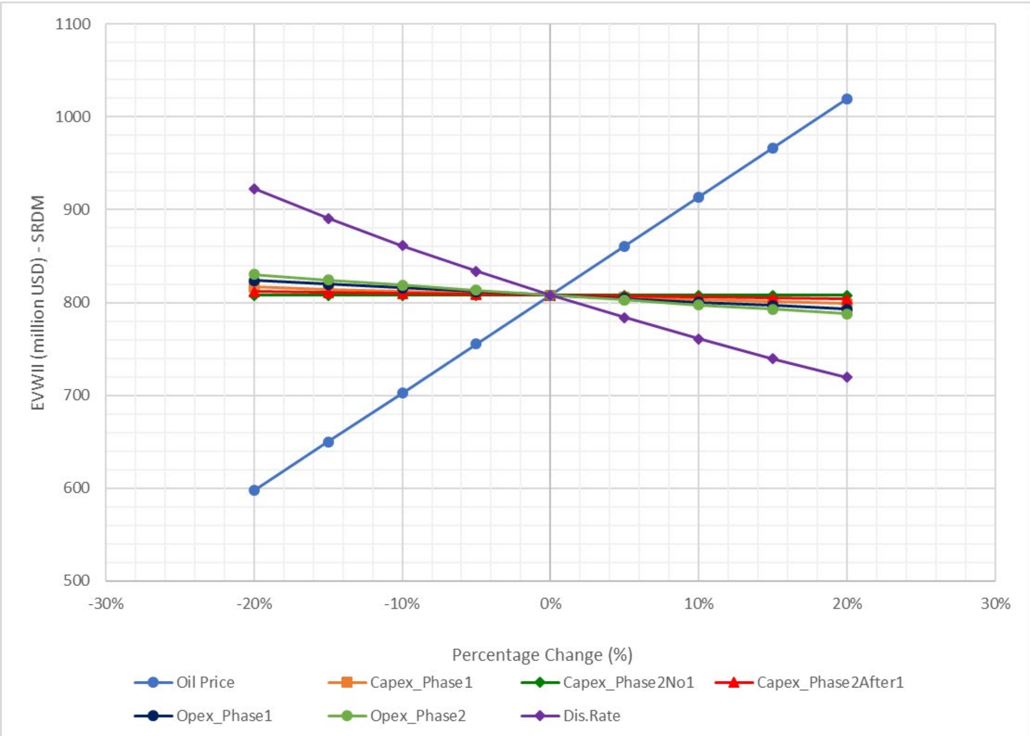


Fig. 29 – Graph of EVWIIs of the Economic Parameters against the percentage change of the corresponding parameters (SRDM approach).

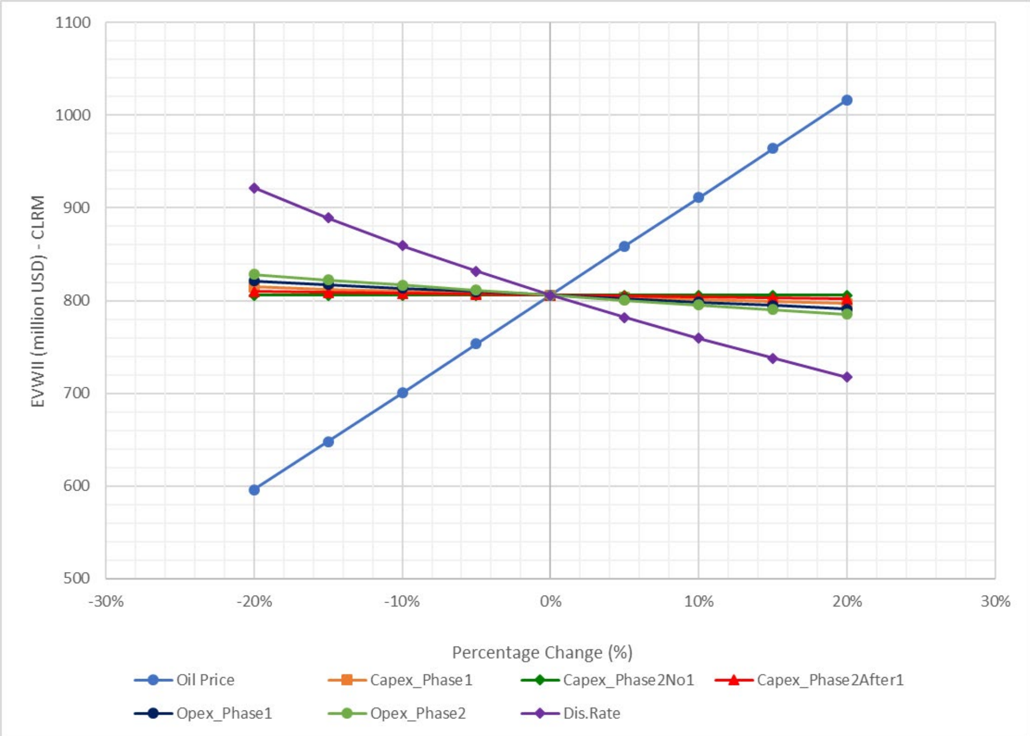


Fig. 30 – Graph of EVWIIs of the Economic Parameters against the percentage change of the corresponding parameters (CLRM approach).

So, by altering the oil price, the EVWII would change more significantly in relative to the change of the other economic parameters. Pertaining to the direction, it is understandable that as oil price increases (decreases), the cashflow would increase (decrease) and thus, the corresponding EVWIIs would increase (decrease). The opposite situation applies to costs and discount rate.

Besides that, EVWIIs corresponding to the changes of plus and minus 10% as well as plus and minus 20% of the economic parameters are computed. The graph of EVWIIs against the percentage change is plotted in Fig. 29 for SRDM and in Fig. 30 for CLRM. These two figures show that the resulting EVWIIs would monotonically increase as the oil price increases. However, for the costs (except for CAPEX_2No1) and discount rate, as the respective value increases, the EVWIIs would decrease. For CAPEX_2No1, the EVWIIs remain constant as its value increases. This means that in this case, changing CAPEX_2No1 would not affect the decision policy at all since none of the realizations would skip primary recovery and directly proceed to secondary recovery.

EVWOIs corresponding to each of the economic parameters are estimated and the respective result is shown in Fig. 31. Fig. 31 shows that the trend of the change of EVWOIs is behaving like the one being illustrated in the case of EVWIIs. In this aspect, higher oil price, lower costs (except for CAPEX_2No1), and lower discount rate would increase EVWOI and vice versa. EVWOI remains constant as CAPEX_2No1 changes. So, the same explanation used for the case of EVWIIs also applies here.

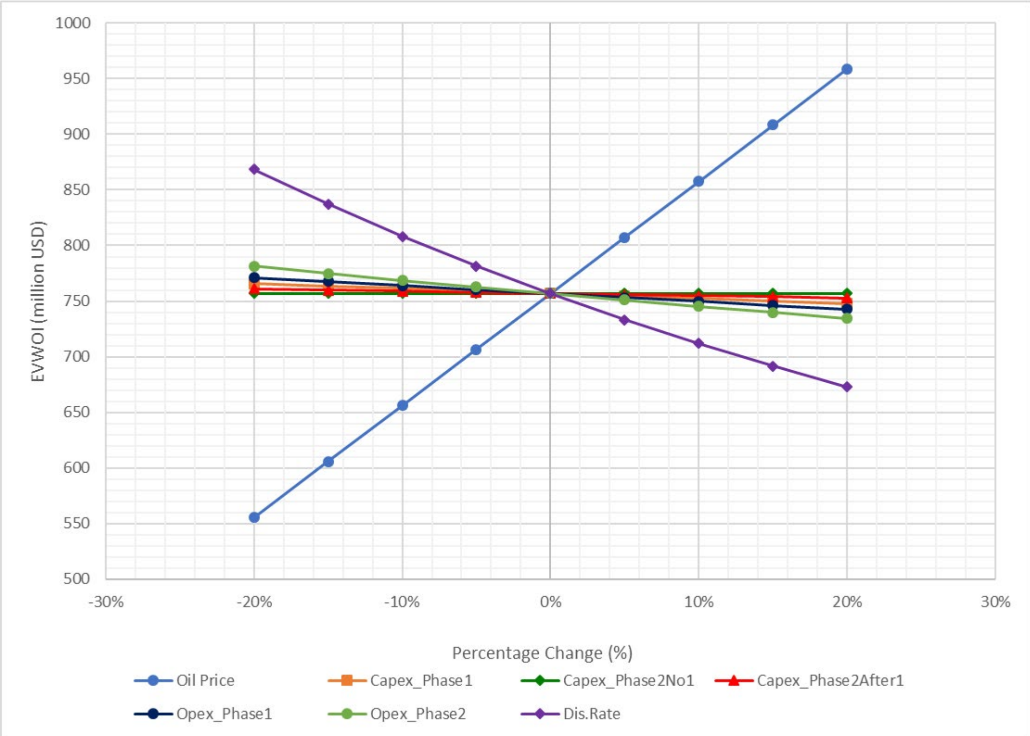


Fig. 31 – Graph of EVWOIs of the Economic Parameters against the percentage change of the corresponding parameters.

b. Difference between the EVWIIs and VOI

The sensitivity tornado plot regarding the difference between EVWII of SRDM and EVWII of CLRM (corresponding to the economic parameters) is shown in Fig. 32. Based on Fig. 32, it shows that the oil price is the most impactful variable and is followed by OPEX of primary recovery, OPEX of secondary recovery, discount rate, CAPEX_2After1, CAPEX_2No1, and CAPEX of primary recovery. By changing the oil price, the EVWII would change more significantly in relative to the change of the other economic parameters. If Fig. 32 is observed carefully, both CAPEX_2No1 and CAPEX of primary recovery do not create any impact to the difference in EVWIIs. As explained, changing CAPEX_2No1 is not impactful as all the realizations skip primary recovery. However, for CAPEX of primary recovery, as it changes, the difference between both EVWIIs stays the same. This indicates that changing CAPEX of primary recovery is not impactful to cause a change in DWIIs. Besides that, there is a non-monotonic behavior shown by certain parameters, such as discount rate and OPEX of secondary recovery in the tornado plot. With respect to this, the graph of the difference between the EVWIIs against the percentage change is plotted in Fig. 33.

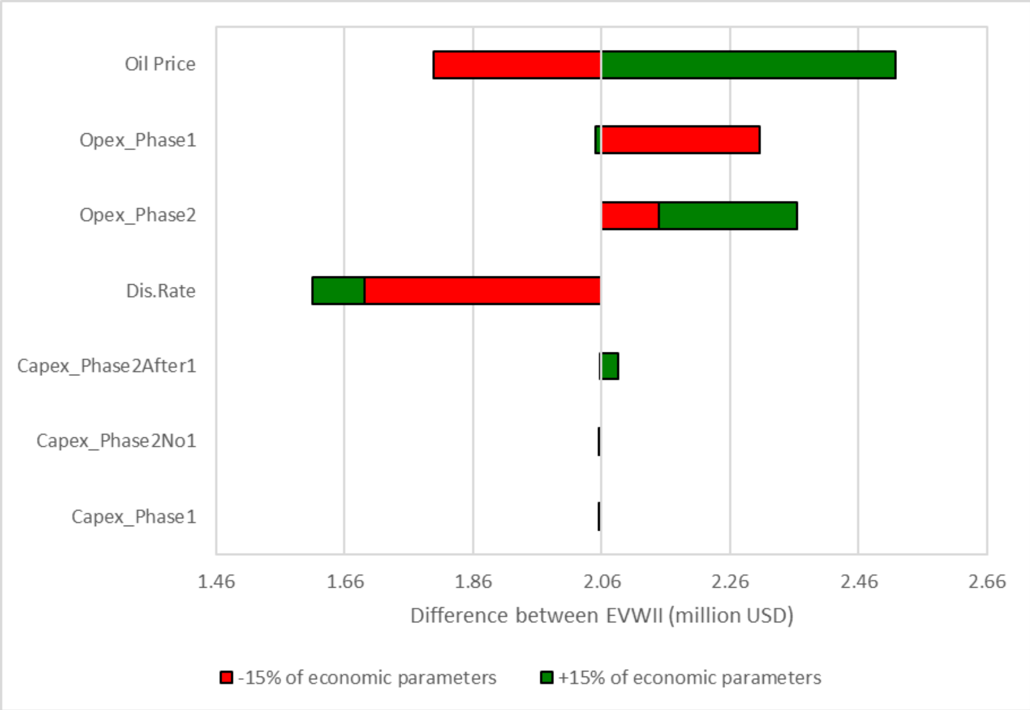


Fig. 32 – Sensitivity Tornado Plot for Difference between EVWIIs corresponding to the Economic Parameters.

Fig. 33 shows that the difference between the EVWIIs fluctuates as the change of the economic parameters increases, except for the cases of CAPEX of primary recovery, and CAPEX_2No1. For CAPEX of primary recovery, since changing it would not alter the resulting decision policies for both SRDM and CLRM, it is perceivable that the difference between EVWIIs remain unchanged. For CAPEX_2No1, it is understandable that as it increases, the resulting difference between two EVWIIs is constant because its corresponding EVWIIs of SRDM and CLRM remain constant as discussed earlier. However, in general, the trend of the difference between the EVWIIs with respect to the change of the economic parameters relies on the difference between the DWIIs corresponding to both SRDM and CLRM. The influence of the change of the economic parameters on the DWIIs is not straightforward as discussed and thus, it results in the fluctuation as shown in Fig. 33. Thus, this might result in the non-monotonic behavior shown in Fig. 32 for some parameters depending on the problem setting.

The graph of VOIs against the change of the economic parameters is respectively plotted in Fig. 34 for SRDM and in Fig. 35 for CLRM. Based on Fig. 34 and Fig. 35, VOIs are generally monotonic as the economic parameters change. This is because both EVWIIs and EVWOIs are monotonic with respect to the change of the parameters as explained.

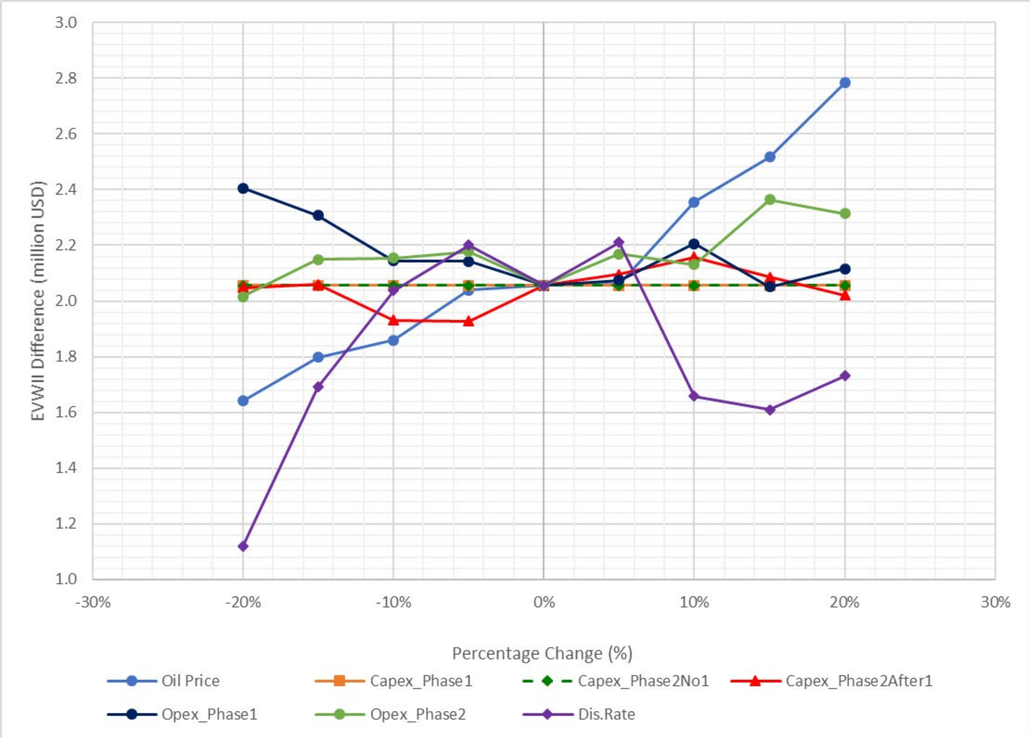


Fig. 33 – Graph of Difference between EVWIIs of the Economic Parameters against the percentage change of the corresponding parameters.

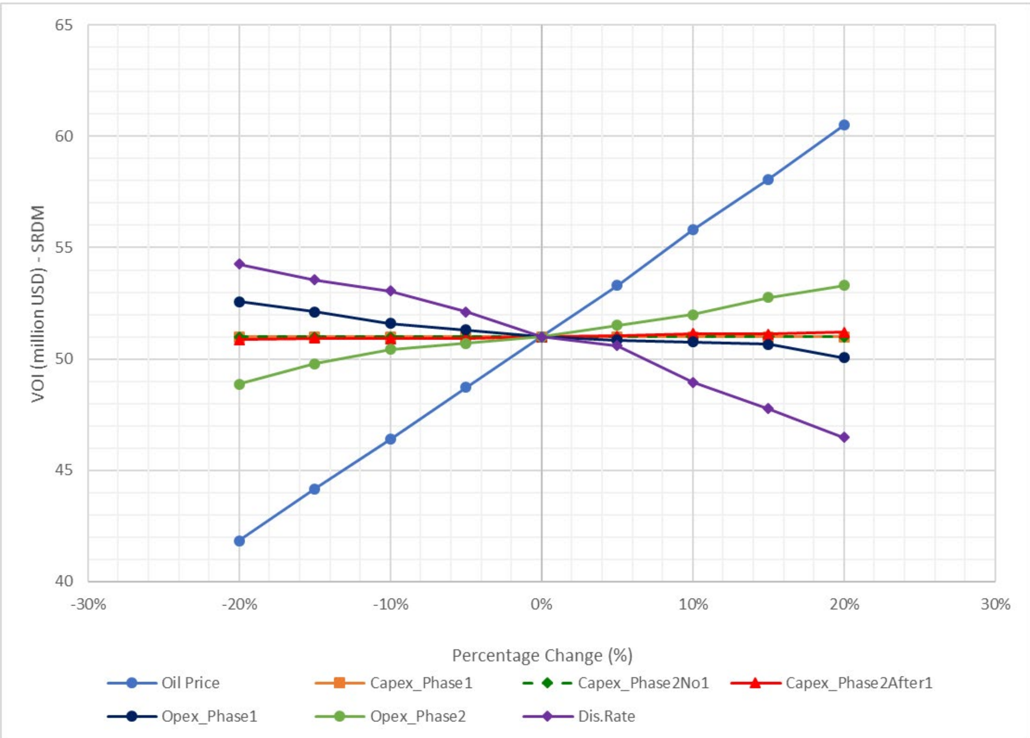


Fig. 34 – Graph of VOIs of the Economic Parameters against the percentage change of the corresponding parameters (SRDM approach).

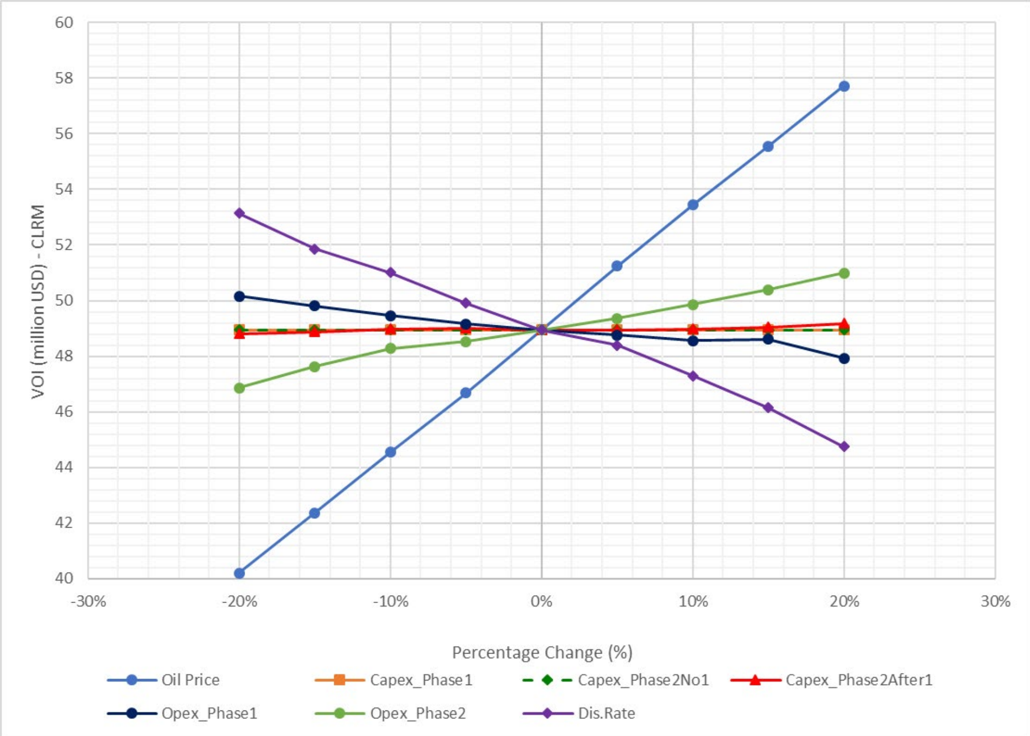


Fig. 35 – Graph of VOIs of the Economic Parameters against the percentage change of the corresponding parameters (CLRM approach).

5.2 Number of Decision Points and Number of Data Points

Aside from the model parameters, sensitivity analysis is also performed on the number of decision points and the number of data points. For the sensitivity analysis on the number of decision points, the decision spaces related to the lifetime of primary recovery and that of secondary recovery need to be changed. In this aspect, the available options to be analyzed consist of the lifetime of primary recovery for n-1 years if their corresponding decision points are placed at Year n. When the decision is made every 10 years and the first decision point is placed at Year 1 (the remaining decision points are placed at Year 11, Year 21, Year 31, and Year 41), the available options only have the lifetimes of primary recovery to be 0 year, 10 years, 20 years, 30 years, 40 years, and 50 years. The same logic applies to the corresponding lifetime of the secondary recovery. For example, the options corresponding to 0 years of primary recovery consist of 0 years, 10 years, 20 years, 30 years, 40 years, and 50 years of secondary recovery.

Although the decision space is changed during the sensitivity analysis on the number of decision points, the information space remains the same. This means that the measured oil production rate is still collected on yearly basis and used accordingly in the regression analysis. As the number of decision points increases, the EVWII would also increase[42]. In other words, when a person has more flexibility in making a decision (indicated by more decision points), the larger the value gain will be. If such trend is shown as the result of the sensitivity analysis, this further proves that the modified LSM algorithm is implemented correctly in the context of IOR initiation problem. Therefore, this sensitivity analysis acts as a cross-check for the validity of the implementation of the algorithm. The same applies to the implementation of CLRM.

Pertaining to the sensitivity analysis on the number of data points[43], the regression analysis is done by using a fixed number of data points (including the data point at current decision point) in which different number of data points can be applied. Hong et al. (2018) initially assumed that the measured oil production rates were modeled as non-Markovian processes as expounded earlier. Regarding this, the number of data points used would change according to the year that a decision point is at. This sensitivity analysis is insightful because in real life, gathering more data would induce additional cost. Therefore, if having more data do not create more value[44], it is more practical and economical to use fewer data points.

[42] Since VOI is dependent on EVWII, it would also increase with the number of decision points. However, VOI is also influenced by EVWOI. Thus, the relationship between VOI and number of data points is also demonstrated in this case for further insights.

[43] This is to reiterate that, for the sensitivity analysis on the number of data points, the number of decision points used is 50.

[44] Results estimated using less data points are close to those estimated using more data points.

5.2.1 Procedure of Sensitivity Analysis

In Hong et al. (2018), the decision is analyzed and made every year throughout the 50 years of the production life of the oil field. Therefore, to conduct the sensitivity analysis, decision is instead analyzed and made every 2 years, every 4 years, every 6 years, every 8 years, and every 10 years. As the sensitivity analysis is being done, the number of options (decision space) would need to be changed. Since the available options to be analyzed would be reduced, the DWOI has to be redetermined and the corresponding EVWOI has to be estimated again. The same logic applies to DWPI and DWIIs (considering both SRDM and CLRM approaches). Therefore, the corresponding EVWPI and EVWIIs have to be estimated again.

Besides that, for the sensitivity analysis on the number of data points, 9 different number of data points are assigned, namely 1 (the measured rates are modeled as Markovian processes), 2, 3, 4, 5, 10, 20, 30, and 40. In this aspect, the number of data points used in the regression analysis would be kept as defined unless the decision maker is at the point of time when he or she has less data points available than the number of data points defined. When such situation happens, all the available data points would be used in the regression analysis.

A different problem setting has been used to perform the sensitivity analysis on both the number of decision points and the number of data points. In this new setting, the SD of OOIP is changed to 42 MMbbl instead of being 35 MMbbl, the SD of ΔE_{R2}^{∞} is changed from 0.05 to 0.06, and the oil price is changed from 50 USD/bbl to 52.5 USD/bbl. Other parameters remain the same. Then, 20000 MCS paths are used for further calculation. With respect to this, EVWII corresponding to SRDM is estimated to be \$862.97 million whereas that of CLRM is \$859.85 million. This yields the difference between the two EVWIIs to be \$3.12 million. For EVWOI, it is \$814.72 million. Thus, the VOI corresponding to SRDM is \$48.25 million and that of CLRM is \$45.13 million.

5.2.2 Results and Discussions

5.2.2.1 Number of Decision Points

Fig. 36 shows that the graph of EVWIIs (corresponding to SRDM) against decision made per 10 years, 8 years, 6 years, 4 years, 2 years, and 1 year. Besides that, Fig. 37 illustrates the similar graph as shown in Fig. 36, but it is done with respect to CLRM. From both figures, it can be observed that EVWIIs increase linearly when the year between two consecutive decisions is shortened (the number of decision points increases). This is because as the number of decision points decreases, the available options would also reduce. As less options are available to be analyzed, the resulting EVWII (considering both SRDM and CLRM) would also decrease. These figures also confirm that higher flexibility in decision making would yield higher EVWIIs.

Apart from this, Fig. 38 demonstrates that the graphs of VOIs (corresponding to SRDM) against decision made per 10 years, 8 years, 6 years, 4 years, 2 years, and 1 year. With this,

Fig. 39 shows the similar graph as shown in Fig. 38, but it is done with respect to CLRM. From both figures, in general, VOIs increase exponentially when the number of decision points increases. However, under scrutiny, between the decisions made per 6 years and 1 year (last 4 bars for both figures), VOIs increase linearly with the number of decision points. Such trend of increase is caused by the change of EVWOIs with respect to the number of decision points. The graph of EVWOIs against decision made per 10 years, 8 years, 6 years, 4 years, 2 years, and 1 year is illustrated in Fig. 40. When the number of decision points increases, EVWOIs increase until decision made is per 4 years and remain constant onwards (not exactly constant but with miniscule difference between the values). EVWOI of decision made per 4 years is same as that of decision made per 2 years, which is \$814.59 million. However, these EVWOIs are different from that of decision made per year, which is \$814.72 million. For this case, the impact of changing the number of decision points to EVWOI results in the behavior of increase as shown in Fig. 38 and Fig. 39. In short, these results further substantiate the correct implementation of the modified LSM algorithm (as well as CLRM) in the context of IOR initiation problem.

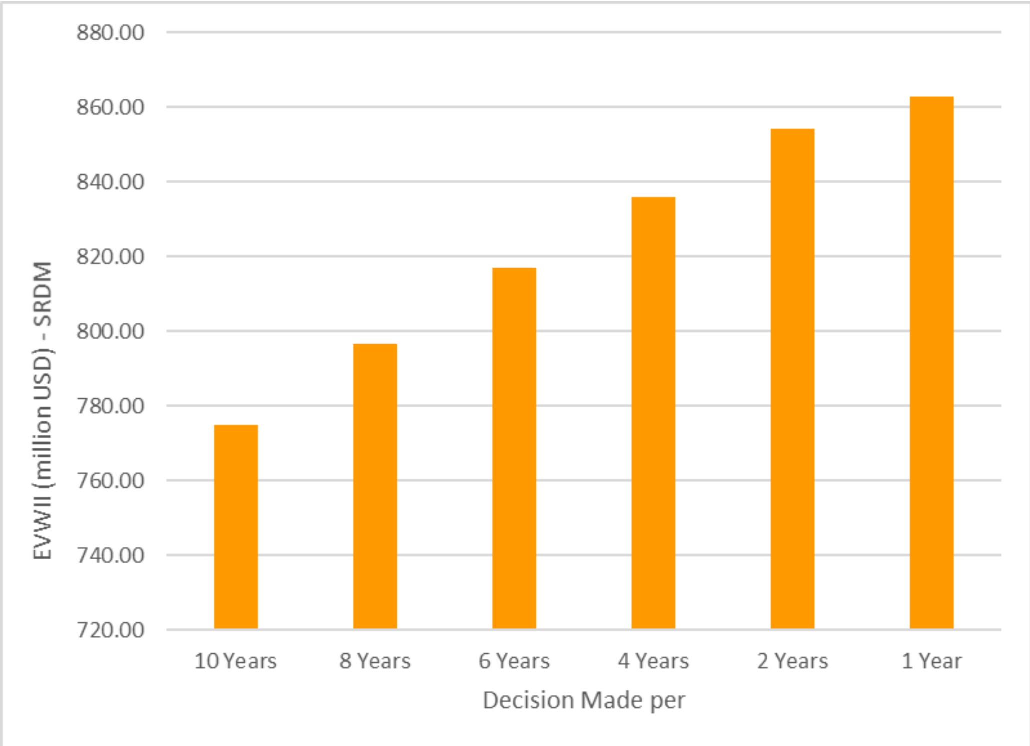


Fig. 36 – Graph of EVWIIs against the number of years between two consecutive decisions (SRDM approach).

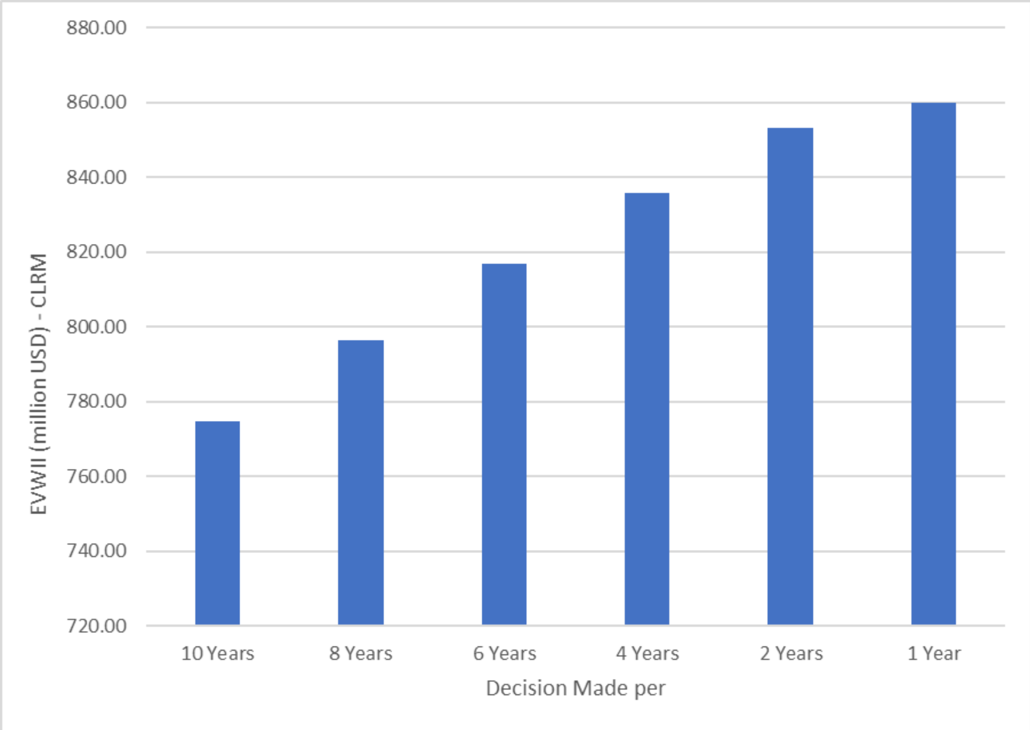


Fig. 37 – Graph of EVWIIs against the number of years between two consecutive decisions (CLRM approach).

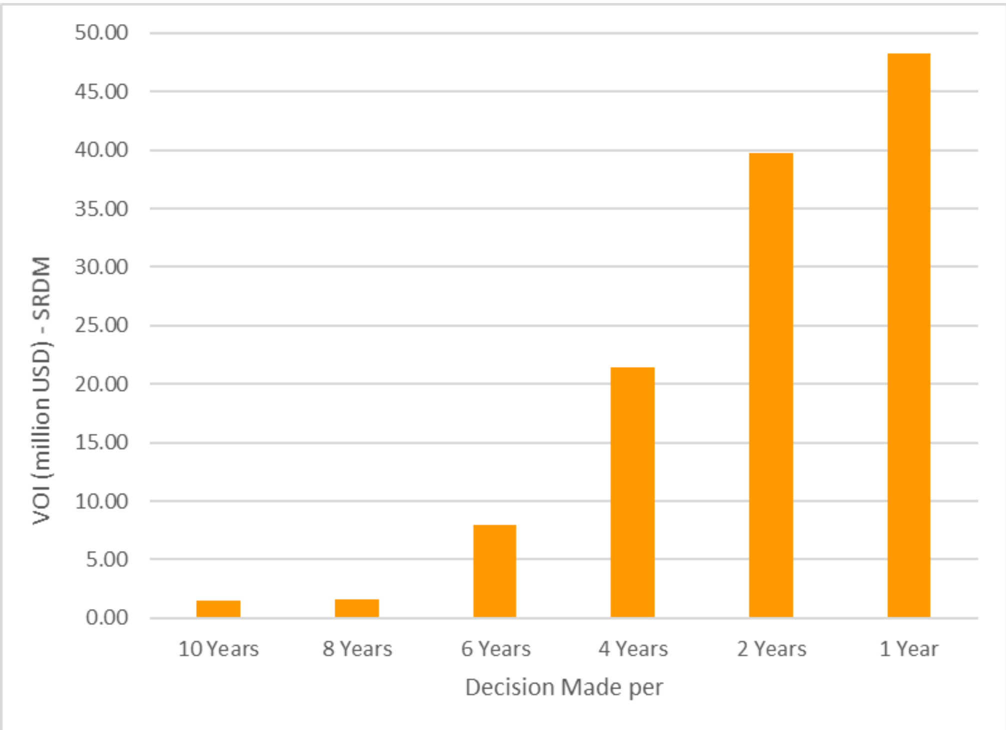


Fig. 38– Graph of VOIs against the number of years between two consecutive decisions (SRDM approach).

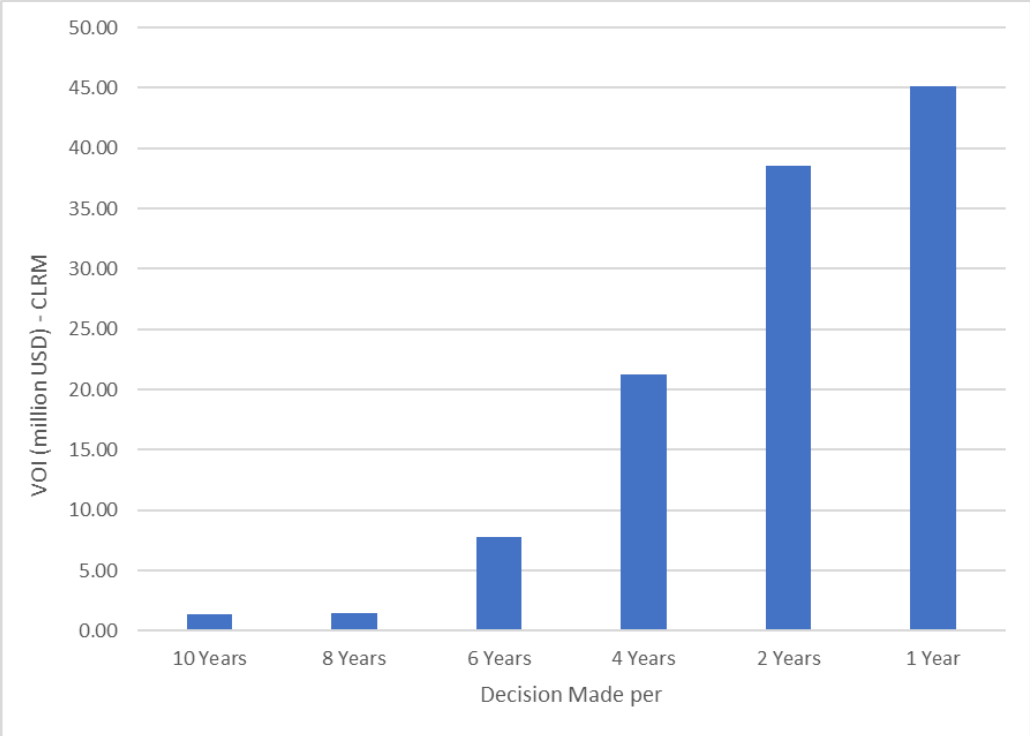


Fig. 39 – Graph of VOIs against the number of years between two consecutive decisions (CLRM approach).

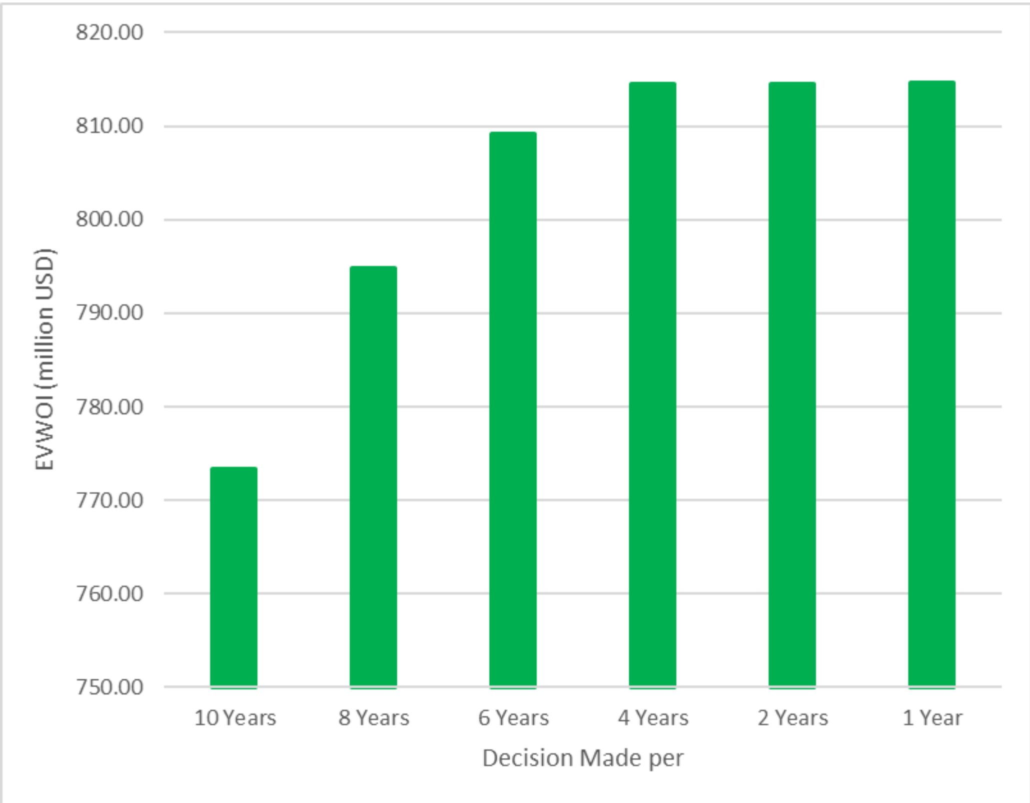


Fig. 40 – Graph of EVWOIs against the number of years between two consecutive decisions.

5.2.2.2 Number of Data Points

Fig. 41 shows that the graph of EVWII (corresponding to SRDM) against the number of data points of 1, 2, 3, 4, 5, 10, 20, 30, and 40 whereas Fig. 42 demonstrates the same case corresponding to CLRM. It can be observed that using fewer data points generally still provides an estimation of EVWII that is close to the one approximated by using all the available data.

To more vividly illustrate this, the percentage difference is computed by finding the percentage of the fraction of difference between EVWII estimated using less data points and EVWII estimated using all available data (non-Markovian processes) to EVWII estimated using all available data. Then, the graph of the percentage difference against the number of data points is plotted in Fig. 43 for SRDM and in Fig. 44 for CLRM.

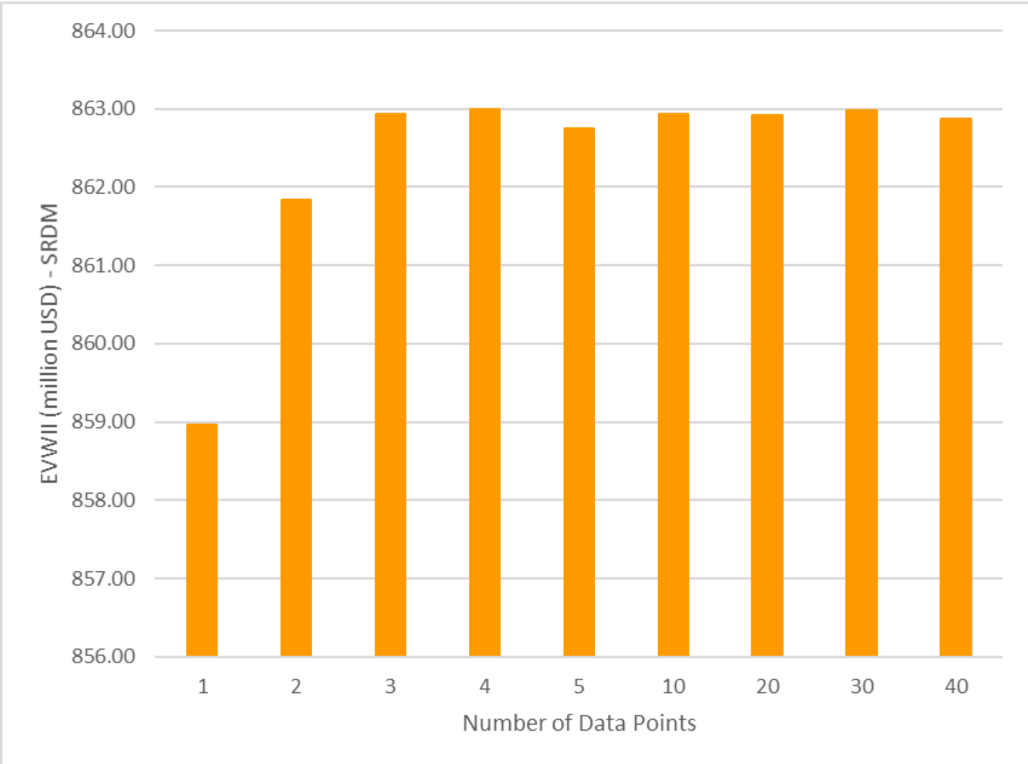


Fig. 41 – Graph of EVWIIs against the number of data points (SRDM approach).

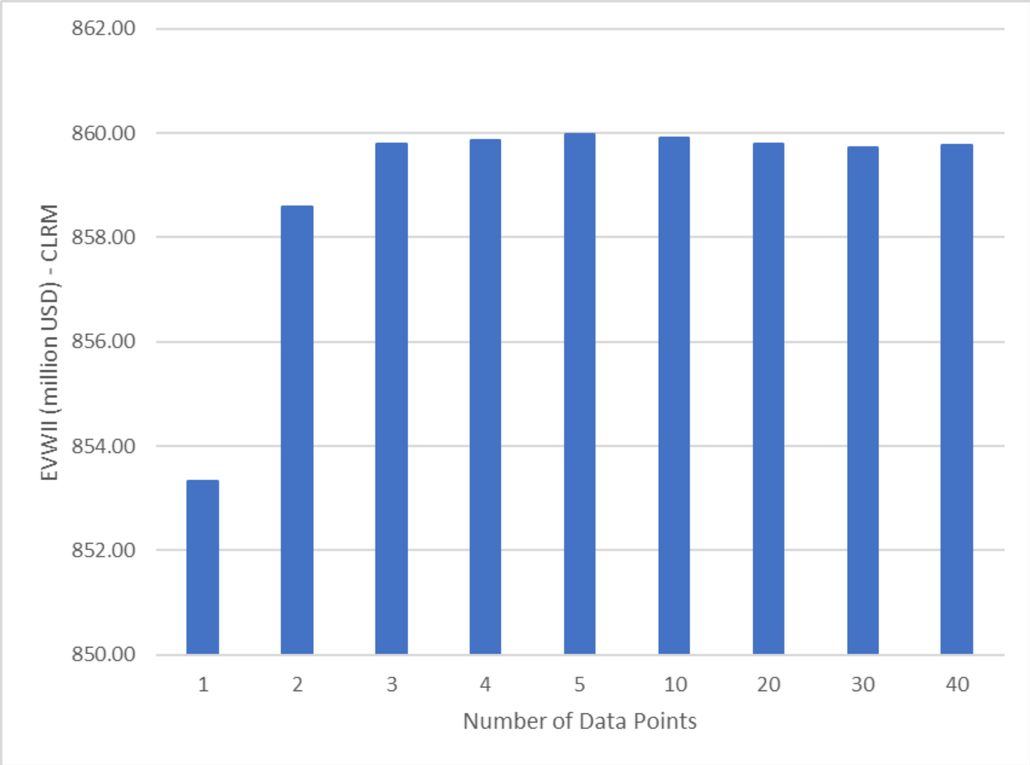


Fig. 42 – Graph of EVWIIs against the number of data points (CLRM approach).

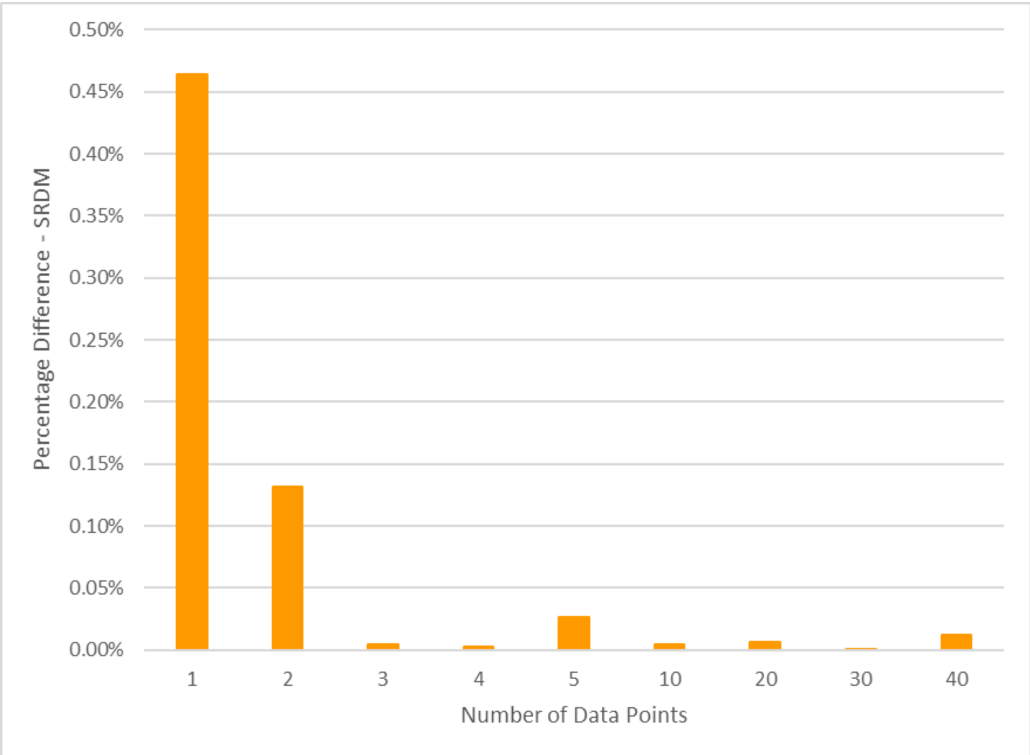


Fig. 43 – Graph of percentage difference against the number of data points (SRDM approach).

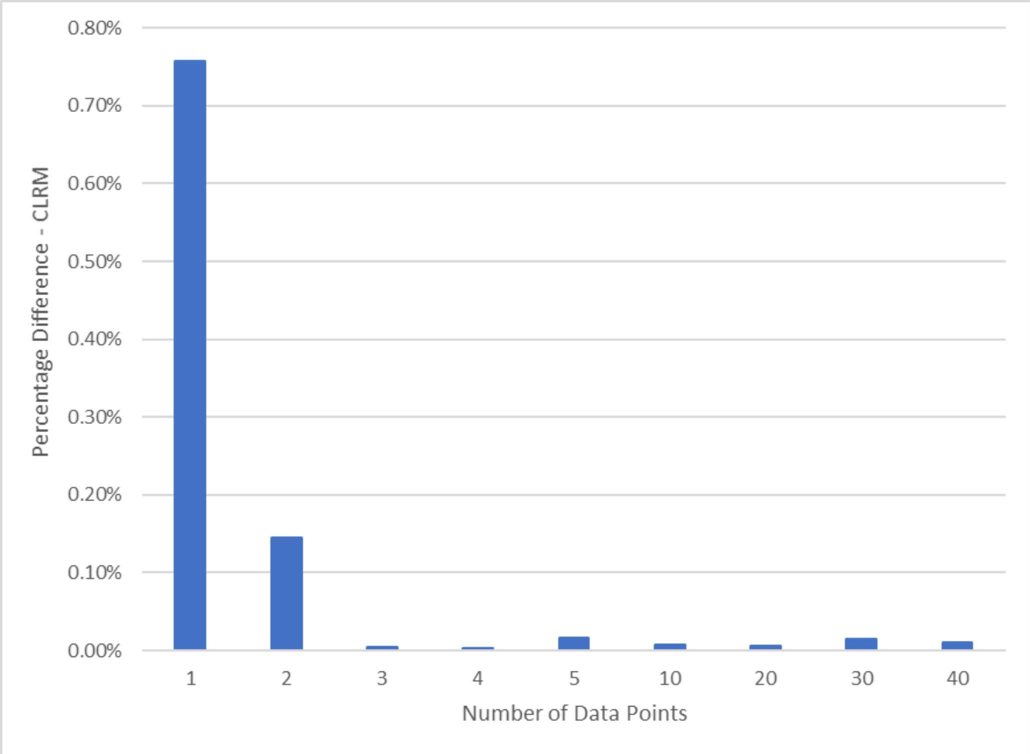


Fig. 44 – Graph of percentage difference against the number of data points (CLRM approach).

From these two figures, modeling the measured oil production rates as Markovian processes would produce the EVWII of SRDM that differs from that of the base case by more than 0.45%. For the case of CLRM, the percentage difference is more than 0.70%. In general, this sensitivity analysis demonstrates that for this IOR initiation problem, using less data point is practically sufficient to provide a good estimation of EVWII. This might be because of the simplicity of the two-factor model. Due to the decline feature of this production model, using fewer data points is enough to determine the trend of decline. Besides that, in this aspect, it is recommended not to model the measured oil rates as Markovian processes as it produces the result that has the highest percentage difference. In addition to this, using 2 data points also generates a relatively high percentage difference. Therefore, from this sensitivity analysis, it can be deduced that using at least 3 data points is recommended for practical purposes in this case.

5.3 SD of Measurement Error

Another sensitivity analysis is performed on the SD of measurement error. Since the mean of the measurement error is assumed to be zero in the problem setting of Hong et al. (2018), increasing (decreasing) the SD of measurement error denotes that more samples of measurement error further from (closer to) the mean are retrieved. In other words, the SD of measurement error is related to the accuracy of the measurement. Regarding this, the values of the samples of measurement error would generally increase (decrease) as the SD increases (decreases). This sensitivity analysis is important because as the SD of measurement error is decreasing, the corresponding VOI would increase and gradually become constant. This indicates that collecting more accurate data do not induce any additional value to the VOI. In general, it is more expensive to collect more accurate data or information because device with higher accuracy is usually cost-consuming. In this context, it is possible that the cost of acquiring a device with small measurement errors exceeds the VOI. Therefore, this sensitivity analysis provides an insight for the decision maker to assess if it is worthwhile to proceed in the information gathering activity that would result in a more accurate information.

5.3.1 Results and Discussions

To do the sensitivity analysis, the SDs of measurement errors of 5%, 10%, 15%, 20%, 25%, and 30% are used. The same problem setting as listed in Table 2, Table 3, and Table 4 has been used to do the sensitivity analysis. Additionally, 25000 paths are used to sample the corresponding variables. In this context, the sampling is only done once. With respect to this, EVWIIs (considering both SRDM and CLRM) and EVWOIs corresponding to each measurement error are estimated. Then, the respective VOIs are determined. The graph of VOI against the SD of measurement error is shown in Fig. 45 for SRDM and the similar case of CLRM is illustrated in Fig. 46.

Based on both figures, as the SD of measurement error decreases, the corresponding VOI increases. This demonstrates that as more accurate data is gathered, this would induce additional value to the VOI. In the case of SRDM, the respective VOI is $53.58 million as the SD of measurement error is 5% whereas that of CLRM is $53.24 million. This means that if a device, which can measure the oil production rates with SD of measurement error of 5%, costs more than these VOIs, it is not worthwhile to acquire this device

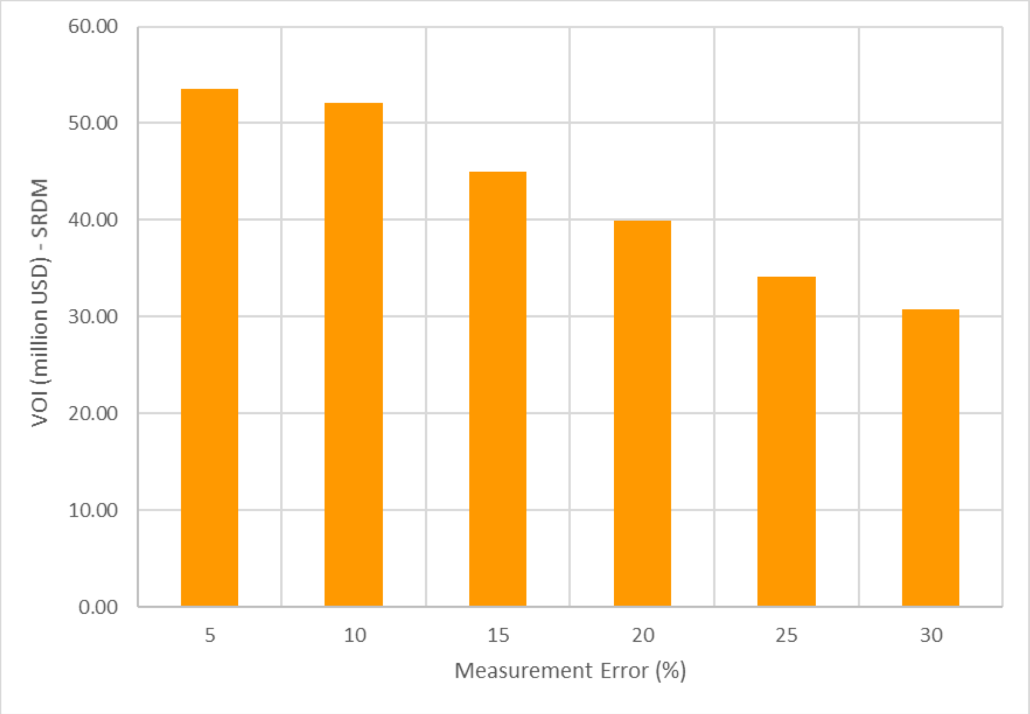


Fig. 45 – Graph of VOIs against the SD of Measurement Error (%) (SRDM approach).

