University of Stavanger

FACULTY OF SCIENCE AND TECHNOLOGY

MASTER'S THESIS

Study programme/specialisation: Petroleum Engineering/Reservoir Engineering	Spring semester, 2021 Open
Author: Lucas Emanuel Andersson	
Supervisors: Professor Reidar B. Bratvold and Associate Professor Aojie Hong	
Title of master's thesis: Valuing abandonment decisions: Maximizing an oil and gas fields Net Present Value	
Credits: 30 ETCS	
Keywords: Real option valuation Least Square Monte Carlo Abandonment Case study Stochastic dynamic programming Optimize Timing Net present value	Number of pages: 60 Stavanger, June 14,2021

Title page for master's thesis
Faculty of Science and Technology

Acknowledgments

I would like to thank Professor Reidar B. Bratvold and Associate Professor Aojie Hong for helping and guiding my work. It has been an educational path within a fascinating subject. Last, my thanks go to my fiancée for her support through my master’s studies.

Abstract

The oil and gas sector in the North Sea is mature and consists primarily of brownfields that have passed their peak production or are close to the end of their lives. Many of these fields will have to be abandoned in the upcoming decade. The decision to abandon an oil and gas field carries a considerable cost, in addition to ending the future revenue stream of the field. If a field is abandoned before the optimal time, value will be lost as a result of revenue and profit losses. Conversely, if it is abandoned too late, value will be destroyed as a result of carrying higher costs than revenues, i.e., as a result of negative profits. Therefore, the timing of the abandonment decision is critical as it can either destroy or create