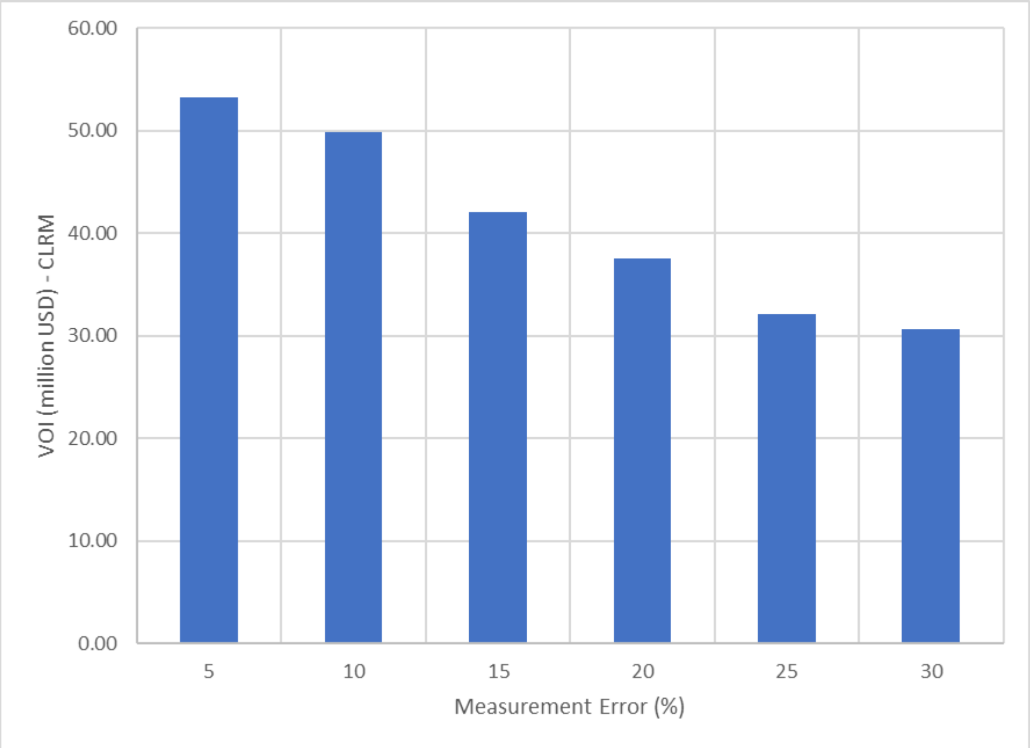


Fig. 46 – Graph of VOIs against the SD of Measurement Error (%) (CLRM approach).

Chapter 6

Integration of Economic Uncertainties

One of the shortcomings of the HBL's model is that it does not model the uncertainties in the economic parameters, which have a significant impact on the decisions as shown in the sensitivity analysis in the previous chapter. Therefore, in this work, economic uncertainties are included in HBL's model. In this aspect, it is essential to note that not all the economic parameters displayed in Table 4 are treated as uncertain. This is because some of these economic parameters are not material as including them would not affect the decision that has been otherwise made. With respect to this, the economic uncertainties included in HBL's model are oil price, OPEXs of both recovery phases, and CAPEX_2After1.

In both CLRM and SRDM approaches, we know that the optimal stopping time given a switch time must be determined. In the context of finding the optimal stopping time given a switch time, there is an underlying fact that we are already in the secondary recovery phase. Thus, in the regression analysis for both CLRM and SRDM approaches, the economic uncertainties considered are oil price and OPEX of secondary recovery because both are material in this situation. Besides that, as we are determining the optimal switch time (with its respective optimal stopping time), we compare the ENPV of "switch option" and that of "continuation option". To enable this comparison to be done on the same basis, the economic uncertainties are oil price, OPEXs of both recovery phases, and CAPEX_2After1. As economic uncertainties are integrated, regression analysis is done by adding these economic parameters (as additional terms) to the regression function. With respect to this, these economic variables are modeled as Markovian processes.

6.1 Economic Parameters as Stochastic Processes

Since the abovementioned economic parameters are treated as uncertain outcomes, they need to be modeled to be variant over time. Hence, these parameters are assumed to follow a stochastic process. There are two commonly used stochastic models to describe the uncertainties in the economic variables, which are the Geometric Brownian Motion (GBM and known as random-walk model) and the Ornstein-Uhlenbeck (OU) Stochastic Process (also known as mean-reverting model and refer to Uhlenbeck and Ornstein (1930) for more details).

Regarding the use of GBM in the pricing of option, it was initially applied to model the stock price in the Black-Scholes model. This model was proposed by Fischer Black and Myron Scholes in 1973 in the context of valuing the European options (Black and Scholes, 1973). Since then, the GBM process has been widely used in the modeling of the prices. Copeland and Antikarov (2001), Brandão et al. (2005), and Smith (2005) discussed the use of GBM to model the prices for the analysis of real-option valuation.

Dias (2004) provided a brief explanation pertaining to the general mechanism of GBM. He used an example of the value of developed field to demonstrate the modeling of this value with the use of GBM. With respect to this, the process of GBM is illustrated in Fig. 47. Basically, in GBM, an uncertain variable at a future time has a lognormal distribution with variance that increases proportionally to time and drift that increases (or reduces) exponentially (Dias, 2004). Thus, the EV would grow exponentially over time (Dias, 2004).

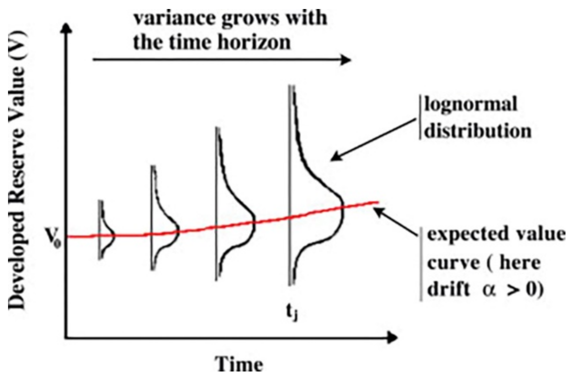


Fig. 47 – Illustration of the GBM process in terms of Developed Field Value[45] (Dias, 2004).

From this illustration, it can be observed that the GBM does not reflect the true behavior of a commodity price. This behavior can be expounded by understanding the microeconomic theory of supply and demand. Based on this theory, there is always an equilibrium price (or can be perceived as long-term mean price) for any product in the market (Dias, 2004; Jafarizadeh and Bratvold, 2012). Therefore, when the price of a product increases and becomes higher than its equilibrium price, the supply of the product would increase. This is because the producers of the product have additional incentives to increase the production in the market (Jafarizadeh and Bratvold, 2012). With this, the price of the product would gradually be driven down to the equilibrium price. By using the same logic, when the price of the product decreases and becomes lower than the equilibrium level, the supply would reduce, and the price would slowly be driven up to the equilibrium level. This behavior of price adjustment is described by the OU stochastic process. In general, the process can be illustrated in Fig. 48 by using oil price as an example (Dias, 2004).

[45] Reprinted from Journal of Petroleum Science and Engineering, Volume 44/ Issues 1-2, Dias, M.A.G, Valuation of exploration and production assets: an overview of real options models / Stochastic processes for oil prices, 102, Copyright 2004, with permission from Elsevier.

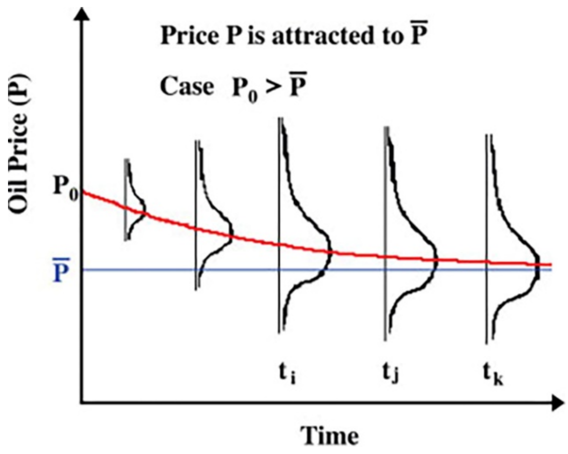


Fig. 48 – Illustration of mean-reverting process in terms of oil price[46] (Dias, 2004).

Based on Fig. 48, when the initial oil price, P_o is higher than the equilibrium oil price, $\overline{P}$ the oil price would eventually decrease until it reaches $\overline{P}$. This is the manifestation of the theory of supply and demand as expounded earlier. In addition to that, it can also be noticed that the mean-reverting model also uses the lognormal distribution, but the variance of the distribution would increase until a period, t_i and remain constant. As Dias (2004) has explained, it is due to the effect of the mean-reversion force that does not allow the values of oil price to be much further from $\overline{P}$.

Apart from this, Pindyck (1999) did a thorough analysis and demonstrated that the data of oil price over 127 years portrayed the behavior of mean-reverting. In addition to this, the mean-reverting model has been regularly used to model the prices in many studies related to petroleum industry (Dixit and Pindyck, 1994; Smith and McCardle, 1999; Dias, 2004; Begg and Smit, 2007; Willigers and Bratvold, 2009). Therefore, for illustrative purpose, this stochastic process is chosen to model the oil price, OPEXs, and CAPEX_2After1 in this extended HBL's model.

[46] Reprinted from Journal of Petroleum Science and Engineering, Volume 44/ Issues 1-2, Dias, M.A.G, Valuation of exploration and production assets: an overview of real options models / Stochastic processes for oil prices, 102, Copyright 2004, with permission from Elsevier.

6.2 Implementation of Mean-Reverting Stochastic Process

A process "S" can be stochastically modeled using the Ornstein-Uhlenbeck process as shown below.

$$dS_t = \theta\,(\mu - S_t)dt + \sigma dW_t \qquad (23)$$

where θ is the speed of mean reversion, μ is the long-term mean which the process reverts, σ is the measure of process volatility, and W_t stands for a Brownian motion in which $dW_t \sim N(0,\sqrt{dt})$. In order to implement this stochastic equation in simulation, it has to be discretized. Gillespie (1996) has opined that only when the discretized time, Δt is sufficiently small, the simulation of the process would work well. Thus, the discretized equation is shown below.

$$S_t = \left(S_{t-1} \times e^{-\theta\Delta t} \right) + \mu\left(1 - e^{-\theta\Delta t}\right) + \left[\sigma \times \sqrt{\frac{1 - e^{-2\theta\Delta t}}{2\theta}} \times dW_t\right] \qquad (24)$$

However, if any commodity price, including oil prices or any cost, is modeled using the above discrete-time expression, negative values might be generated. This is not realistic because negative commodity price never exists. To avoid this problem, the lognormal distribution of the commodity price is used. Thus, in this context, the logarithm of the modeled parameter, namely $\pi_t = \ln[S_t]$, is assumed to follow the mean-reverting process. This process can then be mathematically described as.

$$d\pi_t = \kappa\,[\bar{\pi} - \pi_t]\,dt + \sigma_\pi dz_t \qquad (25)$$

where κ is the speed of mean reversion, $\bar{\pi}$ is the long-term mean that the logarithm of the variable reverts, σ_π stands for the volatility of process, and dz_t describes the increments of the standard Brownian motion. After that, to numerically solve for π_t, the stochastic equation is discretized as shown (by assuming $dz_t \sim N(0,\sqrt{dt})$ in which dt = 1 year).

$$\pi_t = [\pi_{t-1} \times e^{-\kappa\Delta t}] + [\bar{\pi} \times (1 - e^{-\kappa\Delta t})] + \left(\sigma_\pi \times \sqrt{\frac{1 - e^{-2\kappa\Delta t}}{2\kappa}} \times N(0,1)\right) \qquad (26)$$

After calculating π_t, the value of S_t cannot directly be obtained by using the equation of $S_t = e^{\pi_t}$. This is because the mean of the lognormal distribution is added with half of the variance, namely $0.5 \times \mathrm{Var}(\pi_t)$, for the exponential of a normal distribution (Fu et al. 2001). Therefore, the half of the variance is deducted by using the equation below.

$$S_t = e^{\pi_t - [0.5 \times \mathrm{Var}(\pi_t)]} \qquad (27)$$

The variance of the π_t is described as

$$\mathrm{Var}(\pi_t) = [1 - e^{-2\kappa\Delta t}] \times \frac{{\sigma_\pi}^2}{2\kappa} \quad (28)$$

With respect to these, the oil prices are modeled forward in time. Besides that, the modeling of OPEXs of both recovery phases and the CAPEX_2After1 are not that straightforward. With respect to this, the cost multiplier of each of these costs is modeled instead. In this context of performing this forward modeling, the three essential parameters κ, $\overline{\pi}$ and σ_π (generally also known as OU parameters) used in the mean-reverting model need to be determined. These values could be estimated by using least-squares regression as proposed by Smith (2010). This procedure is termed as the calibration of the OU parameters and it is explained in the next section.

6.3 Calibration of the OU Parameters

Since the logarithm of the variables is assumed to follow the mean-reverting process, the least-squares regression, which is suggested by Smith (2010), is conducted on the datasets of $\pi_t = \ln[S_t]$. To calibrate the OU parameters for the modeling of the oil price, a set of oil price data is needed. In this context, there is no so-called "correct" set of data to be used in the calibration. It depends on the preference of a company or individuals if historical data, future data or the combination of both should be used to calibrate these parameters to model the stochastic process. For the purpose of illustration, the annual oil price data, namely NYMEX future prices[47] from 1985 to 2017 (considering only historical data), which is available in the website of U.S. Energy Information Administration (2019), is used.

To begin the procedure of calibration, the following equations are used.

$$x_t = \pi_{t-\Delta t} = \ln[P_{t-\Delta t}] \quad (29)$$

$$y_t = \pi_t = \ln[P_t] \quad (30)$$

Then, the set of y_t data is regressed on the set of x_t data and the following equation is produced

$$y_t = ax_t + b + \delta \quad (31)$$

By using the values of a and b, the OU parameters are estimated.

$$\overline{\pi} = \frac{b}{1 - a} \quad (32)$$

[47] In market, crude oil is traded through two types of contract, namely spot contract and future contract. Spot price is the price corresponding to spot contract in which the crude oil is delivered instantly (Cardenas, 2017). Future price is the price corresponding to future contract in which the crude is delivered within a predetermined time at a specific oil price (Cardenas, 2017).

$$\kappa = \frac{-\ln a}{\Delta t} \tag{33}$$

$$\sigma_\pi = \sigma_\delta \sqrt{\frac{-2\ln a}{\Delta t\,(1 - a^2)}} \tag{34}$$

where δ is the approximation error induced in the least-squares regression, σ_δ stands for the standard deviation of the approximation errors, and Δt is the difference in two time-steps. Refer to Smith (2010) for the derivation of the Equations (32), (33), and (34). Aside from calibrating the oil prices, the similar procedure is supposed to be done on the relevant costs. However, the sets of cost data are not accessible in public domain. Thus, for illustrative purpose, the OU parameters of cost multiplier used here are from Willigers (2009) and it applies to both OPEXs and CAPEX_2After1. Parameters used to simulate the mean-reverting stochastic processes are listed in in Table 10[48].

Parameter	Oil Price	Cost Multiplier
Initial Value	47.48 USD/bbl	1
Equilibrium Value	47.48 USD/bbl	1
Volatility, σ_π	0.2473	0.5
Mean reversion speed, κ	0.0643	0.1
dt, year	1	1

Table 10 – Values of parameters used in the mean-reverting model.

By using the parameters in Table 10, the oil price and the cost multipliers corresponding to the respective costs are modeled forward in time. Fig. 49 shows the probabilistic model of the oil price. With respect to this, in order to correlate these economic parameters with each other, the correlation coefficients between these parameters need to be established. The details of this correlation coefficient follow later.

In general, by having these correlation coefficients, the multi-variate normal distribution (from where 4 different sets of random samples can be retrieved) is developed by assuming the means to zero and the SDs to be 1. These sets of random samples are respectively used as the increments of the standard Brownian motion as shown in Equation (26). Thereafter, the logarithmic values of the oil price and cost multipliers are computed. Then, by applying Equations (27) and (28), the corresponding values of oil price and cost multipliers can be calculated. After generating the samples for oil price and the cost multipliers, the values of the costs need to be computed. In this case, a shift of a year is included in which the oil price at Year n is multiplied with the cost multiplier at Year n+1 to yield the cost at Year n+1.

[48] The initial value is assumed to be the same as equilibrium value for illustrative purpose as well.

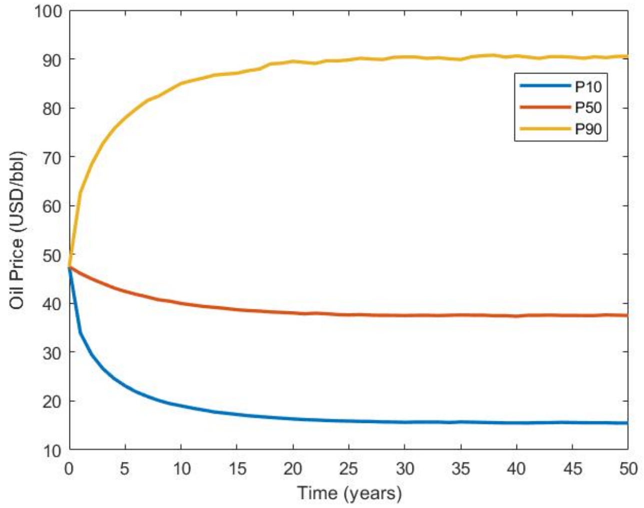


Fig. 49 – Oil prices modeled with 100000 paths using the mean-reverting process.

6.4 Correlation between the Uncertain Economic Parameters

The oil prices and the costs can be modeled using the mean-reverting stochastic process either independently or through correlation. To model the oil prices and the costs through correlation, the correlation coefficients between these economic state variables are established and used to compute the co-variance that would be used to generate a multi-variate normal distribution (as briefly explained in previous section). By doing so, it is ensuring that random samples of economic variables retrieved from this distribution are correlated with each other.

Devore (2010) stated that there were two types of correlation, namely positive and negative correlations. In this case, when variables A are positively correlated to variables B, this means that as variables A increase, variables B also increase. For negative correlation, as variables A increases, variables B would decrease. Besides that, regarding the strength of the correlation, Devore (2010) provided a rule of thumb as shown in Table 11 in which the correlation coefficient is represented by ψ.

Strength	Description
Weak	$-0.5 \leq \psi \leq 0.5$
Moderate	$-0.8 < \psi < -0.5$ $0.8 < \psi < 0.5$
Strong	$\psi \geq -0.8$ $\psi \geq 0.8$

Table 11 – Rule of Thumb regarding the strength of correlation (Devore, 2010).

Based on Table 11, when the correlation coefficient between two different variables is 0.85, it means that there is a strong positive correlation between these two variables. The square of the correlation coefficient, ψ^2 indicates the accuracy of the relationship between two different variables. Consult Devore (2010) for a more comprehensive explanation.

Williggers (2009) did a thorough analysis showing that there was a strong correlation between oil prices and rig rental rates. By offsetting oil price by 1 year with respect to rig rental rates, the correlation coefficient was 0.87 (Willigers, 2009). Thus, the offset of a year established a strong positive correlation between oil prices and rig rental rates. In this case, if we want to find out the correlation coefficient between oil prices and OPEX of primary recovery, the sets of oil price data and OPEX of primary recovery are required. However, the data of OPEX of any kind of project in petroleum industry is not easily available in public domain. Based on the analysis done by Willigers (2009) and for illustrative purpose, a strong positive correlation is assumed between oil prices and OPEX of both recovery phases in this extended HBL's model.

Pertaining to the correlation between oil price and CAPEX, Iqbal and Shetty (2018) explained that an oil and gas firm would increase (reduce) CAPEX when the oil price increases (decreases). Thus, a positive correlation is assumed between these 2 parameters in this work. Additionally, Surovtsev and Sungorov (2016) stated that there was a rule of thumb[49] used to quickly assess the annual OPEX. Based on this rule of thumb, the annual OPEX is assumed to be 4% of total CAPEX. According to this, a positive correlation is thus assumed between OPEX and CAPEX_2After1. Regarding the correlation between OPEXs of both recovery phases, there are not many related literatures found. Hence, for the purpose of illustration, both OPEXs are assumed to positively correlate to each other. The correlation coefficients[50] between the economic parameters used in the mean-reverting stochastic model are tabulated as shown below.

	Oil Price	**OPEX (Primary)**	**OPEX (Secondary)**	**CAPEX_2After1**
Oil Price	1.00	0.87	0.87	0.85
OPEX (Primary)	0.87	1.00	0.55	0.65
OPEX (Secondary)	0.87	0.55	1.00	0.70
CAPEX_2After1	0.85	0.65	0.70	1.00

Table 12 – Correlation coefficients between the economic parameters.

[49] Surovtsev and Sungorov (2016) mentioned that there were no substantiated works done to support this rule of thumb. However, the correlation established between OPEX and CAPEX by this rule of thumb is used in this work for the purpose of illustration.

[50] The economic parameters are assumed to be positively correlated to each other based on some literature reviews. Nonetheless, not many relevant literatures discuss the strength of the correlation among the economic parameters. In fact, the strength of correlation can be determined by using the dataset which are not accessible in public domain. For illustrative purpose, the correlation coefficients in Table 12 are built based on assumption.

6.5 Realistic Example Solved by the Extended HBL's model

A realistic example is solved by the extended HBL's model to demonstrate its valid implementation. Basically, the problem setting of this example is similar to that of in Hong et al. (2018). However, the only change made is by having the mean of E_{R1}^{∞} to be 0.24 instead of having it to be 0.20. The other parameters remain unchanged as listed in Table 2, Table 3, Table 10, and Table 12. Discount rate, CAPEX of the primary recovery, and CAPEX_2No1 remain unchanged as listed in Table 4. 100000 paths are then used in solving this decision problem.

6.5.1 Results and Discussions

The DWOI is to have the primary recovery for 4 years and be followed by the secondary recovery for 8 years. This results in the total lifetime of 12 years. Thus, the EVWOI is found out to be $518.97 million. Besides that, the EVWPI is estimated to be $757.53 million. This leads to the VOPI to be $238.56 million. The EVWII corresponding to the SRDM approach is $624.14 million. The respective VOI is $105.17 million. This indicates that it is not economical to proceed with any information-gathering activity if the cost of the activity is more than $105.17 million. Moreover, this result also illustrates that including the effect of future information and decisions would improve the EV by 20.26%, which is the percentage of the fraction of VOI to EVWOI.

The Cumulative Distribution Functions (CDFs) of NPVs associated with DWOI, DWII, and DWPI are plotted in Fig. 50. From this figure, the DWII moves the CDF of NPV corresponding to DWOI to the right. In this aspect, integrating the effect of future information and decisions in decision making would increase the ENPV. In addition to this, DWPI moves the curve of CDF even further to the right as shown. This is because the NPVs corresponding to DWPI are always higher and this would lead to higher ENPV than those of DWIIs and DWOI.

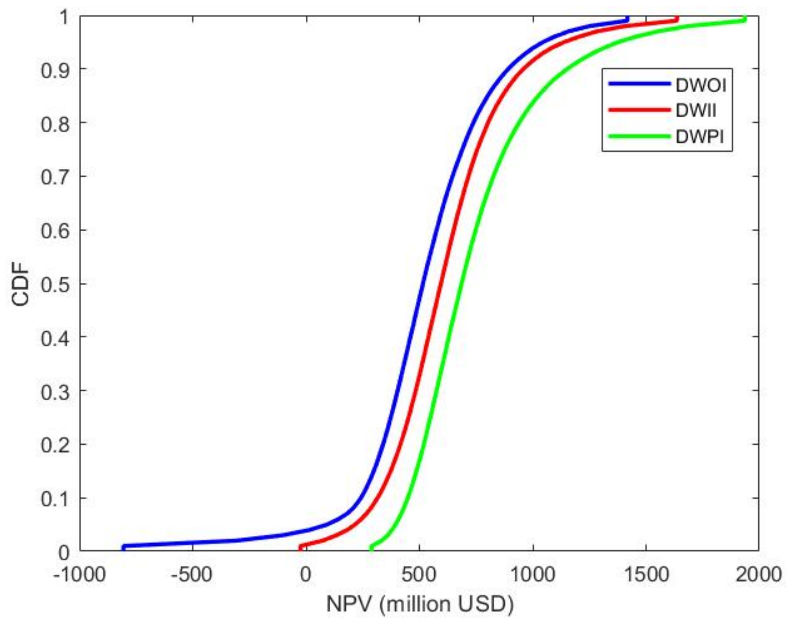


Fig. 50 – Graph of CDFs against NPVs with respect to DWOI, DWII, and DWPI.

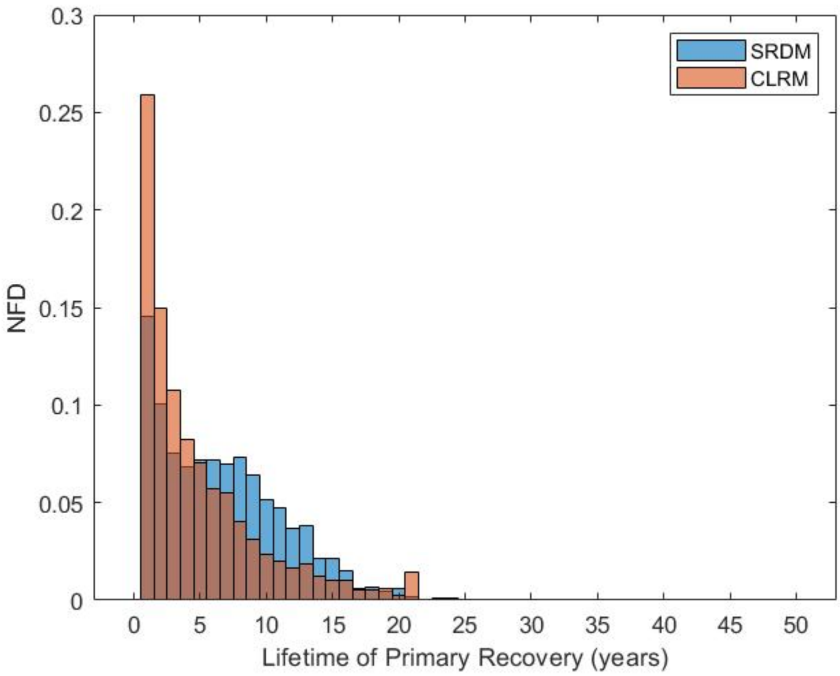
Besides that, the CLRM approach is also extended and used to solve this decision problem. The corresponding EVWII is estimated to be $620.39 million. This yields a special VOI of $101.42 million and an improvement of EV by 19.54%. In this context, SRDM provides an additional value of learning of $3.75 million as compared to CLRM. To demonstrate the comparison between the DWII corresponding to SRDM and that of CLRM, the Normalized Frequency Distributions (NFDs) and the Normalized Cumulative Frequency Distributions (NCFDs) of the lifetime of primary recovery, the lifetime of secondary recovery, and the total lifetime are respectively illustrated in Fig. 51, Fig. 52, and Fig. 53. For the color legend shown in these three figures, the light brown indicates the CLRM approach, the light blue color denotes the SRDM approach, and the brown color shows an overlap between these two approaches (overlap between light brown bars and light blue bars).

Fig. 51 illustrates that for the lifetime of primary recovery corresponding to SRDM, the interval between P10 to P90 ranges from 1 to 12 years. This indicates that there is a chance of 80% to switch from primary to secondary recovery between the end of Year 1 and the end of Year 12. For the case of CLRM, there is in fact 80% chance to perform the switch between the end of Year 1 and the end of Year 11. This shows that the lifetime of primary recovery proposed by using CLRM is slightly shorter than that of SRDM. Besides that, SRDM illustrates there is a chance of 14.58% that it is optimal to switch after 1 year of primary recovery. With respect to this, CLRM even suggests a higher frequency, which is 25.92%. SRDM proposes that there is 6.8% chance that it is optimal to switch after 4 years of primary recovery (as suggested by DWOI). For CLRM, there is higher chance, which is 8.2%, So, by applying either SRDM or CLRM, it is recommended to have the facilities for secondary recovery ready at the end of Year 1 to circumvent from losing flexibility to switch before Year 4. In this case, 32.16% of realizations for SRDM and 51.65% of realizations for CLRM propose to switch before Year 4.

Fig. 52 shows that for the lifetime of secondary recovery corresponding to SRDM, the interval between P10 to P90 ranges from 5 to 23 years. This means that SRDM suggest that there is 80% chance that the secondary recovery would last for 5 to 23 years. For CLRM, the interval between P10 to P90 ranges from 3 to 22 years. The 8 years of secondary recovery as indicated by DWOI only happens in 4.60% of the realizations for SRDM and 5.41% of the realizations for CLRM. For SRDM, the most frequent lifetime of secondary recovery is 12 years, which makes up 5.86% of the realizations. For CLRM, the most frequent lifetime of secondary recovery is 9 years, which also has 5.86% of the realizations. Moreover, for 20.38% of realizations in SRDM, the lifetime of secondary recovery is shorter than 8 years. For CLRM, there are 30.29% of the realizations showing similar result.

Fig. 53 demonstrates that for the total lifetime corresponding to SRDM, the interval between P10 to P90 ranges from 10 to 30 years. For CLRM, the P10 to P90 interval of total lifetime is between 7 to 29 years. Therefore, if SRDM (CLRM) is applied, it is recommended to stop the production anytime from the end of Year 10 (Year 7) to Year 30 (Year 29). In this aspect, this also implies that the production license must last for at least 30 years (29 years). With respect to this, if the production is terminated at the end of Year 12 (as suggested by DWOI), both

SRDM and CLRM suggest that there is certainly a chance of more than 50% of losing the opportunity to have longer production.



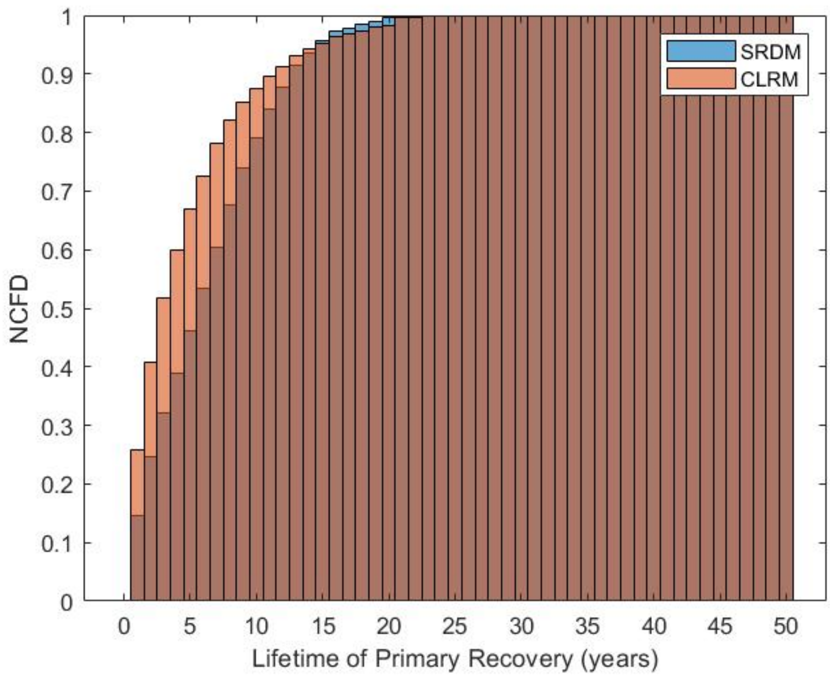


Fig. 51 – Plots of NFD and NCFD of the lifetime of primary recovery with respect to DWIIs solved using CLRM and SRDM.

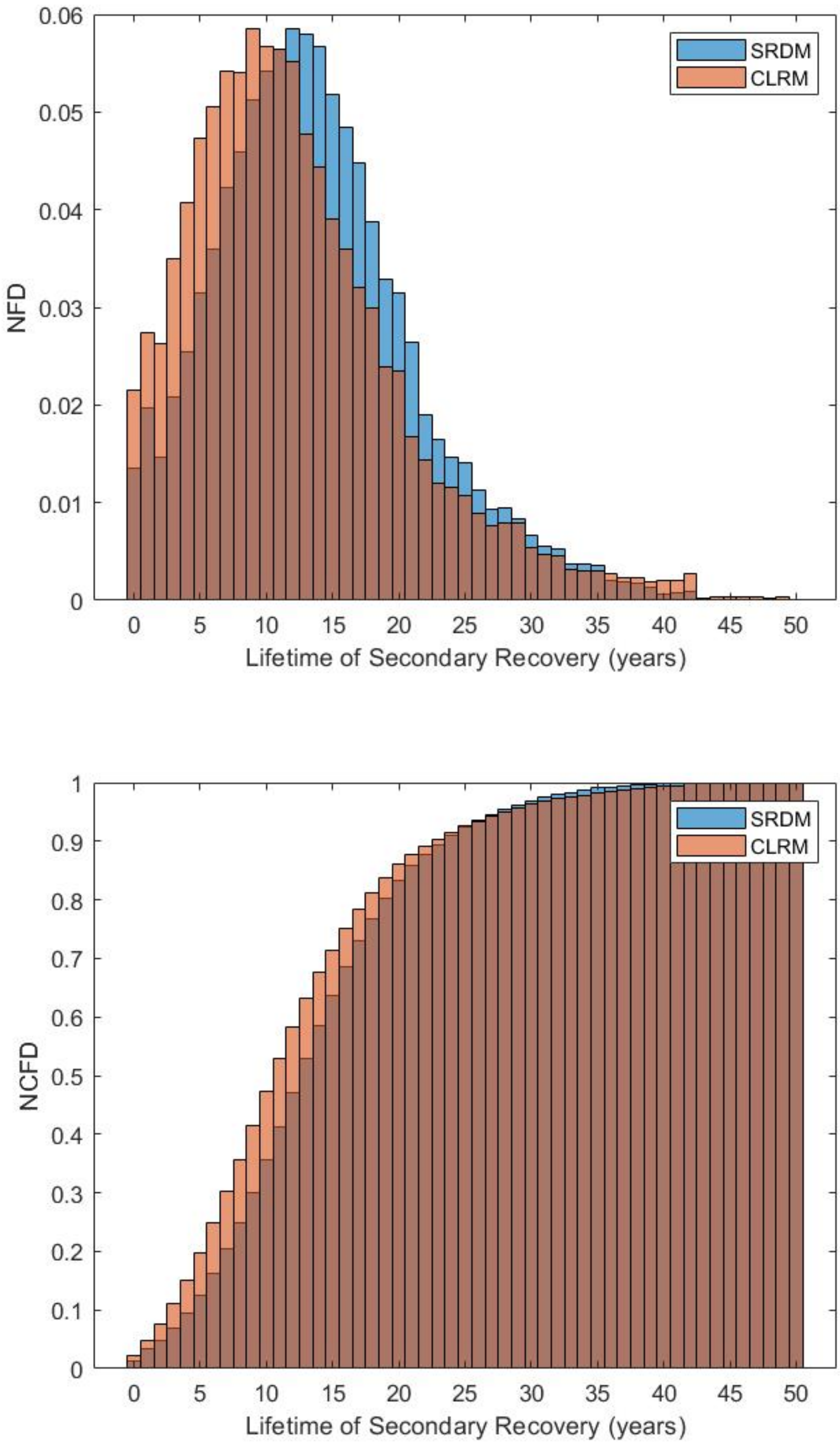


Fig. 52 – Plots of NFD and NCFD of the lifetime of secondary recovery with respect to DWIIs solved using CLRM and SRDM.

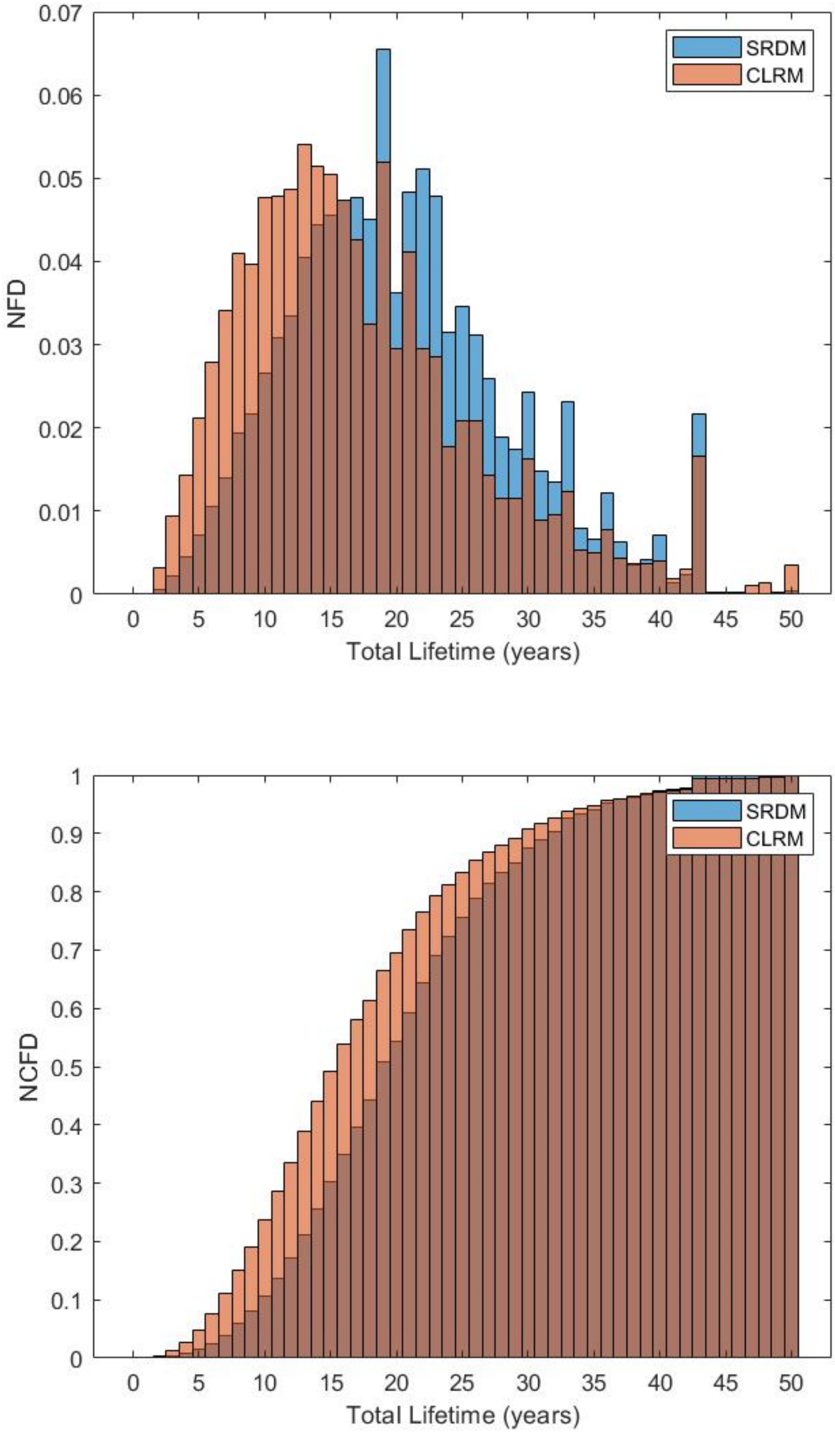


Fig. 53 – Plots of NFD and NCFD of the total lifetime with respect to DWIIs solved using CLRM and SRDM.

Apart from this, the frequencies of different combinations of the lifetime of primary recovery and that of secondary recovery with respect to DWII are correspondingly illustrated in Fig. 54 for SRDM and in Fig. 55 for CLRM. For SRDM, the combinations with higher frequency are to have primary recovery for 1 year and secondary recovery for 7 to 21 years. Besides that, for CLRM, the combinations with higher frequency are to have primary recovery for 1 year and secondary recovery for 4 to 15 years. These results are consistent with the use of the values of means for τ_1 and τ_2. The mean of τ_1 is 16 years whereas that of τ_2 is 7 years and this indicates that oil is recovered faster in secondary recovery (Hong et. al, 2018). Knowing the cashflow is larger at early time (indicating higher NPV), it is thus better to switch from primary recovery to secondary recovery earlier.

In SRDM, it can be noticed that in general, for highly frequent combinations (with a frequency more than 0.45%), as the lifetime of primary recovery increases, the frequencies of having longer lifetime of secondary recovery reduce. This denotes that as the primary recovery phase continues for a longer time, it would be better to continue with it for some time to gather more data to substantiate the decision on when it is best to perform the switch. Once the switch is done, the corresponding lifetime of secondary recovery would be shortened to yield a higher NPV. The same observation is made in the case of CLRM. Additionally, it can be noted that the lifetime of primary recovery of highly frequent combinations corresponding to CLRM is generally shorter than that of SRDM. Thus, this leads to the total lifetime of highly frequent combinations with respect to SRDM is generally more than that of CLRM.

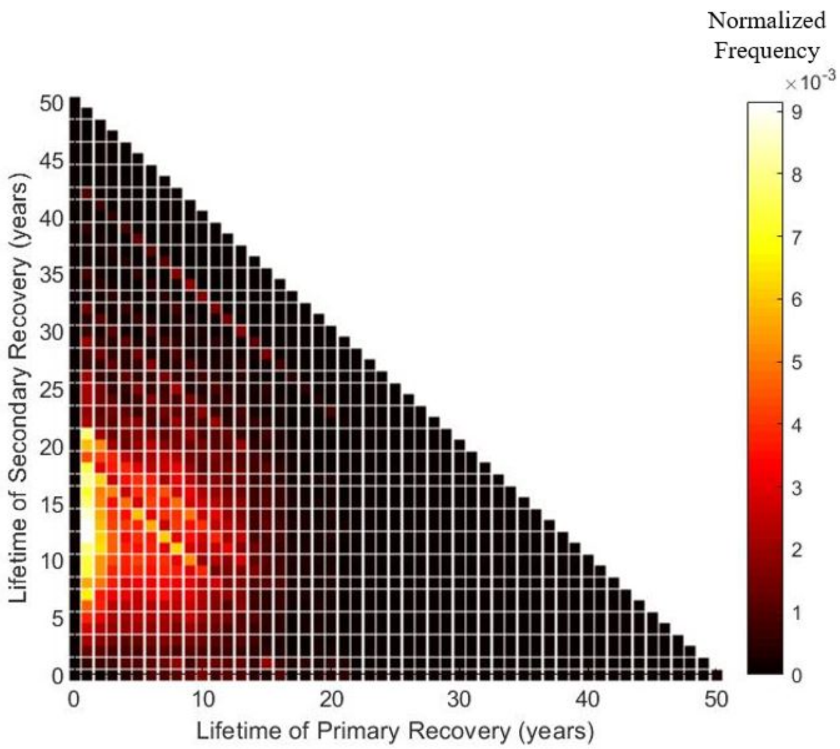


Fig. 54 – Frequencies for different combinations of lifetime of primary recovery and that of secondary recovery corresponding to SRDM.

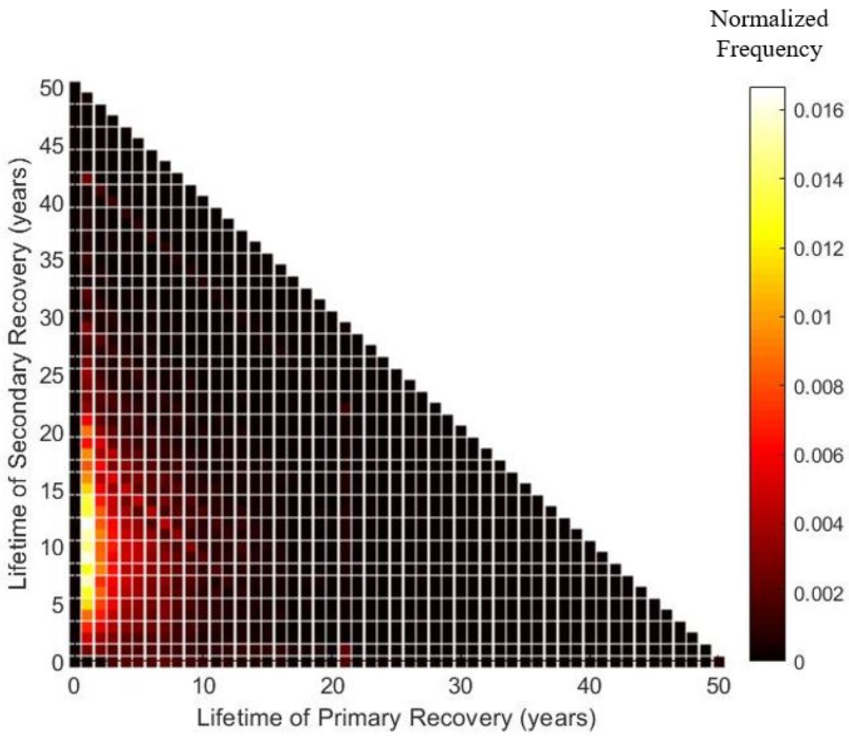


Fig. 55 – Frequencies for different combinations of lifetime of primary recovery and that of secondary recovery corresponding to CLRM.

Fig. 56 shows the mean oil production rates with respect to DWOI, DWIIs corresponding to both SRDM and CLRM. As noted, the mean oil production rate increases significantly after Year 4, decreases between Year 4 and Year 12, and reduces further sharply after Year 12. This is because for DWOI, the decisions to switch after 4 years of primary recovery and to terminate production after 12 years of production (including 8 years of secondary recovery) apply to all realizations. Thus, a drastic change of the mean oil production rate is observed throughout the lifetime of production. However, the mean oil production rates corresponding to both SRDM and CLRM are different as every realization has its corresponding switch time. Thus, the curves of the mean oil production rate with respect to SRDM and CLRM are smoother. In general, the mean oil production rate corresponding to DWOI is higher than those of SRDM and CLRM in the intermediate period. Besides that, the mean oil production rate of CLRM is slightly higher than that of SRDM at the early time, but it eventually becomes lower at later time.

To study the effect of the mean oil production rate on the economical aspect, the graph of the mean Cumulative Discounted Cashflow (CDCF) corresponding to DWOI and DWIIs (considering both SRDM and CLRM) is plotted in Fig. 57. The figure shows that the mean CDFC of DWOI is lower than those of SRDM and CLRM. However, at the early period, the CDFC corresponding to DWII of CLRM is higher than that of SRDM until Year 17 and is lower afterwards. In this context, the overall mean CDCF of CLRM is lower than that of SRDM.

As Hong et al. (2018) explained, this occurred because SRDM might omit some instant gains in the early times to collect more data as these additional data would enhance the decision making later.

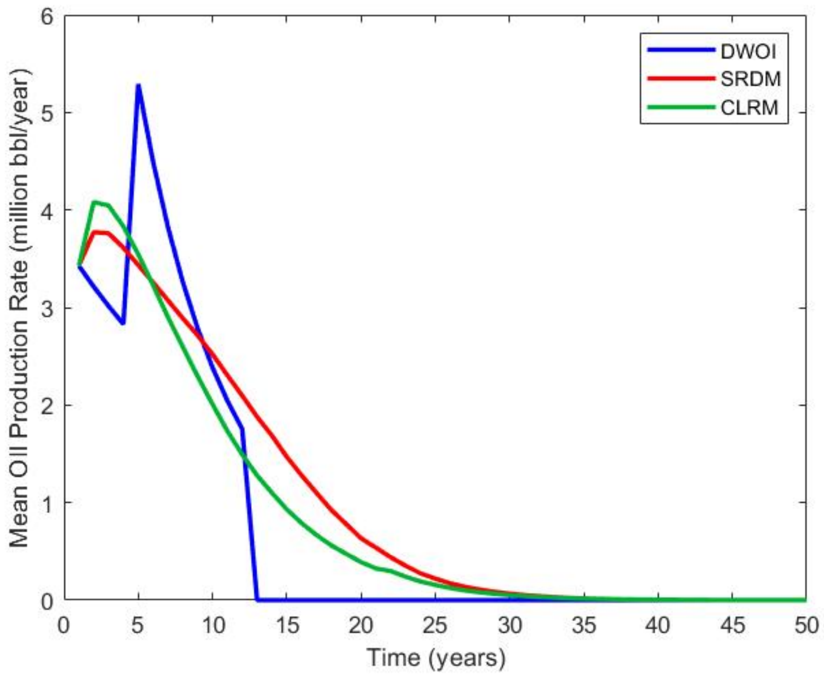


Fig. 56 – Graph of mean oil production rate against time with respect to DWOI and DWIIs corresponding to both CLRM and SRDM.

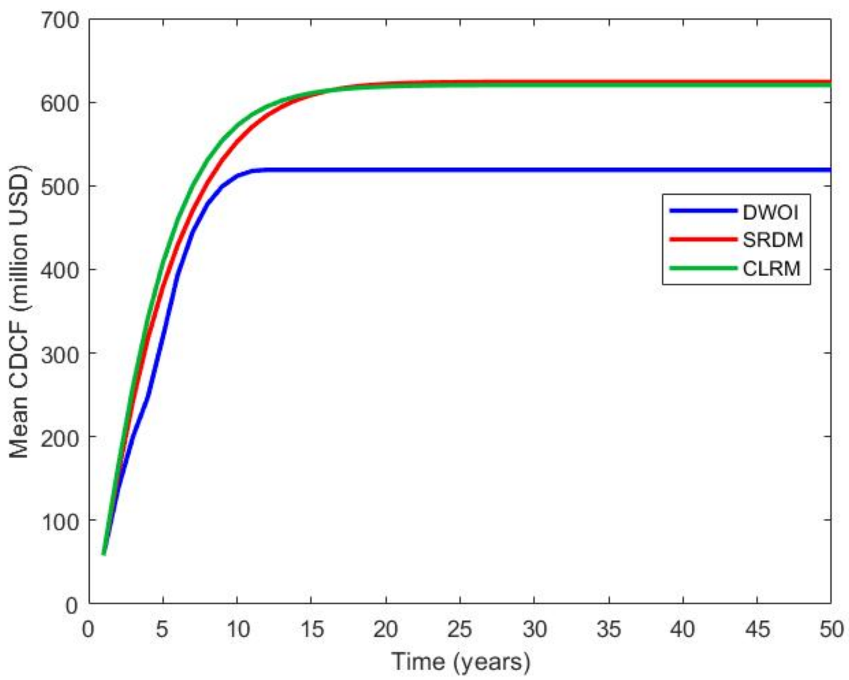


Fig. 57 – Graph of mean cumulative discounted cashflow (CDCF) against time with respect to DWOI and DWIIs corresponding to both CLRM and SRDM.

6.6 Example with Significant Value of Learning

As it has been discussed, the additional value of learning induced by SRDM is the difference between VOI and the special VOI of CLRM. With respect to this, the SRDM approach is absolutely a better approach to be used as compared to CLRM when the value of learning induced is significant. A new problem setting is used to demonstrate a much higher value of learning caused by the SRDM approach. In this context, the problem setting is the same as listed in Table 2, Table 3, Table 10, and Table 12. Discount rate, CAPEX of the primary recovery, and CAPEX_2No1 remain unchanged as listed in Table 4. However, the only change made is by making the SD of E_{R1}^{∞} to be 0.06 instead of being 0.05.

6.6.1 Results and Discussions

By using this problem setting in tandem with 100000 paths, the DWOI is having 0 years of primary recovery and then, 8 years of secondary recovery. This leads to the total lifetime of 9 years. Thus, the corresponding EVWOI is \$471.65 million. The EVWII corresponding to the SRDM approach is \$568.85 million. The respective VOI is \$97.20 million. Moreover, this result also illustrates that including the effect of future information and decisions would improve the EV by 20.61%. Then, the EVWII with respect to CLRM is estimated to be \$527.35 million. This yields a special VOI of \$55.71 million and an improvement of EV by 11.81%. Then, SRDM provides an additional value of learning of \$41.49 million as compared to CLRM.

The comparison between the DWIIs corresponding to SRDM with that of CLRM is done. Therefore, the NFDs and the NCFDs of the lifetime of primary recovery, the lifetime of secondary recovery, and the total lifetime corresponding to both SRDM and CLRM are respectively illustrated in Fig. 58, Fig. 59, and Fig. 60. For the color legend shown in these three figures, the light brown indicates the CLRM approach, the light blue color denotes the SRDM approach, and the brown color shows an overlap between these two approaches (overlap between light brown bars and light blue bars).

Fig. 58 shows that for all realizations, the CLRM approach leads to 0 years of primary recovery. However, for SRDM, the P10-P90 interval for the lifetime of primary recovery ranges from 1 to 13 years. Additionally, Fig. 59 shows that for CLRM, P10-P90 interval for the lifetime of secondary recovery is from 4 to 22 years whereas for SRDM, it ranges from 4 to 23 years. Thus, it can be deduced that the total lifetime corresponding to SRDM would be generally longer than that of CLRM. As shown in Fig. 60, SRDM leads to a chance of 80% to have a total lifetime for 10 to 30 years, which is much longer than that of CLRM. Thus, according to these DWIIs, it can be understood that there is a significant difference between the two VOIs because the SRDM decision policy is significantly different from that of CLRM.

Based on this example, SRDM is more worthwhile to be implemented when DWOI tells us to switch from primary recovery to secondary recovery at the beginning of Year 1. This is because when DWOI suggests the lifetime of primary recovery to be 0 years, this indicates that CLRM would also propose the same decision in terms of the lifetime of primary recovery. However, SRDM might suggest otherwise as it includes the impact of future information and decisions. Thus, SRDM is more robust in the identification of the near optimal decision policy because it would consider the possible future data (learning over time). For CLRM, it would omit future learning (by following the DWOI as explained) as featured in this example.

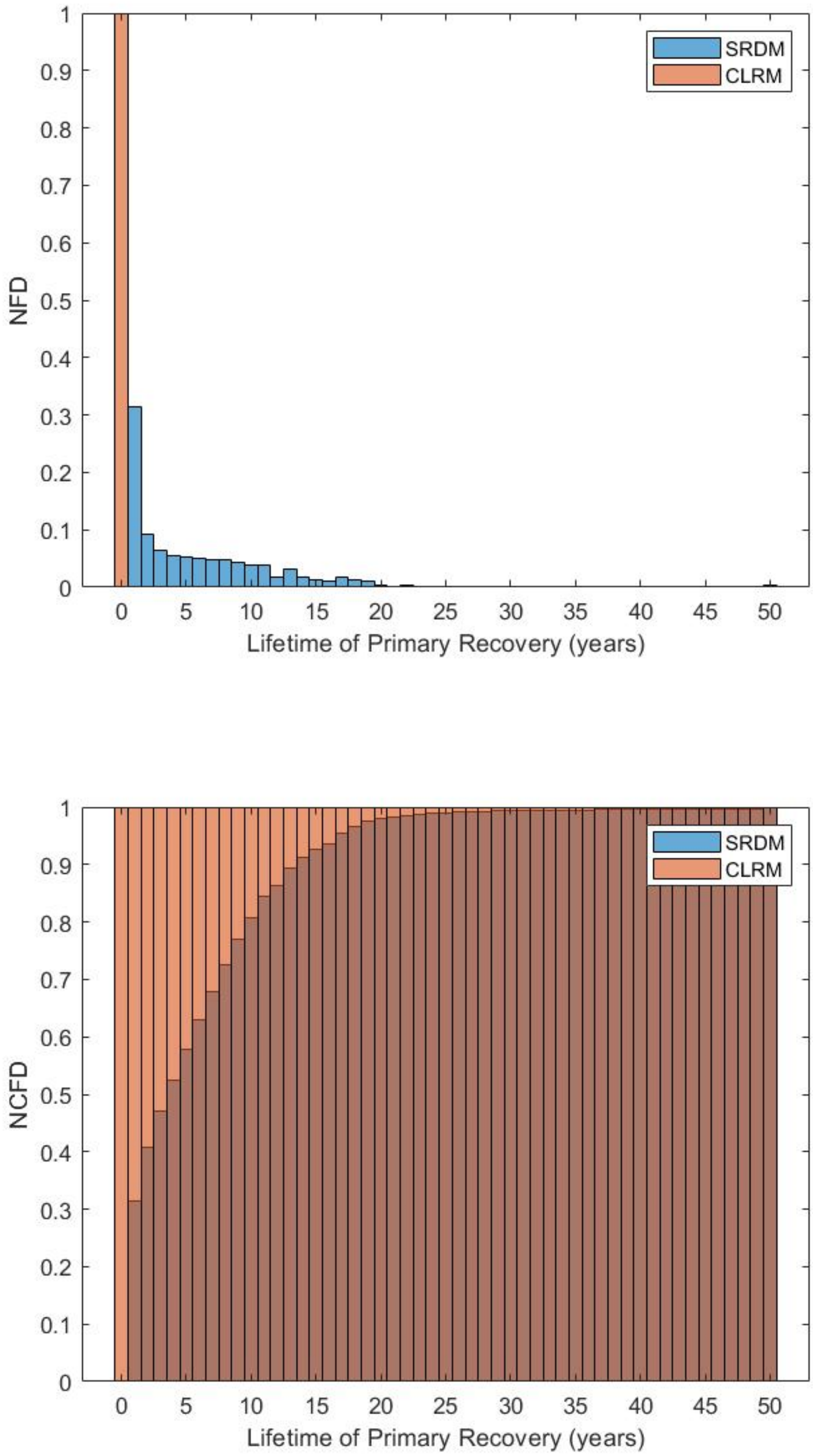


Fig. 58 – Plot of NFD and NCFD of the lifetime of primary recovery with respect to DWIIs solved using CLRM and SRDM.

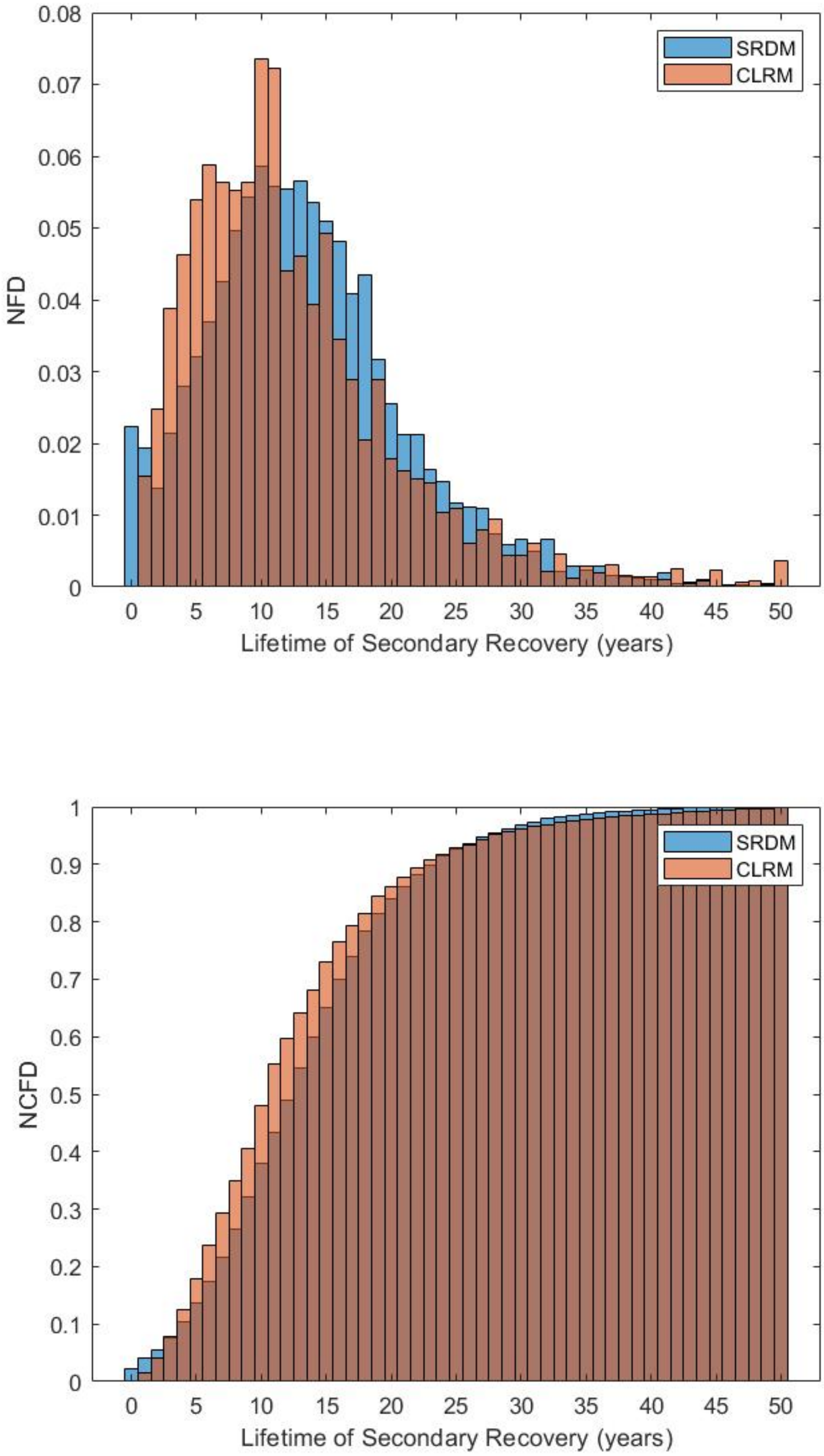


Fig. 59 – Plot of NFD and NCFD of the lifetime of secondary recovery with respect to DWIIs solved using CLRM and SRDM.

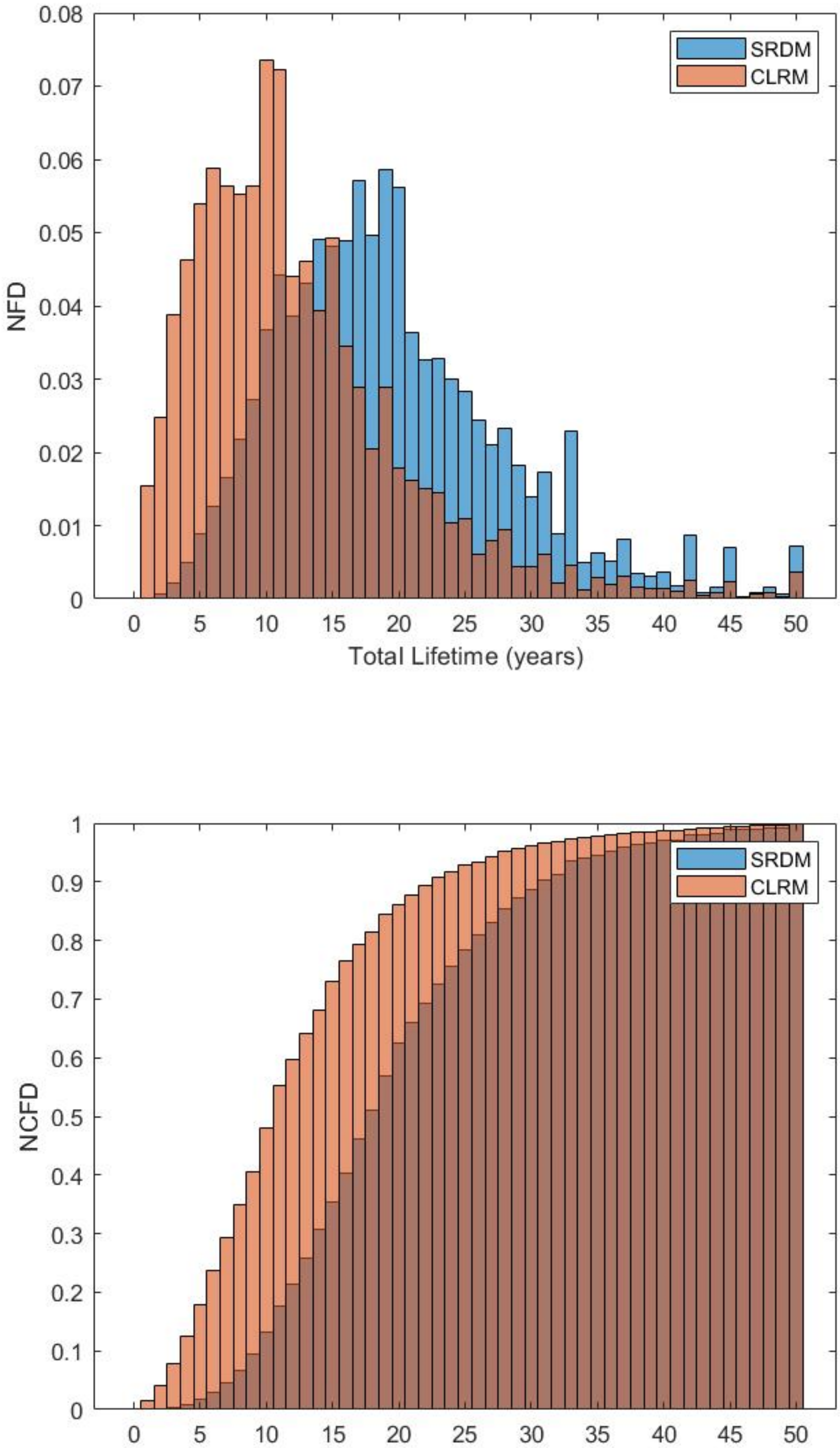


Fig. 60 – Plot of NFD and NCFD of the total lifetime with respect to DWIIs solved using CLRM and SRDM.

Chapter 7

Application of Least-Squares Monte Carlo Method and Reservoir Simulation Model

Hong et al. (2018) illustrated that the modified LSM algorithm could be implemented with a production model based on exponential declines to determine the optimal time to switch from one recovery phase to another. Theoretically, the LSM implementation is independent of production models, but a very computationally intensive model can cause the use of LSM to be computationally prohibitive. Apart from this, Alkhatib et al. (2013) showed that the LSM could be used to a reservoir model in the context of surfactant flooding. Therefore, this chapter briefly demonstrates that how the modified LSM method can be applied[51] with a reservoir simulation model to solve the IOR initiation time problem. Pertaining to this, only the modified LSM algorithm developed in Hong et al. (2018) is implemented to the reservoir simulation model[52].

7.1 General Workflow

To show that the modified LSM can be used alongside with a reservoir simulation model, a reservoir model is built. The detailed description regarding the reservoir model follows later. In this case, the ECLIPSE (2016) software developed by Schlumberger is used to perform the reservoir simulation of the model. Besides that, for HBL's model, it has been explained that different realizations of the petrophysical parameters have been generated to calculate the corresponding measured oil production rates. Therefore, for a reservoir simulation model, different realizations are also required to be used as the input parameters in the modified LSM algorithm. With respect to this, different realizations are created by stochastically changing the permeability of the reservoir model (using normal distribution and its details will be presented later). The generation of the realization is performed by using MATLAB R2019a (2019). Then, the simulation corresponding to each realization is run in ECLIPSE (2016). In this case, an interface between MATLAB R2019a (2019) and ECLIPSE (2016) is created in which each generated realization is written as an -.Inc file to be included into the input file of ECLIPSE (2016), which is -.DATA file. MATLAB R2019a (2019) can then be used to initiate the run of the simulation for different realization. As there is a change in the permeability, ECLIPSE (2016) simulation would result in different oil production rates that can be used as the input

[51] For the work in this chapter, the author also applied the CLRM approach to the reservoir simulation model. However, the focus is on the use of the modified LSM method.

[52] The author would like to emphasize that this chapter is a brief demonstration on the use of the modified LSM method with a reservoir simulation model. The author aims at making this section to serve as a foundation on which future works can be conducted upon. So, for illustrative purpose, only the modified LSM algorithm built in Hong et al. (2018) is implemented in the analysis for this section (not the extended one done in previous chapters).

parameters of the modified LSM algorithm. However, since MATLAB R2019a (2019) is also applied to implement the modified LSM algorithm, another interface between MATLAB R2019a (2019) and ECLIPSE (2016) is developed to import the resulting oil production rates (for each realization) from the output file of ECLIPSE (2016), which is -.RSM file to MATLAB R2019a (2019). Similar procedure is done for different switch times. In this context, the input file of the ECLIPSE (2016) is slightly modified to change the initiation time of the injection.

7.2 Reservoir Model

A 2D, vertical, 2 phases (water and black oil), and 2 layered reservoir simulation model is built to be used in tandem with the modified LSM algorithm. Two vertical wells are set to penetrate at each edge of the reservoir, namely the injector is at the left edge and the producer is at the right edge. Regarding the geometry of the model, the size of a grid block is cubic with length being 50 m, width being 50 m, and height being 1 m. Therefore, the dimension of the reservoir model is 2500 m × 50 m × 100 m. This dimension corresponds to 50 blocks × 1 block × 100 blocks. In this context, both layers have the same length and width. However, the top layer has the height of 40 m whereas the height of the bottom layer is 60 m. With the help of the software of ResInsight (2019), the top view of this reservoir model is shown in Fig. 61 whereas the front view of the reservoir is shown in Fig. 62. The well configuration can be seen in both figures.

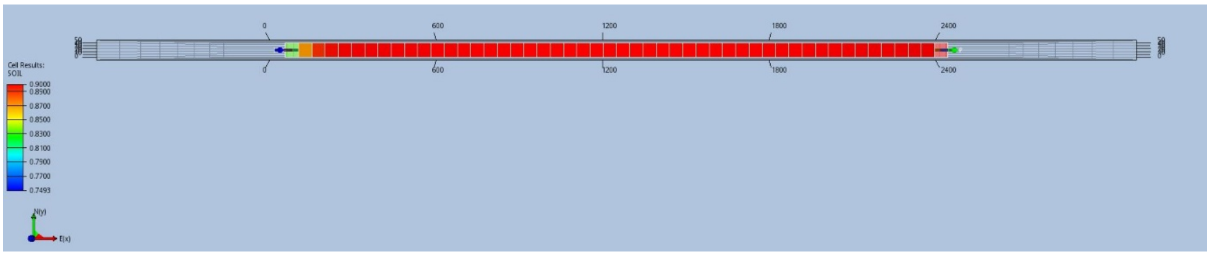


Fig. 61 – The top view of the 2D reservoir model.

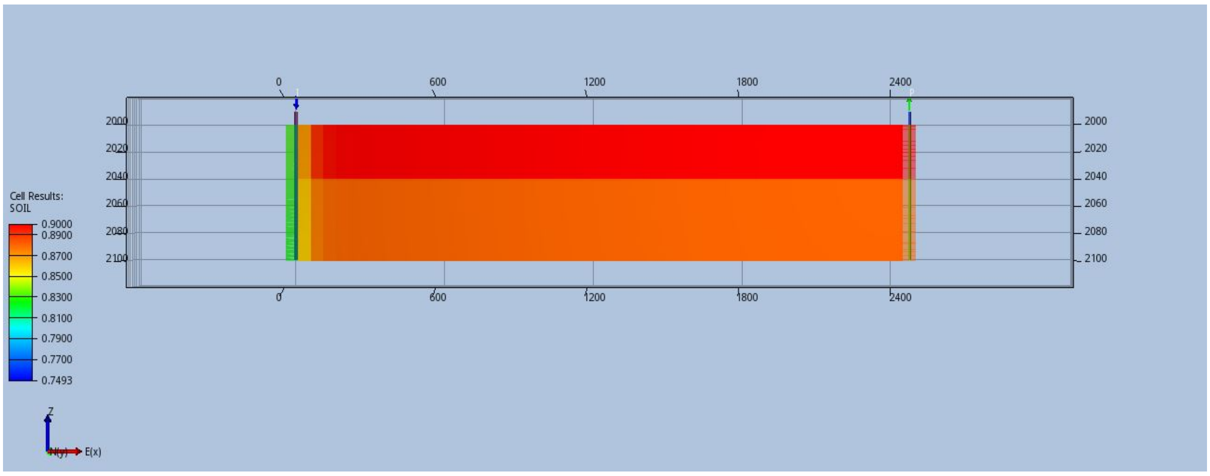


Fig. 62 – The front view of the 2D reservoir model.

The color scale of the oil saturation is provided on both Fig. 61 and Fig. 62. The scale of dimension used for both figures is in the unit of meter. Besides that, the injector is indicated by the blue arrow whereas the producer is indicated by the green arrow.

Apart from this, the horizontal permeability of each layer is assumed to be homogeneous in this case. This mean that the horizontal permeability of each grid block in the corresponding layer is the same. It is important to note that for this example, different realizations of the measured oil production rates are created by randomly generating a set of horizontal permeability for both layers. In this context, a normal distribution[53] is implemented alongside with the respective mean and SD of the horizontal permeability for each layer. Refer to Table 13 for the means and the SDs. The horizontal permeability of each layer is modeled independently[54] without being correlated with each other. Moreover, the vertical permeability of each layer is assumed to be 0.01% of the corresponding horizontal permeability. The purpose of this is to minimize the effect of crossflow of fluids between the layers.

Besides that, the injection and production rates of fluids are set to be the same, which are 15 sm^3/day. With respect to this, water is chosen to be the injection fluid. This denotes that the IOR method used in this example is waterflooding. Other important reservoir parameters and PVT properties are tabulated as shown in Table 14. Based on Table 14, it can be noted that aside from initial water saturation and porosity, the other parameters are assumed the same for these two layers.

Parameter	**Top Layer**	**Bottom Layer**
	Horizontal Permeability (mD)	
Mean	250	178
SD	35	20

Table 13 – Means and SDs for the normal distribution of the horizontal permeability for each corresponding layer.

[53] In general, the lognormal distribution is used to randomly generate the values of permeability to avoid any negative values. However, for this example, normal distribution is used for illustrative purpose. The 100 realizations of permeability generated from the normal distribution has been checked to be positive values before being used in the reservoir simulation.

[54] The permeability of the layers or the permeability of the grid blocks can be correlated with each other by finding the co-variance. Then, by having calculated this co-variance, a multi-variate normal distribution can be established from where different samples of the permeability (or logarithm of permeability) can be retrieved.

Parameter	Values	Units
Initial Reservoir Pressure	234	bar
Oil Density	1000	kg/m^3
Water Density	1000	kg/m^3
Oil Viscosity	6.4	cp
Water Viscosity	0.5	cp
Connate Water Saturation	0.05	
Residual Oil Saturation	0.05	
	Top Layer	**Bottom Layer**
Porosity	0.21	0.218
Initial Water Saturation	0.10	0.12

Table 14 – Values of important reservoir parameters and PVT properties.

7.3 Problem Setting

Regarding the problem setting of this example, a total period of 5 years of production time is assumed. This indicates that there would be 6 different switch times, namely at the beginning of Year 1, Year 2, Year 3, Year 4, Year 5, and Year 6[55]. Besides that, the values of economic parameters used in this example is also listed in Table 15.

Economic Parameters	Values	Units
Oil Price	90	$/bbl
CAPEX (Primary)	1	$ million
CAPEX_2After1 (Secondary)	1.15	$ million
CAPEX_2No1 (Secondary)	1.15	$ million
OPEX (Primary)	1.4	$ million/ year
OPEX (Secondary)	1.6	$ million/ year
Discount Rate	5%	per year

Table 15 – Values of economic parameters for this example[56].

To illustrate how the oil production rates obtained from the reservoir simulation is used as the input parameters for the modified LSM method, 100 different sets of horizontal permeability are generated using the normal distribution as explained. With this, the ECLIPSE (2016) simulation is run for 100 times where each simulation corresponds to each realization. Then, the simulation is repeated for 100 times for different switch times. This denotes that for this

[55] Switching at Year 6 indicates that the primary recovery continues for 5 years.

[56] The values of economic parameters used here might not be realistic (the oil price is assumed to be high whereas the costs are assumed to be low) because the reservoir model used does not recover sufficient amount of oil to produce the positive cashflow if the economic parameters listed in Table 4 are used. Thus, changes are made to the values of economic parameters for illustrative purpose.

example, the ECLIPSE (2016) simulation need to run for 600 times to gather all the necessary input parameters to be used in the modified LSM method. As known, the oil production rates generated by ECLIPSE (2016) are the modeled oil production rates. Thus, to obtain the measured rates, the measurement errors are assumed to be normally distributed with the mean of zero and the SD of 25% of the modeled rates and are added to the modeled rates.

7.4 Results

The DWOI is to have the primary recovery for 5 years without having the secondary recovery. This results in the total lifetime of 5 years. Thus, the EVWOI is found out to be $5.22 million. Regarding perfect information, the EVWPI is $6.37 million. This results in the VOPI to be $1.15 million. The EVWII corresponding to the SRDM approach is $5.437 million. The respective VOI is $0.221 million. Moreover, this result shows that having the effect of future information and decisions would improve the EV by 4.23%. Besides that, the CLRM approach is also used to solve this example. The corresponding EVWII is $5.425 million. This yields a special VOI of $0.209 million and an improvement of EV by 4%. In this aspect, SRDM creates an additional value of learning of $0.012 million as compared to CLRM.

Chapter 8

Discussions and Suggestions for Further Works

Hong et al. (2018) argued that HBL's model could provide a fast[57] analysis to determine the optimal IOR initiation time throughout the lifetime of the production of an oil field. In this context, such analysis can be fast due to the relatively low computational intensiveness of the two-factor production model. Despite the fact that the production model used is simple, applying it in HBL's model still provides useful insights about the optimal choice of the IOR initiation problem (a sequential decision problem). In this analysis, SRDM and CLRM approaches are respectively used to yield the corresponding near-optimal decision policy. As the SRDM approach includes the effect of future data and decisions, it represents a decision maker's optimal, better than CLRM, response to information (Hong et al., 2018). The fact that the SRDM accounts for learning over time in which its impact of this learning and decision making is illustrated by its higher VOI in relative to CLRM. The difference between the VOIs is the value of learning. Upon replicating HBL's model, the validity of the implementation of the modified LSM algorithm in tandem with two-factor production model was shown.

The modified LSM algorithm is based on an ADP approach. Thus, there is always an approximation error induced when applying this method. In this case, Hong et al. (2018) explained that the accuracy of this method relied upon the number of paths (samples) of Monte Carlo and how close a regression function could estimate the actual EV. Regarding the number of Monte Carlo samples, Hong et al. (2018) demonstrated through a sensitivity analysis with an illustrative example that 100000 samples was sufficient to provide a highly accurate VOI. In this work, a similar sensitivity analysis was done (refer to Fig. 3 and Fig. 4) and the results illustrated that 10000 samples could provide a sufficiently good approximation of VOI. Besides that, using a more complicated regression function (not considering the use of a higher order non-linear regression function as explained) would provide a higher accuracy. Applying a more sophisticated regression function would also require a higher computation time. Since it has been shown that the percentage of improvement on EV is not significant for this example problem, the resulting higher accuracy is not material. As a trade-off between accuracy and runtime, the linear regression is still preferred. Additionally, linear regression has been shown to work well in LSM for the oil and gas related studies (Smith, 2005; Jafarizadeh and Bratvold, 2009; Willigers et al., 2011; Thomas and Bratvold, 2015; Hong et al., 2018). Further works can be conducted to identify the effectiveness[58] of a more complicated regression in a

[57] Generally, in the context of IOR initiation time, a decision is made every year. Therefore, the respective decision model is fast if its computation takes less than a year. Therefore, the HBL's model is indeed a fast analysis and not "expensive" to be used as its computation time can be done within a day. However, its runtime is still subject to the complexity of the production model used, the number of Monte Carlo samples, the number of alternatives and decision points.

[58] "Effectiveness" here means the use of more complicated regression function that will result in a significant percentage improvement of EV.

sequential decision problem related to petroleum industry. We suspect that a highly non-linear value function may require terms in the regression function that capture such a non-linearity.

Theoretically, the SRDM approach would provide additional value of learning that makes the corresponding VOI to be higher than the special VOI of CLRM. In this context, for a certain problem setting, an approximation error might produce a result in which the special VOI is higher than the VOI. It has been shown in this work that such unrealistic results still occur as more complicated regression functions are used. More detailed analysis is recommended to mitigate the occurrence of this unrealistic result. In this case, reducing sampling error might help to decrease the approximation error. Thus, different sampling methods, such as Latin hypercube sampling[59], can be implemented. Besides that, as proposed by Alkhatib et al. (2013), sampling techniques can also be improved with methods like polynomial chaos theory and probabilistic collocation[60]. With respect to this, a higher number of samples can be tested along with these sampling methods. However, as the number of samples increases, a longer computation time is expected.

Hong et al. (2018) focused solely on the uncertainties in petrophysical parameters of the two-factor production model. Therefore, HBL's model has been extended in this work to include uncertainties in economic parameters. Although this extension can be simply conducted by adding these variables into the regression, the economic parameters need to be modeled as stochastic processes. In this work, the OU model was selected to capture uncertainties in the economic parameters. The extended HBL's model is deemed to perform well because it provides insightful results of the IOR initiation problem as presented in sections 6.5 and 6.6. Schwartz (1997) illustrated that there were two other models which could be used to describe the behavior of oil prices, namely two-factor model (this is another example of stochastic model and not to be confused with the two-factor production model) and the three-factor model[61]. Thus, for further works, two-factor or three-factor models can be applied to model the economic parameters stochastically. Then, the decision policies produced by implementing different models can be compared and analyzed. Besides that, taxation that is material to the decisions can be considered in the formulation of the economic model, which is not done in both Hong et al. (2018) and this work.

Apart from this, it has been stated that using fewer data points provide a good estimation of results due to the simplicity of the two-factor production model. With respect to this, there is a

---

[59] Refer to McKay et al. (1979) for the details. Besides that, Bratvold and Begg (2010) also provides an introduction of Latin hypercube sampling (LHS). Basically, LHS is a form of stratified sampling in which each CDF of input variable is partitioned into a fixed number of strata (Bratvold and Begg, 2010). The number of strata is the number of iterations needed (Bratvold and Begg, 2010).

[60] Refer to Ghanem and Spanos (1991) for the details of polynomial chaos theory and Tatang et al. (1997) for the details of probabilistic collocation. These methods generally enable the sampling of input variables from the high-probability areas of distribution to be done (Alkhatib et al., 2013).

[61] Jafarizadeh and Bratvold (2012) explained that for a price model, a factor was a variable that showed a random behavior in a market. GBM and OU model are one-factor models as the oil price behaves randomly (Jafarizadeh and Bratvold, 2012). They also expounded that for two-factor or three-factor model, apart from the price, the long-term price, interest rate, and so forth can be modeled as stochastic processes.

possibility that more data points might be needed to provide a more accurate estimation if the measurement error increases. However, this hypothesis is not investigated in this research work. Thus, it can be considered as one of the further studies in future. The sensitivity analysis on the model parameters can also be done on the extended HBL's model to identify the variables that result in the largest impact on the decision policy. It is important to understand that as uncertainties in economic parameters are considered in the extended HBL's model, the corresponding sensitivity analysis on the economic parameters is different (from the one performed in Section 5.1). In this case, it is recommended that the correlation coefficients in Table 12 are changed to analyze the resulting change in the decision policy. Nevertheless, assessing the correlation coefficient matrix (in Table 12) is not very straightforward. The correlations may be easier to be assessed if there are relevant data sets available for the economic parameters. Moreover, sensitivity analysis on the number of decision points, the number of data points, and SD of measurement error can also be conducted on the extended HBL's model to obtain further insights about the decision problem. Hong et al. (2018) also suggested that when there was a change in the correlation coefficient matrix as shown in Table 3, sensitivity analysis could be done to analyze the resulting change in decisions. This suggestion is also applicable to the extended HBL's model.

In this work, the author provides a brief illustration on how reservoir simulation can be applied with the modified LSM method by using a very simple reservoir model. As presented, the modified LSM algorithm is used to inform decisions which are based on a decision criterion, that is the EV. In this case, the objective was to use a simple example to show how the reservoir simulation model could be used to develop a cashflow model for the calculation of EV. Obviously, a number of future studies on the use of more advanced production models are possible. Applying a richer reservoir model, such as a 3D reservoir model with heterogeneity in permeability field, is one possibility. Besides that, the modified LSM algorithm that considers uncertainties in economic parameters can be applied to this more complicated reservoir model. Other reservoir parameters can also be modeled stochastically, for example, initial water saturation, porosity, etc. In addition, apart from measured oil production rates, the measured water production rates can be treated as another state variable in the regression analysis. It was not considered in Hong et al. (2018) and this work due to the formulation of the two-factor production model. However, this suggestion is viable as reservoir simulation would yield the simulated results of water production rates. Different types of IOR can also be tested, for instance, polymer flooding, gas flooding, etc. Upon using a richer production model with the modified LSM algorithm, a sensitivity analysis on the variables discussed in Chapter 5 can also be done to get further insights.

Chapter 9

Conclusions

In this work, the author briefly discussed how the sequential nature in reservoir decision making could help to maximize the value creation from oil and gas reservoirs. The author also mentioned how solving the IOR initiation problem could be understood as a sequential decision making in reservoir management. Thereafter, the author provided a brief review about the theoretical aspects of two optimization methods used to solve this problem, namely SRDM and CLRM. Pertaining to this, SRDM approach could be treated as the application of the modified LSM algorithm in reservoir management. The author also discussed and illustrated the detailed procedure of implementing these two approaches with the two-factor production model. For simplicity, the use of these methods in tandem with the production and economic models is referred to as "HBL's model". The author successfully replicated the application of SRDM and CLRM to solve the IOR initiation problem as shown in Hong et al. (2018) as the results estimated by the replicated model coincided with those presented in Hong et al. (2018). The validity of the implementation of the HBL's model was thus verified. Moreover, this replication also substantiated that HBL's model was indeed a useful and tractable decision model for optimizing the IOR initiation time throughout the lifetime of production.

The author explained and demonstrated different sensitivity analysis by using both the SRDM and CLRM approaches. The first sensitivity analysis in this work was done on the choice of regression function. A more complicated regression function could provide a more accurate estimation of an "actual" EV[62]. In this work, more complex regression functions were tested by including different types of interaction terms in the function and an exponential term with the use of Laguerre polynomials. The interaction terms were included to the regression function because the measured oil production rates were modeled as non-Markovian processes. The sensitivity analysis on the choice of regression function portrayed that having a more complicated regression function did not result in a significant percentage of improvement of the EV[63]. Thus, such improvement is not material to the decisions. As a trade-off between accuracy and computation time, linear regression was chosen for further analysis in this work. Additionally, for certain problem settings, these complicated regression functions could not prevent the occurrence of the unrealistic result (VOI of CLRM is higher than VOI of SRDM) caused by the approximation error. Thus, further works were suggested to mitigate this problem.

The author also illustrated and explained the implementation of sensitivity analysis on the model parameters of the two-factor production model and the economic parameters listed in Table 4. This sensitivity analysis is important as it could help a decision maker to identify the

[62] To emphasize, the "actual" EV here is not the "real-world" EV. It means the approximated EV resulting from a full (not approximate) DP implementation. Therefore, this, of course, is still a model-based approximation to the "real-world" EV.

[63] Higher percentage of improvement of EV indicates higher accuracy of estimation of results.

material variables for the decision policy. Then, more information about these variables could be gathered to reduce the respective uncertainties in decision making if the additional information could create more value. The author also conducted a sensitivity analysis on the number of decision points and the number of data points. The result obtained from the sensitivity analysis on the number of decision points provided a further confirmation that the implementation of the modified LSM algorithm was correct. The same applied to CLRM. From the sensitivity analysis on the number of data points, we can deduce that using fewer data points still generates an estimation of EV that was close to the one approximated by modeling the measured rates as non-Markovian processes. However, for this example problem, modeling the measured oil rates as Markovian processes did not generate a good estimation. Therefore, for practical purpose, lowering the number of data points (to certain extent) can be viable to approximate the EV. The author also outlined the implementation of sensitivity analysis for the SD of measurement error. This analysis provided useful insights to help a decision maker to decide whether he or she should implement a more accurate but also more expensive information-gathering activity.

Furthermore, the author extended the HBL's model by including the uncertainties in the economic parameters listed in Table 4. The extended model provides realistic decision supporting results. Both SRDM and CLRM approaches respectively yielded a decision policy that could identify the near-optimal switch time. However, when compared to the CLRM approach, the SRDM approach resulted in a higher EV and this corresponded to a higher VOI. This comparison highlighted the suboptimality of CLRM. However, the value of learning induced by SRDM is not very significant (or is small) for some of the problem settings. This implies that for some cases, the model choice may not be material[64]. As Hong et al. (2018) argued, the suboptimality of CLRM is case-dependent. In Section 6.6, under a different problem setting, the value of learning was significant because there was a large difference between the decision policy of SRDM and that of CLRM. The terms "large difference" means that for every data path, the CLRM suggested a lifetime of primary recovery for 0 years whereas SRDM proposed otherwise. This resulted in a larger difference between the total lifetime corresponding to both SRDM and CLRM. Thus, this corresponded to a more significant[65] value of learning, which denotes that such value is material to the decisions. Besides that, the author also successfully showed that the modified LSM algorithm could be applied with a reservoir simulation model to generate insightful results. Although the reservoir model used in this study was very simple, the purpose of this brief demonstration was to provide a foundation on which further works could be done.

We can conclude that in terms of solving for the optimal IOR initiation time, the modified LSM algorithm is indeed robust as it can easily be changed to include uncertainties in other state

[64] Relying upon the problem setting, the VOI estimated by using SRDM can be very close to that of CLRM. So, it seems to be indifferent for an individual to implement either SRDM or CLRM. However, in general, SRDM provides a solution that is closer to the actual optimum as compared to CLRM. In other words, SRDM guarantees near-optimality, but CLRM does not.

[65] It is subjective in terms of defining the threshold for the value of learning to be significant as there is no so-called "correct" threshold.

variables which are material to the decisions and conveniently implemented in tandem with different types of production model (whether it be decline-curve based model or reservoir simulation model) to produce insightful results. However, its computation effort is still subject to limited number of alternatives and decision points.

References

Aanonsen, S. I., Nævdal, G., Oliver, D. S., Reynolds, A. C., and Vallès, B. 2009. The Ensemble Kalman Filter in Reservoir Engineering--a Review. *SPE Journal* **14** (3): 393-412. SPE-117274-PA. https://doi.org/10.2118/117274-PA.

Alkhatib, A. M., Babaei, M. and King, P. R. 2013. Decision Making Under Uncertainty: Applying the Least-Squares Monte Carlo Method in Surfactant-Flooding Implementation. *SPE Journal* **18** (4): 721-735. SPE-156647-PA. https://doi.org/10.2118/154467-PA.

Alyaev, S., Hong, A. J., and Bratvold, R. B. 2018. Are you myopic, naïve or farsighted about your geosteering decision? Presented at the Second EAGE/ SPE Geosteering and Well Placement Workshop 2018, Abu Dhabi, United Arab Emirates, 5-8 November. https://doi.org/10.3997/2214-4609.201803217.

Begg, S. H., and Smit, N. 2007. Sensitivity of Project Economics to Uncertainty in Type and Parameters of Price Models. Presented at the SPE Annual Technical Conference and Exhibition, Anaheim, California, USA. 11-14 November. SPE-110812-MS. http://doi.org/10.2118/110812-MS.

Black, F., and Scholes, M. 1973. The Pricing of Options and Corporate Liabilities. *Journal of Political Economy* **81** (3): 637-654. http://dx.doi.org/10.1086/260062.

Boyle, P. P. 1977. Options: A Monte Carlo approach. *Journal of Financial Economics* **4** (3): 323-338. http://dx.doi.org/10.1016/0304-405X(77)90005-8.

Brandão, L. E., Dyer J. S., and Hahn W. J. 2005. Using Binomial Decision Tress to Solve Real-Option Valuation Problems. *Decision Analysis* **2** (2): 69-88. https://doi.org/10.1287/deca.1050.0040.

Bratvold, R. B., Bickel, J. E., and Lohne, H. P. 2009. Value of Information in the Oil and Gas Industry: Past, Present, and Future. *SPE Reservoir Evaluation & Engineering* **12** (4): 630–638. SPE-110378-PA. https://doi.org/10.2118/110378-PA.

Bratvold, R. B., and Begg, S. 2010. *Making Good Decisions*. Richardson, Texas: Society of Petroleum Engineers.

Cardenas, C. 2017. *Offshore Exploratory Drilling Campaigns During Low Oil Price Period: Maximizing Value Creation from Flexibility*. Master Thesis. University of Stavanger, Norway.

Chen, Y., Oliver, D. S., Zhang, D. 2009. Efficient Ensemble-Based Closed-Loop Production Optimization. *SPE Journal* **14** (4): 634–645. SPE-112873-PA. https://doi.org/10.2118/112873-PA.

Copeland, T. and Antikarov, V. 2001. *Real Options*. New York: Texere LLC.

Devore, J. L. 2010. *Probability and Statistics for Engineering and the Sciences*. Boston, Massachusetts: Brooks/ Cole.

Dias, M. A. G. 2004. Valuation of exploration and production assets: an overview of real options models. *Journal of Petroleum Science and Engineering* **44** (1−2), 93-114. https://doi.org/10.1016/j.petrol.2004.02.008.

Dixit, A. K., and Pindyck, R. S. 1994. *Investment under Uncertainty*. Princeton, New Jersey: Princeton University Press.

ECLIPSE Reservoir Engineering Software. 2016. Schlumberger, https://www.software.slb.com/products/eclipse?tab=Overview (accessed 24 March 2019).

Evans, R. 2000. Decision Analysis for Integrated Reservoir Management. Presented at the SPE European Petroleum Conference 2000, Paris, France, 24-25 October. SPE-65148-MS. https://doi.org/10.2118/65148-MS.

Fu, M. C., Laprise, S. B., Madan, D. B., Su, Y., and Wu, R. 2001. Pricing American Options: A Comparison of Monte Carlo Simulation Approaches. *Journal of Computational Finance* **4** (3): 39-88. https://doi.org/10.21314/JCF.2001.066.

Gagniuc, P. A. 2017. *Markov Chains: From Theory to Implementation and Experimentation*. New Jersey: John Wiley and Sons.

Ghanem, R. and Spanos, P. 1991. *Stochastic Finite Elements: A Spectral approach*. New York: Springer-Verlag.

Gillespie, D. T. 1996. Exact numerical simulation of the Ornstein-Uhlenbeck process and its integral. *Phys Rev*. **54** (2): 2084-2091. https://doi.org/10.1103/PhysRevE.54.2084.

Hermant, A. P. and Oilver, D. S. 2011. Data Assimilation Using the Constrained Ensemble Kalman Filter. *SPE Journal* **16** (2): 331-342. https://doi.org/10.2118/125101-PA.

Hillson, D. and Murray-Webster, R. 2005. *Understanding and Managing Risk Attitude*. Burlington, Vermont: Gower Publishing Ltd.

Hong, A. J., Bratvold, R. B., and Lake, L. W. 2018. Fast Analysis of Optimal IOR Switch Time Using a Two-Factor Production Model and Least- Squares Monte Carlo Algorithm (includes associated errata). *SPE Reservoir Evaluation & Engineering* **Preprint** (Preprint). https://doi.org/10.2118/191327-PA.

Howard, R. A. 1960. Dynamic Programming. *Management Science* **12** (5): 317-348. https://doi.org/10.1287/mnsc.12.5.317.

Howard, R. A. 1966. Information Value Theory. *IEEE Transactions on Systems Science and Cybernetics* **2** (1): 22–26. http://doi.org/10.1109/TSSC.1966.300074.

Howard, R. A. 1988. Decision Analysis: Practice and Promise. *Management Science* **34** (6): 679-695. https://doi.org/10.1287/mnsc/34.6.679.

Iqbal, Z., and Shetty, S. 2018. The impact of oil price shocks on capital spending in the oil and gas industry: A VAR analysis at the firm level. *Managerial Finance* **44** (11): 1347-1363. https://doi.org/10.1108/MF-09-2017-0326.

Jafarizadeh, B., and Bratvold, R. B. 2009. Taking Real Options Into Real World: Asset Valuation Through Option Simulation. Presented at the SPE Annual Technical Conference and Exhibition, New Orleans, Louisiana, 4-7 October. SPE-124488-MS. https://doi.org/10.2118/124488-MS.

Jafarizadeh, B., and Bratvold, R. 2012. Two-Factor Oil-Price Model and Real Option Valuation: An Example of Oilfield Abandonment. *SPE Economics & Management* **4** (3), 158 - 170. https://doi.org/10.2118/162862-PA.

Jafarizadeh, B., and Bratvold, R. 2013. Sell Spot or Sell Forward? Analysis of Oil-Trading Decisions With the Two-Factor Price Model and Simulation. *SPE Economics & Management* **5** (3), 80 - 88. https://doi.org/10.2118/165581-PA.

Jansen, J. D., Brouwer, R., and Douma, S. G. 2009. Closed Loop Reservoir Management. Presented at the SPE Reservoir Simulation Symposium, The Woodlands, USA, 2-4 February. SPE-119098-MS. https://doi.org/10.2118/119098-MS.

Lake, L. W. 1989. *Enhanced Oil Recovery*. Eaglewood Cliffs, New Jersey: Prentice Hall.

Longstaff, F., and Schwartz, E. 2001. Valuing American Options by Simulation: a Simple Least-Squares Approach. *Review of Financial Studies* **14** (1): 113–147. https://doi.org/10.1093/rfs/14.1.113.

Martinelli, G., Eidsvik, J., and Hauge, R. 2013. Dynamic decision making for graphical models applied to oil exploration. *European Journal of Operation Research* **230** (3): 688-702. https://doi.org/10.1016/j.ejor.2013.04.057.

Martinkutė-Kaulienė, R. 2012. Exotic Options: a Chooser Option and its Pricing. Business, *Management and Education* **10** (2): 289–301. https://doi.org/10.3846/bme.2012.20.

MATLAB, version R2019a. 2019. Natick, Massachusetts: The Mathworks, Inc.

McKay, M. D., Beckman, R. J. and Conover, W. J. 1979. A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code. *Technometrics* **21** (2): 239-245. https://dx.doi.org/10.2307/1268522.

Moreno, M., and Navas, J. F. 2003. On the Robustness of Least-Squares Monte Carlo (LSM) for Pricing American Derivatives. *Review of Derivatives Research* **6** (2), 107-128. https://doi.org/10.1023/A:1027340210935.

Ng, C. S. W. 2018a. *Dynamic Programming: Using the Least-Squares Monte Carlo Algorithm to Solve the Sequential Decision Problems (Project Report)*. Trondheim, Norway: Norwegian University of Science and Technology.

Ng, C. S. W. 2018b. *Literature Review on the Use of Least-squares Monte Carlo Method in Decision Making of the Implementation of IOR/EOR (Project Report)*. Trondheim, Norway: Norwegian University of Science and Technology.

Parra-Sanchez, C. 2010. *A Life Cycle Optimization Approach to Hydrocarbon Recovery*. Master Thesis. The University of Texas at Austin, USA.

Powell, W. B. 2011. *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. Princeton, New Jersey: John Wiley and Sons.

Pindyck, R. S. 1999. The Long-Run Evolution of Energy Prices. *The Energy Journal* **20** (2): 1-27. doi:10.5547/ISSN0195-6574-EJ-Vol20-No2-1

ResInsight. 2019. Ceetron Solution AS, https://resinsight.org/project/download/ (accessed 25 March 2019)

Schlumberger Oilfield Glossary. 2019a. History Matching, https://www.glossary.oilfield.slb.com/en/Terms/h/history_matching.aspx (assessed 3 June 2019).

Schlumberger Oilfield Glossary. 2019b. Improved Oil Recovery, https://www.glossary.oilfield.slb.com/en/Terms/i/improved_oil_recovery.aspx (assessed 1 May 2019).

Schlumberger. 2014. ECLIPSE Reference Manual, Version 2014.1. Houston, Texas: Schlumberger

Schwartz, E. S. 1997. The Stochastic Behavior of Commodity Prices: Implications for Valuation and Hedging. *The Journal of Finance* **52** (3): 923-973. https://doi.org/10.1111/j.1540-6261.1997.tb02721.x.

Smith, J. E. 2005. Alternative Approaches for Solving Real-Options Problems. *Decision Analysis* **2** (2): 89-102. https://doi.org/10.1287/deca.1050.0041.

Smith, J. E., and McCardle, K. F. 1999. Options in the Real World: Lessons Learned in Evaluating Oil and Gas Investments. *Operations Research* **47** (1): 1-15. https://doi.org/10.1287/opre.47.1.1.

Smith, W. 2010. On the Simulation and Estimation of the Mean-Reverting Ornstein-Uhlenbeck Process. https://commoditymodels.files.wordpress.com/2010/02/estimating-the-parameters-of-a-mean-reverting-ornstein-uhlenbeck-process1.pdf (downloaded 16 April 2019)

Surovtsev, D., and Sungorov, A. 2016. "Vaguely Right or Precisely Wrong?" Making Probabilistic Cost, Time, and Performance Estimates for Bluefields Appraisal.

Presented at the SPE Russian Petroleum Technology Conference and Exhibition 2016, 24-26 October, Moscow, Russia. SPE-181904-MS. https://doi.org/10.2118/181904-MS.

Stentoft, L. 2004. Convergence of the Least Squares Monte Carlo Approach to American Option Valuation. *Management Science* **50** (9) : 1193-1203. https://doi.org/10.1287/mnsc.1030.0155.

Tatang, M., Pan, W., and Prinn, R., et al. 1997. An Efficient Method for Parametric Uncertainty Analysis of Numerical Geophysical Models. *J. Geophys. Res*. **102** (18): 21925-21932. https://doi.org/10.1029/97JD01654.

Thakur, G. C. 1996. What is Reservoir Management? *SPE Journal of Petroleum Technology*. **48** (06): 520-525. SPE-26289-JPT. https://doi.org/10.2118/26289-JPT.

Thomas, P., and Bratvold, R. B. 2015. A Real Options Approach to the Gas Blowdown Decision. Paper presented at the SPE Annual Technical Conference and Exhibition, Houston, Texas, USA, 28-30 September. https://doi.org/10.2118/174868-MS.

Uhlenbeck, G., and Ornstein, L. 1930. On the Theory of the Brownian Motion. *Phys Rev*. **36** (5): 823-841. https://doi.org/10.1103/PhysRev.36.823.

U.S. Energy Information Administration (EIA). 2019. NYMEX Futures Price Data. https://www.eia.gov/dnav/pet/pet_pri_fut_s1_a.htm (accessed 15 March 2019)

Wang, C., Li, G., and Reynolds, A. C. 2009. Production Optimization in Closed-Loop Reservoir Management. *SPE Journal* **14** (3): 506–523. SPE-109805-PA. https://doi.org/10.2118/109805-PA.

Wiggins, M. L., and Startzman, R. A. 1990. An Approach to Reservoir Management. Presented at the SPE Annual Technical Conference and Exhibition 1990, New Orleans, Louisiana, USA, 23–26 September. SPE-20747-MS. https://doi.org/10.2118/20747-MS.

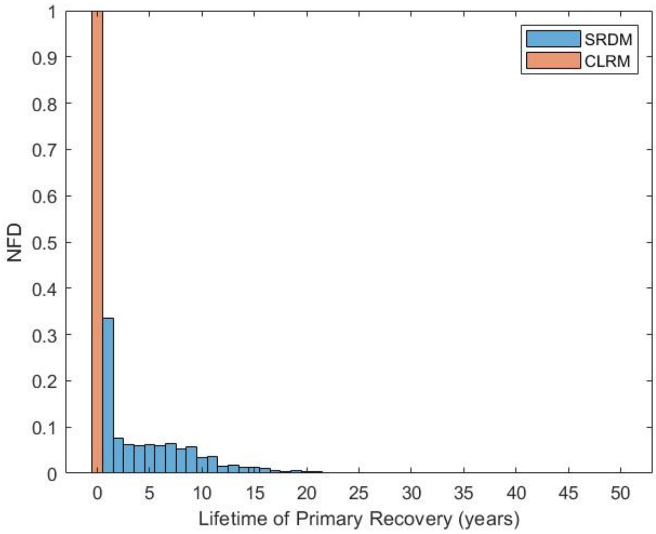
Willigers, B. J. A. 2009. Enhanced Economic Modeling by Correlated Stochastic Models of E&P Costs and Hydrocarbon Prices: The Limitations of Fixed Price Decks and the Versatility of Least-Squares Monte Carlo Simulation. Paper presented at the EUROPEC/EAGE Conference and Exhibition, Amsterdam, The Netherlands. https://doi.org/10.2118/121442-MS.

Willigers, B. J. A., Begg, S. H., and Bratvold, R. B. 2011. Valuation of Swing Contracts by Least-Squares Monte Carlo Simulation. *SPE Economics & Management* **3** (4): 215–225. SPE-133044-PA. https://doi.org/10.2118/133044-PA.

Willigers, B. J. A., and Bratvold, R. B. 2009. Valuing Oil and Gas Options by Least-Squares Monte Carlo Simulation. *SPE Projects, Facilities & Construction* **4** (4): 146-155. https://doi.org/10.2118/116026-PA.

Appendices

Appendix A: Supplementary Figures[66]



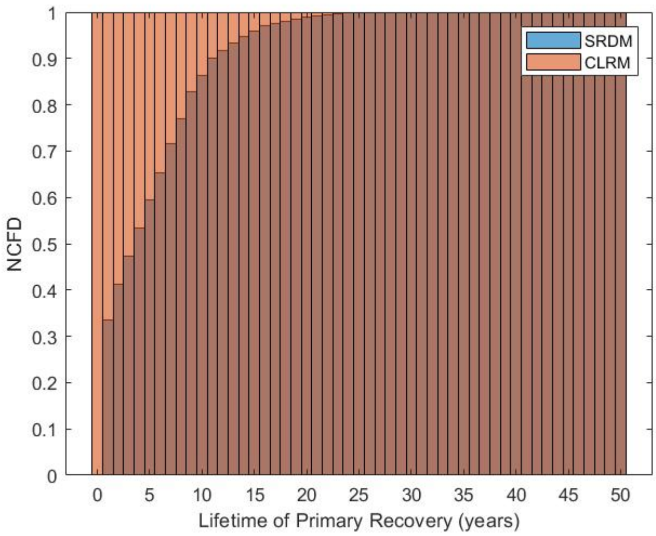
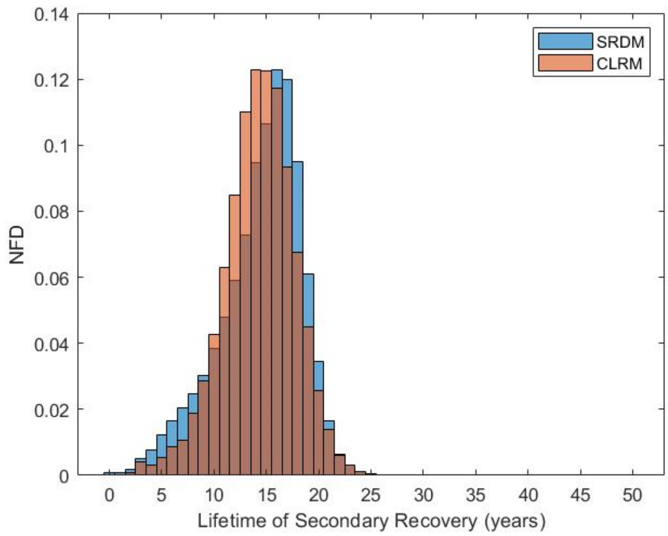


Fig. 63 – Plot of NFD and NCFD of the lifetime of primary recovery with respect to DWIIs solved using CLRM and SRDM for one of the problem settings in Chapter 5.

[66] These supplementary figures illustrate the comparison between the decision policies of SRDM and CLRM based on one of the problem settings discussed in Chapter 5. These figures show the Normalized Frequency Distributions (NFDs) and the Normalized Cumulative Frequency Distributions (NCFDs) of the lifetime of primary recovery, the lifetime of secondary recovery, and the total lifetime of the DWIIs of CLRM and SRDM. For the color legend shown in these figures, the additional brown color shows an overlap between light brown bars and light blue bars. In general, it can be deduced that the total lifetime of DWII corresponding to SRDM is much longer than that of CLRM. This results in a large difference between EVWII of SRDM and that of CLRM.



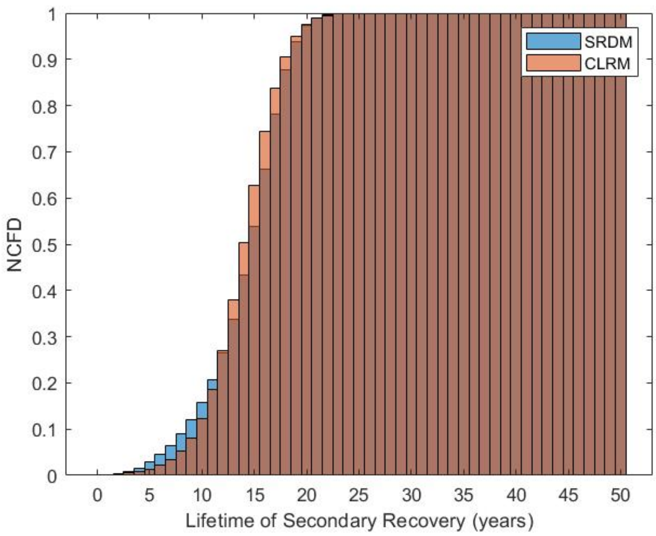
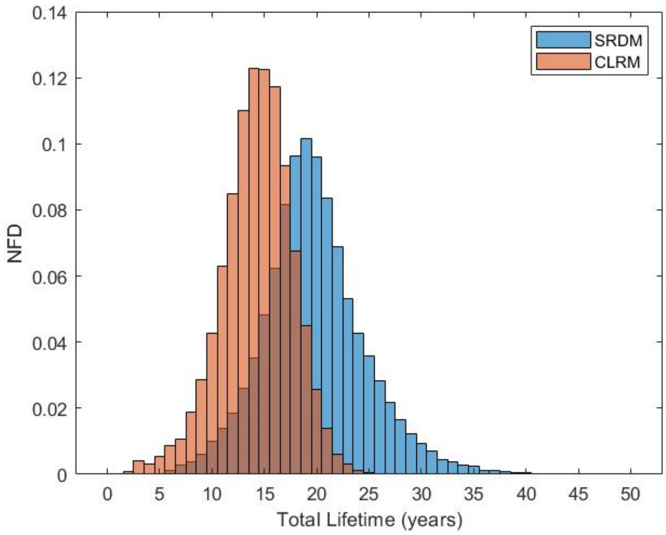


Fig. 64 – Plot of NFD and NCFD of the lifetime of secondary recovery with respect to DWIIs solved using CLRM and SRDM one of the problem settings in Chapter 5.



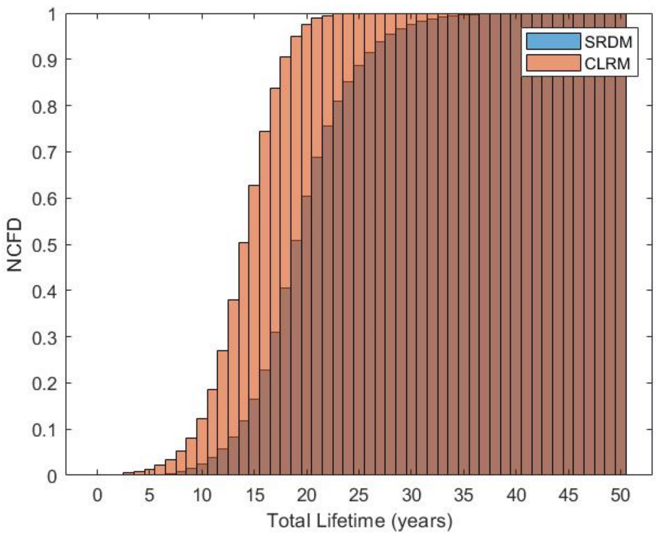


Fig. 65 – Plot of NFD and NCFD of the total lifetime with respect to DWIIs solved using CLRM and SRDM one of the problem settings in Chapter 5.

Appendix B: MATLAB Codes

MATLAB R2019a (2019) has been applied to perform the modified LSM algorithm and other related computation in this research work. All the related codes would be presented along with some brief explanations.

Appendix B1

Appendix B1 consists of the codes of the replicated HBL's model, which are the modified LSM algorithm and the CLRM approach, as discussed in Chapter 3. These codes are also used to perform the sensitivity analysis as discussed in Sections 5.1 and 5.3. For convenience, slight modification can be done on these codes by adding a for-loop to compute the corresponding sensitivity analysis in one attempt.

Appendix B1.1: Modified LSM Algorithm

This code performs the modified LSM algorithm in Hong et al. (2018). This is the replicated model of the modified LSM algorithm.

```
% 1. Monte Carlo Sampling
% This section initializes the necessary input parameters to perform the
% Monte Carlo sampling of the model parameters of the two-factor production model

N_MC = 100000;
FieldLifeTime = 50;

OOIP_Mean = 240;
E_inf_Phase1_Mean = 0.2;
Tau_Phase1_Mean = 16;
dE_inf_Phase2_Mean = 0.15;
Tau_Phase2_Mean = 7;

OOIP_SD = 35;
E_inf_Phase1_SD = 0.05;
Tau_Phase1_SD = 2;
dE_inf_Phase2_SD = 0.05;
Tau_Phase2_SD = 1.5;

ErrorMean4Rate_Pct = 0;
ErrorSD4Rate_Pct = 0.10;

% Initialization of parameters for multivariate normal distribution
mu = [OOIP_Mean, E_inf_Phase1_Mean, Tau_Phase1_Mean, dE_inf_Phase2_Mean, Tau_Phase2_Mean];
ExpSigma = [OOIP_SD, E_inf_Phase1_SD, Tau_Phase1_SD, dE_inf_Phase2_SD, Tau_Phase2_SD];
ExpCorrC = [1 -0.8 0.16 0.56 -0.08; -0.8 1 0.2 -0.7 0.1; 0.16 0.2 1 -0.3 -0.2; 0.56 -0.7 -0.3
1 -0.3; -0.08 0.1 -0.2 -0.3 1];
cov = ExpCorrC.*(ExpSigma'*ExpSigma);
PriorRealizations = mvnrnd(mu, cov, N_MC);
```

```
% Prior realizations of the model parameters
OOIP_prior = PriorRealizations(:,1); OOIP_prior(OOIP_prior<10) = 10;
OOIP_prior(OOIP_prior>1000) = 1000;
E_inf_Phase1_prior = PriorRealizations(:,2); E_inf_Phase1_prior(E_inf_Phase1_prior<0.05) =
0.05; E_inf_Phase1_prior(E_inf_Phase1_prior>0.5) = 0.5;
Tau_Phase1_prior = PriorRealizations(:,3); Tau_Phase1_prior(Tau_Phase1_prior<1) = 1;
Tau_Phase1_prior(Tau_Phase1_prior>30) = 30;
dE_inf_Phase2_prior = PriorRealizations(:,4); dE_inf_Phase2_prior(dE_inf_Phase2_prior<0.01) =
0.01; dE_inf_Phase2_prior(dE_inf_Phase2_prior>0.31) = 0.31;
Tau_Phase2_prior = PriorRealizations(:,5); Tau_Phase2_prior(Tau_Phase2_prior<1) = 1;
Tau_Phase2_prior(Tau_Phase2_prior>13) = 13;


%  2. Two-Factor Production Model
%  This section performs the calculation of the recovery factors and the respective modeled
oil production rates

Recovery_Phase1_real_time = zeros(N_MC,FieldLifeTime);
Recovery_Phase1_real_EndTime = zeros(N_MC,1);
Recovery_Phase2_real_time = zeros(N_MC,1);
Recovery_Phase1n2_real_time = zeros(N_MC,FieldLifeTime);
Recovery_Phase1n2_real_time_ShiftTime = zeros(N_MC,FieldLifeTime,FieldLifeTime+1);
Rate_Phase1n2_real_time_ShiftTime = zeros(N_MC,FieldLifeTime,FieldLifeTime+1);

for i_MC = 1: N_MC
    for time = 1: FieldLifeTime %this is the time shift in rows
    Recovery_Phase1_real_time(i_MC,time) = (E_inf_Phase1_prior(i_MC)).*(1-exp(-
time/Tau_Phase1_prior(i_MC))); %computation of the E_R1
    if time == 1
        Recovery_Phase1_real_EndTime(i_MC) = 0;
    else
        Recovery_Phase1_real_EndTime(i_MC) = Recovery_Phase1_real_time(i_MC,time-1);
    end

    Recovery_Phase2_real_time(i_MC) = Recovery_Phase1_real_EndTime(i_MC) +
dE_inf_Phase2_prior(i_MC).*(1-exp(-1/Tau_Phase2_prior(i_MC)));

    for t = 1: FieldLifeTime %this is the time shift in columns
      %computation of the E_R with both R1 and R2
      if t < time
      Recovery_Phase1n2_real_time(i_MC,t) = Recovery_Phase1_real_time(i_MC,t);
      elseif t == time
      Recovery_Phase1n2_real_time(i_MC,t) = Recovery_Phase2_real_time(i_MC);
      else
      Recovery_Phase1n2_real_time(i_MC,t) = Recovery_Phase1_real_EndTime(i_MC) +
dE_inf_Phase2_prior(i_MC).*(1-exp(-(t-time+1)/Tau_Phase2_prior(i_MC)));
      end
      Recovery_Phase1n2_real_time_ShiftTime(i_MC,t,time) =
Recovery_Phase1n2_real_time(i_MC,t);
      Recovery_Phase1n2_real_time_ShiftTime(i_MC,t,FieldLifeTime+1) =
Recovery_Phase1_real_time(i_MC,t);

      if t == 1 %computation of the flow rate at the first year
      Rate_Phase1n2_real_time_ShiftTime(i_MC,t,time) =
OOIP_prior(i_MC).*Recovery_Phase1n2_real_time_ShiftTime(i_MC,t,time);
      Rate_Phase1n2_real_time_ShiftTime(i_MC,t,FieldLifeTime+1) =
OOIP_prior(i_MC).*Recovery_Phase1n2_real_time_ShiftTime(i_MC,t,FieldLifeTime+1);
      else %computation of the flow rate at the following years
```

```
      Rate_Phase1n2_real_time_ShiftTime(i_MC,t,time) =
OOIP_prior(i_MC).*(Recovery_Phase1n2_real_time_ShiftTime(i_MC,t,time)-
Recovery_Phase1n2_real_time_ShiftTime(i_MC,t-1,time));
      Rate_Phase1n2_real_time_ShiftTime(i_MC,t,FieldLifeTime+1) =
OOIP_prior(i_MC).*(Recovery_Phase1n2_real_time_ShiftTime(i_MC,t,FieldLifeTime+1)-
Recovery_Phase1n2_real_time_ShiftTime(i_MC,t-1,FieldLifeTime+1));
      end

    end
    end
end

% Computation of the measurement error and the measured oil production rates
ErrorSD_Matrix = ErrorSD4Rate_Pct.*Rate_Phase1n2_real_time_ShiftTime;
ObsRate_Phase1n2_real_time_ShiftTime =
normrnd(Rate_Phase1n2_real_time_ShiftTime,ErrorSD_Matrix);


% 3. Calculation of Cashflow and NPV
% This section computes te cashflow and the NPV used in the example problem

N_MC = size(ObsRate_Phase1n2_real_time_ShiftTime,1);
FieldLifeTime = size(ObsRate_Phase1n2_real_time_ShiftTime,2);
CashFlow = zeros(N_MC,FieldLifeTime);
DisCashFlow = zeros(N_MC,FieldLifeTime);
DisCashFlow_ShiftTime = zeros(N_MC,FieldLifeTime,FieldLifeTime+1);
NPV_reals = zeros(N_MC,1);
NPVtable_LTPhase1_LTPhase2_real = zeros(FieldLifeTime+1,FieldLifeTime+1,N_MC);

% Values of Economic Parameters
OilPrice = 50;
Capex_Phase1 = 50;
Capex_Phase2After1 = 40;
Capex_Phase2No1 = 75;
Opex_Phase1 = 20;
Opex_Phase2 = 30;
DisRate = 0.12;

for t = 1: FieldLifeTime
    for time = 1: FieldLifeTime+1
        if t == time
            if time == 1
                CashFlow(:,t) = OilPrice.*ObsRate_Phase1n2_real_time_ShiftTime(:,t,time) -
Opex_Phase2 - Capex_Phase2No1;
                DisCashFlow(:,t) = CashFlow(:,t)/((1+DisRate).^(t));
            else
               CashFlow(:,t) = OilPrice.*ObsRate_Phase1n2_real_time_ShiftTime(:,t,time) -
Opex_Phase2 - Capex_Phase2After1;
               DisCashFlow(:,t) = CashFlow(:,t)/((1+DisRate).^(t));
            end
        elseif t < time
            if t == 1
                CashFlow(:,t) = OilPrice.*ObsRate_Phase1n2_real_time_ShiftTime(:,t,time) -
Opex_Phase1 - Capex_Phase1;
                DisCashFlow(:,t) = CashFlow(:,t)/((1+DisRate).^(t));
            else
                CashFlow(:,t) = OilPrice.*ObsRate_Phase1n2_real_time_ShiftTime(:,t,time) -
Opex_Phase1;
```

```
                DisCashFlow(:,t) = CashFlow(:,t)/((1+DisRate).^(t));
            end
        else
            CashFlow(:,t) = OilPrice.*ObsRate_Phase1n2_real_time_ShiftTime(:,t,time) -
Opex_Phase2;
            DisCashFlow(:,t) = CashFlow(:,t)/((1+DisRate).^(t));
        end

        DisCashFlow_ShiftTime(:,t,time) = DisCashFlow(:,t);

        if time == 1
                NPV_reals(:) = sum(DisCashFlow_ShiftTime(:,1:t,time),2);
                NPVtable_LTPhase1_LTPhase2_real(time,t+1,:) = NPV_reals(:);
        elseif time == FieldLifeTime+1
                NPV_reals(:) = sum(DisCashFlow_ShiftTime(:,1:FieldLifeTime,time),2);
                NPVtable_LTPhase1_LTPhase2_real(time,1,:) = NPV_reals(:);
        else
            if t == time
                NPV_reals(:) = sum(DisCashFlow_ShiftTime(:,1:time,time),2);
                NPVtable_LTPhase1_LTPhase2_real(time,2,:) = NPV_reals(:);
            elseif t < time
                NPV_reals(:) = sum(DisCashFlow_ShiftTime(:,1:time-1,time),2);
                NPVtable_LTPhase1_LTPhase2_real(time,1,:) = NPV_reals(:);
            else
                NPV_reals(:) = sum(DisCashFlow_ShiftTime(:,1:t,time),2);
                NPVtable_LTPhase1_LTPhase2_real(time,t-time+2,:) = NPV_reals(:);
            end
        end
    end
end


% 4. Determination of DWOI, VOWI, DWPI, and VOPI
% This section determines the Decision Without Information and Decision With Perfect
Information
% The corresponding Value Without Information and Value of Perfect Information are estimated

N_MC = size(ObsRate_Phase1n2_real_time_ShiftTime,1);
FieldLifeTime = size(ObsRate_Phase1n2_real_time_ShiftTime,2);
N_Phase1n2LifeTime = sum(1:FieldLifeTime+1);
NPVvector_1real = zeros(N_Phase1n2LifeTime,3);
NPVmatrix_reals = zeros(N_Phase1n2LifeTime,N_MC);
Sum_NPVvector_1real = zeros(N_Phase1n2LifeTime,1);
meanNPVvector = zeros(N_Phase1n2LifeTime,3);
NPV_element_vector = zeros(FieldLifeTime+1,FieldLifeTime+1);
Phase1n2LifeTimeTable = zeros(N_Phase1n2LifeTime,2);

VWPI_real = zeros(N_MC,1);
DWPI_Phase1LifeTime_real = zeros(N_MC,1);
DWPI_Phase2LifeTime_real = zeros(N_MC,1);

for t = 1: FieldLifeTime
    for time = 1: FieldLifeTime+1
        NPV_element_vector(time,t+1) = t;
        if time > 1
            NPV_element_vector(time,FieldLifeTime-time+3:FieldLifeTime+1) = FieldLifeTime+1;
        else
            NPV_element_vector(time,t+1) = t;
```

```
        end
    end
end

[row_NPV_element,col_NPV_element]=find(NPV_element_vector(:,:) < FieldLifeTime+1);

for k = 1: N_Phase1n2LifeTime
    NPVvector_1real(k,1) = col_NPV_element(k)-1;
    NPVvector_1real(k,2) = row_NPV_element(k)-1;
    meanNPVvector(k,1) = col_NPV_element(k)-1;
    meanNPVvector(k,2) = row_NPV_element(k)-1;
end

for i_MC = 1: N_MC
    for k = 1: N_Phase1n2LifeTime
        NPVvector_1real(k,3) =
NPVtable_LTPhase1_LTPhase2_real(col_NPV_element(k),row_NPV_element(k),i_MC);
        NPVmatrix_reals(k,i_MC) = NPVvector_1real(k,3);
        [VWPI_real(i_MC),DWPI_idx] = max(NPVvector_1real(:,3));
        DWPI_Phase1LifeTime_real(i_MC) = NPVvector_1real(DWPI_idx,1);
        DWPI_Phase2LifeTime_real(i_MC) = NPVvector_1real(DWPI_idx,2);
    end
end

meanNPV = mean(NPVmatrix_reals,2);

for k = 1: N_Phase1n2LifeTime
    meanNPVvector(k,3) = meanNPV(k,1);
end

[EVWOI,DWOI_idx] = max(meanNPVvector(:,3));
DWOI_Phase1LifeTime = col_NPV_element(DWOI_idx)-1;
DWOI_Phase2LifeTime = row_NPV_element(DWOI_idx)-1;
DWOI_LifeTime = DWOI_Phase1LifeTime + DWOI_Phase2LifeTime;
DWPI_LifeTime_real = DWPI_Phase1LifeTime_real + DWPI_Phase2LifeTime_real;
Phase1n2LifeTimeTable(:,1) = meanNPVvector(:,1);
Phase1n2LifeTimeTable(:,2) = meanNPVvector(:,2);
EVWPI = mean(VWPI_real);
VOPI = EVWPI - EVWOI;

%  5. Path Table Generation (Modified LSM algorithm)
%  This section determines the optimal stopping time given the switch time at every year
%  Regression Analysis is applied
%  The NPV corresponding to the optimal stopping time is recorded into the PathTable

N_MC = size(ObsRate_Phase1n2_real_time_ShiftTime,1);
FieldLifeTime = size(ObsRate_Phase1n2_real_time_ShiftTime,2);
SRDM_Shift_Stop_OptNPV_real = zeros(N_MC,1);

PathTable = zeros(N_MC,FieldLifeTime+1);
PathTable(:,end) = NPVtable_LTPhase1_LTPhase2_real(FieldLifeTime+1,1,:);
Phase2_StopTime_real_Phase1LifeTime = zeros(N_MC,FieldLifeTime+1);

for k_Shift_time = 1: FieldLifeTime
        NPV_2Phases_matrix = zeros(N_MC,k_Shift_time+1);
    for k_Stop_time = 1: k_Shift_time+1
        NPV_2Phases_matrix(:,k_Stop_time) = NPVtable_LTPhase1_LTPhase2_real(FieldLifeTime-
k_Shift_time+1,k_Stop_time,:);
```

```
    end
    for k_StopTime = 1: k_Shift_time
        if k_StopTime == 1
            x_data_stop_SRDM = ObsRate_Phase1n2_real_time_ShiftTime(:,1:FieldLifeTime-
k_StopTime,FieldLifeTime-k_Shift_time+1);
            X_stop_SRDM = [ones(N_MC,1) x_data_stop_SRDM];
            y_data_stop_matrix = [NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+1)
NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+2)];
            [Stop_value,Stop_idx] =
max([X_stop_SRDM*regress(NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+1),X_stop_SRDM)
X_stop_SRDM*regress(NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+2),X_stop_SRDM)],[],2);
            for i_MC = 1: N_MC
                SRDM_Shift_Stop_OptNPV_real(i_MC,1) =
y_data_stop_matrix(i_MC,Stop_idx(i_MC));
            end
        else
            x_data_stop_SRDM = ObsRate_Phase1n2_real_time_ShiftTime(:,1:FieldLifeTime-
k_StopTime,FieldLifeTime-k_Shift_time+1);
            X_stop_SRDM = [ones(N_MC,1) x_data_stop_SRDM];
            y_data_stop_matrix = [NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+1)
SRDM_Shift_Stop_OptNPV_real(:,1)];
            [Stop_value,Stop_idx] =
max([X_stop_SRDM*regress(NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+1),X_stop_SRDM)
X_stop_SRDM*regress(SRDM_Shift_Stop_OptNPV_real(:,1),X_stop_SRDM)],[],2);
            for i_MC = 1: N_MC
                SRDM_Shift_Stop_OptNPV_real(i_MC,1) =
y_data_stop_matrix(i_MC,Stop_idx(i_MC));
            end
        end
    end
    for i_MC = 1: N_MC
        PathTable(i_MC,FieldLifeTime-k_Shift_time+1) = SRDM_Shift_Stop_OptNPV_real(i_MC,1);
        for k_Stop_time = 1: k_Shift_time+1
            if SRDM_Shift_Stop_OptNPV_real(i_MC,1) == NPV_2Phases_matrix(i_MC,k_Stop_time)
                Phase2_StopTime_real_Phase1LifeTime(i_MC,FieldLifeTime-k_Shift_time+1) =
k_Stop_time-1;
            end
        end
    end
end

% 6. SRDM Approach (Modified LSM algorithm)
% This section determines the optimal decision policy based on SRDM approach

FieldLifeTime = size(ObsRate_Phase1n2_real_time_ShiftTime,2);
y_data_continuation = zeros(N_MC,1);
y_data_shift = zeros(N_MC,1);
y_reg_continuation = zeros(N_MC,1);
y_reg_shift = zeros(N_MC,1);
x_data = ObsRate_Phase1n2_real_time_ShiftTime(:,:,FieldLifeTime+1);

ValueTable = zeros(N_MC,FieldLifeTime+1);
ValueTable(:,end) = PathTable(:,end);
SRDM_DecisionTable = zeros(N_MC,FieldLifeTime+1);
SRDM_DecisionTable(:,end) = 1;

for k_Shift_time = 1:FieldLifeTime
    X = [ones(N_MC,1) x_data(:,1:FieldLifeTime-k_Shift_time)];
```

```
    y_data_shift(:) = PathTable(:,FieldLifeTime-k_Shift_time+1);
    coef_shift = regress(y_data_shift,X);
    y_reg_shift(:) = X*coef_shift;
    y_data_continuation(:) = ValueTable(:,FieldLifeTime-k_Shift_time+2);
    coef_continuation = regress(y_data_continuation,X);
    y_reg_continuation(:) = X*coef_continuation;
    ENPVTable_Comparison = [y_reg_shift, y_reg_continuation];
    NPV_SRDM = [y_data_shift, y_data_continuation];
    [ENPV_SRDM,ENPV_SRDM_idx] = max(ENPVTable_Comparison,[],2);

    for i_MC = 1: N_MC
        ValueTable(i_MC,FieldLifeTime-k_Shift_time+1) = NPV_SRDM(i_MC,ENPV_SRDM_idx(i_MC));
        if y_reg_continuation(i_MC) > y_reg_shift(i_MC)
            SRDM_DecisionTable(i_MC,FieldLifeTime-k_Shift_time+1) = 0;
        else
            SRDM_DecisionTable(i_MC,FieldLifeTime-k_Shift_time+1) = 1;
        end
    end
end

ShiftValue = PathTable(:,1);
ExpShiftValue = mean(ShiftValue);

ContinuationValue = ValueTable(:,1);
ExpContinuationValue = mean(ContinuationValue);
EVWII = ExpContinuationValue;
VOI = EVWII - EVWOI;
VOI_over_EVWOI_pct = VOI/EVWOI*100;

[~,DWI_Phase1LifeTime_idx] = max(SRDM_DecisionTable,[],2);
DWI_Phase1LifeTime = DWI_Phase1LifeTime_idx-1;

DWI_Phase2LifeTime = zeros(N_MC,1);
for i_MC = 1:N_MC
    Phase1LifeTime_idx_real = DWI_Phase1LifeTime_idx(i_MC,1);
    DWI_Phase2LifeTime(i_MC,1) =
Phase2_StopTime_real_Phase1LifeTime(i_MC,Phase1LifeTime_idx_real);
end

DWI_LifeTime = DWI_Phase1LifeTime+DWI_Phase2LifeTime;
```

Appendix B1.2: CLRM Approach

This code determines the optimal decision policy by using CLRM approach. To run this algorithm, sections 1, 2, 3, and 4 from Modified LSM Algorithm (Appendix B1.1) have to be run first.

```
N_MC = size(ObsRate_Phase1n2_real_time_ShiftTime,1);
FieldLifeTime = size(ObsRate_Phase1n2_real_time_ShiftTime,2);
N_Phase1n2LifeTime = sum(1:FieldLifeTime+1);
N_LifeTimeComb = N_Phase1n2LifeTime;
Times = 1:1:FieldLifeTime;
CLRM_OptNPV_real = zeros(N_MC,1);
NPV_Tab_matrix = transpose(NPVmatrix_reals);
Prior_ENPV = meanNPVvector(:,3);
[max_Prior_ENPV, max_Prior_ENPV_idx] = max(meanNPVvector(:,3),[],1);
x_data_switch_CLRM = ObsRate_Phase1n2_real_time_ShiftTime(:,:,FieldLifeTime+1);

CLRM_PhaseLifes_Opt_reals = zeros(N_MC,2);

if max_Prior_ENPV_idx <= Times(FieldLifeTime)+1 %Finding if Switch Time is optimal at Y1
    Shift_time = 1;
    t_dataPhase2 = 1;
    while t_dataPhase2 <= Times(FieldLifeTime-Shift_time+1)
        if t_dataPhase2 == 1 && max_Prior_ENPV_idx == 1 %Switch and Stop at Y1 are optimal
            for i_MC = 1: N_MC
                CLRM_OptNPV_real(i_MC,1) = max_Prior_ENPV;
            end
            break; %The whole loop will break here
        else %Given Switch at Y1 and Stop is not at Y1
            x_data_stop_CLRM = ObsRate_Phase1n2_real_time_ShiftTime(:,:,Shift_time);
            X_stop_CLRM = [ones(N_MC,1) x_data_stop_CLRM(:,1:t_dataPhase2)];
            y_data_stop_CLRM = NPV_Tab_matrix(:,t_dataPhase2+1:FieldLifeTime+1);
            y_reg_stop_CLRM = zeros(N_MC,FieldLifeTime-t_dataPhase2+1);
            for t = 1: FieldLifeTime-t_dataPhase2+1
                y_reg_stop_CLRM(:,t) =
X_stop_CLRM*regress(y_data_stop_CLRM(:,t),X_stop_CLRM);
            end
            [StopTime_reg,StopTime_idx] = max(y_reg_stop_CLRM,[],2);
        for i_MC = 1: N_MC
            if StopTime_reg(i_MC) == y_reg_stop_CLRM(i_MC,1) && CLRM_OptNPV_real(i_MC,1) == 0
                CLRM_OptNPV_real(i_MC,1) = y_data_stop_CLRM(i_MC,1);
                CLRM_PhaseLifes_Opt_reals(i_MC,1) = Shift_time-1;
                CLRM_PhaseLifes_Opt_reals(i_MC,2) = t_dataPhase2+StopTime_idx(i_MC)-1;
            else
                continue;
            end
        end
        end
        t_dataPhase2 = t_dataPhase2+1;
    end
else
% This part is to find the corresponding optimal stop time given Switch time is not at Y1
    for Shift_time = 2: FieldLifeTime %Finding and recording the optimal switch time before
determining the stop time
        N_LifeTimeComb = N_LifeTimeComb-Times(FieldLifeTime-Shift_time+2)-1;
        y_data_switch_CLRM = NPV_Tab_matrix(:,N_Phase1n2LifeTime-
```

```
N_LifeTimeComb+1:N_Phase1n2LifeTime);
        y_reg_switch_CLRM = zeros(N_MC,N_LifeTimeComb);
        ENPVTabComp = zeros(N_MC,N_LifeTimeComb);
        X_switch_CLRM = [ones(N_MC,1) x_data_switch_CLRM(:,1:Shift_time-1)];
        for i_LifeTimeComb = 1: N_LifeTimeComb
            coef_reg_CLRM = regress(y_data_switch_CLRM(:,i_LifeTimeComb),X_switch_CLRM);
            y_reg_switch_CLRM(:,i_LifeTimeComb) = X_switch_CLRM*coef_reg_CLRM;
            ENPVTabComp(:,i_LifeTimeComb) = y_reg_switch_CLRM(:,i_LifeTimeComb);
        end

    [CLRM_reg,CLRM_idx] = max(ENPVTabComp,[],2);

    if Shift_time == FieldLifeTime
        [SwitchDecision_CLRM,SwitchDecision_CLRM_idx] = find(CLRM_idx <= Times(FieldLifeTime-
Shift_time+2)+1);
        [ContinueTime_reg,ContinueTime_idx] = find(CLRM_idx == Times(FieldLifeTime-
Shift_time+2)+1);
        for i_path = 1: length(ContinueTime_reg)
            if CLRM_OptNPV_real(ContinueTime_reg(i_path),1) == 0
               CLRM_OptNPV_real(ContinueTime_reg(i_path),1) =
y_data_switch_CLRM(ContinueTime_reg(i_path),end);
               CLRM_PhaseLifes_Opt_reals(ContinueTime_reg(i_path),1) = FieldLifeTime;
               CLRM_PhaseLifes_Opt_reals(ContinueTime_reg(i_path),2) = 0;
            else
                continue;
            end
        end
    else
        [SwitchDecision_CLRM,SwitchDecision_CLRM_idx] = find(CLRM_idx <= Times(FieldLifeTime-
Shift_time+2));
    end

    x_data_stop_CLRM = ObsRate_Phase1n2_real_time_ShiftTime(:,:,Shift_time);
    t_dataPhase2 = 1;
    while t_dataPhase2 <= Times(FieldLifeTime-Shift_time+1)+1
        X_stop_CLRM = [ones(N_MC,1) x_data_stop_CLRM(:,1:t_dataPhase2+Shift_time-2)];
        y_data_stop_CLRM = y_data_switch_CLRM(:, t_dataPhase2:Times(FieldLifeTime-
Shift_time+1)+1);
        y_reg_stop_CLRM = zeros(N_MC,Times(FieldLifeTime-Shift_time+1)-t_dataPhase2+2);
        for t = 1: Times(FieldLifeTime-Shift_time+1)-t_dataPhase2+2
            y_reg_stop_CLRM(:,t) = X_stop_CLRM*regress(y_data_stop_CLRM(:,t),X_stop_CLRM);
        end
            [StopTime_reg,StopTime_idx] = max(y_reg_stop_CLRM,[],2);
        for i_path = 1: length(SwitchDecision_CLRM)
            if StopTime_reg(SwitchDecision_CLRM(i_path)) ==
y_reg_stop_CLRM(SwitchDecision_CLRM(i_path),1) &&
CLRM_OptNPV_real(SwitchDecision_CLRM(i_path),1) == 0
                CLRM_OptNPV_real(SwitchDecision_CLRM(i_path),1) =
y_data_stop_CLRM(SwitchDecision_CLRM(i_path),1);
                CLRM_PhaseLifes_Opt_reals(SwitchDecision_CLRM(i_path),1) = Shift_time-1;
                CLRM_PhaseLifes_Opt_reals(SwitchDecision_CLRM(i_path),2) =
t_dataPhase2+StopTime_idx(SwitchDecision_CLRM(i_path))-2;
            else
                continue;
            end
        end
            t_dataPhase2 = t_dataPhase2+1;
    end
```

```
    end
end

DWI_CLRM_Phase1LifeTime = CLRM_PhaseLifes_Opt_reals(:,1);
DWI_CLRM_Phase2LifeTime = CLRM_PhaseLifes_Opt_reals(:,2);
DWI_CLRM_LifeTime = DWI_CLRM_Phase1LifeTime + DWI_CLRM_Phase2LifeTime;

EVWI_CLRM = mean(CLRM_OptNPV_real);
VOI_CLRM = EVWI_CLRM-EVWOI;
```

Appendix B2

Appendix B2.1: Regression Function Equation (13)

This code consists of the modified versions of the sections 5 and 6 of Modified LSM Algorithm (Appendix B1.1) and CLRM (Appendix B1.2). With respect to this, the modification is done on the regression analysis. Thus, sections 1 to 4 of Modified LSM Algorithm (Appendix B1.1) have to be run first before executing this code. This code corresponds to the Regression Function of Equation (13) in the report.

```
% 1. Path Table Generation

N_MC = size(ObsRate_Phase1n2_real_time_ShiftTime,1);
FieldLifeTime = size(ObsRate_Phase1n2_real_time_ShiftTime,2);
SRDM_Shift_Stop_OptNPV_real = zeros(N_MC,1);

PathTable = zeros(N_MC,FieldLifeTime+1);
PathTable(:,end) = NPVtable_LTPhase1_LTPhase2_real(FieldLifeTime+1,1,:);
Phase2_StopTime_real_Phase1LifeTime = zeros(N_MC,FieldLifeTime+1);

for k_Shift_time = 1: FieldLifeTime
        NPV_2Phases_matrix = zeros(N_MC,k_Shift_time+1);
    for k_Stop_time = 1: k_Shift_time+1
        NPV_2Phases_matrix(:,k_Stop_time) = NPVtable_LTPhase1_LTPhase2_real(FieldLifeTime-
k_Shift_time+1,k_Stop_time,:);
    end
    for k_StopTime = 1: k_Shift_time
        if k_StopTime == 1
            x_data_stop_SRDM = ObsRate_Phase1n2_real_time_ShiftTime(:,1:FieldLifeTime-
k_StopTime,FieldLifeTime-k_Shift_time+1);
            x_data_sqrt_stop_SRDM = sqrt(x_data_stop_SRDM(:,1:FieldLifeTime-k_StopTime-
1)).*sqrt(x_data_stop_SRDM(:,2:FieldLifeTime-k_StopTime));
            X_stop_SRDM = [ones(N_MC,1) x_data_stop_SRDM x_data_sqrt_stop_SRDM];
            y_data_stop_matrix = [NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+1)
NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+2)];
            [Stop_value,Stop_idx] =
max([X_stop_SRDM*regress(NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+1),X_stop_SRDM)
X_stop_SRDM*regress(NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+2),X_stop_SRDM)],[],2);
            for i_MC = 1: N_MC
                SRDM_Shift_Stop_OptNPV_real(i_MC,1) =
y_data_stop_matrix(i_MC,Stop_idx(i_MC));
            end
        else
            x_data_stop_SRDM = ObsRate_Phase1n2_real_time_ShiftTime(:,1:FieldLifeTime-
k_StopTime,FieldLifeTime-k_Shift_time+1);
            x_data_sqrt_stop_SRDM = sqrt(x_data_stop_SRDM(:,1:FieldLifeTime-k_StopTime-
1)).*sqrt(x_data_stop_SRDM(:,2:FieldLifeTime-k_StopTime));
            X_stop_SRDM = [ones(N_MC,1) x_data_stop_SRDM x_data_sqrt_stop_SRDM];
            y_data_stop_matrix = [NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+1)
SRDM_Shift_Stop_OptNPV_real(:,1)];
            [Stop_value,Stop_idx] =
max([X_stop_SRDM*regress(NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+1),X_stop_SRDM)
X_stop_SRDM*regress(SRDM_Shift_Stop_OptNPV_real(:,1),X_stop_SRDM)],[],2);
```

```
            for i_MC = 1: N_MC
                SRDM_Shift_Stop_OptNPV_real(i_MC,1) =
y_data_stop_matrix(i_MC,Stop_idx(i_MC));
            end
        end
    end
    for i_MC = 1: N_MC
        PathTable(i_MC,FieldLifeTime-k_Shift_time+1) = SRDM_Shift_Stop_OptNPV_real(i_MC,1);
        for k_Stop_time = 1: k_Shift_time+1
            if SRDM_Shift_Stop_OptNPV_real(i_MC,1) == NPV_2Phases_matrix(i_MC,k_Stop_time)
                Phase2_StopTime_real_Phase1LifeTime(i_MC,FieldLifeTime-k_Shift_time+1) =
k_Stop_time-1;
            end
        end
    end
end

% 2. SRDM Approach

FieldLifeTime = size(ObsRate_Phase1n2_real_time_ShiftTime,2);
y_data_continuation = zeros(N_MC,1);
y_data_shift = zeros(N_MC,1);
y_reg_continuation = zeros(N_MC,1);
y_reg_shift = zeros(N_MC,1);
x_data = ObsRate_Phase1n2_real_time_ShiftTime(:,:,FieldLifeTime+1);

ValueTable = zeros(N_MC,FieldLifeTime+1);
ValueTable(:,end) = PathTable(:,end);
SRDM_DecisionTable = zeros(N_MC,FieldLifeTime+1);
SRDM_DecisionTable(:,end) = 1;

for k_Shift_time = 1:FieldLifeTime
    x_data_sqrt = sqrt(x_data(:,1:FieldLifeTime-k_Shift_time-
1)).*sqrt(x_data(:,2:FieldLifeTime-k_Shift_time));
    X = [ones(N_MC,1) x_data(:,1:FieldLifeTime-k_Shift_time) x_data_sqrt];
    y_data_shift(:) = PathTable(:,FieldLifeTime-k_Shift_time+1);
    coef_shift = regress(y_data_shift,X);
    y_reg_shift(:) = X*coef_shift;
    y_data_continuation(:) = ValueTable(:,FieldLifeTime-k_Shift_time+2);
    coef_continuation = regress(y_data_continuation,X);
    y_reg_continuation(:) = X*coef_continuation;
    ENPVTable_Comparison = [y_reg_shift, y_reg_continuation];
    NPV_SRDM = [y_data_shift, y_data_continuation];
    [ENPV_SRDM,ENPV_SRDM_idx] = max(ENPVTable_Comparison,[],2);

    for i_MC = 1: N_MC
        ValueTable(i_MC,FieldLifeTime-k_Shift_time+1) = NPV_SRDM(i_MC,ENPV_SRDM_idx(i_MC));
        if y_reg_continuation(i_MC) > y_reg_shift(i_MC)
            SRDM_DecisionTable(i_MC,FieldLifeTime-k_Shift_time+1) = 0;
        else
            SRDM_DecisionTable(i_MC,FieldLifeTime-k_Shift_time+1) = 1;
        end
    end
end

ShiftValue = PathTable(:,1);
ExpShiftValue = mean(ShiftValue);
```

```
ContinuationValue = ValueTable(:,1);
ExpContinuationValue = mean(ContinuationValue);
EVWII = ExpContinuationValue;
VOI = EVWII - EVWOI;
VOI_over_EVWOI_pct = VOI/EVWOI*100;

[~,DWI_Phase1LifeTime_idx] = max(SRDM_DecisionTable,[],2);
DWI_Phase1LifeTime = DWI_Phase1LifeTime_idx-1;

DWI_Phase2LifeTime = zeros(N_MC,1);
for i_MC = 1:N_MC
    Phase1LifeTime_idx_real = DWI_Phase1LifeTime_idx(i_MC,1);
    DWI_Phase2LifeTime(i_MC,1) =
Phase2_StopTime_real_Phase1LifeTime(i_MC,Phase1LifeTime_idx_real);
end

DWI_LifeTime = DWI_Phase1LifeTime+DWI_Phase2LifeTime;

% 3. CLRM Approach

N_MC = size(ObsRate_Phase1n2_real_time_ShiftTime,1);
FieldLifeTime = size(ObsRate_Phase1n2_real_time_ShiftTime,2);
N_Phase1n2LifeTime = sum(1:FieldLifeTime+1);
N_LifeTimeComb = N_Phase1n2LifeTime;
Times = 1:1:FieldLifeTime;
CLRM_OptNPV_real = zeros(N_MC,1);
NPV_Tab_matrix = transpose(NPVmatrix_reals);
Prior_ENPV = meanNPVvector(:,3);
[max_Prior_ENPV, max_Prior_ENPV_idx] = max(meanNPVvector(:,3),[],1);
x_data_switch_CLRM = ObsRate_Phase1n2_real_time_ShiftTime(:,:,FieldLifeTime+1);

CLRM_PhaseLifes_Opt_reals = zeros(N_MC,2);

if max_Prior_ENPV_idx <= Times(FieldLifeTime)+1 %Finding if Switch Time is optimal at Y1
    Shift_time = 1;
    t_dataPhase2 = 1;
    while t_dataPhase2 <= Times(FieldLifeTime-Shift_time+1)
        if t_dataPhase2 == 1 && max_Prior_ENPV_idx == 1 %Switch and Stop at Y1 are optimal
            for i_MC = 1: N_MC
                CLRM_OptNPV_real(i_MC,1) = max_Prior_ENPV;
            end
            break; %The whole loop will break here
        else %Given Switch at Y1 and Stop is not at Y1
            x_data_stop_CLRM = ObsRate_Phase1n2_real_time_ShiftTime(:,:,Shift_time);
            if t_dataPhase2 == 1
                x_data_sqrt_stop_CLRM = sqrt(x_data_stop_CLRM(:,1:t_dataPhase2));
                X_stop_CLRM = [ones(N_MC,1) x_data_stop_CLRM(:,1:t_dataPhase2)];
            else
                x_data_sqrt_stop_CLRM = sqrt(x_data_stop_CLRM(:,1:t_dataPhase2-
1)).*sqrt(x_data_stop_CLRM(:,2:t_dataPhase2));
                X_stop_CLRM = [ones(N_MC,1) x_data_stop_CLRM(:,1:t_dataPhase2)
x_data_sqrt_stop_CLRM];
            end
            y_data_stop_CLRM = NPV_Tab_matrix(:,t_dataPhase2+1:FieldLifeTime+1);
            y_reg_stop_CLRM = zeros(N_MC,FieldLifeTime-t_dataPhase2+1);
            for t = 1: FieldLifeTime-t_dataPhase2+1
                y_reg_stop_CLRM(:,t) =
X_stop_CLRM*regress(y_data_stop_CLRM(:,t),X_stop_CLRM);
```

```
            end
            [StopTime_reg,StopTime_idx] = max(y_reg_stop_CLRM,[],2);
        for i_MC = 1: N_MC
            if StopTime_reg(i_MC) == y_reg_stop_CLRM(i_MC,1) && CLRM_OptNPV_real(i_MC,1) == 0
                CLRM_OptNPV_real(i_MC,1) = y_data_stop_CLRM(i_MC,1);
                CLRM_PhaseLifes_Opt_reals(i_MC,1) = Shift_time-1;
                CLRM_PhaseLifes_Opt_reals(i_MC,2) = t_dataPhase2+StopTime_idx(i_MC)-1;
            else
                continue;
            end
        end
        end
        t_dataPhase2 = t_dataPhase2+1;
    end
else
% This part is to find the corresponding optimal stop time given Switch time is not at Y1
    for Shift_time = 2: FieldLifeTime %Finding and recording the optimal switch time before
determining the stop time
        N_LifeTimeComb = N_LifeTimeComb-Times(FieldLifeTime-Shift_time+2)-1;
        y_data_switch_CLRM = NPV_Tab_matrix(:,N_Phase1n2LifeTime-
N_LifeTimeComb+1:N_Phase1n2LifeTime);
        y_reg_switch_CLRM = zeros(N_MC,N_LifeTimeComb);
        ENPVTabComp = zeros(N_MC,N_LifeTimeComb);
        if Shift_time == 2
            x_data_sqrt_switch_CLRM = sqrt(x_data_switch_CLRM(:,1:Shift_time-1));
            X_switch_CLRM = [ones(N_MC,1) x_data_switch_CLRM(:,1:Shift_time-1)];
        else
            x_data_sqrt_switch_CLRM = sqrt(x_data_switch_CLRM(:,1:Shift_time-
2)).*sqrt(x_data_switch_CLRM(:,2:Shift_time-1));
            X_switch_CLRM = [ones(N_MC,1) x_data_switch_CLRM(:,1:Shift_time-1)
x_data_sqrt_switch_CLRM];
        end
        for i_LifeTimeComb = 1: N_LifeTimeComb
            coef_reg_CLRM = regress(y_data_switch_CLRM(:,i_LifeTimeComb),X_switch_CLRM);
            y_reg_switch_CLRM(:,i_LifeTimeComb) = X_switch_CLRM*coef_reg_CLRM;
            ENPVTabComp(:,i_LifeTimeComb) = y_reg_switch_CLRM(:,i_LifeTimeComb);
        end

    [CLRM_reg,CLRM_idx] = max(ENPVTabComp,[],2);

    if Shift_time == FieldLifeTime
        [SwitchDecision_CLRM,SwitchDecision_CLRM_idx] = find(CLRM_idx <= Times(FieldLifeTime-
Shift_time+2)+1);
        [ContinueTime_reg,ContinueTime_idx] = find(CLRM_idx == Times(FieldLifeTime-
Shift_time+2)+1);
        for i_path = 1: length(ContinueTime_reg)
            if CLRM_OptNPV_real(ContinueTime_reg(i_path),1) == 0
               CLRM_OptNPV_real(ContinueTime_reg(i_path),1) =
y_data_switch_CLRM(ContinueTime_reg(i_path),end);
               CLRM_PhaseLifes_Opt_reals(ContinueTime_reg(i_path),1) = FieldLifeTime;
               CLRM_PhaseLifes_Opt_reals(ContinueTime_reg(i_path),2) = 0;
            else
                continue;
            end
        end
    else
        [SwitchDecision_CLRM,SwitchDecision_CLRM_idx] = find(CLRM_idx <= Times(FieldLifeTime-
Shift_time+2));
```

```
    end

    x_data_stop_CLRM = ObsRate_Phase1n2_real_time_ShiftTime(:,:,Shift_time);
    t_dataPhase2 = 1;
    while t_dataPhase2 <= Times(FieldLifeTime-Shift_time+1)+1
        if t_dataPhase2 == 1
            x_data_sqrt_stop_CLRM = sqrt(x_data_stop_CLRM(:,1:t_dataPhase2));
            X_stop_CLRM = [ones(N_MC,1) x_data_stop_CLRM(:,1:t_dataPhase2+Shift_time-2)];
        else
            x_data_sqrt_stop_CLRM = sqrt(x_data_stop_CLRM(:,1:t_dataPhase2-
1)).*sqrt(x_data_stop_CLRM(:,2:t_dataPhase2));
            X_stop_CLRM = [ones(N_MC,1) x_data_stop_CLRM(:,1:t_dataPhase2+Shift_time-2)
x_data_sqrt_stop_CLRM];
        end
        y_data_stop_CLRM = y_data_switch_CLRM(:, t_dataPhase2:Times(FieldLifeTime-
Shift_time+1)+1);
        y_reg_stop_CLRM = zeros(N_MC,Times(FieldLifeTime-Shift_time+1)-t_dataPhase2+2);
        for t = 1: Times(FieldLifeTime-Shift_time+1)-t_dataPhase2+2
            y_reg_stop_CLRM(:,t) = X_stop_CLRM*regress(y_data_stop_CLRM(:,t),X_stop_CLRM);
        end
            [StopTime_reg,StopTime_idx] = max(y_reg_stop_CLRM,[],2);
        for i_path = 1: length(SwitchDecision_CLRM)
            if StopTime_reg(SwitchDecision_CLRM(i_path)) ==
y_reg_stop_CLRM(SwitchDecision_CLRM(i_path),1) &&
CLRM_OptNPV_real(SwitchDecision_CLRM(i_path),1) == 0
                CLRM_OptNPV_real(SwitchDecision_CLRM(i_path),1) =
y_data_stop_CLRM(SwitchDecision_CLRM(i_path),1);
                CLRM_PhaseLifes_Opt_reals(SwitchDecision_CLRM(i_path),1) = Shift_time-1;
                CLRM_PhaseLifes_Opt_reals(SwitchDecision_CLRM(i_path),2) =
t_dataPhase2+StopTime_idx(SwitchDecision_CLRM(i_path))-2;
            else
                continue;
            end
        end
            t_dataPhase2 = t_dataPhase2+1;
    end
    end
end

DWI_CLRM_Phase1LifeTime = CLRM_PhaseLifes_Opt_reals(:,1);
DWI_CLRM_Phase2LifeTime = CLRM_PhaseLifes_Opt_reals(:,2);
DWI_CLRM_LifeTime = DWI_CLRM_Phase1LifeTime + DWI_CLRM_Phase2LifeTime;

EVWI_CLRM = mean(CLRM_OptNPV_real);
VOI_CLRM = EVWI_CLRM-EVWOI;
```

Appendix B2.2: Regression Function Equation (14)

This code consists of the modified version of the sections 5 and 6 of Modified LSM Algorithm (Appendix B1.1) and CLRM (Appendix B1.2). With respect to this, the modification is done on the regression analysis. Thus, sections 1 to 4 of Modified LSM Algorithm (Appendix B1.1) have to be run first before executing this code. This code corresponds to the Regression Function of Equation (14) in the report.

```
% 1. Path Table Generation

N_MC = size(ObsRate_Phase1n2_real_time_ShiftTime,1);
FieldLifeTime = size(ObsRate_Phase1n2_real_time_ShiftTime,2);
SRDM_Shift_Stop_OptNPV_real = zeros(N_MC,1);

PathTable = zeros(N_MC,FieldLifeTime+1);
PathTable(:,end) = NPVtable_LTPhase1_LTPhase2_real(FieldLifeTime+1,1,:);
Phase2_StopTime_real_Phase1LifeTime = zeros(N_MC,FieldLifeTime+1);

for k_Shift_time = 1: FieldLifeTime
        NPV_2Phases_matrix = zeros(N_MC,k_Shift_time+1);
    for k_Stop_time = 1: k_Shift_time+1
        NPV_2Phases_matrix(:,k_Stop_time) = NPVtable_LTPhase1_LTPhase2_real(FieldLifeTime-
k_Shift_time+1,k_Stop_time,:);
    end
    for k_StopTime = 1: k_Shift_time
        if k_StopTime == 1
            x_data_stop_SRDM = ObsRate_Phase1n2_real_time_ShiftTime(:,1:FieldLifeTime-
k_StopTime,FieldLifeTime-k_Shift_time+1);
            x_data_sqrt_stop_SRDM = sqrt(x_data_stop_SRDM(:,1:FieldLifeTime-k_StopTime-
2)).*sqrt(x_data_stop_SRDM(:,2:FieldLifeTime-k_StopTime-
1)).*sqrt(x_data_stop_SRDM(:,3:FieldLifeTime-k_StopTime));
            X_stop_SRDM = [ones(N_MC,1) x_data_stop_SRDM x_data_sqrt_stop_SRDM];
            y_data_stop_matrix = [NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+1)
NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+2)];
            [Stop_value,Stop_idx] =
max([X_stop_SRDM*regress(NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+1),X_stop_SRDM)
X_stop_SRDM*regress(NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+2),X_stop_SRDM)],[],2);
            for i_MC = 1: N_MC
                SRDM_Shift_Stop_OptNPV_real(i_MC,1) =
y_data_stop_matrix(i_MC,Stop_idx(i_MC));
            end
        else
            x_data_stop_SRDM = ObsRate_Phase1n2_real_time_ShiftTime(:,1:FieldLifeTime-
k_StopTime,FieldLifeTime-k_Shift_time+1);
            x_data_sqrt_stop_SRDM = sqrt(x_data_stop_SRDM(:,1:FieldLifeTime-k_StopTime-
2)).*sqrt(x_data_stop_SRDM(:,2:FieldLifeTime-k_StopTime-
1)).*sqrt(x_data_stop_SRDM(:,3:FieldLifeTime-k_StopTime));
            X_stop_SRDM = [ones(N_MC,1) x_data_stop_SRDM x_data_sqrt_stop_SRDM];
            y_data_stop_matrix = [NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+1)
SRDM_Shift_Stop_OptNPV_real(:,1)];
            [Stop_value,Stop_idx] =
max([X_stop_SRDM*regress(NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+1),X_stop_SRDM)
X_stop_SRDM*regress(SRDM_Shift_Stop_OptNPV_real(:,1),X_stop_SRDM)],[],2);
            for i_MC = 1: N_MC
                SRDM_Shift_Stop_OptNPV_real(i_MC,1) =
```

```
y_data_stop_matrix(i_MC,Stop_idx(i_MC));
            end
        end
    end
    for i_MC = 1: N_MC
        PathTable(i_MC,FieldLifeTime-k_Shift_time+1) = SRDM_Shift_Stop_OptNPV_real(i_MC,1);
        for k_Stop_time = 1: k_Shift_time+1
            if SRDM_Shift_Stop_OptNPV_real(i_MC,1) == NPV_2Phases_matrix(i_MC,k_Stop_time)
                Phase2_StopTime_real_Phase1LifeTime(i_MC,FieldLifeTime-k_Shift_time+1) =
k_Stop_time-1;
            end
        end
    end
end

% 2. SRDM Approach

FieldLifeTime = size(ObsRate_Phase1n2_real_time_ShiftTime,2);
y_data_continuation = zeros(N_MC,1);
y_data_shift = zeros(N_MC,1);
y_reg_continuation = zeros(N_MC,1);
y_reg_shift = zeros(N_MC,1);
x_data = ObsRate_Phase1n2_real_time_ShiftTime(:,:,FieldLifeTime+1);

ValueTable = zeros(N_MC,FieldLifeTime+1);
ValueTable(:,end) = PathTable(:,end);
SRDM_DecisionTable = zeros(N_MC,FieldLifeTime+1);
SRDM_DecisionTable(:,end) = 1;

for k_Shift_time = 1:FieldLifeTime
    x_data_sqrt = sqrt(x_data(:,1:FieldLifeTime-k_Shift_time-
2)).*sqrt(x_data(:,2:FieldLifeTime-k_Shift_time-1)).*sqrt(x_data(:,3:FieldLifeTime-
k_Shift_time));
    X = [ones(N_MC,1) x_data(:,1:FieldLifeTime-k_Shift_time) x_data_sqrt];
    y_data_shift(:) = PathTable(:,FieldLifeTime-k_Shift_time+1);
    coef_shift = regress(y_data_shift,X);
    y_reg_shift(:) = X*coef_shift;
    y_data_continuation(:) = ValueTable(:,FieldLifeTime-k_Shift_time+2);
    coef_continuation = regress(y_data_continuation,X);
    y_reg_continuation(:) = X*coef_continuation;
    ENPVTable_Comparison = [y_reg_shift, y_reg_continuation];
    NPV_SRDM = [y_data_shift, y_data_continuation];
    [ENPV_SRDM,ENPV_SRDM_idx] = max(ENPVTable_Comparison,[],2);

    for i_MC = 1: N_MC
        ValueTable(i_MC,FieldLifeTime-k_Shift_time+1) = NPV_SRDM(i_MC,ENPV_SRDM_idx(i_MC));
        if y_reg_continuation(i_MC) > y_reg_shift(i_MC)
            SRDM_DecisionTable(i_MC,FieldLifeTime-k_Shift_time+1) = 0;
        else
            SRDM_DecisionTable(i_MC,FieldLifeTime-k_Shift_time+1) = 1;
        end
    end
end

ShiftValue = PathTable(:,1);
ExpShiftValue = mean(ShiftValue);

ContinuationValue = ValueTable(:,1);
```

```
ExpContinuationValue = mean(ContinuationValue);
EVWII = ExpContinuationValue;
VOI = EVWII - EVWOI;
VOI_over_EVWOI_pct = VOI/EVWOI*100;

[~,DWI_Phase1LifeTime_idx] = max(SRDM_DecisionTable,[],2);
DWI_Phase1LifeTime = DWI_Phase1LifeTime_idx-1;

DWI_Phase2LifeTime = zeros(N_MC,1);
for i_MC = 1:N_MC
    Phase1LifeTime_idx_real = DWI_Phase1LifeTime_idx(i_MC,1);
    DWI_Phase2LifeTime(i_MC,1) =
Phase2_StopTime_real_Phase1LifeTime(i_MC,Phase1LifeTime_idx_real);
end

DWI_LifeTime = DWI_Phase1LifeTime+DWI_Phase2LifeTime;

% 3. CLRM Approach

N_MC = size(ObsRate_Phase1n2_real_time_ShiftTime,1);
FieldLifeTime = size(ObsRate_Phase1n2_real_time_ShiftTime,2);
N_Phase1n2LifeTime = sum(1:FieldLifeTime+1);
N_LifeTimeComb = N_Phase1n2LifeTime;
Times = 1:1:FieldLifeTime;
CLRM_OptNPV_real = zeros(N_MC,1);
NPV_Tab_matrix = transpose(NPVmatrix_reals);
Prior_ENPV = meanNPVvector(:,3);
[max_Prior_ENPV, max_Prior_ENPV_idx] = max(meanNPVvector(:,3),[],1);
x_data_switch_CLRM = ObsRate_Phase1n2_real_time_ShiftTime(:,:,FieldLifeTime+1);

CLRM_PhaseLifes_Opt_reals = zeros(N_MC,2);

if max_Prior_ENPV_idx <= Times(FieldLifeTime)+1 %Finding if Switch Time is optimal at Y1
    Shift_time = 1;
    t_dataPhase2 = 1;
    while t_dataPhase2 <= Times(FieldLifeTime-Shift_time+1)
        if t_dataPhase2 == 1 && max_Prior_ENPV_idx == 1 %Switch and Stop at Y1 are optimal
            for i_MC = 1: N_MC
                CLRM_OptNPV_real(i_MC,1) = max_Prior_ENPV;
            end
            break; %The whole loop will break here
        else %Given Switch at Y1 and Stop is not at Y1
            x_data_stop_CLRM = ObsRate_Phase1n2_real_time_ShiftTime(:,:,Shift_time);
            if t_dataPhase2 == 1 && t_dataPhase2 == 2
                x_data_sqrt_stop_CLRM = sqrt(x_data_stop_CLRM(:,1:t_dataPhase2));
                X_stop_CLRM = [ones(N_MC,1) x_data_stop_CLRM(:,1:t_dataPhase2)];
            else
                x_data_sqrt_stop_CLRM = sqrt(x_data_stop_CLRM(:,1:t_dataPhase2-
2)).*sqrt(x_data_stop_CLRM(:,2:t_dataPhase2-1)).*sqrt(x_data_stop_CLRM(:,3:t_dataPhase2));
                X_stop_CLRM = [ones(N_MC,1) x_data_stop_CLRM(:,1:t_dataPhase2)
x_data_sqrt_stop_CLRM];
            end
            y_data_stop_CLRM = NPV_Tab_matrix(:,t_dataPhase2+1:FieldLifeTime+1);
            y_reg_stop_CLRM = zeros(N_MC,FieldLifeTime-t_dataPhase2+1);
            for t = 1: FieldLifeTime-t_dataPhase2+1
                y_reg_stop_CLRM(:,t) =
X_stop_CLRM*regress(y_data_stop_CLRM(:,t),X_stop_CLRM);
            end
```

```
            [StopTime_reg,StopTime_idx] = max(y_reg_stop_CLRM,[],2);
        for i_MC = 1: N_MC
            if StopTime_reg(i_MC) == y_reg_stop_CLRM(i_MC,1) && CLRM_OptNPV_real(i_MC,1) == 0
                CLRM_OptNPV_real(i_MC,1) = y_data_stop_CLRM(i_MC,1);
                CLRM_PhaseLifes_Opt_reals(i_MC,1) = Shift_time-1;
                CLRM_PhaseLifes_Opt_reals(i_MC,2) = t_dataPhase2+StopTime_idx(i_MC)-1;
            else
                continue;
            end
        end
        end
        t_dataPhase2 = t_dataPhase2+1;
    end
else
% This part is to find the corresponding optimal stop time given Switch time is not at Y1
    for Shift_time = 2: FieldLifeTime %Finding and recording the optimal switch time before
determining the stop time
        N_LifeTimeComb = N_LifeTimeComb-Times(FieldLifeTime-Shift_time+2)-1;
        y_data_switch_CLRM = NPV_Tab_matrix(:,N_Phase1n2LifeTime-
N_LifeTimeComb+1:N_Phase1n2LifeTime);
        y_reg_switch_CLRM = zeros(N_MC,N_LifeTimeComb);
        ENPVTabComp = zeros(N_MC,N_LifeTimeComb);
        if Shift_time == 2 && Shift_time == 3
            x_data_sqrt_switch_CLRM = sqrt(x_data_switch_CLRM(:,1:Shift_time-1));
            X_switch_CLRM = [ones(N_MC,1) x_data_switch_CLRM(:,1:Shift_time-1)];
        else
            x_data_sqrt_switch_CLRM = sqrt(x_data_switch_CLRM(:,1:Shift_time-
3)).*sqrt(x_data_switch_CLRM(:,2:Shift_time-2)).*sqrt(x_data_switch_CLRM(:,3:Shift_time-1));
            X_switch_CLRM = [ones(N_MC,1) x_data_switch_CLRM(:,1:Shift_time-1)
x_data_sqrt_switch_CLRM];
        end
        for i_LifeTimeComb = 1: N_LifeTimeComb
            coef_reg_CLRM = regress(y_data_switch_CLRM(:,i_LifeTimeComb),X_switch_CLRM);
            y_reg_switch_CLRM(:,i_LifeTimeComb) = X_switch_CLRM*coef_reg_CLRM;
            ENPVTabComp(:,i_LifeTimeComb) = y_reg_switch_CLRM(:,i_LifeTimeComb);
        end

    [CLRM_reg,CLRM_idx] = max(ENPVTabComp,[],2);

    if Shift_time == FieldLifeTime
        [SwitchDecision_CLRM,SwitchDecision_CLRM_idx] = find(CLRM_idx <= Times(FieldLifeTime-
Shift_time+2)+1);
        [ContinueTime_reg,ContinueTime_idx] = find(CLRM_idx == Times(FieldLifeTime-
Shift_time+2)+1);
        for i_path = 1: length(ContinueTime_reg)
            if CLRM_OptNPV_real(ContinueTime_reg(i_path),1) == 0
               CLRM_OptNPV_real(ContinueTime_reg(i_path),1) =
y_data_switch_CLRM(ContinueTime_reg(i_path),end);
               CLRM_PhaseLifes_Opt_reals(ContinueTime_reg(i_path),1) = FieldLifeTime;
               CLRM_PhaseLifes_Opt_reals(ContinueTime_reg(i_path),2) = 0;
            else
                continue;
            end
        end
    else
        [SwitchDecision_CLRM,SwitchDecision_CLRM_idx] = find(CLRM_idx <= Times(FieldLifeTime-
Shift_time+2));
    end
```

```
    x_data_stop_CLRM = ObsRate_Phase1n2_real_time_ShiftTime(:,:,Shift_time);
    t_dataPhase2 = 1;
    while t_dataPhase2 <= Times(FieldLifeTime-Shift_time+1)+1
        if t_dataPhase2 == 1 && t_dataPhase2 == 2
            x_data_sqrt_stop_CLRM = sqrt(x_data_stop_CLRM(:,1:t_dataPhase2));
            X_stop_CLRM = [ones(N_MC,1) x_data_stop_CLRM(:,1:t_dataPhase2+Shift_time-2)];
        else
            x_data_sqrt_stop_CLRM = sqrt(x_data_stop_CLRM(:,1:t_dataPhase2-
2)).*sqrt(x_data_stop_CLRM(:,2:t_dataPhase2-1)).*sqrt(x_data_stop_CLRM(:,3:t_dataPhase2));
            X_stop_CLRM = [ones(N_MC,1) x_data_stop_CLRM(:,1:t_dataPhase2+Shift_time-2)
x_data_sqrt_stop_CLRM];
        end
        y_data_stop_CLRM = y_data_switch_CLRM(:, t_dataPhase2:Times(FieldLifeTime-
Shift_time+1)+1);
        y_reg_stop_CLRM = zeros(N_MC,Times(FieldLifeTime-Shift_time+1)-t_dataPhase2+2);
        for t = 1: Times(FieldLifeTime-Shift_time+1)-t_dataPhase2+2
            y_reg_stop_CLRM(:,t) = X_stop_CLRM*regress(y_data_stop_CLRM(:,t),X_stop_CLRM);
        end
            [StopTime_reg,StopTime_idx] = max(y_reg_stop_CLRM,[],2);
        for i_path = 1: length(SwitchDecision_CLRM)
            if StopTime_reg(SwitchDecision_CLRM(i_path)) ==
y_reg_stop_CLRM(SwitchDecision_CLRM(i_path),1) &&
CLRM_OptNPV_real(SwitchDecision_CLRM(i_path),1) == 0
                CLRM_OptNPV_real(SwitchDecision_CLRM(i_path),1) =
y_data_stop_CLRM(SwitchDecision_CLRM(i_path),1);
                CLRM_PhaseLifes_Opt_reals(SwitchDecision_CLRM(i_path),1) = Shift_time-1;
                CLRM_PhaseLifes_Opt_reals(SwitchDecision_CLRM(i_path),2) =
t_dataPhase2+StopTime_idx(SwitchDecision_CLRM(i_path))-2;
            else
                continue;
            end
        end
            t_dataPhase2 = t_dataPhase2+1;
    end
    end
end

DWI_CLRM_Phase1LifeTime = CLRM_PhaseLifes_Opt_reals(:,1);
DWI_CLRM_Phase2LifeTime = CLRM_PhaseLifes_Opt_reals(:,2);
DWI_CLRM_LifeTime = DWI_CLRM_Phase1LifeTime + DWI_CLRM_Phase2LifeTime;

EVWI_CLRM = mean(CLRM_OptNPV_real);
VOI_CLRM = EVWI_CLRM-EVWOI;
```

Appendix B2.3: Regression Function Equation (15)

This code consists of the modified version of the sections 5 and 6 of Modified LSM Algorithm (Appendix B1.1) and CLRM (Appendix B1.2). With respect to this, the modification is done on the regression analysis. Thus, sections 1 to 4 of Modified LSM Algorithm (Appendix B1.1) have to be run first before executing this code. This code corresponds to the Regression Function of Equation (15) in the report.

```
% 1. Path Table Generation

N_MC = size(ObsRate_Phase1n2_real_time_ShiftTime,1);
FieldLifeTime = size(ObsRate_Phase1n2_real_time_ShiftTime,2);
SRDM_Shift_Stop_OptNPV_real = zeros(N_MC,1);

PathTable = zeros(N_MC,FieldLifeTime+1);
PathTable(:,end) = NPVtable_LTPhase1_LTPhase2_real(FieldLifeTime+1,1,:);
Phase2_StopTime_real_Phase1LifeTime = zeros(N_MC,FieldLifeTime+1);

for k_Shift_time = 1: FieldLifeTime
        NPV_2Phases_matrix = zeros(N_MC,k_Shift_time+1);
    for k_Stop_time = 1: k_Shift_time+1
        NPV_2Phases_matrix(:,k_Stop_time) = NPVtable_LTPhase1_LTPhase2_real(FieldLifeTime-
k_Shift_time+1,k_Stop_time,:);
    end
    for k_StopTime = 1: k_Shift_time
        if k_StopTime == 1
            x_data_stop_SRDM = ObsRate_Phase1n2_real_time_ShiftTime(:,1:FieldLifeTime-
k_StopTime,FieldLifeTime-k_Shift_time+1);
            x_data_square_stop_SRDM = sqrt(x_data_stop_SRDM(:,1:FieldLifeTime-k_StopTime-
1)).*sqrt(x_data_stop_SRDM(:,2:FieldLifeTime-k_StopTime));
            x_data_cube_stop_SRDM = sqrt(x_data_stop_SRDM(:,1:FieldLifeTime-k_StopTime-
2)).*sqrt(x_data_stop_SRDM(:,2:FieldLifeTime-k_StopTime-
1)).*sqrt(x_data_stop_SRDM(:,3:FieldLifeTime-k_StopTime));
            X_stop_SRDM = [ones(N_MC,1) x_data_stop_SRDM x_data_square_stop_SRDM
x_data_cube_stop_SRDM];
            y_data_stop_matrix = [NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+1)
NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+2)];
            [Stop_value,Stop_idx] =
max([X_stop_SRDM*regress(NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+1),X_stop_SRDM)
X_stop_SRDM*regress(NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+2),X_stop_SRDM)],[],2);
            for i_MC = 1: N_MC
                SRDM_Shift_Stop_OptNPV_real(i_MC,1) =
y_data_stop_matrix(i_MC,Stop_idx(i_MC));
            end
        else
            x_data_stop_SRDM = ObsRate_Phase1n2_real_time_ShiftTime(:,1:FieldLifeTime-
k_StopTime,FieldLifeTime-k_Shift_time+1);
            x_data_square_stop_SRDM = sqrt(x_data_stop_SRDM(:,1:FieldLifeTime-k_StopTime-
1)).*sqrt(x_data_stop_SRDM(:,2:FieldLifeTime-k_StopTime));
            x_data_cube_stop_SRDM = sqrt(x_data_stop_SRDM(:,1:FieldLifeTime-k_StopTime-
2)).*sqrt(x_data_stop_SRDM(:,2:FieldLifeTime-k_StopTime-
1)).*sqrt(x_data_stop_SRDM(:,3:FieldLifeTime-k_StopTime));
            X_stop_SRDM = [ones(N_MC,1) x_data_stop_SRDM x_data_square_stop_SRDM
x_data_cube_stop_SRDM];
            y_data_stop_matrix = [NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+1)
```

```
SRDM_Shift_Stop_OptNPV_real(:,1)];
            [Stop_value,Stop_idx] =
max([X_stop_SRDM*regress(NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+1),X_stop_SRDM)
X_stop_SRDM*regress(SRDM_Shift_Stop_OptNPV_real(:,1),X_stop_SRDM)],[],2);
            for i_MC = 1: N_MC
                SRDM_Shift_Stop_OptNPV_real(i_MC,1) =
y_data_stop_matrix(i_MC,Stop_idx(i_MC));
            end
        end
    end
    for i_MC = 1: N_MC
        PathTable(i_MC,FieldLifeTime-k_Shift_time+1) = SRDM_Shift_Stop_OptNPV_real(i_MC,1);
        for k_Stop_time = 1: k_Shift_time+1
            if SRDM_Shift_Stop_OptNPV_real(i_MC,1) == NPV_2Phases_matrix(i_MC,k_Stop_time)
                Phase2_StopTime_real_Phase1LifeTime(i_MC,FieldLifeTime-k_Shift_time+1) =
k_Stop_time-1;
            end
        end
    end
end

% 2. SRDM Approach

FieldLifeTime = size(ObsRate_Phase1n2_real_time_ShiftTime,2);
y_data_continuation = zeros(N_MC,1);
y_data_shift = zeros(N_MC,1);
y_reg_continuation = zeros(N_MC,1);
y_reg_shift = zeros(N_MC,1);
x_data = ObsRate_Phase1n2_real_time_ShiftTime(:,:,FieldLifeTime+1);

ValueTable = zeros(N_MC,FieldLifeTime+1);
ValueTable(:,end) = PathTable(:,end);
SRDM_DecisionTable = zeros(N_MC,FieldLifeTime+1);
SRDM_DecisionTable(:,end) = 1;

for k_Shift_time = 1:FieldLifeTime
    x_data_square = sqrt(x_data(:,1:FieldLifeTime-k_Shift_time-
1)).*sqrt(x_data(:,2:FieldLifeTime-k_Shift_time));
    x_data_cube = sqrt(x_data(:,1:FieldLifeTime-k_Shift_time-
2)).*sqrt(x_data(:,2:FieldLifeTime-k_Shift_time-1)).*sqrt(x_data(:,3:FieldLifeTime-
k_Shift_time));
    X = [ones(N_MC,1) x_data(:,1:FieldLifeTime-k_Shift_time) x_data_square x_data_cube];
    y_data_shift(:) = PathTable(:,FieldLifeTime-k_Shift_time+1);
    coef_shift = regress(y_data_shift,X);
    y_reg_shift(:) = X*coef_shift;
    y_data_continuation(:) = ValueTable(:,FieldLifeTime-k_Shift_time+2);
    coef_continuation = regress(y_data_continuation,X);
    y_reg_continuation(:) = X*coef_continuation;
    ENPVTable_Comparison = [y_reg_shift, y_reg_continuation];
    NPV_SRDM = [y_data_shift, y_data_continuation];
    [ENPV_SRDM,ENPV_SRDM_idx] = max(ENPVTable_Comparison,[],2);

    for i_MC = 1: N_MC
        ValueTable(i_MC,FieldLifeTime-k_Shift_time+1) = NPV_SRDM(i_MC,ENPV_SRDM_idx(i_MC));
        if y_reg_continuation(i_MC) > y_reg_shift(i_MC)
            SRDM_DecisionTable(i_MC,FieldLifeTime-k_Shift_time+1) = 0;
        else
            SRDM_DecisionTable(i_MC,FieldLifeTime-k_Shift_time+1) = 1;
```

```
        end
    end
end

ShiftValue = PathTable(:,1);
ExpShiftValue = mean(ShiftValue);

ContinuationValue = ValueTable(:,1);
ExpContinuationValue = mean(ContinuationValue);
EVWII = ExpContinuationValue;
VOI = EVWII - EVWOI;
VOI_over_EVWOI_pct = VOI/EVWOI*100;

[~,DWI_Phase1LifeTime_idx] = max(SRDM_DecisionTable,[],2);
DWI_Phase1LifeTime = DWI_Phase1LifeTime_idx-1;

DWI_Phase2LifeTime = zeros(N_MC,1);
for i_MC = 1:N_MC
    Phase1LifeTime_idx_real = DWI_Phase1LifeTime_idx(i_MC,1);
    DWI_Phase2LifeTime(i_MC,1) =
Phase2_StopTime_real_Phase1LifeTime(i_MC,Phase1LifeTime_idx_real);
end

DWI_LifeTime = DWI_Phase1LifeTime+DWI_Phase2LifeTime;

% 3. CLRM Approach

N_MC = size(ObsRate_Phase1n2_real_time_ShiftTime,1);
FieldLifeTime = size(ObsRate_Phase1n2_real_time_ShiftTime,2);
N_Phase1n2LifeTime = sum(1:FieldLifeTime+1);
N_LifeTimeComb = N_Phase1n2LifeTime;
Times = 1:1:FieldLifeTime;
CLRM_OptNPV_real = zeros(N_MC,1);
NPV_Tab_matrix = transpose(NPVmatrix_reals);
Prior_ENPV = meanNPVvector(:,3);
[max_Prior_ENPV, max_Prior_ENPV_idx] = max(meanNPVvector(:,3),[],1);
x_data_switch_CLRM = ObsRate_Phase1n2_real_time_ShiftTime(:,:,FieldLifeTime+1);

CLRM_PhaseLifes_Opt_reals = zeros(N_MC,2);

if max_Prior_ENPV_idx <= Times(FieldLifeTime)+1 %Finding if Switch Time is optimal at Y1
    Shift_time = 1;
    t_dataPhase2 = 1;
    while t_dataPhase2 <= Times(FieldLifeTime-Shift_time+1)
        if t_dataPhase2 == 1 && max_Prior_ENPV_idx == 1 %Switch and Stop at Y1 are optimal
            for i_MC = 1: N_MC
                CLRM_OptNPV_real(i_MC,1) = max_Prior_ENPV;
            end
            break; %The whole loop will break here
        else %Given Switch at Y1 and Stop is not at Y1
            x_data_stop_CLRM = ObsRate_Phase1n2_real_time_ShiftTime(:,:,Shift_time);
            if t_dataPhase2 == 1
                x_data_square_stop_CLRM = x_data_stop_CLRM(:,1:t_dataPhase2);
                X_stop_CLRM = [ones(N_MC,1) x_data_stop_CLRM(:,1:t_dataPhase2)];
            elseif t_dataPhase2 == 2
                x_data_square_stop_CLRM = sqrt(x_data_stop_CLRM(:,1:t_dataPhase2-
1)).*sqrt(x_data_stop_CLRM(:,2:t_dataPhase2));
                X_stop_CLRM = [ones(N_MC,1) x_data_stop_CLRM(:,1:t_dataPhase2)
```

```
x_data_square_stop_CLRM];
            else
                x_data_square_stop_CLRM = sqrt(x_data_stop_CLRM(:,1:t_dataPhase2-
1)).*sqrt(x_data_stop_CLRM(:,2:t_dataPhase2));
                x_data_cube_stop_CLRM = sqrt(x_data_stop_CLRM(:,1:t_dataPhase2-
2)).*sqrt(x_data_stop_CLRM(:,2:t_dataPhase2-1)).*sqrt(x_data_stop_CLRM(:,3:t_dataPhase2));
                X_stop_CLRM = [ones(N_MC,1) x_data_stop_CLRM(:,1:t_dataPhase2)
x_data_square_stop_CLRM x_data_cube_stop_CLRM];
            end
            y_data_stop_CLRM = NPV_Tab_matrix(:,t_dataPhase2+1:FieldLifeTime+1);
            y_reg_stop_CLRM = zeros(N_MC,FieldLifeTime-t_dataPhase2+1);
            for t = 1: FieldLifeTime-t_dataPhase2+1
                y_reg_stop_CLRM(:,t) =
X_stop_CLRM*regress(y_data_stop_CLRM(:,t),X_stop_CLRM);
            end
            [StopTime_reg,StopTime_idx] = max(y_reg_stop_CLRM,[],2);
        for i_MC = 1: N_MC
            if StopTime_reg(i_MC) == y_reg_stop_CLRM(i_MC,1) && CLRM_OptNPV_real(i_MC,1) == 0
                CLRM_OptNPV_real(i_MC,1) = y_data_stop_CLRM(i_MC,1);
                CLRM_PhaseLifes_Opt_reals(i_MC,1) = Shift_time-1;
                CLRM_PhaseLifes_Opt_reals(i_MC,2) = t_dataPhase2+StopTime_idx(i_MC)-1;
            else
                continue;
            end
        end
        end
        t_dataPhase2 = t_dataPhase2+1;
    end
else
% This part is to find the corresponding optimal stop time given Switch time is not at Y1
    for Shift_time = 2: FieldLifeTime %Finding and recording the optimal switch time before
determining the stop time
        N_LifeTimeComb = N_LifeTimeComb-Times(FieldLifeTime-Shift_time+2)-1;
        y_data_switch_CLRM = NPV_Tab_matrix(:,N_Phase1n2LifeTime-
N_LifeTimeComb+1:N_Phase1n2LifeTime);
        y_reg_switch_CLRM = zeros(N_MC,N_LifeTimeComb);
        ENPVTabComp = zeros(N_MC,N_LifeTimeComb);
        if Shift_time == 2
            x_data_square_switch_CLRM = x_data_switch_CLRM(:,1:Shift_time-1);
            X_switch_CLRM = [ones(N_MC,1) x_data_switch_CLRM(:,1:Shift_time-1)];
        elseif Shift_time == 3
            x_data_square_switch_CLRM = sqrt(x_data_switch_CLRM(:,1:Shift_time-
2)).*sqrt(x_data_switch_CLRM(:,2:Shift_time-1));
            X_switch_CLRM = [ones(N_MC,1) x_data_switch_CLRM(:,1:Shift_time-1)
x_data_square_switch_CLRM];
        else
            x_data_square_switch_CLRM = sqrt(x_data_switch_CLRM(:,1:Shift_time-
2)).*sqrt(x_data_switch_CLRM(:,2:Shift_time-1));
            x_data_cube_switch_CLRM = sqrt(x_data_switch_CLRM(:,1:Shift_time-
3)).*sqrt(x_data_switch_CLRM(:,2:Shift_time-2)).*sqrt(x_data_switch_CLRM(:,3:Shift_time-1));
            X_switch_CLRM = [ones(N_MC,1) x_data_switch_CLRM(:,1:Shift_time-1)
x_data_square_switch_CLRM x_data_cube_switch_CLRM];
        end
        for i_LifeTimeComb = 1: N_LifeTimeComb
            coef_reg_CLRM = regress(y_data_switch_CLRM(:,i_LifeTimeComb),X_switch_CLRM);
            y_reg_switch_CLRM(:,i_LifeTimeComb) = X_switch_CLRM*coef_reg_CLRM;
            ENPVTabComp(:,i_LifeTimeComb) = y_reg_switch_CLRM(:,i_LifeTimeComb);
        end
```

```
    [CLRM_reg,CLRM_idx] = max(ENPVTabComp,[],2);

    if Shift_time == FieldLifeTime
        [SwitchDecision_CLRM,SwitchDecision_CLRM_idx] = find(CLRM_idx <= Times(FieldLifeTime-
Shift_time+2)+1);
        [ContinueTime_reg,ContinueTime_idx] = find(CLRM_idx == Times(FieldLifeTime-
Shift_time+2)+1);
        for i_path = 1: length(ContinueTime_reg)
            if CLRM_OptNPV_real(ContinueTime_reg(i_path),1) == 0
               CLRM_OptNPV_real(ContinueTime_reg(i_path),1) =
y_data_switch_CLRM(ContinueTime_reg(i_path),end);
               CLRM_PhaseLifes_Opt_reals(ContinueTime_reg(i_path),1) = FieldLifeTime;
               CLRM_PhaseLifes_Opt_reals(ContinueTime_reg(i_path),2) = 0;
            else
                continue;
            end
        end
    else
        [SwitchDecision_CLRM,SwitchDecision_CLRM_idx] = find(CLRM_idx <= Times(FieldLifeTime-
Shift_time+2));
    end

    x_data_stop_CLRM = ObsRate_Phase1n2_real_time_ShiftTime(:,:,Shift_time);
    t_dataPhase2 = 1;
    while t_dataPhase2 <= Times(FieldLifeTime-Shift_time+1)+1
        if t_dataPhase2 == 1
            x_data_square_stop_CLRM = x_data_stop_CLRM(:,1:t_dataPhase2);
            X_stop_CLRM = [ones(N_MC,1) x_data_stop_CLRM(:,1:t_dataPhase2+Shift_time-2)];
        elseif t_dataPhase2 == 2
            x_data_square_stop_CLRM = sqrt(x_data_stop_CLRM(:,1:t_dataPhase2-
1)).*sqrt(x_data_stop_CLRM(:,2:t_dataPhase2));
            X_stop_CLRM = [ones(N_MC,1) x_data_stop_CLRM(:,1:t_dataPhase2+Shift_time-2)
x_data_square_stop_CLRM];
        else
            x_data_square_stop_CLRM = sqrt(x_data_stop_CLRM(:,1:t_dataPhase2-
1)).*sqrt(x_data_stop_CLRM(:,2:t_dataPhase2));
            x_data_cube_stop_CLRM = sqrt(x_data_stop_CLRM(:,1:t_dataPhase2-
2)).*sqrt(x_data_stop_CLRM(:,2:t_dataPhase2-1)).*sqrt(x_data_stop_CLRM(:,3:t_dataPhase2));
            X_stop_CLRM = [ones(N_MC,1) x_data_stop_CLRM(:,1:t_dataPhase2+Shift_time-2)
x_data_square_stop_CLRM x_data_cube_stop_CLRM];
        end
        y_data_stop_CLRM = y_data_switch_CLRM(:, t_dataPhase2:Times(FieldLifeTime-
Shift_time+1)+1);
        y_reg_stop_CLRM = zeros(N_MC,Times(FieldLifeTime-Shift_time+1)-t_dataPhase2+2);
        for t = 1: Times(FieldLifeTime-Shift_time+1)-t_dataPhase2+2
            y_reg_stop_CLRM(:,t) = X_stop_CLRM*regress(y_data_stop_CLRM(:,t),X_stop_CLRM);
        end
            [StopTime_reg,StopTime_idx] = max(y_reg_stop_CLRM,[],2);
        for i_path = 1: length(SwitchDecision_CLRM)
            if StopTime_reg(SwitchDecision_CLRM(i_path)) ==
y_reg_stop_CLRM(SwitchDecision_CLRM(i_path),1) &&
CLRM_OptNPV_real(SwitchDecision_CLRM(i_path),1) == 0
                CLRM_OptNPV_real(SwitchDecision_CLRM(i_path),1) =
y_data_stop_CLRM(SwitchDecision_CLRM(i_path),1);
                CLRM_PhaseLifes_Opt_reals(SwitchDecision_CLRM(i_path),1) = Shift_time-1;
                CLRM_PhaseLifes_Opt_reals(SwitchDecision_CLRM(i_path),2) =
t_dataPhase2+StopTime_idx(SwitchDecision_CLRM(i_path))-2;
```

```
            else
                continue;
            end
        end
            t_dataPhase2 = t_dataPhase2+1;
    end
    end
end

DWI_CLRM_Phase1LifeTime = CLRM_PhaseLifes_Opt_reals(:,1);
DWI_CLRM_Phase2LifeTime = CLRM_PhaseLifes_Opt_reals(:,2);
DWI_CLRM_LifeTime = DWI_CLRM_Phase1LifeTime + DWI_CLRM_Phase2LifeTime;

EVWI_CLRM = mean(CLRM_OptNPV_real);
VOI_CLRM = EVWI_CLRM-EVWOI;
```

Appendix B2.4: Regression Function Equation (16)

This code consists of the modified version of the sections 5 and 6 of Modified LSM Algorithm (Appendix B1.1) and CLRM (Appendix B1.2). With respect to this, the modification is done on the regression analysis. Thus, sections 1 to 4 of Modified LSM Algorithm (Appendix B1.1) have to be run first before executing this code. This code corresponds to the Regression Function of Equation (16) in the report.

```
% 1. Path Table Generation

N_MC = size(ObsRate_Phase1n2_real_time_ShiftTime,1);
FieldLifeTime = size(ObsRate_Phase1n2_real_time_ShiftTime,2);
SRDM_Shift_Stop_OptNPV_real = zeros(N_MC,1);

PathTable = zeros(N_MC,FieldLifeTime+1);
PathTable(:,end) = NPVtable_LTPhase1_LTPhase2_real(FieldLifeTime+1,1,:);
Phase2_StopTime_real_Phase1LifeTime = zeros(N_MC,FieldLifeTime+1);

for k_Shift_time = 1: FieldLifeTime
        NPV_2Phases_matrix = zeros(N_MC,k_Shift_time+1);
    for k_Stop_time = 1: k_Shift_time+1
        NPV_2Phases_matrix(:,k_Stop_time) = NPVtable_LTPhase1_LTPhase2_real(FieldLifeTime-
k_Shift_time+1,k_Stop_time,:);
    end
    for k_StopTime = 1: k_Shift_time
        if k_StopTime == 1
            x_data_stop_SRDM = ObsRate_Phase1n2_real_time_ShiftTime(:,1:FieldLifeTime-
k_StopTime,FieldLifeTime-k_Shift_time+1);
            x_data_square_stop_SRDM = x_data_stop_SRDM(:,1:FieldLifeTime-k_StopTime-
1).*x_data_stop_SRDM(:,2:FieldLifeTime-k_StopTime);
            X_stop_SRDM = [ones(N_MC,1) x_data_stop_SRDM x_data_square_stop_SRDM];
            y_data_stop_matrix = [NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+1)
NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+2)];
            [Stop_value,Stop_idx] =
max([X_stop_SRDM*regress(NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+1),X_stop_SRDM)
X_stop_SRDM*regress(NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+2),X_stop_SRDM)],[],2);
            for i_MC = 1: N_MC
                SRDM_Shift_Stop_OptNPV_real(i_MC,1) =
y_data_stop_matrix(i_MC,Stop_idx(i_MC));
            end
        else
            x_data_stop_SRDM = ObsRate_Phase1n2_real_time_ShiftTime(:,1:FieldLifeTime-
k_StopTime,FieldLifeTime-k_Shift_time+1);
            x_data_square_stop_SRDM = x_data_stop_SRDM(:,1:FieldLifeTime-k_StopTime-
1).*x_data_stop_SRDM(:,2:FieldLifeTime-k_StopTime);
            X_stop_SRDM = [ones(N_MC,1) x_data_stop_SRDM x_data_square_stop_SRDM];
            y_data_stop_matrix = [NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+1)
SRDM_Shift_Stop_OptNPV_real(:,1)];
            [Stop_value,Stop_idx] =
max([X_stop_SRDM*regress(NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+1),X_stop_SRDM)
X_stop_SRDM*regress(SRDM_Shift_Stop_OptNPV_real(:,1),X_stop_SRDM)],[],2);
            for i_MC = 1: N_MC
                SRDM_Shift_Stop_OptNPV_real(i_MC,1) =
y_data_stop_matrix(i_MC,Stop_idx(i_MC));
            end
```

```
        end
    end
    for i_MC = 1: N_MC
        PathTable(i_MC,FieldLifeTime-k_Shift_time+1) = SRDM_Shift_Stop_OptNPV_real(i_MC,1);
        for k_Stop_time = 1: k_Shift_time+1
            if SRDM_Shift_Stop_OptNPV_real(i_MC,1) == NPV_2Phases_matrix(i_MC,k_Stop_time)
                Phase2_StopTime_real_Phase1LifeTime(i_MC,FieldLifeTime-k_Shift_time+1) =
k_Stop_time-1;
            end
        end
    end
end

% 2. SRDM Approach

FieldLifeTime = size(ObsRate_Phase1n2_real_time_ShiftTime,2);
y_data_continuation = zeros(N_MC,1);
y_data_shift = zeros(N_MC,1);
y_reg_continuation = zeros(N_MC,1);
y_reg_shift = zeros(N_MC,1);
x_data = ObsRate_Phase1n2_real_time_ShiftTime(:,:,FieldLifeTime+1);

ValueTable = zeros(N_MC,FieldLifeTime+1);
ValueTable(:,end) = PathTable(:,end);
SRDM_DecisionTable = zeros(N_MC,FieldLifeTime+1);
SRDM_DecisionTable(:,end) = 1;

for k_Shift_time = 1:FieldLifeTime
    x_data_square = x_data(:,1:FieldLifeTime-k_Shift_time-1).*x_data(:,2:FieldLifeTime-
k_Shift_time);
    X = [ones(N_MC,1) x_data(:,1:FieldLifeTime-k_Shift_time) x_data_square];
    y_data_shift(:) = PathTable(:,FieldLifeTime-k_Shift_time+1);
    coef_shift = regress(y_data_shift,X);
    y_reg_shift(:) = X*coef_shift;
    y_data_continuation(:) = ValueTable(:,FieldLifeTime-k_Shift_time+2);
    coef_continuation = regress(y_data_continuation,X);
    y_reg_continuation(:) = X*coef_continuation;
    ENPVTable_Comparison = [y_reg_shift, y_reg_continuation];
    NPV_SRDM = [y_data_shift, y_data_continuation];
    [ENPV_SRDM,ENPV_SRDM_idx] = max(ENPVTable_Comparison,[],2);

    for i_MC = 1: N_MC
        ValueTable(i_MC,FieldLifeTime-k_Shift_time+1) = NPV_SRDM(i_MC,ENPV_SRDM_idx(i_MC));
        if y_reg_continuation(i_MC) > y_reg_shift(i_MC)
            SRDM_DecisionTable(i_MC,FieldLifeTime-k_Shift_time+1) = 0;
        else
            SRDM_DecisionTable(i_MC,FieldLifeTime-k_Shift_time+1) = 1;
        end
    end
end

ShiftValue = PathTable(:,1);
ExpShiftValue = mean(ShiftValue);

ContinuationValue = ValueTable(:,1);
ExpContinuationValue = mean(ContinuationValue);
EVWII = ExpContinuationValue;
VOI = EVWII - EVWOI;
```

```
VOI_over_EVWOI_pct = VOI/EVWOI*100;

[~,DWI_Phase1LifeTime_idx] = max(SRDM_DecisionTable,[],2);
DWI_Phase1LifeTime = DWI_Phase1LifeTime_idx-1;

DWI_Phase2LifeTime = zeros(N_MC,1);
for i_MC = 1:N_MC
    Phase1LifeTime_idx_real = DWI_Phase1LifeTime_idx(i_MC,1);
    DWI_Phase2LifeTime(i_MC,1) =
Phase2_StopTime_real_Phase1LifeTime(i_MC,Phase1LifeTime_idx_real);
end

DWI_LifeTime = DWI_Phase1LifeTime+DWI_Phase2LifeTime;

% 3. CLRM Approach

N_MC = size(ObsRate_Phase1n2_real_time_ShiftTime,1);
FieldLifeTime = size(ObsRate_Phase1n2_real_time_ShiftTime,2);
N_Phase1n2LifeTime = sum(1:FieldLifeTime+1);
N_LifeTimeComb = N_Phase1n2LifeTime;
Times = 1:1:FieldLifeTime;
CLRM_OptNPV_real = zeros(N_MC,1);
NPV_Tab_matrix = transpose(NPVmatrix_reals);
Prior_ENPV = meanNPVvector(:,3);
[max_Prior_ENPV, max_Prior_ENPV_idx] = max(meanNPVvector(:,3),[],1);
x_data_switch_CLRM = ObsRate_Phase1n2_real_time_ShiftTime(:,:,FieldLifeTime+1);

CLRM_PhaseLifes_Opt_reals = zeros(N_MC,2);

if max_Prior_ENPV_idx <= Times(FieldLifeTime)+1 %Finding if Switch Time is optimal at Y1
    Shift_time = 1;
    t_dataPhase2 = 1;
    while t_dataPhase2 <= Times(FieldLifeTime-Shift_time+1)
        if t_dataPhase2 == 1 && max_Prior_ENPV_idx == 1 %Switch and Stop at Y1 are optimal
            for i_MC = 1: N_MC
                CLRM_OptNPV_real(i_MC,1) = max_Prior_ENPV;
            end
            break; %The whole loop will break here
        else %Given Switch at Y1 and Stop is not at Y1
            x_data_stop_CLRM = ObsRate_Phase1n2_real_time_ShiftTime(:,:,Shift_time);
            if t_dataPhase2 == 1
                x_data_square_stop_CLRM = x_data_stop_CLRM(:,1:t_dataPhase2);
                X_stop_CLRM = [ones(N_MC,1) x_data_stop_CLRM(:,1:t_dataPhase2)];
            else
                x_data_square_stop_CLRM = x_data_stop_CLRM(:,1:t_dataPhase2-
1).*x_data_stop_CLRM(:,2:t_dataPhase2);
                X_stop_CLRM = [ones(N_MC,1) x_data_stop_CLRM(:,1:t_dataPhase2)
x_data_square_stop_CLRM];
            end
            y_data_stop_CLRM = NPV_Tab_matrix(:,t_dataPhase2+1:FieldLifeTime+1);
            y_reg_stop_CLRM = zeros(N_MC,FieldLifeTime-t_dataPhase2+1);
            for t = 1: FieldLifeTime-t_dataPhase2+1
                y_reg_stop_CLRM(:,t) =
X_stop_CLRM*regress(y_data_stop_CLRM(:,t),X_stop_CLRM);
            end
            [StopTime_reg,StopTime_idx] = max(y_reg_stop_CLRM,[],2);
        for i_MC = 1: N_MC
            if StopTime_reg(i_MC) == y_reg_stop_CLRM(i_MC,1) && CLRM_OptNPV_real(i_MC,1) == 0
```

```
                CLRM_OptNPV_real(i_MC,1) = y_data_stop_CLRM(i_MC,1);
                CLRM_PhaseLifes_Opt_reals(i_MC,1) = Shift_time-1;
                CLRM_PhaseLifes_Opt_reals(i_MC,2) = t_dataPhase2+StopTime_idx(i_MC)-1;
            else
                continue;
            end
        end
        end
        t_dataPhase2 = t_dataPhase2+1;
    end
else
% This part is to find the corresponding optimal stop time given Switch time is not at Y1
    for Shift_time = 2: FieldLifeTime %Finding and recording the optimal switch time before
determining the stop time
        N_LifeTimeComb = N_LifeTimeComb-Times(FieldLifeTime-Shift_time+2)-1;
        y_data_switch_CLRM = NPV_Tab_matrix(:,N_Phase1n2LifeTime-
N_LifeTimeComb+1:N_Phase1n2LifeTime);
        y_reg_switch_CLRM = zeros(N_MC,N_LifeTimeComb);
        ENPVTabComp = zeros(N_MC,N_LifeTimeComb);
        if Shift_time == 2
            x_data_square_switch_CLRM = x_data_switch_CLRM(:,1:Shift_time-1);
            X_switch_CLRM = [ones(N_MC,1) x_data_switch_CLRM(:,1:Shift_time-1)];
        else
            x_data_square_switch_CLRM = x_data_switch_CLRM(:,1:Shift_time-
2).*x_data_switch_CLRM(:,2:Shift_time-1);
            X_switch_CLRM = [ones(N_MC,1) x_data_switch_CLRM(:,1:Shift_time-1)
x_data_square_switch_CLRM];
        end
        for i_LifeTimeComb = 1: N_LifeTimeComb
            coef_reg_CLRM = regress(y_data_switch_CLRM(:,i_LifeTimeComb),X_switch_CLRM);
            y_reg_switch_CLRM(:,i_LifeTimeComb) = X_switch_CLRM*coef_reg_CLRM;
            ENPVTabComp(:,i_LifeTimeComb) = y_reg_switch_CLRM(:,i_LifeTimeComb);
        end

    [CLRM_reg,CLRM_idx] = max(ENPVTabComp,[],2);

    if Shift_time == FieldLifeTime
        [SwitchDecision_CLRM,SwitchDecision_CLRM_idx] = find(CLRM_idx <= Times(FieldLifeTime-
Shift_time+2)+1);
        [ContinueTime_reg,ContinueTime_idx] = find(CLRM_idx == Times(FieldLifeTime-
Shift_time+2)+1);
        for i_path = 1: length(ContinueTime_reg)
            if CLRM_OptNPV_real(ContinueTime_reg(i_path),1) == 0
               CLRM_OptNPV_real(ContinueTime_reg(i_path),1) =
y_data_switch_CLRM(ContinueTime_reg(i_path),end);
               CLRM_PhaseLifes_Opt_reals(ContinueTime_reg(i_path),1) = FieldLifeTime;
               CLRM_PhaseLifes_Opt_reals(ContinueTime_reg(i_path),2) = 0;
            else
                continue;
            end
        end
    else
        [SwitchDecision_CLRM,SwitchDecision_CLRM_idx] = find(CLRM_idx <= Times(FieldLifeTime-
Shift_time+2));
    end

    x_data_stop_CLRM = ObsRate_Phase1n2_real_time_ShiftTime(:,:,Shift_time);
    t_dataPhase2 = 1;
```

```
    while t_dataPhase2 <= Times(FieldLifeTime-Shift_time+1)+1
        if t_dataPhase2 == 1
            x_data_square_stop_CLRM = x_data_stop_CLRM(:,1:t_dataPhase2);
            X_stop_CLRM = [ones(N_MC,1) x_data_stop_CLRM(:,1:t_dataPhase2+Shift_time-2)];
        else
            x_data_square_stop_CLRM = x_data_stop_CLRM(:,1:t_dataPhase2-
1).*x_data_stop_CLRM(:,2:t_dataPhase2);
            X_stop_CLRM = [ones(N_MC,1) x_data_stop_CLRM(:,1:t_dataPhase2+Shift_time-2)
x_data_square_stop_CLRM];
        end
        y_data_stop_CLRM = y_data_switch_CLRM(:, t_dataPhase2:Times(FieldLifeTime-
Shift_time+1)+1);
        y_reg_stop_CLRM = zeros(N_MC,Times(FieldLifeTime-Shift_time+1)-t_dataPhase2+2);
        for t = 1: Times(FieldLifeTime-Shift_time+1)-t_dataPhase2+2
            y_reg_stop_CLRM(:,t) = X_stop_CLRM*regress(y_data_stop_CLRM(:,t),X_stop_CLRM);
        end
            [StopTime_reg,StopTime_idx] = max(y_reg_stop_CLRM,[],2);
        for i_path = 1: length(SwitchDecision_CLRM)
            if StopTime_reg(SwitchDecision_CLRM(i_path)) ==
y_reg_stop_CLRM(SwitchDecision_CLRM(i_path),1) &&
CLRM_OptNPV_real(SwitchDecision_CLRM(i_path),1) == 0
                CLRM_OptNPV_real(SwitchDecision_CLRM(i_path),1) =
y_data_stop_CLRM(SwitchDecision_CLRM(i_path),1);
                CLRM_PhaseLifes_Opt_reals(SwitchDecision_CLRM(i_path),1) = Shift_time-1;
                CLRM_PhaseLifes_Opt_reals(SwitchDecision_CLRM(i_path),2) =
t_dataPhase2+StopTime_idx(SwitchDecision_CLRM(i_path))-2;
            else
                continue;
            end
        end
            t_dataPhase2 = t_dataPhase2+1;
    end
    end
end

DWI_CLRM_Phase1LifeTime = CLRM_PhaseLifes_Opt_reals(:,1);
DWI_CLRM_Phase2LifeTime = CLRM_PhaseLifes_Opt_reals(:,2);
DWI_CLRM_LifeTime = DWI_CLRM_Phase1LifeTime + DWI_CLRM_Phase2LifeTime;

EVWI_CLRM = mean(CLRM_OptNPV_real);
VOI_CLRM = EVWI_CLRM-EVWOI;
```

Appendix B2.5: Regression Function Equation (17)

This code consists of the modified version of the sections 5 and 6 of Modified LSM Algorithm (Appendix B1.1) and CLRM (Appendix B1.2). With respect to this, the modification is done on the regression analysis. Thus, sections 1 to 4 of Modified LSM Algorithm (Appendix B1.1) have to be run first before executing this code. This code corresponds to the Regression Function of Equation (17) in the report.

```
% 1. Path Table Generation

N_MC = size(ObsRate_Phase1n2_real_time_ShiftTime,1);
FieldLifeTime = size(ObsRate_Phase1n2_real_time_ShiftTime,2);
SRDM_Shift_Stop_OptNPV_real = zeros(N_MC,1);

PathTable = zeros(N_MC,FieldLifeTime+1);
PathTable(:,end) = NPVtable_LTPhase1_LTPhase2_real(FieldLifeTime+1,1,:);
Phase2_StopTime_real_Phase1LifeTime = zeros(N_MC,FieldLifeTime+1);

for k_Shift_time = 1: FieldLifeTime
        NPV_2Phases_matrix = zeros(N_MC,k_Shift_time+1);
    for k_Stop_time = 1: k_Shift_time+1
        NPV_2Phases_matrix(:,k_Stop_time) = NPVtable_LTPhase1_LTPhase2_real(FieldLifeTime-
k_Shift_time+1,k_Stop_time,:);
    end
    for k_StopTime = 1: k_Shift_time
        if k_StopTime == 1
            x_data_stop_SRDM = ObsRate_Phase1n2_real_time_ShiftTime(:,1:FieldLifeTime-
k_StopTime,FieldLifeTime-k_Shift_time+1);
            x_data_cube_stop_SRDM = x_data_stop_SRDM(:,1:FieldLifeTime-k_StopTime-
2).*x_data_stop_SRDM(:,2:FieldLifeTime-k_StopTime-1).*x_data_stop_SRDM(:,3:FieldLifeTime-
k_StopTime);
            X_stop_SRDM = [ones(N_MC,1) x_data_stop_SRDM x_data_cube_stop_SRDM];
            y_data_stop_matrix = [NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+1)
NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+2)];
            [Stop_value,Stop_idx] =
max([X_stop_SRDM*regress(NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+1),X_stop_SRDM)
X_stop_SRDM*regress(NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+2),X_stop_SRDM)],[],2);
            for i_MC = 1: N_MC
                SRDM_Shift_Stop_OptNPV_real(i_MC,1) =
y_data_stop_matrix(i_MC,Stop_idx(i_MC));
            end
        else
            x_data_stop_SRDM = ObsRate_Phase1n2_real_time_ShiftTime(:,1:FieldLifeTime-
k_StopTime,FieldLifeTime-k_Shift_time+1);
            x_data_cube_stop_SRDM = x_data_stop_SRDM(:,1:FieldLifeTime-k_StopTime-
2).*x_data_stop_SRDM(:,2:FieldLifeTime-k_StopTime-1).*x_data_stop_SRDM(:,3:FieldLifeTime-
k_StopTime);
            X_stop_SRDM = [ones(N_MC,1) x_data_stop_SRDM x_data_cube_stop_SRDM];
            y_data_stop_matrix = [NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+1)
SRDM_Shift_Stop_OptNPV_real(:,1)];
            [Stop_value,Stop_idx] =
max([X_stop_SRDM*regress(NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+1),X_stop_SRDM)
X_stop_SRDM*regress(SRDM_Shift_Stop_OptNPV_real(:,1),X_stop_SRDM)],[],2);
            for i_MC = 1: N_MC
                SRDM_Shift_Stop_OptNPV_real(i_MC,1) =
```

```
y_data_stop_matrix(i_MC,Stop_idx(i_MC));
            end
        end
    end
    for i_MC = 1: N_MC
        PathTable(i_MC,FieldLifeTime-k_Shift_time+1) = SRDM_Shift_Stop_OptNPV_real(i_MC,1);
        for k_Stop_time = 1: k_Shift_time+1
            if SRDM_Shift_Stop_OptNPV_real(i_MC,1) == NPV_2Phases_matrix(i_MC,k_Stop_time)
                Phase2_StopTime_real_Phase1LifeTime(i_MC,FieldLifeTime-k_Shift_time+1) = 
k_Stop_time-1;
            end
        end
    end
end

% 2. SRDM Approach

FieldLifeTime = size(ObsRate_Phase1n2_real_time_ShiftTime,2);
y_data_continuation = zeros(N_MC,1);
y_data_shift = zeros(N_MC,1);
y_reg_continuation = zeros(N_MC,1);
y_reg_shift = zeros(N_MC,1);
x_data = ObsRate_Phase1n2_real_time_ShiftTime(:,:,FieldLifeTime+1);

ValueTable = zeros(N_MC,FieldLifeTime+1);
ValueTable(:,end) = PathTable(:,end);
SRDM_DecisionTable = zeros(N_MC,FieldLifeTime+1);
SRDM_DecisionTable(:,end) = 1;

for k_Shift_time = 1:FieldLifeTime
    x_data_cube = x_data(:,1:FieldLifeTime-k_Shift_time-2).*x_data(:,2:FieldLifeTime-
k_Shift_time-1).*x_data(:,3:FieldLifeTime-k_Shift_time);
    X = [ones(N_MC,1) x_data(:,1:FieldLifeTime-k_Shift_time) x_data_cube];
    y_data_shift(:) = PathTable(:,FieldLifeTime-k_Shift_time+1);
    coef_shift = regress(y_data_shift,X);
    y_reg_shift(:) = X*coef_shift;
    y_data_continuation(:) = ValueTable(:,FieldLifeTime-k_Shift_time+2);
    coef_continuation = regress(y_data_continuation,X);
    y_reg_continuation(:) = X*coef_continuation;
    ENPVTable_Comparison = [y_reg_shift, y_reg_continuation];
    NPV_SRDM = [y_data_shift, y_data_continuation];
    [ENPV_SRDM,ENPV_SRDM_idx] = max(ENPVTable_Comparison,[],2);

    for i_MC = 1: N_MC
        ValueTable(i_MC,FieldLifeTime-k_Shift_time+1) = NPV_SRDM(i_MC,ENPV_SRDM_idx(i_MC));
        if y_reg_continuation(i_MC) > y_reg_shift(i_MC)
            SRDM_DecisionTable(i_MC,FieldLifeTime-k_Shift_time+1) = 0;
        else
            SRDM_DecisionTable(i_MC,FieldLifeTime-k_Shift_time+1) = 1;
        end
    end
end

ShiftValue = PathTable(:,1);
ExpShiftValue = mean(ShiftValue);

ContinuationValue = ValueTable(:,1);
ExpContinuationValue = mean(ContinuationValue);
```

```
EVWII = ExpContinuationValue;
VOI = EVWII - EVWOI;
VOI_over_EVWOI_pct = VOI/EVWOI*100;

[~,DWI_Phase1LifeTime_idx] = max(SRDM_DecisionTable,[],2);
DWI_Phase1LifeTime = DWI_Phase1LifeTime_idx-1;

DWI_Phase2LifeTime = zeros(N_MC,1);
for i_MC = 1:N_MC
    Phase1LifeTime_idx_real = DWI_Phase1LifeTime_idx(i_MC,1);
    DWI_Phase2LifeTime(i_MC,1) =
Phase2_StopTime_real_Phase1LifeTime(i_MC,Phase1LifeTime_idx_real);
end

DWI_LifeTime = DWI_Phase1LifeTime+DWI_Phase2LifeTime;

% 3. CLRM Approach

N_MC = size(ObsRate_Phase1n2_real_time_ShiftTime,1);
FieldLifeTime = size(ObsRate_Phase1n2_real_time_ShiftTime,2);
N_Phase1n2LifeTime = sum(1:FieldLifeTime+1);
N_LifeTimeComb = N_Phase1n2LifeTime;
Times = 1:1:FieldLifeTime;
CLRM_OptNPV_real = zeros(N_MC,1);
NPV_Tab_matrix = transpose(NPVmatrix_reals);
Prior_ENPV = meanNPVvector(:,3);
[max_Prior_ENPV, max_Prior_ENPV_idx] = max(meanNPVvector(:,3),[],1);
x_data_switch_CLRM = ObsRate_Phase1n2_real_time_ShiftTime(:,:,FieldLifeTime+1);

CLRM_PhaseLifes_Opt_reals = zeros(N_MC,2);

if max_Prior_ENPV_idx <= Times(FieldLifeTime)+1 %Finding if Switch Time is optimal at Y1
    Shift_time = 1;
    t_dataPhase2 = 1;
    while t_dataPhase2 <= Times(FieldLifeTime-Shift_time+1)
        if t_dataPhase2 == 1 && max_Prior_ENPV_idx == 1 %Switch and Stop at Y1 are optimal
            for i_MC = 1: N_MC
                CLRM_OptNPV_real(i_MC,1) = max_Prior_ENPV;
            end
            break; %The whole loop will break here
        else %Given Switch at Y1 and Stop is not at Y1
            x_data_stop_CLRM = ObsRate_Phase1n2_real_time_ShiftTime(:,:,Shift_time);
            if t_dataPhase2 == 1 && t_dataPhase2 == 2
                x_data_cube_stop_CLRM = x_data_stop_CLRM(:,1:t_dataPhase2);
                X_stop_CLRM = [ones(N_MC,1) x_data_stop_CLRM(:,1:t_dataPhase2)];
            else
                x_data_cube_stop_CLRM = x_data_stop_CLRM(:,1:t_dataPhase2-
2).*x_data_stop_CLRM(:,2:t_dataPhase2-1).*x_data_stop_CLRM(:,3:t_dataPhase2);
                X_stop_CLRM = [ones(N_MC,1) x_data_stop_CLRM(:,1:t_dataPhase2)
x_data_cube_stop_CLRM];
            end
            y_data_stop_CLRM = NPV_Tab_matrix(:,t_dataPhase2+1:FieldLifeTime+1);
            y_reg_stop_CLRM = zeros(N_MC,FieldLifeTime-t_dataPhase2+1);
            for t = 1: FieldLifeTime-t_dataPhase2+1
                y_reg_stop_CLRM(:,t) =
X_stop_CLRM*regress(y_data_stop_CLRM(:,t),X_stop_CLRM);
            end
            [StopTime_reg,StopTime_idx] = max(y_reg_stop_CLRM,[],2);
```

```
        for i_MC = 1: N_MC
            if StopTime_reg(i_MC) == y_reg_stop_CLRM(i_MC,1) && CLRM_OptNPV_real(i_MC,1) == 0
                CLRM_OptNPV_real(i_MC,1) = y_data_stop_CLRM(i_MC,1);
                CLRM_PhaseLifes_Opt_reals(i_MC,1) = Shift_time-1;
                CLRM_PhaseLifes_Opt_reals(i_MC,2) = t_dataPhase2+StopTime_idx(i_MC)-1;
            else
                continue;
            end
        end
        end
        t_dataPhase2 = t_dataPhase2+1;
    end
else
% This part is to find the corresponding optimal stop time given Switch time is not at Y1
    for Shift_time = 2: FieldLifeTime %Finding and recording the optimal switch time before
determining the stop time
        N_LifeTimeComb = N_LifeTimeComb-Times(FieldLifeTime-Shift_time+2)-1;
        y_data_switch_CLRM = NPV_Tab_matrix(:,N_Phase1n2LifeTime-
N_LifeTimeComb+1:N_Phase1n2LifeTime);
        y_reg_switch_CLRM = zeros(N_MC,N_LifeTimeComb);
        ENPVTabComp = zeros(N_MC,N_LifeTimeComb);
        if Shift_time == 2 && Shift_time == 3
            x_data_cube_switch_CLRM = x_data_switch_CLRM(:,1:Shift_time-1);
            X_switch_CLRM = [ones(N_MC,1) x_data_switch_CLRM(:,1:Shift_time-1)];
        else
            x_data_cube_switch_CLRM = x_data_switch_CLRM(:,1:Shift_time-
3).*x_data_switch_CLRM(:,2:Shift_time-2).*x_data_switch_CLRM(:,3:Shift_time-1);
            X_switch_CLRM = [ones(N_MC,1) x_data_switch_CLRM(:,1:Shift_time-1)
x_data_cube_switch_CLRM];
        end
        for i_LifeTimeComb = 1: N_LifeTimeComb
            coef_reg_CLRM = regress(y_data_switch_CLRM(:,i_LifeTimeComb),X_switch_CLRM);
            y_reg_switch_CLRM(:,i_LifeTimeComb) = X_switch_CLRM*coef_reg_CLRM;
            ENPVTabComp(:,i_LifeTimeComb) = y_reg_switch_CLRM(:,i_LifeTimeComb);
        end

    [CLRM_reg,CLRM_idx] = max(ENPVTabComp,[],2);

    if Shift_time == FieldLifeTime
        [SwitchDecision_CLRM,SwitchDecision_CLRM_idx] = find(CLRM_idx <= Times(FieldLifeTime-
Shift_time+2)+1);
        [ContinueTime_reg,ContinueTime_idx] = find(CLRM_idx == Times(FieldLifeTime-
Shift_time+2)+1);
        for i_path = 1: length(ContinueTime_reg)
            if CLRM_OptNPV_real(ContinueTime_reg(i_path),1) == 0
               CLRM_OptNPV_real(ContinueTime_reg(i_path),1) =
y_data_switch_CLRM(ContinueTime_reg(i_path),end);
               CLRM_PhaseLifes_Opt_reals(ContinueTime_reg(i_path),1) = FieldLifeTime;
               CLRM_PhaseLifes_Opt_reals(ContinueTime_reg(i_path),2) = 0;
            else
                continue;
            end
        end
    else
        [SwitchDecision_CLRM,SwitchDecision_CLRM_idx] = find(CLRM_idx <= Times(FieldLifeTime-
Shift_time+2));
    end
```

```
    x_data_stop_CLRM = ObsRate_Phase1n2_real_time_ShiftTime(:,:,Shift_time);
    t_dataPhase2 = 1;
    while t_dataPhase2 <= Times(FieldLifeTime-Shift_time+1)+1
        if t_dataPhase2 == 1 && t_dataPhase2 == 2
            x_data_cube_stop_CLRM = x_data_stop_CLRM(:,1:t_dataPhase2);
            X_stop_CLRM = [ones(N_MC,1) x_data_stop_CLRM(:,1:t_dataPhase2+Shift_time-2)];
        else
            x_data_cube_stop_CLRM = x_data_stop_CLRM(:,1:t_dataPhase2-
2).*x_data_stop_CLRM(:,2:t_dataPhase2-1).*x_data_stop_CLRM(:,3:t_dataPhase2);
            X_stop_CLRM = [ones(N_MC,1) x_data_stop_CLRM(:,1:t_dataPhase2+Shift_time-2)
x_data_cube_stop_CLRM];
        end
        y_data_stop_CLRM = y_data_switch_CLRM(:, t_dataPhase2:Times(FieldLifeTime-
Shift_time+1)+1);
        y_reg_stop_CLRM = zeros(N_MC,Times(FieldLifeTime-Shift_time+1)-t_dataPhase2+2);
        for t = 1: Times(FieldLifeTime-Shift_time+1)-t_dataPhase2+2
            y_reg_stop_CLRM(:,t) = X_stop_CLRM*regress(y_data_stop_CLRM(:,t),X_stop_CLRM);
        end
            [StopTime_reg,StopTime_idx] = max(y_reg_stop_CLRM,[],2);
        for i_path = 1: length(SwitchDecision_CLRM)
            if StopTime_reg(SwitchDecision_CLRM(i_path)) ==
y_reg_stop_CLRM(SwitchDecision_CLRM(i_path),1) &&
CLRM_OptNPV_real(SwitchDecision_CLRM(i_path),1) == 0
                CLRM_OptNPV_real(SwitchDecision_CLRM(i_path),1) =
y_data_stop_CLRM(SwitchDecision_CLRM(i_path),1);
                CLRM_PhaseLifes_Opt_reals(SwitchDecision_CLRM(i_path),1) = Shift_time-1;
                CLRM_PhaseLifes_Opt_reals(SwitchDecision_CLRM(i_path),2) =
t_dataPhase2+StopTime_idx(SwitchDecision_CLRM(i_path))-2;
            else
                continue;
            end
        end
            t_dataPhase2 = t_dataPhase2+1;
    end
    end
end

DWI_CLRM_Phase1LifeTime = CLRM_PhaseLifes_Opt_reals(:,1);
DWI_CLRM_Phase2LifeTime = CLRM_PhaseLifes_Opt_reals(:,2);
DWI_CLRM_LifeTime = DWI_CLRM_Phase1LifeTime + DWI_CLRM_Phase2LifeTime;

EVWI_CLRM = mean(CLRM_OptNPV_real);
VOI_CLRM = EVWI_CLRM-EVWOI;
```

Appendix B2.6: Regression Function Equation (18)

This code consists of the modified version of the sections 5 and 6 of Modified LSM Algorithm (Appendix B1.1) and CLRM (Appendix B1.2). With respect to this, the modification is done on the regression analysis. Thus, sections 1 to 4 of Modified LSM Algorithm (Appendix B1.1) have to be run first before executing this code. This code corresponds to the Regression Function of Equation (18) in the report.

```
% 1. Path Table Generation

N_MC = size(ObsRate_Phase1n2_real_time_ShiftTime,1);
FieldLifeTime = size(ObsRate_Phase1n2_real_time_ShiftTime,2);
SRDM_Shift_Stop_OptNPV_real = zeros(N_MC,1);

PathTable = zeros(N_MC,FieldLifeTime+1);
PathTable(:,end) = NPVtable_LTPhase1_LTPhase2_real(FieldLifeTime+1,1,:);
Phase2_StopTime_real_Phase1LifeTime = zeros(N_MC,FieldLifeTime+1);

for k_Shift_time = 1: FieldLifeTime
        NPV_2Phases_matrix = zeros(N_MC,k_Shift_time+1);
    for k_Stop_time = 1: k_Shift_time+1
        NPV_2Phases_matrix(:,k_Stop_time) = NPVtable_LTPhase1_LTPhase2_real(FieldLifeTime-
k_Shift_time+1,k_Stop_time,:);
    end
    for k_StopTime = 1: k_Shift_time
        if k_StopTime == 1
            x_data_stop_SRDM = ObsRate_Phase1n2_real_time_ShiftTime(:,1:FieldLifeTime-
k_StopTime,FieldLifeTime-k_Shift_time+1);
            x_data_square_stop_SRDM = x_data_stop_SRDM(:,1:FieldLifeTime-k_StopTime-
1).*x_data_stop_SRDM(:,2:FieldLifeTime-k_StopTime);
            x_data_cube_stop_SRDM = x_data_stop_SRDM(:,1:FieldLifeTime-k_StopTime-
2).*x_data_stop_SRDM(:,2:FieldLifeTime-k_StopTime-1).*x_data_stop_SRDM(:,3:FieldLifeTime-
k_StopTime);
            X_stop_SRDM = [ones(N_MC,1) x_data_stop_SRDM x_data_square_stop_SRDM
x_data_cube_stop_SRDM];
            y_data_stop_matrix = [NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+1)
NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+2)];
            [Stop_value,Stop_idx] =
max([X_stop_SRDM*regress(NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+1),X_stop_SRDM)
X_stop_SRDM*regress(NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+2),X_stop_SRDM)],[],2);
            for i_MC = 1: N_MC
                SRDM_Shift_Stop_OptNPV_real(i_MC,1) =
y_data_stop_matrix(i_MC,Stop_idx(i_MC));
            end
        else
            x_data_stop_SRDM = ObsRate_Phase1n2_real_time_ShiftTime(:,1:FieldLifeTime-
k_StopTime,FieldLifeTime-k_Shift_time+1);
            x_data_square_stop_SRDM = x_data_stop_SRDM(:,1:FieldLifeTime-k_StopTime-
1).*x_data_stop_SRDM(:,2:FieldLifeTime-k_StopTime);
            x_data_cube_stop_SRDM = x_data_stop_SRDM(:,1:FieldLifeTime-k_StopTime-
2).*x_data_stop_SRDM(:,2:FieldLifeTime-k_StopTime-1).*x_data_stop_SRDM(:,3:FieldLifeTime-
k_StopTime);
            X_stop_SRDM = [ones(N_MC,1) x_data_stop_SRDM x_data_square_stop_SRDM
x_data_cube_stop_SRDM];
            y_data_stop_matrix = [NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+1)
```

```
SRDM_Shift_Stop_OptNPV_real(:,1)];
            [Stop_value,Stop_idx] =
max([X_stop_SRDM*regress(NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+1),X_stop_SRDM)
X_stop_SRDM*regress(SRDM_Shift_Stop_OptNPV_real(:,1),X_stop_SRDM)],[],2);
            for i_MC = 1: N_MC
                SRDM_Shift_Stop_OptNPV_real(i_MC,1) =
y_data_stop_matrix(i_MC,Stop_idx(i_MC));
            end
        end
    end
    for i_MC = 1: N_MC
        PathTable(i_MC,FieldLifeTime-k_Shift_time+1) = SRDM_Shift_Stop_OptNPV_real(i_MC,1);
        for k_Stop_time = 1: k_Shift_time+1
            if SRDM_Shift_Stop_OptNPV_real(i_MC,1) == NPV_2Phases_matrix(i_MC,k_Stop_time)
                Phase2_StopTime_real_Phase1LifeTime(i_MC,FieldLifeTime-k_Shift_time+1) =
k_Stop_time-1;
            end
        end
    end
end

% 2. SRDM Approach

FieldLifeTime = size(ObsRate_Phase1n2_real_time_ShiftTime,2);
y_data_continuation = zeros(N_MC,1);
y_data_shift = zeros(N_MC,1);
y_reg_continuation = zeros(N_MC,1);
y_reg_shift = zeros(N_MC,1);
x_data = ObsRate_Phase1n2_real_time_ShiftTime(:,:,FieldLifeTime+1);

ValueTable = zeros(N_MC,FieldLifeTime+1);
ValueTable(:,end) = PathTable(:,end);
SRDM_DecisionTable = zeros(N_MC,FieldLifeTime+1);
SRDM_DecisionTable(:,end) = 1;

for k_Shift_time = 1:FieldLifeTime
    x_data_square = x_data(:,1:FieldLifeTime-k_Shift_time-1).*x_data(:,2:FieldLifeTime-
k_Shift_time);
    x_data_cube = x_data(:,1:FieldLifeTime-k_Shift_time-2).*x_data(:,2:FieldLifeTime-
k_Shift_time-1).*x_data(:,3:FieldLifeTime-k_Shift_time);
    X = [ones(N_MC,1) x_data(:,1:FieldLifeTime-k_Shift_time) x_data_square x_data_cube];
    y_data_shift(:) = PathTable(:,FieldLifeTime-k_Shift_time+1);
    coef_shift = regress(y_data_shift,X);
    y_reg_shift(:) = X*coef_shift;
    y_data_continuation(:) = ValueTable(:,FieldLifeTime-k_Shift_time+2);
    coef_continuation = regress(y_data_continuation,X);
    y_reg_continuation(:) = X*coef_continuation;
    ENPVTable_Comparison = [y_reg_shift, y_reg_continuation];
    NPV_SRDM = [y_data_shift, y_data_continuation];
    [ENPV_SRDM,ENPV_SRDM_idx] = max(ENPVTable_Comparison,[],2);

    for i_MC = 1: N_MC
        ValueTable(i_MC,FieldLifeTime-k_Shift_time+1) = NPV_SRDM(i_MC,ENPV_SRDM_idx(i_MC));
        if y_reg_continuation(i_MC) > y_reg_shift(i_MC)
            SRDM_DecisionTable(i_MC,FieldLifeTime-k_Shift_time+1) = 0;
        else
            SRDM_DecisionTable(i_MC,FieldLifeTime-k_Shift_time+1) = 1;
        end
```

```
    end
end

ShiftValue = PathTable(:,1);
ExpShiftValue = mean(ShiftValue);

ContinuationValue = ValueTable(:,1);
ExpContinuationValue = mean(ContinuationValue);
EVWII = ExpContinuationValue;
VOI = EVWII - EVWOI;
VOI_over_EVWOI_pct = VOI/EVWOI*100;

[~,DWI_Phase1LifeTime_idx] = max(SRDM_DecisionTable,[],2);
DWI_Phase1LifeTime = DWI_Phase1LifeTime_idx-1;

DWI_Phase2LifeTime = zeros(N_MC,1);
for i_MC = 1:N_MC
    Phase1LifeTime_idx_real = DWI_Phase1LifeTime_idx(i_MC,1);
    DWI_Phase2LifeTime(i_MC,1) =
Phase2_StopTime_real_Phase1LifeTime(i_MC,Phase1LifeTime_idx_real);
end

DWI_LifeTime = DWI_Phase1LifeTime+DWI_Phase2LifeTime;

% 3. CLRM Approach

N_MC = size(ObsRate_Phase1n2_real_time_ShiftTime,1);
FieldLifeTime = size(ObsRate_Phase1n2_real_time_ShiftTime,2);
N_Phase1n2LifeTime = sum(1:FieldLifeTime+1);
N_LifeTimeComb = N_Phase1n2LifeTime;
Times = 1:1:FieldLifeTime;
CLRM_OptNPV_real = zeros(N_MC,1);
NPV_Tab_matrix = transpose(NPVmatrix_reals);
Prior_ENPV = meanNPVvector(:,3);
[max_Prior_ENPV, max_Prior_ENPV_idx] = max(meanNPVvector(:,3),[],1);
x_data_switch_CLRM = ObsRate_Phase1n2_real_time_ShiftTime(:,:,FieldLifeTime+1);

CLRM_PhaseLifes_Opt_reals = zeros(N_MC,2);

if max_Prior_ENPV_idx <= Times(FieldLifeTime)+1 %Finding if Switch Time is optimal at Y1
    Shift_time = 1;
    t_dataPhase2 = 1;
    while t_dataPhase2 <= Times(FieldLifeTime-Shift_time+1)
        if t_dataPhase2 == 1 && max_Prior_ENPV_idx == 1 %Switch and Stop at Y1 are optimal
            for i_MC = 1: N_MC
                CLRM_OptNPV_real(i_MC,1) = max_Prior_ENPV;
            end
            break; %The whole loop will break here
        else %Given Switch at Y1 and Stop is not at Y1
            x_data_stop_CLRM = ObsRate_Phase1n2_real_time_ShiftTime(:,:,Shift_time);
            if t_dataPhase2 == 1
                x_data_square_stop_CLRM = x_data_stop_CLRM(:,1:t_dataPhase2);
                X_stop_CLRM = [ones(N_MC,1) x_data_stop_CLRM(:,1:t_dataPhase2)];
            elseif t_dataPhase2 == 2
                x_data_square_stop_CLRM = x_data_stop_CLRM(:,1:t_dataPhase2-
1).*x_data_stop_CLRM(:,2:t_dataPhase2);
                X_stop_CLRM = [ones(N_MC,1) x_data_stop_CLRM(:,1:t_dataPhase2)
x_data_square_stop_CLRM];
```

```
            else
                x_data_square_stop_CLRM = x_data_stop_CLRM(:,1:t_dataPhase2-
1).*x_data_stop_CLRM(:,2:t_dataPhase2);
                x_data_cube_stop_CLRM = x_data_stop_CLRM(:,1:t_dataPhase2-
2).*x_data_stop_CLRM(:,2:t_dataPhase2-1).*x_data_stop_CLRM(:,3:t_dataPhase2);
                X_stop_CLRM = [ones(N_MC,1) x_data_stop_CLRM(:,1:t_dataPhase2)
x_data_square_stop_CLRM x_data_cube_stop_CLRM];
            end
            y_data_stop_CLRM = NPV_Tab_matrix(:,t_dataPhase2+1:FieldLifeTime+1);
            y_reg_stop_CLRM = zeros(N_MC,FieldLifeTime-t_dataPhase2+1);
            for t = 1: FieldLifeTime-t_dataPhase2+1
                y_reg_stop_CLRM(:,t) =
X_stop_CLRM*regress(y_data_stop_CLRM(:,t),X_stop_CLRM);
            end
            [StopTime_reg,StopTime_idx] = max(y_reg_stop_CLRM,[],2);
        for i_MC = 1: N_MC
            if StopTime_reg(i_MC) == y_reg_stop_CLRM(i_MC,1) && CLRM_OptNPV_real(i_MC,1) == 0
                CLRM_OptNPV_real(i_MC,1) = y_data_stop_CLRM(i_MC,1);
                CLRM_PhaseLifes_Opt_reals(i_MC,1) = Shift_time-1;
                CLRM_PhaseLifes_Opt_reals(i_MC,2) = t_dataPhase2+StopTime_idx(i_MC)-1;
            else
                continue;
            end
        end
        end
        t_dataPhase2 = t_dataPhase2+1;
    end
else
% This part is to find the corresponding optimal stop time given Switch time is not at Y1
    for Shift_time = 2: FieldLifeTime %Finding and recording the optimal switch time before
determining the stop time
        N_LifeTimeComb = N_LifeTimeComb-Times(FieldLifeTime-Shift_time+2)-1;
        y_data_switch_CLRM = NPV_Tab_matrix(:,N_Phase1n2LifeTime-
N_LifeTimeComb+1:N_Phase1n2LifeTime);
        y_reg_switch_CLRM = zeros(N_MC,N_LifeTimeComb);
        ENPVTabComp = zeros(N_MC,N_LifeTimeComb);
        if Shift_time == 2
            x_data_square_switch_CLRM = x_data_switch_CLRM(:,1:Shift_time-1);
            X_switch_CLRM = [ones(N_MC,1) x_data_switch_CLRM(:,1:Shift_time-1)];
        elseif Shift_time == 3
            x_data_square_switch_CLRM = x_data_switch_CLRM(:,1:Shift_time-
2).*x_data_switch_CLRM(:,2:Shift_time-1);
            X_switch_CLRM = [ones(N_MC,1) x_data_switch_CLRM(:,1:Shift_time-1)
x_data_square_switch_CLRM];
        else
            x_data_square_switch_CLRM = x_data_switch_CLRM(:,1:Shift_time-
2).*x_data_switch_CLRM(:,2:Shift_time-1);
            x_data_cube_switch_CLRM = x_data_switch_CLRM(:,1:Shift_time-
3).*x_data_switch_CLRM(:,2:Shift_time-2).*x_data_switch_CLRM(:,3:Shift_time-1); %%Check
here!!!!
            X_switch_CLRM = [ones(N_MC,1) x_data_switch_CLRM(:,1:Shift_time-1)
x_data_square_switch_CLRM x_data_cube_switch_CLRM];
        end
        for i_LifeTimeComb = 1: N_LifeTimeComb
            coef_reg_CLRM = regress(y_data_switch_CLRM(:,i_LifeTimeComb),X_switch_CLRM);
            y_reg_switch_CLRM(:,i_LifeTimeComb) = X_switch_CLRM*coef_reg_CLRM;
            ENPVTabComp(:,i_LifeTimeComb) = y_reg_switch_CLRM(:,i_LifeTimeComb);
        end
```

```
    [CLRM_reg,CLRM_idx] = max(ENPVTabComp,[],2);

    if Shift_time == FieldLifeTime
        [SwitchDecision_CLRM,SwitchDecision_CLRM_idx] = find(CLRM_idx <= Times(FieldLifeTime-
Shift_time+2)+1);
        [ContinueTime_reg,ContinueTime_idx] = find(CLRM_idx == Times(FieldLifeTime-
Shift_time+2)+1);
        for i_path = 1: length(ContinueTime_reg)
            if CLRM_OptNPV_real(ContinueTime_reg(i_path),1) == 0
               CLRM_OptNPV_real(ContinueTime_reg(i_path),1) =
y_data_switch_CLRM(ContinueTime_reg(i_path),end);
               CLRM_PhaseLifes_Opt_reals(ContinueTime_reg(i_path),1) = FieldLifeTime;
               CLRM_PhaseLifes_Opt_reals(ContinueTime_reg(i_path),2) = 0;
            else
                continue;
            end
        end
    else
        [SwitchDecision_CLRM,SwitchDecision_CLRM_idx] = find(CLRM_idx <= Times(FieldLifeTime-
Shift_time+2));
    end

    x_data_stop_CLRM = ObsRate_Phase1n2_real_time_ShiftTime(:,:,Shift_time);
    t_dataPhase2 = 1;
    while t_dataPhase2 <= Times(FieldLifeTime-Shift_time+1)+1
        if t_dataPhase2 == 1
            x_data_square_stop_CLRM = x_data_stop_CLRM(:,1:t_dataPhase2);
            X_stop_CLRM = [ones(N_MC,1) x_data_stop_CLRM(:,1:t_dataPhase2+Shift_time-2)];
        elseif t_dataPhase2 == 2
            x_data_square_stop_CLRM = x_data_stop_CLRM(:,1:t_dataPhase2-
1).*x_data_stop_CLRM(:,2:t_dataPhase2);
            X_stop_CLRM = [ones(N_MC,1) x_data_stop_CLRM(:,1:t_dataPhase2+Shift_time-2)
x_data_square_stop_CLRM];
        else
            x_data_square_stop_CLRM = x_data_stop_CLRM(:,1:t_dataPhase2-
1).*x_data_stop_CLRM(:,2:t_dataPhase2);
            x_data_cube_stop_CLRM = x_data_stop_CLRM(:,1:t_dataPhase2-
2).*x_data_stop_CLRM(:,2:t_dataPhase2-1).*x_data_stop_CLRM(:,3:t_dataPhase2);
            X_stop_CLRM = [ones(N_MC,1) x_data_stop_CLRM(:,1:t_dataPhase2+Shift_time-2)
x_data_square_stop_CLRM x_data_cube_stop_CLRM];
        end
        y_data_stop_CLRM = y_data_switch_CLRM(:, t_dataPhase2:Times(FieldLifeTime-
Shift_time+1)+1);
        y_reg_stop_CLRM = zeros(N_MC,Times(FieldLifeTime-Shift_time+1)-t_dataPhase2+2);
        for t = 1: Times(FieldLifeTime-Shift_time+1)-t_dataPhase2+2
            y_reg_stop_CLRM(:,t) = X_stop_CLRM*regress(y_data_stop_CLRM(:,t),X_stop_CLRM);
        end
            [StopTime_reg,StopTime_idx] = max(y_reg_stop_CLRM,[],2);
        for i_path = 1: length(SwitchDecision_CLRM)
            if StopTime_reg(SwitchDecision_CLRM(i_path)) ==
y_reg_stop_CLRM(SwitchDecision_CLRM(i_path),1) &&
CLRM_OptNPV_real(SwitchDecision_CLRM(i_path),1) == 0
                CLRM_OptNPV_real(SwitchDecision_CLRM(i_path),1) =
y_data_stop_CLRM(SwitchDecision_CLRM(i_path),1);
                CLRM_PhaseLifes_Opt_reals(SwitchDecision_CLRM(i_path),1) = Shift_time-1;
                CLRM_PhaseLifes_Opt_reals(SwitchDecision_CLRM(i_path),2) =
t_dataPhase2+StopTime_idx(SwitchDecision_CLRM(i_path))-2;
```

```
            else
                continue;
            end
        end
            t_dataPhase2 = t_dataPhase2+1;
    end
    end
end

DWI_CLRM_Phase1LifeTime = CLRM_PhaseLifes_Opt_reals(:,1);
DWI_CLRM_Phase2LifeTime = CLRM_PhaseLifes_Opt_reals(:,2);
DWI_CLRM_LifeTime = DWI_CLRM_Phase1LifeTime + DWI_CLRM_Phase2LifeTime;

EVWI_CLRM = mean(CLRM_OptNPV_real);
VOI_CLRM = EVWI_CLRM-EVWOI;
```

Appendix B2.7: Laguerre Polynomials without dependency terms (without renormalization)

This code consists of the modified version of the sections 5 and 6 of Modified LSM Algorithm (Appendix B1.1) and CLRM (Appendix B1.2). With respect to this the modification is done on the regression analysis. Thus, sections 1 to 4 of Modified LSM Algorithm (Appendix B1.1) have to be run first before executing this code. This code corresponds to the Laguerre Polynomials without dependency terms (no renormalization) in the report.

```
% 1. Path Table Generation

N_MC = size(ObsRate_Phase1n2_real_time_ShiftTime,1);
FieldLifeTime = size(ObsRate_Phase1n2_real_time_ShiftTime,2);
SRDM_Shift_Stop_OptNPV_real = zeros(N_MC,1);

PathTable = zeros(N_MC,FieldLifeTime+1);
PathTable(:,end) = NPVtable_LTPhase1_LTPhase2_real(FieldLifeTime+1,1,:);
Phase2_StopTime_real_Phase1LifeTime = zeros(N_MC,FieldLifeTime+1);

for k_Shift_time = 1: FieldLifeTime
        NPV_2Phases_matrix = zeros(N_MC,k_Shift_time+1);
    for k_Stop_time = 1: k_Shift_time+1
        NPV_2Phases_matrix(:,k_Stop_time) = NPVtable_LTPhase1_LTPhase2_real(FieldLifeTime-
k_Shift_time+1,k_Stop_time,:);
    end
    for k_StopTime = 1: k_Shift_time
        if k_StopTime == 1
            x_data_stop_SRDM = ObsRate_Phase1n2_real_time_ShiftTime(:,1:FieldLifeTime-
k_StopTime,FieldLifeTime-k_Shift_time+1);
            Laguerre_PathTable_0 = exp(-x_data_stop_SRDM/2);
            Laguerre_PathTable_1 = exp(-x_data_stop_SRDM/2).*(1-x_data_stop_SRDM);
            Laguerre_PathTable_2 = exp(-x_data_stop_SRDM/2).*(1-
2.*x_data_stop_SRDM+0.5.*(x_data_stop_SRDM.^2));

            X_stop_SRDM = [ones(N_MC,1) Laguerre_PathTable_0 Laguerre_PathTable_1
Laguerre_PathTable_2];
            y_data_stop_matrix = [NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+1)
NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+2)];
            [Stop_value,Stop_idx] =
max([X_stop_SRDM*regress(NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+1),X_stop_SRDM)
X_stop_SRDM*regress(NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+2),X_stop_SRDM)],[],2);
            for i_MC = 1: N_MC
                SRDM_Shift_Stop_OptNPV_real(i_MC,1) =
y_data_stop_matrix(i_MC,Stop_idx(i_MC));
            end
        else
            x_data_stop_SRDM = ObsRate_Phase1n2_real_time_ShiftTime(:,1:FieldLifeTime-
k_StopTime,FieldLifeTime-k_Shift_time+1);
            Laguerre_PathTable_0 = exp(-x_data_stop_SRDM/2);
            Laguerre_PathTable_1 = exp(-x_data_stop_SRDM/2).*(1-x_data_stop_SRDM);
            Laguerre_PathTable_2 = exp(-x_data_stop_SRDM/2).*(1-
2.*x_data_stop_SRDM+0.5.*(x_data_stop_SRDM.^2));

            X_stop_SRDM = [ones(N_MC,1) Laguerre_PathTable_0 Laguerre_PathTable_1
Laguerre_PathTable_2];
```

```
            y_data_stop_matrix = [NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+1)
SRDM_Shift_Stop_OptNPV_real(:,1)];
            [Stop_value,Stop_idx] =
max([X_stop_SRDM*regress(NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+1),X_stop_SRDM)
X_stop_SRDM*regress(SRDM_Shift_Stop_OptNPV_real(:,1),X_stop_SRDM)],[],2);
            for i_MC = 1: N_MC
                SRDM_Shift_Stop_OptNPV_real(i_MC,1) =
y_data_stop_matrix(i_MC,Stop_idx(i_MC));
            end
        end
    end
    for i_MC = 1: N_MC
        PathTable(i_MC,FieldLifeTime-k_Shift_time+1) = SRDM_Shift_Stop_OptNPV_real(i_MC,1);
        for k_Stop_time = 1: k_Shift_time+1
            if SRDM_Shift_Stop_OptNPV_real(i_MC,1) == NPV_2Phases_matrix(i_MC,k_Stop_time)
                Phase2_StopTime_real_Phase1LifeTime(i_MC,FieldLifeTime-k_Shift_time+1) =
k_Stop_time-1;
            end
        end
    end
end

% 2. SRDM Apprach

FieldLifeTime = size(ObsRate_Phase1n2_real_time_ShiftTime,2);
y_data_continuation = zeros(N_MC,1);
y_data_shift = zeros(N_MC,1);
y_reg_continuation = zeros(N_MC,1);
y_reg_shift = zeros(N_MC,1);
x_data = ObsRate_Phase1n2_real_time_ShiftTime(:,:,FieldLifeTime+1);

ValueTable = zeros(N_MC,FieldLifeTime+1);
ValueTable(:,end) = PathTable(:,end);
SRDM_DecisionTable = zeros(N_MC,FieldLifeTime+1);
SRDM_DecisionTable(:,end) = 1;

for k_Shift_time = 1:FieldLifeTime
    Laguerre_SRDM_0 = exp(-x_data(:,1:FieldLifeTime-k_Shift_time)/2);
    Laguerre_SRDM_1 = exp(-x_data(:,1:FieldLifeTime-k_Shift_time)/2).*(1-
x_data(:,1:FieldLifeTime-k_Shift_time));
    Laguerre_SRDM_2 = exp(-x_data(:,1:FieldLifeTime-k_Shift_time)/2).*(1-
2.*x_data(:,1:FieldLifeTime-k_Shift_time)+0.5.*x_data(:,1:FieldLifeTime-k_Shift_time).^2);

    X = [ones(N_MC,1) Laguerre_SRDM_0 Laguerre_SRDM_1 Laguerre_SRDM_2];
    y_data_shift(:) = PathTable(:,FieldLifeTime-k_Shift_time+1);
    coef_shift = regress(y_data_shift,X);
    y_reg_shift(:) = X*coef_shift;
    y_data_continuation(:) = ValueTable(:,FieldLifeTime-k_Shift_time+2);
    coef_continuation = regress(y_data_continuation,X);
    y_reg_continuation(:) = X*coef_continuation;
    ENPVTable_Comparison = [y_reg_shift, y_reg_continuation];
    NPV_SRDM = [y_data_shift, y_data_continuation];
    [ENPV_SRDM,ENPV_SRDM_idx] = max(ENPVTable_Comparison,[],2);

    for i_MC = 1: N_MC
        ValueTable(i_MC,FieldLifeTime-k_Shift_time+1) = NPV_SRDM(i_MC,ENPV_SRDM_idx(i_MC));
        if y_reg_continuation(i_MC) > y_reg_shift(i_MC)
            SRDM_DecisionTable(i_MC,FieldLifeTime-k_Shift_time+1) = 0;
```

```
        else
            SRDM_DecisionTable(i_MC,FieldLifeTime-k_Shift_time+1) = 1;
        end
    end
end

ShiftValue = PathTable(:,1);
ExpShiftValue = mean(ShiftValue);

ContinuationValue = ValueTable(:,1);
ExpContinuationValue = mean(ContinuationValue);
EVWII = ExpContinuationValue;
VOI = EVWII - EVWOI;
VOI_over_EVWOI_pct = VOI/EVWOI*100;

[~,DWI_Phase1LifeTime_idx] = max(SRDM_DecisionTable,[],2);
DWI_Phase1LifeTime = DWI_Phase1LifeTime_idx-1;

DWI_Phase2LifeTime = zeros(N_MC,1);
for i_MC = 1:N_MC
    Phase1LifeTime_idx_real = DWI_Phase1LifeTime_idx(i_MC,1);
    DWI_Phase2LifeTime(i_MC,1) =
Phase2_StopTime_real_Phase1LifeTime(i_MC,Phase1LifeTime_idx_real);
end

DWI_LifeTime = DWI_Phase1LifeTime+DWI_Phase2LifeTime;

% 3. CLRM Approach

N_MC = size(ObsRate_Phase1n2_real_time_ShiftTime,1);
FieldLifeTime = size(ObsRate_Phase1n2_real_time_ShiftTime,2);
N_Phase1n2LifeTime = sum(1:FieldLifeTime+1);
N_LifeTimeComb = N_Phase1n2LifeTime;
Times = 1:1:FieldLifeTime;
CLRM_OptNPV_real = zeros(N_MC,1);
NPV_Tab_matrix = transpose(NPVmatrix_reals);
Prior_ENPV = meanNPVvector(:,3);
[max_Prior_ENPV, max_Prior_ENPV_idx] = max(meanNPVvector(:,3),[],1);
x_data_switch_CLRM = ObsRate_Phase1n2_real_time_ShiftTime(:,:,FieldLifeTime+1);

CLRM_PhaseLifes_Opt_reals = zeros(N_MC,2);

if max_Prior_ENPV_idx <= Times(FieldLifeTime)+1 %Finding if Switch Time is optimal at Y1
    Shift_time = 1;
    t_dataPhase2 = 1;
    while t_dataPhase2 <= Times(FieldLifeTime-Shift_time+1)
        if t_dataPhase2 == 1 && max_Prior_ENPV_idx == 1 %Switch and Stop at Y1 are optimal
            for i_MC = 1: N_MC
                CLRM_OptNPV_real(i_MC,1) = max_Prior_ENPV;
            end
            break; %The whole loop will break here
        else %Given Switch at Y1 and Stop is not at Y1
            x_data_stop_CLRM = ObsRate_Phase1n2_real_time_ShiftTime(:,:,Shift_time);
            Laguerre_CLRM_0 = exp(-x_data_stop_CLRM(:,1:t_dataPhase2)/2);
            Laguerre_CLRM_1 = exp(-x_data_stop_CLRM(:,1:t_dataPhase2)/2).*(1-
x_data_stop_CLRM(:,1:t_dataPhase2));
            Laguerre_CLRM_2 = exp(-x_data_stop_CLRM(:,1:t_dataPhase2)/2).*(1-
2.*x_data_stop_CLRM(:,1:t_dataPhase2)+0.5.*(x_data_stop_CLRM(:,1:t_dataPhase2).^2));
```

```
            X_stop_CLRM = [ones(N_MC,1) Laguerre_CLRM_0 Laguerre_CLRM_1 Laguerre_CLRM_2];
            y_data_stop_CLRM = NPV_Tab_matrix(:,t_dataPhase2+1:FieldLifeTime+1);
            y_reg_stop_CLRM = zeros(N_MC,FieldLifeTime-t_dataPhase2+1);
            for t = 1: FieldLifeTime-t_dataPhase2+1
                y_reg_stop_CLRM(:,t) =
X_stop_CLRM*regress(y_data_stop_CLRM(:,t),X_stop_CLRM);
            end
            [StopTime_reg,StopTime_idx] = max(y_reg_stop_CLRM,[],2);
        for i_MC = 1: N_MC
            if StopTime_reg(i_MC) == y_reg_stop_CLRM(i_MC,1) && CLRM_OptNPV_real(i_MC,1) == 0
                CLRM_OptNPV_real(i_MC,1) = y_data_stop_CLRM(i_MC,1);
                CLRM_PhaseLifes_Opt_reals(i_MC,1) = Shift_time-1;
                CLRM_PhaseLifes_Opt_reals(i_MC,2) = t_dataPhase2+StopTime_idx(i_MC)-1;
            else
                continue;
            end
        end
        end
        t_dataPhase2 = t_dataPhase2+1;
    end
else
% This part is to find the corresponding optimal stop time given Switch time is not at Y1
    for Shift_time = 2: FieldLifeTime %Finding and recording the optimal switch time before
determining the stop time
        N_LifeTimeComb = N_LifeTimeComb-Times(FieldLifeTime-Shift_time+2)-1;
        y_data_switch_CLRM = NPV_Tab_matrix(:,N_Phase1n2LifeTime-
N_LifeTimeComb+1:N_Phase1n2LifeTime);
        y_reg_switch_CLRM = zeros(N_MC,N_LifeTimeComb);
        ENPVTabComp = zeros(N_MC,N_LifeTimeComb);
        Laguerre_CLRM_0 = exp(-x_data_switch_CLRM(:,1:Shift_time-1)/2);
        Laguerre_CLRM_1 = exp(-x_data_switch_CLRM(:,1:Shift_time-1)/2).*(1-
x_data_switch_CLRM(:,1:Shift_time-1));
        Laguerre_CLRM_2 = exp(-x_data_switch_CLRM(:,1:Shift_time-1)/2).*(1-
2.*x_data_switch_CLRM(:,1:Shift_time-1)+0.5.*x_data_switch_CLRM(:,1:Shift_time-1).^2);

        X_switch_CLRM = [ones(N_MC,1) Laguerre_CLRM_0 Laguerre_CLRM_1 Laguerre_CLRM_2];
        for i_LifeTimeComb = 1: N_LifeTimeComb
            coef_reg_CLRM = regress(y_data_switch_CLRM(:,i_LifeTimeComb),X_switch_CLRM);
            y_reg_switch_CLRM(:,i_LifeTimeComb) = X_switch_CLRM*coef_reg_CLRM;
            ENPVTabComp(:,i_LifeTimeComb) = y_reg_switch_CLRM(:,i_LifeTimeComb);
        end

    [CLRM_reg,CLRM_idx] = max(ENPVTabComp,[],2);

    if Shift_time == FieldLifeTime
        [SwitchDecision_CLRM,SwitchDecision_CLRM_idx] = find(CLRM_idx <= Times(FieldLifeTime-
Shift_time+2)+1);
        [ContinueTime_reg,ContinueTime_idx] = find(CLRM_idx == Times(FieldLifeTime-
Shift_time+2)+1);
        for i_path = 1: length(ContinueTime_reg)
            if CLRM_OptNPV_real(ContinueTime_reg(i_path),1) == 0
               CLRM_OptNPV_real(ContinueTime_reg(i_path),1) =
y_data_switch_CLRM(ContinueTime_reg(i_path),end);
               CLRM_PhaseLifes_Opt_reals(ContinueTime_reg(i_path),1) = FieldLifeTime;
               CLRM_PhaseLifes_Opt_reals(ContinueTime_reg(i_path),2) = 0;
            else
                continue;
```

```
            end
        end
    else
        [SwitchDecision_CLRM,SwitchDecision_CLRM_idx] = find(CLRM_idx <= Times(FieldLifeTime-
Shift_time+2));
    end

    x_data_stop_CLRM = ObsRate_Phase1n2_real_time_ShiftTime(:,:,Shift_time);
    t_dataPhase2 = 1;
    while t_dataPhase2 <= Times(FieldLifeTime-Shift_time+1)+1
        Laguerre_CLRM_0 = exp(-x_data_stop_CLRM(:,1:t_dataPhase2+Shift_time-2)/2);
        Laguerre_CLRM_1 = exp(-x_data_stop_CLRM(:,1:t_dataPhase2+Shift_time-2)/2).*(1-
x_data_stop_CLRM(:,1:t_dataPhase2+Shift_time-2));
        Laguerre_CLRM_2 = exp(-x_data_stop_CLRM(:,1:t_dataPhase2+Shift_time-2)/2).*(1-
2.*x_data_stop_CLRM(:,1:t_dataPhase2+Shift_time-
2)+0.5.*x_data_stop_CLRM(:,1:t_dataPhase2+Shift_time-2).^2);

        X_stop_CLRM = [ones(N_MC,1) Laguerre_CLRM_0 Laguerre_CLRM_1 Laguerre_CLRM_2];
        y_data_stop_CLRM = y_data_switch_CLRM(:, t_dataPhase2:Times(FieldLifeTime-
Shift_time+1)+1);
        y_reg_stop_CLRM = zeros(N_MC,Times(FieldLifeTime-Shift_time+1)-t_dataPhase2+2);
        for t = 1: Times(FieldLifeTime-Shift_time+1)-t_dataPhase2+2
            y_reg_stop_CLRM(:,t) = X_stop_CLRM*regress(y_data_stop_CLRM(:,t),X_stop_CLRM);
        end
            [StopTime_reg,StopTime_idx] = max(y_reg_stop_CLRM,[],2);
        for i_path = 1: length(SwitchDecision_CLRM)
            if StopTime_reg(SwitchDecision_CLRM(i_path)) ==
y_reg_stop_CLRM(SwitchDecision_CLRM(i_path),1) &&
CLRM_OptNPV_real(SwitchDecision_CLRM(i_path),1) == 0
                CLRM_OptNPV_real(SwitchDecision_CLRM(i_path),1) =
y_data_stop_CLRM(SwitchDecision_CLRM(i_path),1);
                CLRM_PhaseLifes_Opt_reals(SwitchDecision_CLRM(i_path),1) = Shift_time-1;
                CLRM_PhaseLifes_Opt_reals(SwitchDecision_CLRM(i_path),2) =
t_dataPhase2+StopTime_idx(SwitchDecision_CLRM(i_path))-2;
            else
                continue;
            end
        end
            t_dataPhase2 = t_dataPhase2+1;
    end
    end
end

DWI_CLRM_Phase1LifeTime = CLRM_PhaseLifes_Opt_reals(:,1);
DWI_CLRM_Phase2LifeTime = CLRM_PhaseLifes_Opt_reals(:,2);
DWI_CLRM_LifeTime = DWI_CLRM_Phase1LifeTime + DWI_CLRM_Phase2LifeTime;

EVWI_CLRM = mean(CLRM_OptNPV_real);
VOI_CLRM = EVWI_CLRM-EVWOI;
```

Appendix B2.8: Laguerre Polynomials with dependency terms (without renormalization)

This code consists of the modified version of the sections 5 and 6 of Modified LSM Algorithm (Appendix B1.1) and CLRM (Appendix B1.2). With respect to this the modification is done on the regression analysis. Thus, sections 1 to 4 of Modified LSM Algorithm (Appendix B1.1) have to be run first before executing this code. This code corresponds to the Laguerre Polynomials with dependency terms (no renormalization) in the report. The dependency terms correspond to those included in Equation (13).

```
% 1. Path Table Generation

N_MC = size(ObsRate_Phase1n2_real_time_ShiftTime,1);
FieldLifeTime = size(ObsRate_Phase1n2_real_time_ShiftTime,2);
SRDM_Shift_Stop_OptNPV_real = zeros(N_MC,1);

PathTable = zeros(N_MC,FieldLifeTime+1);
PathTable(:,end) = NPVtable_LTPhase1_LTPhase2_real(FieldLifeTime+1,1,:);
Phase2_StopTime_real_Phase1LifeTime = zeros(N_MC,FieldLifeTime+1);

for k_Shift_time = 1: FieldLifeTime
        NPV_2Phases_matrix = zeros(N_MC,k_Shift_time+1);
    for k_Stop_time = 1: k_Shift_time+1
        NPV_2Phases_matrix(:,k_Stop_time) = NPVtable_LTPhase1_LTPhase2_real(FieldLifeTime-
k_Shift_time+1,k_Stop_time,:);
    end
    for k_StopTime = 1: k_Shift_time
        if k_StopTime == 1
            x_data_stop_SRDM = ObsRate_Phase1n2_real_time_ShiftTime(:,1:FieldLifeTime-
k_StopTime,FieldLifeTime-k_Shift_time+1);
            Laguerre_PathTable_0 = exp(-x_data_stop_SRDM/2);
            Laguerre_PathTable_1 = exp(-x_data_stop_SRDM/2).*(1-x_data_stop_SRDM);
            Laguerre_PathTable_2 = exp(-x_data_stop_SRDM/2).*(1-
2.*x_data_stop_SRDM+0.5.*(x_data_stop_SRDM.^2));

            x_data_sqrt_stop_SRDM = sqrt(x_data_stop_SRDM(:,1:FieldLifeTime-k_StopTime-
1)).*sqrt(x_data_stop_SRDM(:,2:FieldLifeTime-k_StopTime));
            X_stop_SRDM = [ones(N_MC,1) Laguerre_PathTable_0 Laguerre_PathTable_1
Laguerre_PathTable_2 x_data_sqrt_stop_SRDM];
            y_data_stop_matrix = [NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+1)
NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+2)];
            [Stop_value,Stop_idx] =
max([X_stop_SRDM*regress(NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+1),X_stop_SRDM)
X_stop_SRDM*regress(NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+2),X_stop_SRDM)],[],2);
            for i_MC = 1: N_MC
                SRDM_Shift_Stop_OptNPV_real(i_MC,1) =
y_data_stop_matrix(i_MC,Stop_idx(i_MC));
            end
        else
            x_data_stop_SRDM = ObsRate_Phase1n2_real_time_ShiftTime(:,1:FieldLifeTime-
k_StopTime,FieldLifeTime-k_Shift_time+1);
            Laguerre_PathTable_0 = exp(-x_data_stop_SRDM/2);
            Laguerre_PathTable_1 = exp(-x_data_stop_SRDM/2).*(1-x_data_stop_SRDM);
            Laguerre_PathTable_2 = exp(-x_data_stop_SRDM/2).*(1-
```

```
2.*x_data_stop_SRDM+0.5.*(x_data_stop_SRDM.^2));

            x_data_sqrt_stop_SRDM = sqrt(x_data_stop_SRDM(:,1:FieldLifeTime-k_StopTime-
1)).*sqrt(x_data_stop_SRDM(:,2:FieldLifeTime-k_StopTime));
            X_stop_SRDM = [ones(N_MC,1) Laguerre_PathTable_0 Laguerre_PathTable_1
Laguerre_PathTable_2 x_data_sqrt_stop_SRDM];
            y_data_stop_matrix = [NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+1)
SRDM_Shift_Stop_OptNPV_real(:,1)];
            [Stop_value,Stop_idx] =
max([X_stop_SRDM*regress(NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+1),X_stop_SRDM)
X_stop_SRDM*regress(SRDM_Shift_Stop_OptNPV_real(:,1),X_stop_SRDM)],[],2);
            for i_MC = 1: N_MC
                SRDM_Shift_Stop_OptNPV_real(i_MC,1) =
y_data_stop_matrix(i_MC,Stop_idx(i_MC));
            end
        end
    end
    for i_MC = 1: N_MC
        PathTable(i_MC,FieldLifeTime-k_Shift_time+1) = SRDM_Shift_Stop_OptNPV_real(i_MC,1);
        for k_Stop_time = 1: k_Shift_time+1
            if SRDM_Shift_Stop_OptNPV_real(i_MC,1) == NPV_2Phases_matrix(i_MC,k_Stop_time)
                Phase2_StopTime_real_Phase1LifeTime(i_MC,FieldLifeTime-k_Shift_time+1) =
k_Stop_time-1;
            end
        end
    end
end

% 2. SRDM Approach

FieldLifeTime = size(ObsRate_Phase1n2_real_time_ShiftTime,2);
y_data_continuation = zeros(N_MC,1);
y_data_shift = zeros(N_MC,1);
y_reg_continuation = zeros(N_MC,1);
y_reg_shift = zeros(N_MC,1);
x_data = ObsRate_Phase1n2_real_time_ShiftTime(:,:,FieldLifeTime+1);

ValueTable = zeros(N_MC,FieldLifeTime+1);
ValueTable(:,end) = PathTable(:,end);
SRDM_DecisionTable = zeros(N_MC,FieldLifeTime+1);
SRDM_DecisionTable(:,end) = 1;

for k_Shift_time = 1:FieldLifeTime
    Laguerre_SRDM_0 = exp(-x_data(:,1:FieldLifeTime-k_Shift_time)/2);
    Laguerre_SRDM_1 = exp(-x_data(:,1:FieldLifeTime-k_Shift_time)/2).*(1-
x_data(:,1:FieldLifeTime-k_Shift_time));
    Laguerre_SRDM_2 = exp(-x_data(:,1:FieldLifeTime-k_Shift_time)/2).*(1-
2.*x_data(:,1:FieldLifeTime-k_Shift_time)+0.5.*x_data(:,1:FieldLifeTime-k_Shift_time).^2);
    x_data_sqrt = sqrt(x_data(:,1:FieldLifeTime-k_Shift_time-
1)).*sqrt(x_data(:,2:FieldLifeTime-k_Shift_time));
    X = [ones(N_MC,1) Laguerre_SRDM_0 Laguerre_SRDM_1 Laguerre_SRDM_2 x_data_sqrt];
    y_data_shift(:) = PathTable(:,FieldLifeTime-k_Shift_time+1);
    coef_shift = regress(y_data_shift,X);
    y_reg_shift(:) = X*coef_shift;
    y_data_continuation(:) = ValueTable(:,FieldLifeTime-k_Shift_time+2);
    coef_continuation = regress(y_data_continuation,X);
    y_reg_continuation(:) = X*coef_continuation;
    ENPVTable_Comparison = [y_reg_shift, y_reg_continuation];
```

```
    NPV_SRDM = [y_data_shift, y_data_continuation];
    [ENPV_SRDM,ENPV_SRDM_idx] = max(ENPVTable_Comparison,[],2);

    for i_MC = 1: N_MC
        ValueTable(i_MC,FieldLifeTime-k_Shift_time+1) = NPV_SRDM(i_MC,ENPV_SRDM_idx(i_MC));
        if y_reg_continuation(i_MC) > y_reg_shift(i_MC)
            SRDM_DecisionTable(i_MC,FieldLifeTime-k_Shift_time+1) = 0;
        else
            SRDM_DecisionTable(i_MC,FieldLifeTime-k_Shift_time+1) = 1;
        end
    end
end

ShiftValue = PathTable(:,1);
ExpShiftValue = mean(ShiftValue);

ContinuationValue = ValueTable(:,1);
ExpContinuationValue = mean(ContinuationValue);
EVWII = ExpContinuationValue;
VOI = EVWII - EVWOI;
VOI_over_EVWOI_pct = VOI/EVWOI*100;

[~,DWI_Phase1LifeTime_idx] = max(SRDM_DecisionTable,[],2);
DWI_Phase1LifeTime = DWI_Phase1LifeTime_idx-1;

DWI_Phase2LifeTime = zeros(N_MC,1);
for i_MC = 1:N_MC
    Phase1LifeTime_idx_real = DWI_Phase1LifeTime_idx(i_MC,1);
    DWI_Phase2LifeTime(i_MC,1) =
Phase2_StopTime_real_Phase1LifeTime(i_MC,Phase1LifeTime_idx_real);
end

DWI_LifeTime = DWI_Phase1LifeTime+DWI_Phase2LifeTime;

% 3. CLRM Approach

N_MC = size(ObsRate_Phase1n2_real_time_ShiftTime,1);
FieldLifeTime = size(ObsRate_Phase1n2_real_time_ShiftTime,2);
N_Phase1n2LifeTime = sum(1:FieldLifeTime+1);
N_LifeTimeComb = N_Phase1n2LifeTime;
Times = 1:1:FieldLifeTime;
CLRM_OptNPV_real = zeros(N_MC,1);
NPV_Tab_matrix = transpose(NPVmatrix_reals);
Prior_ENPV = meanNPVvector(:,3);
[max_Prior_ENPV, max_Prior_ENPV_idx] = max(meanNPVvector(:,3),[],1);
x_data_switch_CLRM = ObsRate_Phase1n2_real_time_ShiftTime(:,:,FieldLifeTime+1);

CLRM_PhaseLifes_Opt_reals = zeros(N_MC,2);

if max_Prior_ENPV_idx <= Times(FieldLifeTime)+1 %Finding if Switch Time is optimal at Y1
    Shift_time = 1;
    t_dataPhase2 = 1;
    while t_dataPhase2 <= Times(FieldLifeTime-Shift_time+1)
        if t_dataPhase2 == 1 && max_Prior_ENPV_idx == 1 %Switch and Stop at Y1 are optimal
            for i_MC = 1: N_MC
                CLRM_OptNPV_real(i_MC,1) = max_Prior_ENPV;
            end
            break; %The whole loop will break here
```

```
        else %Given Switch at Y1 and Stop is not at Y1
            x_data_stop_CLRM = ObsRate_Phase1n2_real_time_ShiftTime(:,:,Shift_time);
            Laguerre_CLRM_0 = exp(-x_data_stop_CLRM(:,1:t_dataPhase2)/2);
            Laguerre_CLRM_1 = exp(-x_data_stop_CLRM(:,1:t_dataPhase2)/2).*(1-
x_data_stop_CLRM(:,1:t_dataPhase2));
            Laguerre_CLRM_2 = exp(-x_data_stop_CLRM(:,1:t_dataPhase2)/2).*(1-
2.*x_data_stop_CLRM(:,1:t_dataPhase2)+0.5.*x_data_stop_CLRM(:,1:t_dataPhase2).^2);

            if t_dataPhase2 == 1
                x_data_sqrt_stop_CLRM = sqrt(x_data_stop_CLRM(:,1:t_dataPhase2));
                X_stop_CLRM = [ones(N_MC,1) Laguerre_CLRM_0 Laguerre_CLRM_1 Laguerre_CLRM_2];
            else
                x_data_sqrt_stop_CLRM = sqrt(x_data_stop_CLRM(:,1:t_dataPhase2-
1)).*sqrt(x_data_stop_CLRM(:,2:t_dataPhase2));
                X_stop_CLRM = [ones(N_MC,1) Laguerre_CLRM_0 Laguerre_CLRM_1 Laguerre_CLRM_2
x_data_sqrt_stop_CLRM];
            end
            y_data_stop_CLRM = NPV_Tab_matrix(:,t_dataPhase2+1:FieldLifeTime+1);
            y_reg_stop_CLRM = zeros(N_MC,FieldLifeTime-t_dataPhase2+1);
            for t = 1: FieldLifeTime-t_dataPhase2+1
                y_reg_stop_CLRM(:,t) =
X_stop_CLRM*regress(y_data_stop_CLRM(:,t),X_stop_CLRM);
            end
            [StopTime_reg,StopTime_idx] = max(y_reg_stop_CLRM,[],2);
        for i_MC = 1: N_MC
            if StopTime_reg(i_MC) == y_reg_stop_CLRM(i_MC,1) && CLRM_OptNPV_real(i_MC,1) == 0
                CLRM_OptNPV_real(i_MC,1) = y_data_stop_CLRM(i_MC,1);
                CLRM_PhaseLifes_Opt_reals(i_MC,1) = Shift_time-1;
                CLRM_PhaseLifes_Opt_reals(i_MC,2) = t_dataPhase2+StopTime_idx(i_MC)-1;
            else
                continue;
            end
        end
        end
        t_dataPhase2 = t_dataPhase2+1;
    end
else
% This part is to find the corresponding optimal stop time given Switch time is not at Y1
    for Shift_time = 2: FieldLifeTime %Finding and recording the optimal switch time before
determining the stop time
        N_LifeTimeComb = N_LifeTimeComb-Times(FieldLifeTime-Shift_time+2)-1;
        y_data_switch_CLRM = NPV_Tab_matrix(:,N_Phase1n2LifeTime-
N_LifeTimeComb+1:N_Phase1n2LifeTime);
        y_reg_switch_CLRM = zeros(N_MC,N_LifeTimeComb);
        ENPVTabComp = zeros(N_MC,N_LifeTimeComb);
        Laguerre_CLRM_0 = exp(-x_data_switch_CLRM(:,1:Shift_time-1)/2);
        Laguerre_CLRM_1 = exp(-x_data_switch_CLRM(:,1:Shift_time-1)/2).*(1-
x_data_switch_CLRM(:,1:Shift_time-1));
        Laguerre_CLRM_2 = exp(-x_data_switch_CLRM(:,1:Shift_time-1)/2).*(1-
2.*x_data_switch_CLRM(:,1:Shift_time-1)+0.5.*x_data_switch_CLRM(:,1:Shift_time-1).^2);

        if Shift_time == 2
            x_data_sqrt_switch_CLRM = sqrt(x_data_switch_CLRM(:,1:Shift_time-1));
            X_switch_CLRM = [ones(N_MC,1) Laguerre_CLRM_0 Laguerre_CLRM_1 Laguerre_CLRM_2];
        else
            x_data_sqrt_switch_CLRM = sqrt(x_data_switch_CLRM(:,1:Shift_time-
2)).*sqrt(x_data_switch_CLRM(:,2:Shift_time-1));
            X_switch_CLRM = [ones(N_MC,1) Laguerre_CLRM_0 Laguerre_CLRM_1 Laguerre_CLRM_2
```

```
x_data_sqrt_switch_CLRM];
        end
        for i_LifeTimeComb = 1: N_LifeTimeComb
            coef_reg_CLRM = regress(y_data_switch_CLRM(:,i_LifeTimeComb),X_switch_CLRM);
            y_reg_switch_CLRM(:,i_LifeTimeComb) = X_switch_CLRM*coef_reg_CLRM;
            ENPVTabComp(:,i_LifeTimeComb) = y_reg_switch_CLRM(:,i_LifeTimeComb);
        end

    [CLRM_reg,CLRM_idx] = max(ENPVTabComp,[],2);

    if Shift_time == FieldLifeTime
        [SwitchDecision_CLRM,SwitchDecision_CLRM_idx] = find(CLRM_idx <= Times(FieldLifeTime-
Shift_time+2)+1);
        [ContinueTime_reg,ContinueTime_idx] = find(CLRM_idx == Times(FieldLifeTime-
Shift_time+2)+1);
        for i_path = 1: length(ContinueTime_reg)
            if CLRM_OptNPV_real(ContinueTime_reg(i_path),1) == 0
               CLRM_OptNPV_real(ContinueTime_reg(i_path),1) =
y_data_switch_CLRM(ContinueTime_reg(i_path),end);
               CLRM_PhaseLifes_Opt_reals(ContinueTime_reg(i_path),1) = FieldLifeTime;
               CLRM_PhaseLifes_Opt_reals(ContinueTime_reg(i_path),2) = 0;
            else
                continue;
            end
        end
    else
        [SwitchDecision_CLRM,SwitchDecision_CLRM_idx] = find(CLRM_idx <= Times(FieldLifeTime-
Shift_time+2));
    end

    x_data_stop_CLRM = ObsRate_Phase1n2_real_time_ShiftTime(:,:,Shift_time);
    t_dataPhase2 = 1;
    while t_dataPhase2 <= Times(FieldLifeTime-Shift_time+1)+1
        Laguerre_CLRM_0 = exp(-x_data_stop_CLRM(:,1:t_dataPhase2+Shift_time-2)/2);
        Laguerre_CLRM_1 = exp(-x_data_stop_CLRM(:,1:t_dataPhase2+Shift_time-2)/2).*(1-
x_data_stop_CLRM(:,1:t_dataPhase2+Shift_time-2));
        Laguerre_CLRM_2 = exp(-x_data_stop_CLRM(:,1:t_dataPhase2+Shift_time-2)/2).*(1-
2.*x_data_stop_CLRM(:,1:t_dataPhase2+Shift_time-
2)+0.5.*x_data_stop_CLRM(:,1:t_dataPhase2+Shift_time-2).^2);

        if t_dataPhase2 == 1
            x_data_sqrt_stop_CLRM = sqrt(x_data_stop_CLRM(:,1:t_dataPhase2));
            X_stop_CLRM = [ones(N_MC,1) Laguerre_CLRM_0 Laguerre_CLRM_1 Laguerre_CLRM_2];
        else
            x_data_sqrt_stop_CLRM = sqrt(x_data_stop_CLRM(:,1:t_dataPhase2-
1)).*sqrt(x_data_stop_CLRM(:,2:t_dataPhase2));
            X_stop_CLRM = [ones(N_MC,1) Laguerre_CLRM_0 Laguerre_CLRM_1 Laguerre_CLRM_2
x_data_sqrt_stop_CLRM];
        end
        y_data_stop_CLRM = y_data_switch_CLRM(:, t_dataPhase2:Times(FieldLifeTime-
Shift_time+1)+1);
        y_reg_stop_CLRM = zeros(N_MC,Times(FieldLifeTime-Shift_time+1)-t_dataPhase2+2);
        for t = 1: Times(FieldLifeTime-Shift_time+1)-t_dataPhase2+2
            y_reg_stop_CLRM(:,t) = X_stop_CLRM*regress(y_data_stop_CLRM(:,t),X_stop_CLRM);
        end
            [StopTime_reg,StopTime_idx] = max(y_reg_stop_CLRM,[],2);
        for i_path = 1: length(SwitchDecision_CLRM)
            if StopTime_reg(SwitchDecision_CLRM(i_path)) ==
```

```
y_reg_stop_CLRM(SwitchDecision_CLRM(i_path),1) &&
CLRM_OptNPV_real(SwitchDecision_CLRM(i_path),1) == 0
                CLRM_OptNPV_real(SwitchDecision_CLRM(i_path),1) =
y_data_stop_CLRM(SwitchDecision_CLRM(i_path),1);
                CLRM_PhaseLifes_Opt_reals(SwitchDecision_CLRM(i_path),1) = Shift_time-1;
                CLRM_PhaseLifes_Opt_reals(SwitchDecision_CLRM(i_path),2) =
t_dataPhase2+StopTime_idx(SwitchDecision_CLRM(i_path))-2;
            else
                continue;
            end
        end
            t_dataPhase2 = t_dataPhase2+1;
    end
    end
end

DWI_CLRM_Phase1LifeTime = CLRM_PhaseLifes_Opt_reals(:,1);
DWI_CLRM_Phase2LifeTime = CLRM_PhaseLifes_Opt_reals(:,2);
DWI_CLRM_LifeTime = DWI_CLRM_Phase1LifeTime + DWI_CLRM_Phase2LifeTime;

EVWI_CLRM = mean(CLRM_OptNPV_real);
VOI_CLRM = EVWI_CLRM-EVWOI;
```

Appendix B2.9: Renormalization of Measured Oil Rates

This code performs the renormalization of the measured oil rates which are used in the regression analysis for Laguerre Polynomials (with renormalization). The code of applying Laguerre Polynomials (with renormalization) is not included here. However, it can be easily produced by changing the measured oil rates to the normalized oil rates in the codes presented in Appendix B2.7 and Appendix B2.8.

```
%"1" indicates using Average Rate (correspond to DWOI)
%"2" indicates using Mean Rate
Choice_of_AbitraryValue = 1;

% Computing the average rate corresponding to DWOI
Average_Rate_Base_Case =
mean(ObsRate_Phase1n2_real_time_ShiftTime(:,DWOI_Phase2LifeTime,DWOI_Phase1LifeTime+1));

% Computing the mean of maximum and minimum measured oil rates
mean_rate = 0.5.*(max(ObsRate_Phase1n2_real_time_ShiftTime,[],'all') +
min(ObsRate_Phase1n2_real_time_ShiftTime,[],'all'));

if Choice_of_AbitraryValue == 1
    arbitrary_value = Average_Rate_Base_Case;
elseif Choice_of_AbitraryValue == 2
    arbitrary_value = mean_rate;
else
    arbitrary_value = 7; %unit of MMbbl/year
end

Renormalized_ObsRate_Phase1n2_real_time_ShiftTime =
(1/arbitrary_value).*ObsRate_Phase1n2_real_time_ShiftTime;
```

Appendix B3

There are five codes presented under Appendix B3. These codes are used for the sensitivity analysis on the number of decision points as discussed in Section 5.2. These codes need to be run in the order of how they are presented here to get the correct results. Prior to running these five codes, sections 1 to 3 of Modified LSM Algorithm (Appendix B1.1) have to be run first to conduct the sampling and the calculation of NPV.

Appendix B3.1: LifeTimeTable for corresponding Decision Points

This code modifies the Lifetime Table based on the current number of decision points. Thus, this modified Lifetime Table consists of the primary and secondary lifetimes corresponding to the available options (which is determined by the number of decision points being analyzed).

```
FieldLifeTime = 50;
Year = 10; %Change this value to define the desired number of decision points

% 1. Preallocation of Essential Parameters

if Year == 2
    N_DecisionPoint = 25;
    LT_DecisionPoint = 0:2:FieldLifeTime;
elseif Year == 4
    N_DecisionPoint = 12;
    LT_DecisionPoint = 0:4:FieldLifeTime;
elseif Year == 6
    N_DecisionPoint = 8;
    LT_DecisionPoint = 0:6:FieldLifeTime;
elseif Year == 8
    N_DecisionPoint = 6;
    LT_DecisionPoint = 0:8:FieldLifeTime;
elseif Year == 10
    N_DecisionPoint = 5;
    LT_DecisionPoint = 0:10:FieldLifeTime;
end

N_Phase1n2LifeTime_DiffDP = sum(0:1:N_DecisionPoint+1);
LifeTime_Table_DiffDP = zeros(N_Phase1n2LifeTime_DiffDP,2);
meanNPVvector_DiffDP = zeros(N_Phase1n2LifeTime_DiffDP,3);

% 2. Defining LifeTime Phase 1
NPV_element_vector_LT1 = zeros(N_DecisionPoint+1,N_DecisionPoint+1);

for i_DecisionPoint = 1:N_DecisionPoint
    for k_DecisionPoint = 1:N_DecisionPoint+1
        NPV_element_vector_LT1(k_DecisionPoint,i_DecisionPoint) =
LT_DecisionPoint(1,k_DecisionPoint);
        if k_DecisionPoint > 1
            NPV_element_vector_LT1(k_DecisionPoint,N_DecisionPoint-
k_DecisionPoint+3:N_DecisionPoint+1) = LT_DecisionPoint(1,N_DecisionPoint+1)+1;
        else
            NPV_element_vector_LT1(k_DecisionPoint,i_DecisionPoint+1) =
LT_DecisionPoint(1,k_DecisionPoint);
```

```
        end
    end
end

[row_NPV_LT1_element,col_NPV_LT1_element]=find(NPV_element_vector_LT1(:,:) <=
LT_DecisionPoint(1,N_DecisionPoint+1));

for k = 1: N_Phase1n2LifeTime_DiffDP
    meanNPVvector_DiffDP(k,1) =
NPV_element_vector_LT1(col_NPV_LT1_element(k),row_NPV_LT1_element(k));
end

% 3. Defining LifeTime Phase 2
NPV_element_vector_LT2 = zeros(N_DecisionPoint+1,N_DecisionPoint+1);

for i_DecisionPoint = 1:N_DecisionPoint
    for k_DecisionPoint = 1:N_DecisionPoint+1
        NPV_element_vector_LT2(k_DecisionPoint,i_DecisionPoint) =
LT_DecisionPoint(1,i_DecisionPoint);
        if k_DecisionPoint > 1
            NPV_element_vector_LT2(k_DecisionPoint,N_DecisionPoint-
k_DecisionPoint+3:N_DecisionPoint+1) = LT_DecisionPoint(1,N_DecisionPoint+1)+1;
        else
            NPV_element_vector_LT2(k_DecisionPoint,i_DecisionPoint+1) =
LT_DecisionPoint(1,i_DecisionPoint+1);
        end
    end
end

[row_NPV_LT2_element,col_NPV_LT2_element]=find(NPV_element_vector_LT2(:,:) <
LT_DecisionPoint(1,N_DecisionPoint+1));

for k = 1: N_Phase1n2LifeTime_DiffDP-1
    meanNPVvector_DiffDP(k,2) =
NPV_element_vector_LT2(col_NPV_LT2_element(k),row_NPV_LT2_element(k));
end

% 4. Defining LifeTime Table

LifeTime_Table_DiffDP(:,1) = meanNPVvector_DiffDP(:,1);
LifeTime_Table_DiffDP(:,2) = meanNPVvector_DiffDP(:,2);
```

Appendix B3.2: Reshaping NPV (Sensitivity Analysis on the Number of Decision Points)

This code mainly reshapes the initial NPV table generated based on the current number of decision points. The initial NPV table refers to the NPV table that is established by having the decision point per year.

```
N_MC = size(ObsRate_Phase1n2_real_time_ShiftTime,1);
FieldLifeTime = size(ObsRate_Phase1n2_real_time_ShiftTime,2);

NPVtable_LTPhase1_LTPhase2_real_DiffDP = zeros(N_DecisionPoint+1,N_DecisionPoint+1,N_MC);
ObsRate_Phase1n2_real_time_ShiftTime_DiffDP = ObsRate_Phase1n2_real_time_ShiftTime;

% NPV for corresponding Decision Points
for i_Option = 1 : N_DecisionPoint+1
    for k_Option = 1 : N_DecisionPoint+1
    NPVtable_LTPhase1_LTPhase2_real_DiffDP(i_Option,k_Option,:) =
NPVtable_LTPhase1_LTPhase2_real(LT_DecisionPoint(1,i_Option)+1,LT_DecisionPoint(1,k_Option)+1
,:);
    end
end
```

Appendix B3.3: Determination of DWOI, EVWOI, DWPI, and VWPI (Sensitivity Analysis on the Number of Decision Points)

Based on the current number of decision points, this code determines the Decision Without Information and Decision With Perfect Information. The corresponding Expected Value Without Information and Value of Perfect Information are estimated.

```
N_Phase1n2LifeTime = size(LifeTime_Table_DiffDP,1);
NPVvector_1real_DiffDP = zeros(N_Phase1n2LifeTime,3);
NPVmatrix_reals_DiffDP = zeros(N_Phase1n2LifeTime,N_MC);
Sum_NPVvector_1real_DiffDP = zeros(N_Phase1n2LifeTime,1);

VWPI_real_DiffDP = zeros(N_MC,1);
DWPI_Phase1LifeTime_real_DiffDP = zeros(N_MC,1);
DWPI_Phase2LifeTime_real_DiffDP = zeros(N_MC,1);

NPVvector_1real_DiffDP(:,1) = LifeTime_Table_DiffDP(:,1);
NPVvector_1real_DiffDP(:,2) = LifeTime_Table_DiffDP(:,2);
meanNPVvector_DiffDP(:,1) = LifeTime_Table_DiffDP(:,1);
meanNPVvector_DiffDP(:,2) = LifeTime_Table_DiffDP(:,2);

for i_MC = 1: N_MC
    for k = 1: N_Phase1n2LifeTime
        NPVvector_1real_DiffDP(k,3) =
NPVtable_LTPhase1_LTPhase2_real(LifeTime_Table_DiffDP(k,1)+1,LifeTime_Table_DiffDP(k,2)+1,i_M
C);
        NPVmatrix_reals_DiffDP(k,i_MC) = NPVvector_1real_DiffDP(k,3);
        [VWPI_real_DiffDP(i_MC),DWPI_DiffDP_idx] = max(NPVvector_1real_DiffDP(:,3));
```

```
        DWPI_Phase1LifeTime_real_DiffDP(i_MC) = NPVvector_1real_DiffDP(DWPI_DiffDP_idx,1);
        DWPI_Phase2LifeTime_real_DiffDP(i_MC) = NPVvector_1real_DiffDP(DWPI_DiffDP_idx,2);
    end
end

meanNPV_DiffDP = mean(NPVmatrix_reals_DiffDP,2);
meanNPVvector_DiffDP(:,3) = meanNPV_DiffDP(:,1);

[EVWOI_DiffDP,DWOI_DiffDP_idx] = max(meanNPVvector_DiffDP(:,3));
DWOI_DiffDP_Phase1LifeTime = LifeTime_Table_DiffDP(DWOI_DiffDP_idx,1);
DWOI_DiffDP_Phase2LifeTime = LifeTime_Table_DiffDP(DWOI_DiffDP_idx,2);
DWOI_DiffDP_LifeTime = DWOI_DiffDP_Phase1LifeTime + DWOI_DiffDP_Phase2LifeTime;
DWPI_LifeTime_real_DiffDP = DWPI_Phase1LifeTime_real_DiffDP +
DWPI_Phase2LifeTime_real_DiffDP;
EVWPI_DiffDP = mean(VWPI_real_DiffDP);
VOPI_DiffDP = EVWPI_DiffDP - EVWOI_DiffDP;
```

Appendix B3.4: SRDM Approach (Sensitivity Analysis on the Number of Decision Points)

This code is the modified version of Sections 5 and 6 of Modified LSM Algorithm (Appendix B1.1). It determines the optimal decision policy by applying different number of decision points as defined.

```
% 1. Path Table Generation
N_MC = size(ObsRate_Phase1n2_real_time_ShiftTime,1);
FieldLifeTime = size(ObsRate_Phase1n2_real_time_ShiftTime,2);
SRDM_Shift_Stop_OptNPV_real = zeros(N_MC,1);
Shift_time = fliplr(LT_DecisionPoint);

% Determining the optimal stopping time for each decision point (switch time)
PathTable = zeros(N_MC,N_DecisionPoint+1);
PathTable(:,end) = NPVtable_LTPhase1_LTPhase2_real_DiffDP(N_DecisionPoint+1,1,:);
Phase2_StopTime_real_Phase1LifeTime_DiffDP = zeros(N_MC,N_DecisionPoint+1);

for k_DecisionPoint = 2: N_DecisionPoint+1

        NPV_2Phases_matrix = zeros(N_MC,k_DecisionPoint);

    for k_Stop_time = 1: k_DecisionPoint
        NPV_2Phases_matrix(:,k_Stop_time) =
NPVtable_LTPhase1_LTPhase2_real_DiffDP(N_DecisionPoint-k_DecisionPoint+2,k_Stop_time,:);
    end

    for k_StopTime = 1: k_DecisionPoint-1
        if k_StopTime == 1
            x_data_stop_SRDM =
ObsRate_Phase1n2_real_time_ShiftTime_DiffDP(:,1:Shift_time(k_StopTime+1),Shift_time(k_Decisio
nPoint)+1);
            X_stop_SRDM = [ones(N_MC,1) x_data_stop_SRDM];
            y_data_stop_matrix = [NPV_2Phases_matrix(:,k_DecisionPoint-k_StopTime)
NPV_2Phases_matrix(:,k_DecisionPoint)];
```

```
            [Stop_value,Stop_idx] =
max([X_stop_SRDM*regress(NPV_2Phases_matrix(:,k_DecisionPoint-k_StopTime),X_stop_SRDM)
X_stop_SRDM*regress(NPV_2Phases_matrix(:,k_DecisionPoint),X_stop_SRDM)],[],2);
            for i_MC = 1: N_MC
                SRDM_Shift_Stop_OptNPV_real(i_MC,1) =
y_data_stop_matrix(i_MC,Stop_idx(i_MC));
            end
        else
            x_data_stop_SRDM =
ObsRate_Phase1n2_real_time_ShiftTime_DiffDP(:,1:Shift_time(k_StopTime+1),Shift_time(k_Decisio
nPoint)+1);
            X_stop_SRDM = [ones(N_MC,1) x_data_stop_SRDM];
            y_data_stop_matrix = [NPV_2Phases_matrix(:,k_DecisionPoint-k_StopTime)
SRDM_Shift_Stop_OptNPV_real(:,1)];
            [Stop_value,Stop_idx] =
max([X_stop_SRDM*regress(NPV_2Phases_matrix(:,k_DecisionPoint-k_StopTime),X_stop_SRDM)
X_stop_SRDM*regress(SRDM_Shift_Stop_OptNPV_real(:,1),X_stop_SRDM)],[],2);
            for i_MC = 1: N_MC
                SRDM_Shift_Stop_OptNPV_real(i_MC,1) =
y_data_stop_matrix(i_MC,Stop_idx(i_MC));
            end
        end
    end

    for i_MC = 1: N_MC
        PathTable(i_MC,N_DecisionPoint-k_DecisionPoint+2) =
SRDM_Shift_Stop_OptNPV_real(i_MC,1);
        for k_Stop_time = 1: k_DecisionPoint
            if SRDM_Shift_Stop_OptNPV_real(i_MC,1) == NPV_2Phases_matrix(i_MC,k_Stop_time)
                Phase2_StopTime_real_Phase1LifeTime_DiffDP(i_MC,N_DecisionPoint-
k_DecisionPoint+2) = LT_DecisionPoint(k_Stop_time);
            end
        end
    end
end

% 2. SRDM Approach (Modified LSM Algorithm)

y_data_continuation = zeros(N_MC,1);
y_data_shift = zeros(N_MC,1);
y_reg_continuation = zeros(N_MC,1);
y_reg_shift = zeros(N_MC,1);

% Determining the optimal switch time
ValueTable = zeros(N_MC,N_DecisionPoint+1);
ValueTable(:,end) = PathTable(:,end);
SRDM_DecisionTable = zeros(N_MC,N_DecisionPoint+1);
SRDM_DecisionTable(:,end) = 1;
x_data =
ObsRate_Phase1n2_real_time_ShiftTime_DiffDP(:,:,LT_DecisionPoint(N_DecisionPoint+1)+1);

for k_Shift_time = 2: N_DecisionPoint+1
    X = [ones(N_MC,1) x_data(:,1:Shift_time(k_Shift_time))];
    y_data_shift(:) = PathTable(:,N_DecisionPoint-k_Shift_time+2);
    coef_shift = regress(y_data_shift,X);
    y_reg_shift(:) = X*coef_shift;
    y_data_continuation(:) = ValueTable(:,N_DecisionPoint-k_Shift_time+3);
    coef_continuation = regress(y_data_continuation,X);
```

```
    y_reg_continuation(:) = X*coef_continuation;
    ENPVTable_Comparison = [y_reg_shift, y_reg_continuation];
    NPV_SRDM = [y_data_shift, y_data_continuation];
    [ENPV_SRDM,ENPV_SRDM_idx] = max(ENPVTable_Comparison,[],2);

    for i_MC = 1: N_MC
        ValueTable(i_MC,N_DecisionPoint-k_Shift_time+2) = NPV_SRDM(i_MC,ENPV_SRDM_idx(i_MC));
        if y_reg_continuation(i_MC) > y_reg_shift(i_MC)
            SRDM_DecisionTable(i_MC,N_DecisionPoint-k_Shift_time+2) = 0;
        else
            SRDM_DecisionTable(i_MC,N_DecisionPoint-k_Shift_time+2) = 1;
        end
    end
end

ShiftValue = PathTable(:,1);
ExpShiftValue = mean(ShiftValue);

ContinuationValue = ValueTable(:,1);
ExpContinuationValue = mean(ContinuationValue);
EVWII_DiffDP = ExpContinuationValue;
VOI_DiffDP = EVWII_DiffDP - EVWOI_DiffDP;
VOI_over_EVWOI_pct = VOI_DiffDP/EVWOI_DiffDP*100;

[~,DWI_Phase1LifeTime_DiffDP_idx] = max(SRDM_DecisionTable,[],2);

DWI_Phase1LifeTime_DiffDP = LT_DecisionPoint(DWI_Phase1LifeTime_DiffDP_idx);
DWI_Phase1LifeTime_DiffDP = transpose(DWI_Phase1LifeTime_DiffDP);

DWI_Phase2LifeTime_DiffDP = zeros(N_MC,1);

for i_MC = 1:N_MC
    Phase1LifeTime_idx_real = DWI_Phase1LifeTime_DiffDP_idx(i_MC,1);
    DWI_Phase2LifeTime_DiffDP(i_MC,1) =
Phase2_StopTime_real_Phase1LifeTime_DiffDP(i_MC,Phase1LifeTime_idx_real);
end

DWI_LifeTime_DiffDP = DWI_Phase1LifeTime_DiffDP+DWI_Phase2LifeTime_DiffDP;
```

Appendix B3.5: CLRM Approach (Sensitivity Analysis on the Number of Decision Points)

This code consists of the modified version of CLRM (Appendix B1.2). It determines the optimal decision policy by applying different number of decision points as defined.

```
N_MC = size(ObsRate_Phase1n2_real_time_ShiftTime_DiffDP,1);
FieldLifeTime = size(ObsRate_Phase1n2_real_time_ShiftTime_DiffDP,2);
NPV_Tab_matrix_DiffDP = transpose(NPVmatrix_reals_DiffDP);
N_Phase1n2LifeTime_DiffDP = size(NPV_Tab_matrix_DiffDP,2);
N_LifeTimeComb_DiffDP = N_Phase1n2LifeTime_DiffDP;
N_Options_DiffDP = fliplr(1:1:N_DecisionPoint+1);
CLRM_OptNPV_real_DiffDP = zeros(N_MC,1);
CLRM_PhaseLifes_Opt_reals_DiffDP = zeros(N_MC,2);
Prior_ENPV_DiffDP = meanNPVvector_DiffDP(:,3);
[max_Prior_ENPV_DiffDP, max_Prior_ENPV_DiffDP_idx] = max(meanNPVvector_DiffDP(:,3),[],1);
x_data_switch_CLRM =
ObsRate_Phase1n2_real_time_ShiftTime_DiffDP(:,:,LT_DecisionPoint(N_DecisionPoint+1)+1);

if max_Prior_ENPV_DiffDP_idx <= N_DecisionPoint+1
    t_dataPhase2 = 1;
    while t_dataPhase2 <= N_DecisionPoint
        if t_dataPhase2 == 1 && max_Prior_ENPV_DiffDP_idx == 1 %Switch and Stop at Y1 are
optimal
            for i_MC = 1: N_MC
                CLRM_OptNPV_real_DiffDP(i_MC,1) = max_Prior_ENPV_DiffDP;
            end
            break; %The whole loop will break here
        else %Given Switch at Y1 and Stop is not at Y1
            x_data_stop_CLRM =
ObsRate_Phase1n2_real_time_ShiftTime_DiffDP(:,:,LT_DecisionPoint(1)+1);
            X_stop_CLRM = [ones(N_MC,1)
x_data_stop_CLRM(:,1:LT_DecisionPoint(t_dataPhase2+1))];
            y_data_stop_CLRM = NPV_Tab_matrix_DiffDP(:,t_dataPhase2+1:N_DecisionPoint+1);
            y_reg_stop_CLRM = zeros(N_MC,N_DecisionPoint-t_dataPhase2+1);
            for t = 1: N_DecisionPoint-t_dataPhase2+1
                coef_reg_stop_CLRM = regress(y_data_stop_CLRM(:,t),X_stop_CLRM);
                y_reg_stop_CLRM(:,t) = X_stop_CLRM*coef_reg_stop_CLRM;
            end
            [StopTime_reg,StopTime_idx] = max(y_reg_stop_CLRM,[],2);
        for i_MC = 1: N_MC
            if StopTime_reg(i_MC) == y_reg_stop_CLRM(i_MC,1) &&
CLRM_OptNPV_real_DiffDP(i_MC,1) == 0
                CLRM_OptNPV_real_DiffDP(i_MC,1) = y_data_stop_CLRM(i_MC,1);
                CLRM_PhaseLifes_Opt_reals_DiffDP(i_MC,1) = 0;
                CLRM_PhaseLifes_Opt_reals_DiffDP(i_MC,2) =
LT_DecisionPoint(t_dataPhase2+StopTime_idx(i_MC));
            else
                continue;
            end
        end
        end
        t_dataPhase2 = t_dataPhase2+1;
    end
else
    for k_DecisionPoint = 1: N_DecisionPoint-1
```

```
        N_LifeTimeComb_DiffDP = N_LifeTimeComb_DiffDP-N_Options_DiffDP(k_DecisionPoint);
        y_data_switch_CLRM = NPV_Tab_matrix_DiffDP(:,N_Phase1n2LifeTime_DiffDP-
N_LifeTimeComb_DiffDP+1:N_Phase1n2LifeTime_DiffDP);
        y_reg_switch_CLRM = zeros(N_MC,N_LifeTimeComb_DiffDP);
        ENPVTabComp = zeros(N_MC,N_LifeTimeComb_DiffDP);
        X_switch_CLRM = [ones(N_MC,1)
x_data_switch_CLRM(:,1:LT_DecisionPoint(k_DecisionPoint+1))];
        for i_LifeTimeComb = 1: N_LifeTimeComb_DiffDP
            coef_reg_CLRM = regress(y_data_switch_CLRM(:,i_LifeTimeComb),X_switch_CLRM);
            y_reg_switch_CLRM(:,i_LifeTimeComb) = X_switch_CLRM*coef_reg_CLRM;
            ENPVTabComp(:,i_LifeTimeComb) = y_reg_switch_CLRM(:,i_LifeTimeComb);
        end

    [CLRM_reg,CLRM_idx] = max(ENPVTabComp,[],2);

    if k_DecisionPoint == N_DecisionPoint-1
        [SwitchDecision_CLRM,SwitchDecision_CLRM_idx] = find(CLRM_idx <=
N_Options_DiffDP(k_DecisionPoint+1)+1);
        [ContinueTime_reg,ContinueTime_idx] = find(CLRM_idx ==
N_Options_DiffDP(k_DecisionPoint+1)+1);
        for i_path = 1: length(ContinueTime_reg)
            if CLRM_OptNPV_real_DiffDP(ContinueTime_reg(i_path),1) == 0
               CLRM_OptNPV_real_DiffDP(ContinueTime_reg(i_path),1) =
y_data_switch_CLRM(ContinueTime_reg(i_path),end);
               CLRM_PhaseLifes_Opt_reals_DiffDP(ContinueTime_reg(i_path),1) =
LT_DecisionPoint(N_DecisionPoint+1);
               CLRM_PhaseLifes_Opt_reals_DiffDP(ContinueTime_reg(i_path),2) = 0;
            else
                continue;
            end
        end
    else
        [SwitchDecision_CLRM,SwitchDecision_CLRM_idx] = find(CLRM_idx <=
N_Options_DiffDP(k_DecisionPoint+1));
    end

    x_data_stop_CLRM =
ObsRate_Phase1n2_real_time_ShiftTime_DiffDP(:,:,LT_DecisionPoint(k_DecisionPoint+1)+1);
    t_dataPhase2 = 1;
    while t_dataPhase2 <= N_Options_DiffDP(k_DecisionPoint+1)
        X_stop_CLRM = [ones(N_MC,1)
x_data_stop_CLRM(:,1:LT_DecisionPoint(t_dataPhase2+k_DecisionPoint))];
        y_data_stop_CLRM = y_data_switch_CLRM(:,
t_dataPhase2:N_Options_DiffDP(k_DecisionPoint+1));
        y_reg_stop_CLRM = zeros(N_MC,N_Options_DiffDP(k_DecisionPoint+1)-t_dataPhase2+1);
        for t = 1: N_Options_DiffDP(k_DecisionPoint+1)-t_dataPhase2+1
            y_reg_stop_CLRM(:,t) = X_stop_CLRM*regress(y_data_stop_CLRM(:,t),X_stop_CLRM);
        end
            [StopTime_reg,StopTime_idx] = max(y_reg_stop_CLRM,[],2);
        for i_path = 1: length(SwitchDecision_CLRM)
            if StopTime_reg(SwitchDecision_CLRM(i_path)) ==
y_reg_stop_CLRM(SwitchDecision_CLRM(i_path),1) &&
CLRM_OptNPV_real_DiffDP(SwitchDecision_CLRM(i_path),1) == 0
                CLRM_OptNPV_real_DiffDP(SwitchDecision_CLRM(i_path),1) =
y_data_stop_CLRM(SwitchDecision_CLRM(i_path),1);
                CLRM_PhaseLifes_Opt_reals_DiffDP(SwitchDecision_CLRM(i_path),1) =
LT_DecisionPoint(k_DecisionPoint+1);
                CLRM_PhaseLifes_Opt_reals_DiffDP(SwitchDecision_CLRM(i_path),2) =
```

```
LT_DecisionPoint(t_dataPhase2+StopTime_idx(SwitchDecision_CLRM(i_path))-1);
            else
                continue;
            end
        end
            t_dataPhase2 = t_dataPhase2+1;
    end
    end
end

DWI_CLRM_Phase1LifeTime_DiffDP = CLRM_PhaseLifes_Opt_reals_DiffDP(:,1);
DWI_CLRM_Phase2LifeTime_DiffDP = CLRM_PhaseLifes_Opt_reals_DiffDP(:,2);
DWI_CLRM_LifeTime_DiffDP = DWI_CLRM_Phase1LifeTime_DiffDP + DWI_CLRM_Phase2LifeTime_DiffDP;

EVWI_CLRM_DiffDP = mean(CLRM_OptNPV_real_DiffDP);
VOI_CLRM_DiffDP = EVWI_CLRM_DiffDP-EVWOI_DiffDP;
```

Appendix B4

There are two codes presented under Appendix B4. These codes are used for the sensitivity analysis on the number of data points as discussed in Section 5.2.

Appendix B4.1: SRDM Approach (Sensitivity Analysis on the Number of Data Points)

This code determines the optimal policy by using different number of data points. It consists of the modified version of Sections 5 and 6 of Modified LSM Algorithm (Appendix B1.1). To run this algorithm, sections 1, 2, 3, and 4 from Modified LSM Algorithm (Appendix B1.1) have to be run first to conduct the sampling and the calculation of NPV.

```
% 1. Path Table Generation

N_DataPoints = 40; %Change the number of data points here
N_MC = size(ObsRate_Phase1n2_real_time_ShiftTime,1);
FieldLifeTime = size(ObsRate_Phase1n2_real_time_ShiftTime,2);
SRDM_Shift_Stop_OptNPV_real = zeros(N_MC,1);

PathTable = zeros(N_MC,FieldLifeTime+1);
PathTable(:,end) = NPVtable_LTPhase1_LTPhase2_real(FieldLifeTime+1,1,:);
Phase2_StopTime_real_Phase1LifeTime = zeros(N_MC,FieldLifeTime+1);

for k_Shift_time = 1: FieldLifeTime
        NPV_2Phases_matrix = zeros(N_MC,k_Shift_time+1);
    for k_Stop_time = 1: k_Shift_time+1
        NPV_2Phases_matrix(:,k_Stop_time) = NPVtable_LTPhase1_LTPhase2_real(FieldLifeTime-
k_Shift_time+1,k_Stop_time,:);
    end
    for k_StopTime = 1: k_Shift_time
        if k_StopTime == 1
            if N_DataPoints > FieldLifeTime-k_StopTime
                x_data_stop_SRDM = ObsRate_Phase1n2_real_time_ShiftTime(:,1:FieldLifeTime-
k_StopTime,FieldLifeTime-k_Shift_time+1);
            else
                x_data_stop_SRDM = ObsRate_Phase1n2_real_time_ShiftTime(:,FieldLifeTime-
k_StopTime-N_DataPoints+1:FieldLifeTime-k_StopTime,FieldLifeTime-k_Shift_time+1);
            end
            X_stop_SRDM = [ones(N_MC,1) x_data_stop_SRDM];
            y_data_stop_matrix = [NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+1)
NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+2)];
            [Stop_value,Stop_idx] =
max([X_stop_SRDM*regress(NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+1),X_stop_SRDM)
X_stop_SRDM*regress(NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+2),X_stop_SRDM)],[],2);
            for i_MC = 1: N_MC
                SRDM_Shift_Stop_OptNPV_real(i_MC,1) =
y_data_stop_matrix(i_MC,Stop_idx(i_MC));
            end
        else
            if N_DataPoints > FieldLifeTime-k_StopTime
                x_data_stop_SRDM = ObsRate_Phase1n2_real_time_ShiftTime(:,1:FieldLifeTime-
k_StopTime,FieldLifeTime-k_Shift_time+1);
```

```
            else
                x_data_stop_SRDM = ObsRate_Phase1n2_real_time_ShiftTime(:,FieldLifeTime-
k_StopTime-N_DataPoints+1:FieldLifeTime-k_StopTime,FieldLifeTime-k_Shift_time+1);
            end
            X_stop_SRDM = [ones(N_MC,1) x_data_stop_SRDM];
            y_data_stop_matrix = [NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+1)
SRDM_Shift_Stop_OptNPV_real(:,1)];
            [Stop_value,Stop_idx] =
max([X_stop_SRDM*regress(NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+1),X_stop_SRDM)
X_stop_SRDM*regress(SRDM_Shift_Stop_OptNPV_real(:,1),X_stop_SRDM)],[],2);
            for i_MC = 1: N_MC
                SRDM_Shift_Stop_OptNPV_real(i_MC,1) =
y_data_stop_matrix(i_MC,Stop_idx(i_MC));
            end
        end
    end
    for i_MC = 1: N_MC
        PathTable(i_MC,FieldLifeTime-k_Shift_time+1) = SRDM_Shift_Stop_OptNPV_real(i_MC,1);
        for k_Stop_time = 1: k_Shift_time+1
            if SRDM_Shift_Stop_OptNPV_real(i_MC,1) == NPV_2Phases_matrix(i_MC,k_Stop_time)
                Phase2_StopTime_real_Phase1LifeTime(i_MC,FieldLifeTime-k_Shift_time+1) =
k_Stop_time-1;
            end
        end
    end
end

% 2. SRDM Approach (Modified LSM Algorithm)

FieldLifeTime = size(ObsRate_Phase1n2_real_time_ShiftTime,2);
y_data_continuation = zeros(N_MC,1);
y_data_shift = zeros(N_MC,1);
y_reg_continuation = zeros(N_MC,1);
y_reg_shift = zeros(N_MC,1);
x_data = ObsRate_Phase1n2_real_time_ShiftTime(:,:,FieldLifeTime+1);
N_DataPoints = 40;

ValueTable = zeros(N_MC,FieldLifeTime+1);
ValueTable(:,end) = PathTable(:,end);
SRDM_DecisionTable = zeros(N_MC,FieldLifeTime+1);
SRDM_DecisionTable(:,end) = 1;

for k_Shift_time = 1:FieldLifeTime
    if N_DataPoints > FieldLifeTime-k_Shift_time
        X = [ones(N_MC,1) x_data(:,1:FieldLifeTime-k_Shift_time)];
    else
        X = [ones(N_MC,1) x_data(:,FieldLifeTime-k_Shift_time-N_DataPoints+1:FieldLifeTime-
k_Shift_time)];
    end
    y_data_shift(:) = PathTable(:,FieldLifeTime-k_Shift_time+1);
    coef_shift = regress(y_data_shift,X);
    y_reg_shift(:) = X*coef_shift;
    y_data_continuation(:) = ValueTable(:,FieldLifeTime-k_Shift_time+2);
    coef_continuation = regress(y_data_continuation,X);
    y_reg_continuation(:) = X*coef_continuation;
    ENPVTable_Comparison = [y_reg_shift, y_reg_continuation];
    NPV_SRDM = [y_data_shift, y_data_continuation];
    [ENPV_SRDM,ENPV_SRDM_idx] = max(ENPVTable_Comparison,[],2);
```

```
    for i_MC = 1: N_MC
        ValueTable(i_MC,FieldLifeTime-k_Shift_time+1) = NPV_SRDM(i_MC,ENPV_SRDM_idx(i_MC));
        if y_reg_continuation(i_MC) > y_reg_shift(i_MC)
            SRDM_DecisionTable(i_MC,FieldLifeTime-k_Shift_time+1) = 0;
        else
            SRDM_DecisionTable(i_MC,FieldLifeTime-k_Shift_time+1) = 1;
        end
    end
end

ShiftValue = PathTable(:,1);
ExpShiftValue = mean(ShiftValue);

ContinuationValue = ValueTable(:,1);
ExpContinuationValue = mean(ContinuationValue);
EVWII = ExpContinuationValue;
VOI = EVWII - EVWOI;
VOI_over_EVWOI_pct = VOI/EVWOI*100;

[~,DWI_Phase1LifeTime_idx] = max(SRDM_DecisionTable,[],2);
DWI_Phase1LifeTime = DWI_Phase1LifeTime_idx-1;

DWI_Phase2LifeTime = zeros(N_MC,1);
for i_MC = 1:N_MC
    Phase1LifeTime_idx_real = DWI_Phase1LifeTime_idx(i_MC,1);
    DWI_Phase2LifeTime(i_MC,1) =
Phase2_StopTime_real_Phase1LifeTime(i_MC,Phase1LifeTime_idx_real);
end

DWI_LifeTime = DWI_Phase1LifeTime+DWI_Phase2LifeTime;
```

Appendix B4.2: CLRM Approach (Sensitivity Analysis on the Number of Data Points)

This code determines the optimal policy by using different number of data points. It consists of the modified version of CLRM (Appendix B1.2). To run this algorithm, sections 1, 2, 3, and 4 from Modified LSM Algorithm (Appendix B1.1) have to be run first to conduct the sampling and the calculation of NPV.

```
N_MC = size(ObsRate_Phase1n2_real_time_ShiftTime,1);
FieldLifeTime = size(ObsRate_Phase1n2_real_time_ShiftTime,2);
N_Phase1n2LifeTime = sum(1:FieldLifeTime+1);
N_LifeTimeComb = N_Phase1n2LifeTime;
Times = 1:1:FieldLifeTime;
CLRM_OptNPV_real = zeros(N_MC,1);
NPV_Tab_matrix = transpose(NPVmatrix_reals);
Prior_ENPV = meanNPVvector(:,3);
[max_Prior_ENPV, max_Prior_ENPV_idx] = max(meanNPVvector(:,3),[],1);
x_data_switch_CLRM = ObsRate_Phase1n2_real_time_ShiftTime(:,:,FieldLifeTime+1);

CLRM_PhaseLifes_Opt_reals = zeros(N_MC,2);
N_DataPoints = 40; %Change the number of data points here

if max_Prior_ENPV_idx <= Times(FieldLifeTime)+1 %Finding if Switch Time is optimal at Y1
    Shift_time = 1;
    t_dataPhase2 = 1;
    while t_dataPhase2 <= Times(FieldLifeTime-Shift_time+1)
        if t_dataPhase2 == 1 && max_Prior_ENPV_idx == 1 %Switch and Stop at Y1 are optimal
            for i_MC = 1: N_MC
                CLRM_OptNPV_real(i_MC,1) = max_Prior_ENPV;
            end
            break; %The whole loop will break here
        else %Given Switch at Y1 and Stop is not at Y1
            x_data_stop_CLRM = ObsRate_Phase1n2_real_time_ShiftTime(:,:,Shift_time);
            if N_DataPoints >= t_dataPhase2
                X_stop_CLRM = [ones(N_MC,1) x_data_stop_CLRM(:,1:t_dataPhase2)];
            else
                X_stop_CLRM = [ones(N_MC,1) x_data_stop_CLRM(:,t_dataPhase2-
N_DataPoints+1:t_dataPhase2)];
            end
            y_data_stop_CLRM = NPV_Tab_matrix(:,t_dataPhase2+1:FieldLifeTime+1);
            y_reg_stop_CLRM = zeros(N_MC,FieldLifeTime-t_dataPhase2+1);
            for t = 1: FieldLifeTime-t_dataPhase2+1
                y_reg_stop_CLRM(:,t) =
X_stop_CLRM*regress(y_data_stop_CLRM(:,t),X_stop_CLRM);
            end
            [StopTime_reg,StopTime_idx] = max(y_reg_stop_CLRM,[],2);
        for i_MC = 1: N_MC
            if StopTime_reg(i_MC) == y_reg_stop_CLRM(i_MC,1) && CLRM_OptNPV_real(i_MC,1) == 0
                CLRM_OptNPV_real(i_MC,1) = y_data_stop_CLRM(i_MC,1);
                CLRM_PhaseLifes_Opt_reals(i_MC,1) = Shift_time-1;
                CLRM_PhaseLifes_Opt_reals(i_MC,2) = t_dataPhase2+StopTime_idx(i_MC)-1;
            else
                continue;
            end
        end
```

```
        end
        t_dataPhase2 = t_dataPhase2+1;
    end
else
% This part is to find the corresponding optimal stop time given Switch time is not at Y1
    for Shift_time = 2: FieldLifeTime %Finding and recording the optimal switch time before
determining the stop time
        N_LifeTimeComb = N_LifeTimeComb-Times(FieldLifeTime-Shift_time+2)-1;
        y_data_switch_CLRM = NPV_Tab_matrix(:,N_Phase1n2LifeTime-
N_LifeTimeComb+1:N_Phase1n2LifeTime);
        y_reg_switch_CLRM = zeros(N_MC,N_LifeTimeComb);
        ENPVTabComp = zeros(N_MC,N_LifeTimeComb);
        if N_DataPoints >= Shift_time-1
            X_switch_CLRM = [ones(N_MC,1) x_data_switch_CLRM(:,1:Shift_time-1)];
        else
            X_switch_CLRM = [ones(N_MC,1) x_data_switch_CLRM(:,Shift_time-
N_DataPoints:Shift_time-1)];
        end
        for i_LifeTimeComb = 1: N_LifeTimeComb
            coef_reg_CLRM = regress(y_data_switch_CLRM(:,i_LifeTimeComb),X_switch_CLRM);
            y_reg_switch_CLRM(:,i_LifeTimeComb) = X_switch_CLRM*coef_reg_CLRM;
            ENPVTabComp(:,i_LifeTimeComb) = y_reg_switch_CLRM(:,i_LifeTimeComb);
        end

    [CLRM_reg,CLRM_idx] = max(ENPVTabComp,[],2);

    if Shift_time == FieldLifeTime
        [SwitchDecision_CLRM,SwitchDecision_CLRM_idx] = find(CLRM_idx <= Times(FieldLifeTime-
Shift_time+2)+1);
        [ContinueTime_reg,ContinueTime_idx] = find(CLRM_idx == Times(FieldLifeTime-
Shift_time+2)+1);
        for i_path = 1: length(ContinueTime_reg)
            if CLRM_OptNPV_real(ContinueTime_reg(i_path),1) == 0
               CLRM_OptNPV_real(ContinueTime_reg(i_path),1) =
y_data_switch_CLRM(ContinueTime_reg(i_path),end);
               CLRM_PhaseLifes_Opt_reals(ContinueTime_reg(i_path),1) = FieldLifeTime;
               CLRM_PhaseLifes_Opt_reals(ContinueTime_reg(i_path),2) = 0;
            else
                continue;
            end
        end
    else
        [SwitchDecision_CLRM,SwitchDecision_CLRM_idx] = find(CLRM_idx <= Times(FieldLifeTime-
Shift_time+2));
    end

    x_data_stop_CLRM = ObsRate_Phase1n2_real_time_ShiftTime(:,:,Shift_time);
    t_dataPhase2 = 1;
    while t_dataPhase2 <= Times(FieldLifeTime-Shift_time+1)+1
        if N_DataPoints >= t_dataPhase2+Shift_time-2
            X_stop_CLRM = [ones(N_MC,1) x_data_stop_CLRM(:,1:t_dataPhase2+Shift_time-2)];
        else
            X_stop_CLRM = [ones(N_MC,1) x_data_stop_CLRM(:,t_dataPhase2+Shift_time-
N_DataPoints-1:t_dataPhase2+Shift_time-2)];
        end
        y_data_stop_CLRM = y_data_switch_CLRM(:, t_dataPhase2:Times(FieldLifeTime-
Shift_time+1)+1);
        y_reg_stop_CLRM = zeros(N_MC,Times(FieldLifeTime-Shift_time+1)-t_dataPhase2+2);
```

```
        for t = 1: Times(FieldLifeTime-Shift_time+1)-t_dataPhase2+2
            y_reg_stop_CLRM(:,t) = X_stop_CLRM*regress(y_data_stop_CLRM(:,t),X_stop_CLRM);
        end
            [StopTime_reg,StopTime_idx] = max(y_reg_stop_CLRM,[],2);
        for i_path = 1: length(SwitchDecision_CLRM)
            if StopTime_reg(SwitchDecision_CLRM(i_path)) ==
y_reg_stop_CLRM(SwitchDecision_CLRM(i_path),1) &&
CLRM_OptNPV_real(SwitchDecision_CLRM(i_path),1) == 0
                CLRM_OptNPV_real(SwitchDecision_CLRM(i_path),1) =
y_data_stop_CLRM(SwitchDecision_CLRM(i_path),1);
                CLRM_PhaseLifes_Opt_reals(SwitchDecision_CLRM(i_path),1) = Shift_time-1;
                CLRM_PhaseLifes_Opt_reals(SwitchDecision_CLRM(i_path),2) =
t_dataPhase2+StopTime_idx(SwitchDecision_CLRM(i_path))-2;
            else
                continue;
            end
        end
            t_dataPhase2 = t_dataPhase2+1;
    end
    end
end

DWI_CLRM_Phase1LifeTime = CLRM_PhaseLifes_Opt_reals(:,1);
DWI_CLRM_Phase2LifeTime = CLRM_PhaseLifes_Opt_reals(:,2);
DWI_CLRM_LifeTime = DWI_CLRM_Phase1LifeTime + DWI_CLRM_Phase2LifeTime;

EVWI_CLRM = mean(CLRM_OptNPV_real);
VOI_CLRM = EVWI_CLRM-EVWOI;
```

Appendix B5

There are five codes presented under Appendix B5. These codes generally apply the modified LSM algorithm by including the economic uncertainties. These codes need to be run in the order of how they are presented here to get the correct results. Prior to running these five codes, sections 1 to 2 of Modified LSM Algorithm (Appendix B1.1) have to be run first to conduct the sampling.

Appendix B5.1: Calibration of the Parameters for Ornstein-Uhlenbeck Stochastic Process

This code conducts the calibration of the OU parameters used in the OU Model based on the least-squares approach proposed by Smith (2010). This code is originally developed by Smith (2010) and slightly modified for this research work. A set of oil price has to be loaded before running this code.

```
d_t = 1; %year

if (size(OilPrice,2) > size(OilPrice,1))
    OilPrice = OilPrice';
end

% regression is done on the logarithm of oil price
[regression_coefficient,dummy,residual] =
regress(log(OilPrice(2:end)),[ones(size(OilPrice(1:end-1))) log(OilPrice(1:end-1))]);

a = regression_coefficient(1);
b = regression_coefficient(2);

theta = -log(b)/d_t;
mu = a/(1-b);
sigma = std(residual)*sqrt((-2*log(b))/(d_t*(1-b^2)));
```

Appendix B5.2: Mean Reverting Process (Ornstein-Uhlenbeck Process)

This code performs the stochastic modeling of the economic parameters based on the OU model. The OU parameters obtained from running the previous code are used as the input parameters here.

```
N_MC = size(ObsRate_Phase1n2_real_time_ShiftTime,1);
FieldLifeTime = size(ObsRate_Phase1n2_real_time_ShiftTime,2);

% 1. OU Parameters for Oil Price - Inputs here are from calibration done by least-squares
approach
OilPrice = 47.48;
theta_oilprice = 0.0643; %speed of mean reversion
mu_oilprice = 47.48; %reverting mean
sigma_oilprice = 0.2473; %process volatility %percent
dt_oilprice = 1; %year
Pi_OilPrice_Matrix = zeros(N_MC,FieldLifeTime);
OilPrice_Matrix = zeros(N_MC,FieldLifeTime);

% 2. OU Parameters for Cost Multiplier
% The parameters here are from Willigers (2008)
% These same parameters are used for Opex_P1, Opex_P2 & Capex_2After1
Initial_cost_multiplier = 1;
theta_cost_multiplier = 0.1; %speed of mean reversion
mu_cost_multiplier = 1; %reverting mean
sigma_cost_multiplier = 0.5; %process volatility %percent
dt_cost_multiplier = 1; %year
Cost_Multiplier_Opex_Phase1 = zeros(N_MC,FieldLifeTime);
Cost_Multiplier_Opex_Phase2 = zeros(N_MC,FieldLifeTime);
Cost_Multiplier_Capex_Phase2After1 = zeros(N_MC,FieldLifeTime);
Pi_Opex_Phase1_Matrix = zeros(N_MC,FieldLifeTime);
Pi_Opex_Phase2_Matrix = zeros(N_MC,FieldLifeTime);
Pi_Capex_Phase2After1_Matrix = zeros(N_MC,FieldLifeTime);

% 3. Generating random samples using multivariate normal distibution

mvnd_mean = [0,0,0,0];
mvnd_SD = [1,1,1,1];
exp_CorrC = [1 0.87 0.87 0.85; 0.87 1 0.55 0.65; 0.87 0.55 1 0.7; 0.85 0.65 0.7 1];
% There are 4 by 4 Correlation Coefficient Matrix used in generating the multivariate uniform
distribution
% Since no data, except for oil price, is available, these coefficients are mainly assumed
cov = exp_CorrC.*(mvnd_SD'*mvnd_SD);

mvnd_samples = zeros(N_MC,4,FieldLifeTime);
reshaped_mvnd_samples = zeros(N_MC,FieldLifeTime,4);

for t = 1: FieldLifeTime %these loops are to generate 4 different types samples from mvnrnd
for each year throughout the FieldLifeTime
    mvnd_samples(:,:,t) = mvnrnd(mvnd_mean,cov,N_MC);
    reshaped_mvnd_samples(:,t,:) = mvnd_samples(:,:,t);
end
```

```
% 4. Generation of the Values of Economic Parameters

for t = 1: FieldLifeTime %the reshaped_mnvd_samples are used in the OU model
    if t == 1
        Pi_OilPrice_Matrix(:,t) = (log(OilPrice)*exp(-theta_oilprice*dt_oilprice)) +
(log(mu_oilprice)*(1-exp(-theta_oilprice*dt_oilprice))) + (sigma_oilprice*sqrt((1-exp(-
2*theta_oilprice*dt_oilprice))/(2*theta_oilprice))*reshaped_mvnd_samples(:,t,1));
        OilPrice_Matrix(:,t) = exp(Pi_OilPrice_Matrix(:,t)-
(0.5*(sigma_oilprice^2/(2*theta_oilprice))*(1-exp(-2*theta_oilprice*t))));
        Pi_Opex_Phase1_Matrix(:,t) = (log(Initial_cost_multiplier)*exp(-
theta_cost_multiplier*dt_cost_multiplier)) + (log(mu_cost_multiplier)*(1-exp(-
theta_cost_multiplier*dt_cost_multiplier))) + (sigma_cost_multiplier*sqrt((1-exp(-
2*theta_cost_multiplier*dt_cost_multiplier))/(2*theta_cost_multiplier))*reshaped_mvnd_samples
(:,t,2));
        Cost_Multiplier_Opex_Phase1(:,t) = exp(Pi_Opex_Phase1_Matrix(:,t)-
(0.5*(sigma_cost_multiplier^2/(2*theta_cost_multiplier))*(1-exp(-
2*theta_cost_multiplier*t))));
        Pi_Opex_Phase2_Matrix(:,t) = (log(Initial_cost_multiplier)*exp(-
theta_cost_multiplier*dt_cost_multiplier)) + (log(mu_cost_multiplier)*(1-exp(-
theta_cost_multiplier*dt_cost_multiplier))) + (sigma_cost_multiplier*sqrt((1-exp(-
2*theta_cost_multiplier*dt_cost_multiplier))/(2*theta_cost_multiplier))*reshaped_mvnd_samples
(:,t,3));
        Cost_Multiplier_Opex_Phase2(:,t) = exp(Pi_Opex_Phase2_Matrix(:,t)-
(0.5*(sigma_cost_multiplier^2/(2*theta_cost_multiplier))*(1-exp(-
2*theta_cost_multiplier*t))));
        Pi_Capex_Phase2After1_Matrix(:,t) = (log(Initial_cost_multiplier)*exp(-
theta_cost_multiplier*dt_cost_multiplier)) + (log(mu_cost_multiplier)*(1-exp(-
theta_cost_multiplier*dt_cost_multiplier))) + (sigma_cost_multiplier*sqrt((1-exp(-
2*theta_cost_multiplier*dt_cost_multiplier))/(2*theta_cost_multiplier))*reshaped_mvnd_samples
(:,t,4));
        Cost_Multiplier_Capex_Phase2After1(:,t) = exp(Pi_Capex_Phase2After1_Matrix(:,t)-
(0.5*(sigma_cost_multiplier^2/(2*theta_cost_multiplier))*(1-exp(-
2*theta_cost_multiplier*t))));
    else
        Pi_OilPrice_Matrix(:,t) = (Pi_OilPrice_Matrix(:,t-1)*exp(-
theta_oilprice*dt_oilprice)) + (log(mu_oilprice)*(1-exp(-theta_oilprice*dt_oilprice))) +
(sigma_oilprice*sqrt((1-exp(-
2*theta_oilprice*dt_oilprice))/(2*theta_oilprice))*reshaped_mvnd_samples(:,t,1));
        OilPrice_Matrix(:,t) = exp(Pi_OilPrice_Matrix(:,t)-
(0.5*(sigma_oilprice^2/(2*theta_oilprice))*(1-exp(-2*theta_oilprice*t))));
        Pi_Opex_Phase1_Matrix(:,t) = (Pi_Opex_Phase1_Matrix(:,t-1)*exp(-
theta_cost_multiplier*dt_cost_multiplier)) + (log(mu_cost_multiplier)*(1-exp(-
theta_cost_multiplier*dt_cost_multiplier))) + (sigma_cost_multiplier*sqrt((1-exp(-
2*theta_cost_multiplier*dt_cost_multiplier))/(2*theta_cost_multiplier))*reshaped_mvnd_samples
(:,t,2));
        Cost_Multiplier_Opex_Phase1(:,t) = exp(Pi_Opex_Phase1_Matrix(:,t)-
(0.5*(sigma_cost_multiplier^2/(2*theta_cost_multiplier))*(1-exp(-
2*theta_cost_multiplier*t))));
        Pi_Opex_Phase2_Matrix(:,t) = (Pi_Opex_Phase2_Matrix(:,t-1)*exp(-
theta_cost_multiplier*dt_cost_multiplier)) + (log(mu_cost_multiplier)*(1-exp(-
theta_cost_multiplier*dt_cost_multiplier))) + (sigma_cost_multiplier*sqrt((1-exp(-
2*theta_cost_multiplier*dt_cost_multiplier))/(2*theta_cost_multiplier))*reshaped_mvnd_samples
(:,t,3));
        Cost_Multiplier_Opex_Phase2(:,t) = exp(Pi_Opex_Phase2_Matrix(:,t)-
(0.5*(sigma_cost_multiplier^2/(2*theta_cost_multiplier))*(1-exp(-
2*theta_cost_multiplier*t))));
        Pi_Capex_Phase2After1_Matrix(:,t) = (Pi_Capex_Phase2After1_Matrix(:,t-1)*exp(-
theta_cost_multiplier*dt_cost_multiplier)) + (log(mu_cost_multiplier)*(1-exp(-
```

```
theta_cost_multiplier*dt_cost_multiplier))) + (sigma_cost_multiplier*sqrt((1-exp(-
2*theta_cost_multiplier*dt_cost_multiplier))/(2*theta_cost_multiplier))*reshaped_mvnd_samples
(:,t,4));
        Cost_Multiplier_Capex_Phase2After1(:,t) = exp(Pi_Capex_Phase2After1_Matrix(:,t)-
(0.5*(sigma_cost_multiplier^2/(2*theta_cost_multiplier))*(1-exp(-
2*theta_cost_multiplier*t))));
    end
end

Opex_Phase1_Matrix = zeros(N_MC,FieldLifeTime);
Opex_Phase2_Matrix = zeros(N_MC,FieldLifeTime);
Capex_Phase2After1_Matrix = zeros(N_MC,FieldLifeTime);

for t = 1: FieldLifeTime
    if t == 1 %for Costs at Y1, Oil Price at Year 0 is used
        Opex_Phase1_Matrix(:,t) = OilPrice.*Cost_Multiplier_Opex_Phase1(:,t);
        Opex_Phase2_Matrix(:,t) = OilPrice.*Cost_Multiplier_Opex_Phase2(:,t);
        Capex_Phase2After1_Matrix(:,t) = OilPrice.*Cost_Multiplier_Capex_Phase2After1(:,t);
    else %for Costs at Year n+1, Oil Price at Year n is used
        Opex_Phase1_Matrix(:,t) = OilPrice_Matrix(:,t-1).*Cost_Multiplier_Opex_Phase1(:,t);
        Opex_Phase2_Matrix(:,t) = OilPrice_Matrix(:,t-1).*Cost_Multiplier_Opex_Phase2(:,t);
        Capex_Phase2After1_Matrix(:,t) = OilPrice_Matrix(:,t-
1).*Cost_Multiplier_Capex_Phase2After1(:,t);
    end
end
```

Appendix B5.3: Calculation of Cashflow and NPV (Economic Uncertainties)

This code computes the cashflow and the NPV used in the example problem which economic uncertainties are included. This code is the modified version of section 3 of Modified LSM Algorithm (Appendix B1.1). After running this code, section 4 of Modified LSM Algorithm (Appendix B1.1) has to be run before proceeding to either CLRM or SRDM.

```
FieldLifeTime = size(ObsRate_Phase1n2_real_time_ShiftTime,2);
N_MC = size(ObsRate_Phase1n2_real_time_ShiftTime,1);

% Values of Economic Parameters
Capex_Phase1 = 50;
Capex_Phase2No1 = 75;
DisRate = 0.12;

N_Phase1n2LifeTime = sum(1:FieldLifeTime+1);
CashFlow = zeros(N_MC,FieldLifeTime);
DisCashFlow = zeros(N_MC,FieldLifeTime);
DisCashFlow_ShiftTime = zeros(N_MC,FieldLifeTime,FieldLifeTime+1);
NPV_reals = zeros(N_MC,1);
NPVtable_LTPhase1_LTPhase2_real = zeros(FieldLifeTime+1,FieldLifeTime+1,N_MC);

for t = 1: FieldLifeTime
    for time = 1: FieldLifeTime+1
        if t == time
            if time == 1
                CashFlow(:,t) =
OilPrice_Matrix(:,t).*ObsRate_Phase1n2_real_time_ShiftTime(:,t,time) -
Opex_Phase2_Matrix(:,t) - Capex_Phase2No1;
                DisCashFlow(:,t) = CashFlow(:,t)/((1+DisRate).^(t));
            else
                CashFlow(:,t) =
OilPrice_Matrix(:,t).*ObsRate_Phase1n2_real_time_ShiftTime(:,t,time) -
Opex_Phase2_Matrix(:,t) - Capex_Phase2After1_Matrix(:,t);
                DisCashFlow(:,t) = CashFlow(:,t)/((1+DisRate).^(t));
            end
        elseif t < time
            if t == 1
                CashFlow(:,t) =
OilPrice_Matrix(:,t).*ObsRate_Phase1n2_real_time_ShiftTime(:,t,time) -
Opex_Phase1_Matrix(:,t) - Capex_Phase1;
                DisCashFlow(:,t) = CashFlow(:,t)/((1+DisRate).^(t));
            else
                CashFlow(:,t) =
OilPrice_Matrix(:,t).*ObsRate_Phase1n2_real_time_ShiftTime(:,t,time) -
Opex_Phase1_Matrix(:,t);
                DisCashFlow(:,t) = CashFlow(:,t)/((1+DisRate).^(t));
            end
        else
            CashFlow(:,t) =
OilPrice_Matrix(:,t).*ObsRate_Phase1n2_real_time_ShiftTime(:,t,time) -
Opex_Phase2_Matrix(:,t);
            DisCashFlow(:,t) = CashFlow(:,t)/((1+DisRate).^(t));
        end
```

```
            DisCashFlow_ShiftTime(:,t,time) = DisCashFlow(:,t);

            if time == 1
                    NPV_reals(:) = sum(DisCashFlow_ShiftTime(:,1:t,time),2);
                    NPVtable_LTPhase1_LTPhase2_real(time,t+1,:) = NPV_reals(:);
            elseif time == FieldLifeTime+1
                    NPV_reals(:) = sum(DisCashFlow_ShiftTime(:,1:FieldLifeTime,time),2);
                    NPVtable_LTPhase1_LTPhase2_real(time,1,:) = NPV_reals(:);
            else
                if t == time
                    NPV_reals(:) = sum(DisCashFlow_ShiftTime(:,1:time,time),2);
                    NPVtable_LTPhase1_LTPhase2_real(time,2,:) = NPV_reals(:);
                elseif t < time
                    NPV_reals(:) = sum(DisCashFlow_ShiftTime(:,1:time-1,time),2);
                    NPVtable_LTPhase1_LTPhase2_real(time,1,:) = NPV_reals(:);
                else
                    NPV_reals(:) = sum(DisCashFlow_ShiftTime(:,1:t,time),2);
                    NPVtable_LTPhase1_LTPhase2_real(time,t-time+2,:) = NPV_reals(:);
                end
            end
        end
    end
```

Appendix B5.4: SRDM Approach (Economic Uncertainties)

This code determines the optimal decision policy using SRDM approach by including economic uncertainties. It consists of the modified version of Sections 5 and 6 of Modified LSM Algorithm (Appendix B1.1).

```
%  1. Path Table Generation (Modified LSM algorithm)
%  This section determines the optimal stopping time given the switch time at every year
%  Regression Analysis is applied
%  The NPV corresponding to the optimal stopping time is recorded into the PathTable

N_MC = size(ObsRate_Phase1n2_real_time_ShiftTime,1);
FieldLifeTime = size(ObsRate_Phase1n2_real_time_ShiftTime,2);
SRDM_Shift_Stop_OptNPV_real = zeros(N_MC,1);

PathTable = zeros(N_MC,FieldLifeTime+1);
PathTable(:,end) = NPVtable_LTPhase1_LTPhase2_real(FieldLifeTime+1,1,:);
Phase2_StopTime_real_Phase1LifeTime = zeros(N_MC,FieldLifeTime+1);

for k_Shift_time = 1: FieldLifeTime
        NPV_2Phases_matrix = zeros(N_MC,k_Shift_time+1);
    for k_Stop_time = 1: k_Shift_time+1
        NPV_2Phases_matrix(:,k_Stop_time) = NPVtable_LTPhase1_LTPhase2_real(FieldLifeTime-
k_Shift_time+1,k_Stop_time,:);
    end
    for k_StopTime = 1: k_Shift_time
        if k_StopTime == 1
            x_data_stop_SRDM = ObsRate_Phase1n2_real_time_ShiftTime(:,1:FieldLifeTime-
k_StopTime,FieldLifeTime-k_Shift_time+1);
            oilprice_data = OilPrice_Matrix(:,FieldLifeTime-k_StopTime);
            opex_data = Opex_Phase2_Matrix(:,FieldLifeTime-k_StopTime);
            X_stop_SRDM = [ones(N_MC,1) x_data_stop_SRDM oilprice_data opex_data];
            y_data_stop_matrix = [NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+1)
NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+2)];
            [Stop_value,Stop_idx] =
max([X_stop_SRDM*regress(NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+1),X_stop_SRDM)
X_stop_SRDM*regress(NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+2),X_stop_SRDM)],[],2);
            for i_MC = 1: N_MC
                SRDM_Shift_Stop_OptNPV_real(i_MC,1) =
y_data_stop_matrix(i_MC,Stop_idx(i_MC));
            end
        else
            x_data_stop_SRDM = ObsRate_Phase1n2_real_time_ShiftTime(:,1:FieldLifeTime-
k_StopTime,FieldLifeTime-k_Shift_time+1);
            if FieldLifeTime-k_StopTime == 0
                oilprice_data = OilPrice_Matrix(:,1:FieldLifeTime-k_StopTime);
                opex_data = Opex_Phase2_Matrix(:,1:FieldLifeTime-k_StopTime);
            else
                oilprice_data = OilPrice_Matrix(:,FieldLifeTime-k_StopTime);
                opex_data = Opex_Phase2_Matrix(:,FieldLifeTime-k_StopTime);
            end
            X_stop_SRDM = [ones(N_MC,1) x_data_stop_SRDM oilprice_data opex_data];
            y_data_stop_matrix = [NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+1)
SRDM_Shift_Stop_OptNPV_real(:,1)];
            [Stop_value,Stop_idx] =
max([X_stop_SRDM*regress(NPV_2Phases_matrix(:,k_Shift_time-k_StopTime+1),X_stop_SRDM)
```

```
X_stop_SRDM*regress(SRDM_Shift_Stop_OptNPV_real(:,1),X_stop_SRDM)],[],2);
            for i_MC = 1: N_MC
                SRDM_Shift_Stop_OptNPV_real(i_MC,1) =
y_data_stop_matrix(i_MC,Stop_idx(i_MC));
            end
        end
    end
    for i_MC = 1: N_MC
        PathTable(i_MC,FieldLifeTime-k_Shift_time+1) = SRDM_Shift_Stop_OptNPV_real(i_MC,1);
        for k_Stop_time = 1: k_Shift_time+1
            if SRDM_Shift_Stop_OptNPV_real(i_MC,1) == NPV_2Phases_matrix(i_MC,k_Stop_time)
                Phase2_StopTime_real_Phase1LifeTime(i_MC,FieldLifeTime-k_Shift_time+1) =
k_Stop_time-1;
            end
        end
    end
end

% 2. SRDM Approach (Modified LSM algorithm)
% This section determines the optimal decision policy based on SRDM approach

FieldLifeTime = size(ObsRate_Phase1n2_real_time_ShiftTime,2);
y_data_continuation = zeros(N_MC,1);
y_data_shift = zeros(N_MC,1);
y_reg_continuation = zeros(N_MC,1);
y_reg_shift = zeros(N_MC,1);
x_data = ObsRate_Phase1n2_real_time_ShiftTime(:,:,FieldLifeTime+1);

ValueTable = zeros(N_MC,FieldLifeTime+1);
ValueTable(:,end) = PathTable(:,end);
SRDM_DecisionTable = zeros(N_MC,FieldLifeTime+1);
SRDM_DecisionTable(:,end) = 1;
oilprice_data = OilPrice_Matrix(:,:);
opex_phase1_data = Opex_Phase1_Matrix(:,:);
opex_phase2_data = Opex_Phase2_Matrix(:,:);
capex_phase2after1_data = Capex_Phase2After1_Matrix(:,:);

for k_Shift_time = 1:FieldLifeTime
    if FieldLifeTime-k_Shift_time == 0
        X_shift = [ones(N_MC,1) x_data(:,1:FieldLifeTime-k_Shift_time)
oilprice_data(:,1:FieldLifeTime-k_Shift_time) opex_phase1_data(:,1:FieldLifeTime-
k_Shift_time) opex_phase2_data(:,1:FieldLifeTime-k_Shift_time)
capex_phase2after1_data(:,1:FieldLifeTime-k_Shift_time)];
        X_continuation = [ones(N_MC,1) x_data(:,1:FieldLifeTime-k_Shift_time)
oilprice_data(:,1:FieldLifeTime-k_Shift_time) opex_phase1_data(:,1:FieldLifeTime-
k_Shift_time) opex_phase2_data(:,1:FieldLifeTime-k_Shift_time)
capex_phase2after1_data(:,1:FieldLifeTime-k_Shift_time)];
    else
        X_shift = [ones(N_MC,1) x_data(:,1:FieldLifeTime-k_Shift_time)
oilprice_data(:,FieldLifeTime-k_Shift_time) opex_phase1_data(:,FieldLifeTime-k_Shift_time)
opex_phase2_data(:,FieldLifeTime-k_Shift_time) capex_phase2after1_data(:,FieldLifeTime-
k_Shift_time)];
        X_continuation = [ones(N_MC,1) x_data(:,1:FieldLifeTime-k_Shift_time)
oilprice_data(:,FieldLifeTime-k_Shift_time) opex_phase1_data(:,FieldLifeTime-k_Shift_time)
opex_phase2_data(:,FieldLifeTime-k_Shift_time) capex_phase2after1_data(:,FieldLifeTime-
k_Shift_time)];
    end
    y_data_shift(:) = PathTable(:,FieldLifeTime-k_Shift_time+1);
```

```
    coef_shift = regress(y_data_shift,X_shift);
    y_reg_shift(:) = X_shift*coef_shift;
    y_data_continuation(:) = ValueTable(:,FieldLifeTime-k_Shift_time+2);
    coef_continuation = regress(y_data_continuation,X_continuation);
    y_reg_continuation(:) = X_continuation*coef_continuation;
    ENPVTable_Comparison = [y_reg_shift, y_reg_continuation];
    NPV_SRDM = [y_data_shift, y_data_continuation];
    [ENPV_SRDM,ENPV_SRDM_idx] = max(ENPVTable_Comparison,[],2);

    for i_MC = 1: N_MC
        ValueTable(i_MC,FieldLifeTime-k_Shift_time+1) = NPV_SRDM(i_MC,ENPV_SRDM_idx(i_MC));
        if y_reg_continuation(i_MC) > y_reg_shift(i_MC)
            SRDM_DecisionTable(i_MC,FieldLifeTime-k_Shift_time+1) = 0;
        else
            SRDM_DecisionTable(i_MC,FieldLifeTime-k_Shift_time+1) = 1;
        end
    end
end

ShiftValue = PathTable(:,1);
ExpShiftValue = mean(ShiftValue);

ContinuationValue = ValueTable(:,1);
ExpContinuationValue = mean(ContinuationValue);
EVWII = ExpContinuationValue;
VOI = EVWII - EVWOI;
VOI_over_EVWOI_pct = VOI/EVWOI*100;

[~,DWI_Phase1LifeTime_idx] = max(SRDM_DecisionTable,[],2);
DWI_Phase1LifeTime = DWI_Phase1LifeTime_idx-1;

DWI_Phase2LifeTime = zeros(N_MC,1);
for i_MC = 1:N_MC
    Phase1LifeTime_idx_real = DWI_Phase1LifeTime_idx(i_MC,1);
    DWI_Phase2LifeTime(i_MC,1) =
Phase2_StopTime_real_Phase1LifeTime(i_MC,Phase1LifeTime_idx_real);
end

DWI_LifeTime = DWI_Phase1LifeTime+DWI_Phase2LifeTime;
```

Appendix B5.5: CLRM Approach (Economic Uncertainties)

This section determines the optimal decision policy based on CLRM approach by including economic uncertainties. This code consists of the modified version of CLRM (Appendix B1.2).

```
N_MC = size(ObsRate_Phase1n2_real_time_ShiftTime,1);
FieldLifeTime = size(ObsRate_Phase1n2_real_time_ShiftTime,2);
N_Phase1n2LifeTime = sum(1:FieldLifeTime+1);
N_LifeTimeComb = N_Phase1n2LifeTime;
Times = 1:1:FieldLifeTime;
CLRM_OptNPV_real = zeros(N_MC,1);
NPV_Tab_matrix = transpose(NPVmatrix_reals);
Prior_ENPV = meanNPVvector(:,3);
[max_Prior_ENPV, max_Prior_ENPV_idx] = max(meanNPVvector(:,3),[],1);
x_data_switch_CLRM = ObsRate_Phase1n2_real_time_ShiftTime(:,:,FieldLifeTime+1);
oilprice_data = OilPrice_Matrix(:,:);
opex_phase1_data = Opex_Phase1_Matrix(:,:);
opex_phase2_data = Opex_Phase2_Matrix(:,:);
capex_phase2after1_data = Capex_Phase2After1_Matrix(:,:);

CLRM_PhaseLifes_Opt_reals = zeros(N_MC,2);

if max_Prior_ENPV_idx <= Times(FieldLifeTime)+1 %Finding if Switch Time is optimal at Y1
    Shift_time = 1;
    t_dataPhase2 = 1;
    while t_dataPhase2 <= Times(FieldLifeTime-Shift_time+1)
        if t_dataPhase2 == 1 && max_Prior_ENPV_idx == 1 %Switch and Stop at Y1 are optimal
            for i_MC = 1: N_MC
                CLRM_OptNPV_real(i_MC,1) = max_Prior_ENPV;
            end
            break; %The whole loop will break here
        else %Given Switch at Y1 and Stop is not at Y1
            x_data_stop_CLRM = ObsRate_Phase1n2_real_time_ShiftTime(:,:,Shift_time);
            X_stop_CLRM = [ones(N_MC,1) x_data_stop_CLRM(:,1:t_dataPhase2)
oilprice_data(:,t_dataPhase2) opex_phase2_data(:,t_dataPhase2)];
            y_data_stop_CLRM = NPV_Tab_matrix(:,t_dataPhase2+1:FieldLifeTime+1);
            y_reg_stop_CLRM = zeros(N_MC,FieldLifeTime-t_dataPhase2+1);
            for t = 1: FieldLifeTime-t_dataPhase2+1
                y_reg_stop_CLRM(:,t) =
X_stop_CLRM*regress(y_data_stop_CLRM(:,t),X_stop_CLRM);
            end
            [StopTime_reg,StopTime_idx] = max(y_reg_stop_CLRM,[],2);
        for i_MC = 1: N_MC
            if StopTime_reg(i_MC) == y_reg_stop_CLRM(i_MC,1) && CLRM_OptNPV_real(i_MC,1) == 0
                CLRM_OptNPV_real(i_MC,1) = y_data_stop_CLRM(i_MC,1);
                CLRM_PhaseLifes_Opt_reals(i_MC,1) = Shift_time-1;
                CLRM_PhaseLifes_Opt_reals(i_MC,2) = t_dataPhase2+StopTime_idx(i_MC)-1;
            else
                continue;
            end
        end
        end
        t_dataPhase2 = t_dataPhase2+1;
    end
else
% This part is to find the corresponding optimal stop time given Switch time is not at Y1
```

```
    for Shift_time = 2: FieldLifeTime %Finding and recording the optimal switch time before
determining the stop time
        N_LifeTimeComb = N_LifeTimeComb-Times(FieldLifeTime-Shift_time+2)-1;
        y_data_switch_CLRM = NPV_Tab_matrix(:,N_Phase1n2LifeTime-
N_LifeTimeComb+1:N_Phase1n2LifeTime);
        y_reg_switch_CLRM = zeros(N_MC,N_LifeTimeComb);
        ENPVTabComp = zeros(N_MC,N_LifeTimeComb);
        X_switch_CLRM = [ones(N_MC,1) x_data_switch_CLRM(:,1:Shift_time-1)
oilprice_data(:,Shift_time-1) opex_phase1_data(:,Shift_time-1) opex_phase2_data(:,Shift_time-
1) capex_phase2after1_data(:,Shift_time-1)];
        for i_LifeTimeComb = 1: N_LifeTimeComb
            coef_reg_CLRM = regress(y_data_switch_CLRM(:,i_LifeTimeComb),X_switch_CLRM);
            y_reg_switch_CLRM(:,i_LifeTimeComb) = X_switch_CLRM*coef_reg_CLRM;
            ENPVTabComp(:,i_LifeTimeComb) = y_reg_switch_CLRM(:,i_LifeTimeComb);
        end

    [CLRM_reg,CLRM_idx] = max(ENPVTabComp,[],2);

    if Shift_time == FieldLifeTime
        [SwitchDecision_CLRM,SwitchDecision_CLRM_idx] = find(CLRM_idx <= Times(FieldLifeTime-
Shift_time+2)+1);
        [ContinueTime_reg,ContinueTime_idx] = find(CLRM_idx == Times(FieldLifeTime-
Shift_time+2)+1);
        for i_path = 1: length(ContinueTime_reg)
            if CLRM_OptNPV_real(ContinueTime_reg(i_path),1) == 0
               CLRM_OptNPV_real(ContinueTime_reg(i_path),1) =
y_data_switch_CLRM(ContinueTime_reg(i_path),end);
               CLRM_PhaseLifes_Opt_reals(ContinueTime_reg(i_path),1) = FieldLifeTime;
               CLRM_PhaseLifes_Opt_reals(ContinueTime_reg(i_path),2) = 0;
            else
                continue;
            end
        end
    else
        [SwitchDecision_CLRM,SwitchDecision_CLRM_idx] = find(CLRM_idx <= Times(FieldLifeTime-
Shift_time+2));
    end

    x_data_stop_CLRM = ObsRate_Phase1n2_real_time_ShiftTime(:,:,Shift_time);
    t_dataPhase2 = 1;
    while t_dataPhase2 <= Times(FieldLifeTime-Shift_time+1)+1
        X_stop_CLRM = [ones(N_MC,1) x_data_stop_CLRM(:,1:t_dataPhase2+Shift_time-2)
oilprice_data(:,t_dataPhase2+Shift_time-2) opex_phase2_data(:,t_dataPhase2+Shift_time-2)];
        y_data_stop_CLRM = y_data_switch_CLRM(:, t_dataPhase2:Times(FieldLifeTime-
Shift_time+1)+1);
        y_reg_stop_CLRM = zeros(N_MC,Times(FieldLifeTime-Shift_time+1)-t_dataPhase2+2);
        for t = 1: Times(FieldLifeTime-Shift_time+1)-t_dataPhase2+2
            y_reg_stop_CLRM(:,t) = X_stop_CLRM*regress(y_data_stop_CLRM(:,t),X_stop_CLRM);
        end
            [StopTime_reg,StopTime_idx] = max(y_reg_stop_CLRM,[],2);
        for i_path = 1: length(SwitchDecision_CLRM)
            if StopTime_reg(SwitchDecision_CLRM(i_path)) ==
y_reg_stop_CLRM(SwitchDecision_CLRM(i_path),1) &&
CLRM_OptNPV_real(SwitchDecision_CLRM(i_path),1) == 0
                CLRM_OptNPV_real(SwitchDecision_CLRM(i_path),1) =
y_data_stop_CLRM(SwitchDecision_CLRM(i_path),1);
                CLRM_PhaseLifes_Opt_reals(SwitchDecision_CLRM(i_path),1) = Shift_time-1;
                CLRM_PhaseLifes_Opt_reals(SwitchDecision_CLRM(i_path),2) =
```

```
t_dataPhase2+StopTime_idx(SwitchDecision_CLRM(i_path))-2;
            else
                continue;
            end
        end
            t_dataPhase2 = t_dataPhase2+1;
    end
    end
end

DWI_CLRM_Phase1LifeTime = CLRM_PhaseLifes_Opt_reals(:,1);
DWI_CLRM_Phase2LifeTime = CLRM_PhaseLifes_Opt_reals(:,2);
DWI_CLRM_LifeTime = DWI_CLRM_Phase1LifeTime + DWI_CLRM_Phase2LifeTime;

EVWI_CLRM = mean(CLRM_OptNPV_real);
VOI_CLRM = EVWI_CLRM-EVWOI;
```

Appendix B6

Appendix B6.1: Probabilistic Plot of the Oil Price Model

This code makes the plots of P10, P50 and P90 of the simulated oil price.

```
N_MC = 100000;
FieldLifeTime = 50;
P10_OilPrice = zeros(1,FieldLifeTime+1);
P50_OilPrice = zeros(1,FieldLifeTime+1);
P90_OilPrice = zeros(1,FieldLifeTime+1);

LifeTime = 0:1:FieldLifeTime;

for i_LifeTime = 2: FieldLifeTime+1
    P10_OilPrice(:,i_LifeTime) = prctile(OilPrice_Matrix(:,i_LifeTime-1),10);
    P50_OilPrice(:,i_LifeTime) = prctile(OilPrice_Matrix(:,i_LifeTime-1),50);
    P90_OilPrice(:,i_LifeTime) = prctile(OilPrice_Matrix(:,i_LifeTime-1),90);
end

P10_OilPrice(:,1) = OilPrice;
P50_OilPrice(:,1) = OilPrice;
P90_OilPrice(:,1) = OilPrice;

figure
P10 = plot(LifeTime,P10_OilPrice); hold on;
P50 = plot(LifeTime,P50_OilPrice); hold on;
P90 = plot(LifeTime,P90_OilPrice); hold on;

set(P10,'LineWidth',2);
set(P50,'LineWidth',2);
set(P90,'LineWidth',2);

legend('P10','P50','P90');
xlabel('Time (years)');
ylabel('Oil Price (USD/bbl)');
```

Appendix B6.2: Cumulative Distribution Functions (CDF) Plot of NPV corresponding to DWOI, DWII (SRDM), and DWPI

```
N_MC = 100000;
FieldLifeTime = 50;
N_percentile = 99;
N_Phase1n2LifeTime = sum(0:FieldLifeTime+1);

NPV_DWOI = NPVtable_LTPhase1_LTPhase2_real(DWOI_Phase1LifeTime+1,DWOI_Phase2LifeTime+1,:);

NPV_DWII = zeros(N_MC,1);
NPV_DWPI = zeros(N_MC,1);

for i_MC = 1: N_MC
    NPV_DWII(i_MC,1) =
NPVtable_LTPhase1_LTPhase2_real(DWI_Phase1LifeTime(i_MC,1)+1,DWI_Phase2LifeTime(i_MC,1)+1,i_M
C);
    NPV_DWPI(i_MC,1) =
NPVtable_LTPhase1_LTPhase2_real(DWPI_Phase1LifeTime_real(i_MC,1)+1,DWPI_Phase2LifeTime_real(i
_MC,1)+1,i_MC);
end

Percentile_DWOI = zeros(N_percentile+2,1);
Percentile_DWII = zeros(N_percentile+2,1);
Percentile_DWPI = zeros(N_percentile+2,1);

for i_percentile = 2: N_percentile+1
    Percentile_DWOI(i_percentile,1) = prctile(NPV_DWOI,i_percentile-1);
    Percentile_DWII(i_percentile,1) = prctile(NPV_DWII,i_percentile-1);
    Percentile_DWPI(i_percentile,1)= prctile(NPV_DWPI,i_percentile-1);
end

Percentile_DWOI(1,1) = Percentile_DWOI(2,1);
Percentile_DWII(1,1) = Percentile_DWII(2,1);
Percentile_DWPI(1,1)= Percentile_DWPI(2,1);

Percentile_DWOI(101,1) = Percentile_DWOI(100,1);
Percentile_DWII(101,1) = Percentile_DWII(100,1);
Percentile_DWPI(101,1)= Percentile_DWPI(100,1);

figure;
y = 0:0.01:1;
cdf_DWOI = plot(Percentile_DWOI,y); hold on;
cdf_DWII = plot(Percentile_DWII,y); hold on;
cdf_DWPI = plot(Percentile_DWPI,y); hold on;

set(cdf_DWOI,'LineWidth',2);
set(cdf_DWII,'LineWidth',2);
set(cdf_DWPI,'LineWidth',2);

set(cdf_DWOI,'Color','blue');
set(cdf_DWII,'Color','red');
set(cdf_DWPI,'Color','green');

legend('DWOI','DWII','DWPI');
```

```
xlabel('NPV (million USD)');
ylabel('CDF');
```

Appendix B6.3: Normalized Frequency Distributions (NFD) of DWII

This code plots the Normalized Frequency Distributions (NFD) of DWIIs corresponding to SRDM and CLRM.

```
N_MC = 100000;
FieldLifeTime = 50;
N_Phase1n2LifeTime = sum(0:FieldLifeTime+1);

% 1. Lifetime of Primary Recovery
figure;
hist_DWII_SRDM_LT1 =
histogram(DWI_Phase1LifeTime,51,'Normalization','probability','BinWidth',1,'BinLimits',[-0.5
50.5]); hold on;
hist_DWII_CLRM_LT1 =
histogram(DWI_CLRM_Phase1LifeTime,51,'Normalization','probability','BinWidth',1,'BinLimits',[
-0.5 50.5]); hold on;
legend('SRDM','CLRM');
xlabel('Lifetime of Primary Recovery (years)');
ylabel('NFD');

% 2. Lifetime of Secondary Recovery
figure;
hist_DWII_SRDM_LT2 =
histogram(DWI_Phase2LifeTime,51,'Normalization','probability','BinWidth',1,'BinLimits',[-0.5
50.5]); hold on;
hist_DWII_CLRM_LT2 =
histogram(DWI_CLRM_Phase2LifeTime,51,'Normalization','probability','BinWidth',1,'BinLimits',[
-0.5 50.5]); hold on;
legend('SRDM','CLRM');
xlabel('Lifetime of Secondary Recovery (years)');
ylabel('NFD');

% 3. Total Lifetime
figure;
hist_DWII_SRDM_TLT =
histogram(DWI_LifeTime,51,'Normalization','probability','BinWidth',1,'BinLimits',[-0.5
50.5]); hold on;
hist_DWII_CLRM_tLT2 =
histogram(DWI_CLRM_LifeTime,51,'Normalization','probability','BinWidth',1,'BinLimits',[-0.5
50.5]); hold on;
legend('SRDM','CLRM');
xlabel('Total Lifetime (years)');
ylabel('NFD');
```

Appendix B6.4: Normalized Cumulative Frequency Distributions (NCFD) of DWII

This code plots the Normalized Cumulative Frequency Distributions (NCFD) of DWIIs corresponding to SRDM and CLRM.

```
N_MC = 100000;
FieldLifeTime = 50;
N_Phase1n2LifeTime = sum(0:FieldLifeTime+1);

% 1. Lifetime of Primary Recovery
figure;
hist_DWII_SRDM_LT1 =
histogram(DWI_Phase1LifeTime,51,'Normalization','cdf','BinWidth',1,'BinLimits',[-0.5 50.5]);
hold on;
hist_DWII_CLRM_LT1 =
histogram(DWI_CLRM_Phase1LifeTime,51,'Normalization','cdf','BinWidth',1,'BinLimits',[-0.5
50.5]); hold on;
legend('SRDM','CLRM');
xlabel('Lifetime of Primary Recovery (years)');
ylim([0 1]);
ylabel('NCFD');

% 2. Lifetime of Secondary Recovery
figure;
hist_DWII_SRDM_LT2 =
histogram(DWI_Phase2LifeTime,51,'Normalization','cdf','BinWidth',1,'BinLimits',[-0.5 50.5]);
hold on;
hist_DWII_CLRM_LT2 =
histogram(DWI_CLRM_Phase2LifeTime,51,'Normalization','cdf','BinWidth',1,'BinLimits',[-0.5
50.5]); hold on;
legend('SRDM','CLRM');
xlabel('Lifetime of Secondary Recovery (years)');
ylim([0 1]);
ylabel('NCFD');

% 3. Total Lifetime
figure;
hist_DWII_SRDM_TLT =
histogram(DWI_LifeTime,51,'Normalization','cdf','BinWidth',1,'BinLimits',[-0.5 50.5]); hold
on;
hist_DWII_CLRM_tLT2 =
histogram(DWI_CLRM_LifeTime,51,'Normalization','cdf','BinWidth',1,'BinLimits',[-0.5 50.5]);
hold on;
legend('SRDM','CLRM');
xlabel('Total Lifetime (years)');
ylim([0 1]);
ylabel('NCFD');
```

Appendix B6.5: Plot of Different Combinations of Lifetimes of Primary and Secondary Recoveries

This code makes the normalized frequency plot of different combinations of lifetimes of primary and secondary recoveries This code is originally built by Hong et al. (2018). It is slightly modified in this work.

```
% 1. SRDM Approach

Phase1n2LifeTime_Prob = zeros(N_Phase1n2LifeTime,1);
for i_Phase1n2LifeTime = 1:N_Phase1n2LifeTime %% Combinations of LT1 and LT2
    Counter = 0;
    for i_real = 1:N_MC
        if DWI_Phase1LifeTime(i_real,1) == Phase1n2LifeTimeTable(i_Phase1n2LifeTime,1) &&
DWI_Phase2LifeTime(i_real,1) == Phase1n2LifeTimeTable(i_Phase1n2LifeTime,2)
            Counter = Counter+1;
        end
    end
    Phase1n2LifeTime_Prob(i_Phase1n2LifeTime,1) = Counter;
end
Phase1n2LifeTime_Prob = Phase1n2LifeTime_Prob/N_MC;

rho_SRDM = corr(DWI_Phase1LifeTime, DWI_Phase2LifeTime);
figure;
scatter(Phase1n2LifeTimeTable(:,1),Phase1n2LifeTimeTable(:,2),[],Phase1n2LifeTime_Prob,'fille
d','s');
colormap hot
colorbar;
xlabel('Lifetime of Primary Recovery (years)');
ylabel('Lifetime of Secondary Recovery (years)');

% 2. CLRM Approach

Phase1n2LifeTime_Prob = zeros(N_Phase1n2LifeTime,1);
for i_Phase1n2LifeTime = 1:N_Phase1n2LifeTime %% Combinations of LT1 and LT2
    Counter = 0;
    for i_real = 1:N_MC
        if DWI_CLRM_Phase1LifeTime(i_real,1) == Phase1n2LifeTimeTable(i_Phase1n2LifeTime,1)
&& DWI_CLRM_Phase2LifeTime(i_real,1) == Phase1n2LifeTimeTable(i_Phase1n2LifeTime,2)
            Counter = Counter+1;
        end
    end
    Phase1n2LifeTime_Prob(i_Phase1n2LifeTime,1) = Counter;
end
Phase1n2LifeTime_Prob = Phase1n2LifeTime_Prob/N_MC;

rho_CLRM = corr(DWI_CLRM_Phase1LifeTime, DWI_CLRM_Phase2LifeTime);
figure;
scatter(Phase1n2LifeTimeTable(:,1),Phase1n2LifeTimeTable(:,2),[],Phase1n2LifeTime_Prob,'fille
d','s');
colormap hot
colorbar
xlabel('Lifetime of Primary Recovery (years)');
ylabel('Lifetime of Secondary Recovery (years)');
```

Appendix B6.6: Mean Oil Rates Plot

This code plots the graph of mean oil rates against time (year) for DWOI, SRDM and CLRM.

```
N_MC = 100000;
FieldLifeTime = 50;
N_Phase1n2LifeTime = sum(0:FieldLifeTime+1);
LifeTime = 1:1:FieldLifeTime;

Mean_ObsRates_DWOI = mean(ObsRate_Phase1n2_real_time_ShiftTime(:,:,DWOI_Phase1LifeTime+1));
Mean_ObsRates_DWOI(:,DWOI_LifeTime+1:FieldLifeTime) = 0;

ObsRates_DWI_SRDM = zeros(N_MC,FieldLifeTime);
ObsRates_DWI_CLRM = zeros(N_MC,FieldLifeTime);

for i_MC = 1: N_MC
  for i_time = 1: FieldLifeTime
    ObsRates_DWI_SRDM(i_MC,i_time) =
ObsRate_Phase1n2_real_time_ShiftTime(i_MC,i_time,DWI_Phase1LifeTime(i_MC,1)+1);
    ObsRates_DWI_CLRM(i_MC,i_time) =
ObsRate_Phase1n2_real_time_ShiftTime(i_MC,i_time,DWI_CLRM_Phase1LifeTime(i_MC,1)+1);
  end
    ObsRates_DWI_SRDM(i_MC,DWI_LifeTime(i_MC,1)+1:FieldLifeTime) = 0;
    ObsRates_DWI_CLRM(i_MC,DWI_CLRM_LifeTime(i_MC,1)+1:FieldLifeTime) = 0;
end

Mean_ObsRates_DWI_SRDM = mean(ObsRates_DWI_SRDM);
Mean_ObsRates_DWI_CLRM = mean(ObsRates_DWI_CLRM);

figure;
mean_rates_DWOI_plot = plot(LifeTime,Mean_ObsRates_DWOI); hold on;
mean_rates_DWII_SRDM_plot = plot(LifeTime,Mean_ObsRates_DWI_SRDM); hold on;
mean_rates_DWII_CLRM_plot = plot(LifeTime,Mean_ObsRates_DWI_CLRM); hold on;

legend('DWOI','SRDM','CLRM');
set(mean_rates_DWOI_plot,'LineWidth',2);
set(mean_rates_DWII_SRDM_plot,'LineWidth',2);
set(mean_rates_DWII_CLRM_plot,'LineWidth',2);

set(mean_rates_DWOI_plot,'Color','blue');
set(mean_rates_DWII_SRDM_plot,'Color','red');
set(mean_rates_DWII_CLRM_plot,'Color',[0 0.7 0.2]);

legend('DWOI','SRDM','CLRM');
xlabel('Time (years)');
ylabel('Mean OIl Production Rate (million bbl/year)');
```

Appendix B6.7: Mean Cumulative Discounted Cashflow Plot for DWOI and DWIIs of SRDM and CLRM

This code plots the graph of mean cumulative discounted cashflow against time (year) for DWOI, DWIIs of SRDM and CLRM.

```
N_MC = 100000;
FieldLifeTime = 50;
N_Phase1n2LifeTime = sum(0:FieldLifeTime+1);
LifeTime = 1:1:FieldLifeTime;

CDCF_DWOI = DisCashFlow_ShiftTime(:,:,DWOI_Phase1LifeTime+1);

for i_time = 2: FieldLifeTime
    if i_time > DWOI_LifeTime
    CDCF_DWOI(:,i_time) = CDCF_DWOI(:,DWOI_LifeTime);
    else
    CDCF_DWOI(:,i_time) = CDCF_DWOI(:,i_time)+CDCF_DWOI(:,i_time-1);
    end
end

Mean_CDCF_DWOI = mean(CDCF_DWOI);

CDCF_DWI_SRDM = zeros(N_MC,FieldLifeTime);
CDCF_DWI_CLRM = zeros(N_MC,FieldLifeTime);

for i_MC = 1: N_MC
    CDCF_DWI_SRDM(i_MC,:) = DisCashFlow_ShiftTime(i_MC,:,DWI_Phase1LifeTime(i_MC,1)+1);
    CDCF_DWI_CLRM(i_MC,:) = DisCashFlow_ShiftTime(i_MC,:,DWI_CLRM_Phase1LifeTime(i_MC,1)+1);
end

for i_MC = 1: N_MC
    for i_time = 2: FieldLifeTime
       if i_time > DWI_LifeTime(i_MC,1)
       CDCF_DWI_SRDM(i_MC,i_time) = CDCF_DWI_SRDM(i_MC,DWI_LifeTime(i_MC,1));
       else
       CDCF_DWI_SRDM(i_MC,i_time) = CDCF_DWI_SRDM(i_MC,i_time)+CDCF_DWI_SRDM(i_MC,i_time-1);
       end
    end
end

for i_MC = 1: N_MC
    for i_time = 2: FieldLifeTime
       if i_time > DWI_CLRM_LifeTime(i_MC,1)
       CDCF_DWI_CLRM(i_MC,i_time) = CDCF_DWI_CLRM(i_MC,DWI_CLRM_LifeTime(i_MC,1));
       else
       CDCF_DWI_CLRM(i_MC,i_time) = CDCF_DWI_CLRM(i_MC,i_time)+CDCF_DWI_CLRM(i_MC,i_time-1);
       end
    end
end

Mean_CDCF_DWI_SRDM = mean(CDCF_DWI_SRDM);
Mean_CDCF_DWI_CLRM = mean(CDCF_DWI_CLRM);

figure;
mean_CDCF_DWOI_plot = plot(LifeTime,Mean_CDCF_DWOI); hold on;
```

```
mean_CDCF_DWII_SRDM_plot = plot(LifeTime,Mean_CDCF_DWI_SRDM); hold on;
mean_CDCF_DWII_CLRM_plot = plot(LifeTime,Mean_CDCF_DWI_CLRM); hold on;

set(mean_CDCF_DWOI_plot,'LineWidth',2);
set(mean_CDCF_DWII_SRDM_plot,'LineWidth',2);
set(mean_CDCF_DWII_CLRM_plot,'LineWidth',2);

set(mean_CDCF_DWOI_plot,'Color','blue');
set(mean_CDCF_DWII_SRDM_plot,'Color','red');
set(mean_CDCF_DWII_CLRM_plot,'Color',[0 0.7 0.2]);

legend('DWOI','SRDM','CLRM');
xlabel('Time (years)');
ylabel('Mean CDCF (million USD)');
```

Appendix B7

There are six codes under Appendix B7. These codes are used to illustrate how the modified LSM algorithm can be applied to a reservoir simulation model. These codes have to be executed in the order of how they are presented to get the correct results. After running these 6 codes, sections 3 to 6 from the modified LSM algorithm (Appendix B1.1) and CLRM (Appendix B1.2) should be run to determine the optimal decision policies.

Appendix B7.1: Generation of Different Realizations of Permeability

This code generates different values of horizontal permeability for top and bottom layers using normal distribution.

```
N_realization = 100;
PermX = zeros(N_realization,2);
PERMX_INIT_a = zeros(2500,1);

mean_PermX_Layer1 = 250;
SD_PermX_Layer1 = 35;
mean_PermX_Layer2 = 178;
SD_PermX_Layer2 = 20;

for i_realization = 1: N_realization
    filename = ['C:\Users\cuthb\OneDrive -
NTNU\2D_LSM_Reservoir_Model\PERMX\PERMX',num2str(i_realization),'.inc'];
    PermX(i_realization,1) = normrnd(mean_PermX_Layer1,SD_PermX_Layer1);
    PermX(i_realization,2) = normrnd(mean_PermX_Layer2,SD_PermX_Layer2);

    PERMX_INIT_a(1:1000,1) = PermX(i_realization,1);
    PERMX_INIT_a(1001:2500,1) = PermX(i_realization,2);

    fileID = fopen(filename,'w');
    fprintf(fileID,'%s\n','PERMX');
    fprintf(fileID,'%g\n',PERMX_INIT_a);
    fprintf(fileID,'%s','/');
    fclose(fileID);
end
```

Appendix B7.2: Generation of Files for Reservoir Simulation

This code generates different '.DATA' files accordingly along with the 'PERMX.inc' files to be used for ECLIPSE (2016) simulation.

```
Year = 6;
Realizations = 100;

for i_Year = 1: Year

BaseFiles = ['C:\Users\cuthb\OneDrive -
```

```
NTNU\2D_LSM_Reservoir_Model\Switch_at_Year_',num2str(i_Year),'\BaseFile'];

for i = 1:Realizations
    RealizationFolder =['C:\Users\cuthb\OneDrive - 
NTNU\2D_LSM_Reservoir_Model\Switch_at_Year_',num2str(i_Year),'\Simulation\R',num2str(i)];
    copyfile(BaseFiles,RealizationFolder);
    OldName = [RealizationFolder,'\2D_Model_Switch_Y',num2str(i_Year),'.DATA'];
    NewName = 
[RealizationFolder,'\2D_Model_Switch_Y',num2str(i_Year),'_R',num2str(i),'.DATA'];
    movefile(OldName,NewName)
    PERMXfrom = ['C:\Users\cuthb\OneDrive - 
NTNU\2D_LSM_Reservoir_Model\PERMX\PERMX',num2str(i),'.inc'];
    PERMXto = [RealizationFolder,'\PERMX.inc'];
    copyfile(PERMXfrom,PERMXto);
end

end
```

Appendix B7.3: Run Eclipse

This code calls for the running of the ECLIPSE (2016) simulation with the help of runecl ().

```
Year = 6;
Foldername = 'C:\Users\cuthb\OneDrive - NTNU\2D_LSM_Reservoir_Model';

for i_Year = 1: Year
    for i_realization = 1: N_realization
        filename = 
[Foldername,'\Switch_at_Year_',num2str(i_realization),'\Simulation\R',num2str(i_realization)]
;
        runecl(filename);
    end
end
```

The function of runecl () is shown below.

```
function run = runecl(filename)
% Call Eclipse to run a data-file in Matlab
    cmd=['!C:\ecl\macros\eclrun.exe eclipse ' filename];
    eval(cmd);

end
```

Appendix B7.4: Extraction and Modification of RSM File

This code mainly extracts the output data (RSM file) for each realization and changes it to '-txt.' file to be imported into MATLAB R2019a (2019) for further analysis.

```
Year = 6;
N_realization = 100;
Foldername = 'C:\Users\cuthb\OneDrive - NTNU\2D_LSM_Reservoir_Model';

for i_Year = 1: Year
    for i_realization = 1: N_realization
        RealizationFile_from =
[Foldername,'\Data\Switch_at_Year_',num2str(i_Year),'\Simulation\R',num2str(i_realization),'\
2D_MODEL_SWITCH_Y',num2str(i_Year),'_R',num2str(i_realization),'.RSM'];
        RealizationFile_to =
[Foldername,'\Switch_at_Year_',num2str(i_Year),'\TextFile\2D_MODEL_SWITCH_Y',num2str(i_Year),
'_R',num2str(i_realization),'.txt'];
        copyfile(RealizationFile_from,RealizationFile_to);

    end
end
```

Appendix B7.5: LSM Input Data Generation (only applicable to Reservoir Model described in this work)

This code generates the input data used for the modified LSM method from the output file of the Reservoir Simulation. The RSM file only provides the cumulative production (cubic meter) for each year. Since the state variable is measured oil rates (MMbbl/year), calculation of oil rate and unit conversion are done by this code.

```
Year = 5;
N_realization = 100;

% DataPoint defines the number of rows corresponding to the data at particular year
% For instance, the data at Year 1 is in row 9
% DataPoint is very case-dependent and needs to be modified
% accordingly if different reservoir model is used.
DataPoint = [9;10;11;12;13];
Data_Matrix = zeros(N_realization,Year,Year+1);
Foldername = 'C:\Users\cuthb\OneDrive - NTNU\2D_LSM_Reservoir_Model';

for i_Year = 1: Year+1
    for i_realization = 1: N_realization
        Filename =
[Foldername,'\Switch_at_Year_',num2str(i_Year),'\TextFile\2D_MODEL_SWITCH_Y',num2str(i_Year),
'_R',num2str(i_realization),'.txt'];
        DataFile = importdata(Filename);

        % Columns 4 and 5 correspond the cumulative production for top and bottom layer
        Y1_Data_metric = (DataFile.data(DataPoint(1),4) +
DataFile.data(DataPoint(1),5))/1000000;
```

```
        Y2_Data_metric = ((DataFile.data(DataPoint(2),4) - DataFile.data(DataPoint(1),4)) +
(DataFile.data(DataPoint(2),5) - DataFile.data(DataPoint(1),5)))/1000000;
        Y3_Data_metric = ((DataFile.data(DataPoint(3),4) - DataFile.data(DataPoint(2),4)) +
(DataFile.data(DataPoint(3),5) - DataFile.data(DataPoint(2),5)))/1000000;
        Y4_Data_metric = ((DataFile.data(DataPoint(4),4) - DataFile.data(DataPoint(3),4)) +
(DataFile.data(DataPoint(4),5) - DataFile.data(DataPoint(3),5)))/1000000;
        Y5_Data_metric = ((DataFile.data(DataPoint(5),4) - DataFile.data(DataPoint(4),4)) +
(DataFile.data(DataPoint(5),5) - DataFile.data(DataPoint(4),5)))/1000000;

        % Unit conversion from cubic meter to bbl
        Y1_Data = Y1_Data_metric.*6.29;
        Y2_Data = Y2_Data_metric.*6.29;
        Y3_Data = Y3_Data_metric.*6.29;
        Y4_Data = Y4_Data_metric.*6.29;
        Y5_Data = Y5_Data_metric.*6.29;

        Data_Matrix(i_realization,1,i_Year) = Y1_Data;
        Data_Matrix(i_realization,2,i_Year) = Y2_Data;
        Data_Matrix(i_realization,3,i_Year) = Y3_Data;
        Data_Matrix(i_realization,4,i_Year) = Y4_Data;
        Data_Matrix(i_realization,5,i_Year) = Y5_Data;
    end
end
```

Appendix B7.6: Initialization of Input Parameters

This code computes the measured oil rates used for the modified LSM algorithm by using the modeled rates from ECLIPSE (2016) reservoir simulation and initializes the economic parameters.

```
% 1. Computation of Measured Oil Production Rates
Rate_Phase1n2_real_time_ShiftTime = Data_Matrix;
ErrorMean4Rate_Pct = 0;
ErrorSD4Rate_Pct = 0.25;
ErrorSD_Matrix = ErrorSD4Rate_Pct.*Rate_Phase1n2_real_time_ShiftTime;
ObsRate_Phase1n2_real_time_ShiftTime =
normrnd(Rate_Phase1n2_real_time_ShiftTime,ErrorSD_Matrix);

N_MC = size(ObsRate_Phase1n2_real_time_ShiftTime,1);
FieldLifeTime = size(ObsRate_Phase1n2_real_time_ShiftTime,2);

% 2. Initialization of Values of Economic Parameters
OilPrice = 90;
Capex_Phase1 = 1;
Capex_Phase2After1 = 1.15;
Capex_Phase2No1 = 1.15;
Opex_Phase1 = 1.4;
Opex_Phase2 = 1.6;
DisRate = 0.05;
```

Appendix C: ECLIPSE Data File

The ECLIPSE (2016) data file is presented below. However, this data file only includes the switch time at Year 1. However, to create the data file for another switch time, it can be obtained by shifting the positions of "`WELSPECS`" of the injector and " `WCONINJE`" according to the timestep, which is represented as "`TSTEP`". For example, in this case, if the data file of having switch time at Year 3 is needed (indicates that waterflooding is initiated at the beginning of Year 3), "`WELSPECS`" of the injector and " `WCONINJE`" are placed after 2 "`TSTEP`" (because each `TSTEP`" is defined as 365 days or a year in this data file). In addition, the water/oil saturation functions (`SWOF`) used in this file is the modified example of data set presented in ECLIPSE Reference Manual prepared by Schlumberger (2014).

ECLIPSE Data File: 2D and 2 Layered Reservoir Model with the Waterflooding Study and Switch Time at Year 1

```
-- 2D water flooding model (50m*50m*1m and 50*1*100 Blocks)
-- Total Dimension of model is 2500m*50m*100m
-- There are two layers in this reservoir model
-- This Data File corresponds to the switch time at Year 1
-- Model made by Cuthbert Shang Wui Ng

======================================================================

RUNSPEC
TITLE
 2D Reservoir Model for Modified LSM Algorithm
DIMENS
   50   1   100  /

OIL

WATER

METRIC

TABDIMS
    2    1   20    2    2   2 /

WELLDIMS
    2    100    1    2 /

START
   1 'JAN' 2019  /

NSTACK
   100 /

UNIFOUT

UNIFIN

GRID
```

```
DEBUG
38*0 1  /

INIT

DXV
  50*50/

DYV
  1*50 /

DZ
  5000*1
/

INCLUDE
'PERMX.inc'  /
/

COPY
PERMX PERMY /
PERMX PERMZ /
/

MULTIPLY
  PERMZ 0.0001 /
/

BOX
1 50 1 1 1 100 /

TOPS
50*2000
50*2001
50*2002
50*2003
50*2004
50*2005
50*2006
50*2007
50*2008
50*2009
50*2010
50*2011
50*2012
50*2013
50*2014
50*2015
50*2016
50*2017
50*2018
50*2019
50*2020
50*2021
50*2022
50*2023
50*2024
50*2025
```

```
50*2026
50*2027
50*2028
50*2029
50*2030
50*2031
50*2032
50*2033
50*2034
50*2035
50*2036
50*2037
50*2038
50*2039
50*2040
50*2041
50*2042
50*2043
50*2044
50*2045
50*2046
50*2047
50*2048
50*2049
50*2050
50*2051
50*2052
50*2053
50*2054
50*2055
50*2056
50*2057
50*2058
50*2059
50*2060
50*2061
50*2062
50*2063
50*2064
50*2065
50*2066
50*2067
50*2068
50*2069
50*2070
50*2071
50*2072
50*2073
50*2074
50*2075
50*2076
50*2077
50*2078
50*2079
50*2080
50*2081
50*2082
50*2083
```

```
50*2084
50*2085
50*2086
50*2087
50*2088
50*2089
50*2090
50*2091
50*2092
50*2093
50*2094
50*2095
50*2096
50*2097
50*2098
50*2099
/

PORO
  2000*0.210
  3000*0.218
/

RPTGRID
--
'DX'
'DY'
'DZ'
'PERMX'
'PERMY'
 /

DEBUG
 0 0 1 0 1 0 1  /

PROPS

SWOF
0.05 0.0000  1.0000 11.000
0.15 0.0100  0.7000 8.5000
0.22 0.0350  0.5500 7.0000
0.30 0.0700  0.4000 4.0000
0.40 0.1500  0.1250 3.0000
0.50 0.2000  0.0649 2.7000
0.60 0.3300  0.0048 2.0000
0.80 0.6500  0.0030 1.0000
0.90 0.8300  0.0001 0.7200
1.00 1.0000  0.0    0.0000/
/


PVTW
  234  1.0  4.28e-5  0.5 /
/

PVCDO
  234      1.0      6.65e-5      6.4   192.e-5    /
```

```
RSCONSTT
  0.00  43.8 /

ROCK
  234    1E-8 /

PCW
5000*73 /

DENSITY
 1000  1000  0.824 /

RPTPROPS
/

REGIONS

FIPNUM
 2000*1 3000*2/

SATNUM
 2000*1 3000*2/


SOLUTION

PRESSURE
5000*234/

SWAT
  2000*0.10
  3000*0.12
/

RPTSOL
-- Initialisation Print Output
--
'PRES' 'SWAT' /

SUMMARY

EXCEL
SEPARATE

ROPT
/

ROPR
1 2/

SCHEDULE

TUNING
/
/ LITMIN DDPLIM DDSLIM
24 1 50 3 24 16 0.0001 0.0001 /

RPTSCHED
```

```
 'PRES' 'SWAT' 'RESTART=1' 'CPU=2' /

WELSPECS
'I'  'G'   1    1  2100  'WAT'  /
'P'  'G'   50   1  2100  'OIL'  /
/

COMPDAT
'I'  1*  1*   1  100 'OPEN'  1*    1*   1.0 /
'P'  1*  1*   1  100 'OPEN'  1*    1*   1.0 /
/

WCONPROD
'P' 'OPEN' 'LRAT' 3* 15/
/

WCONINJE
'I' 'WAT' 'OPEN' 'RATE' 15 /
/


TSTEP
 365
/

TSTEP
 365
/

TSTEP
 365
/

TSTEP
 365
/

TSTEP
 365
/

END
```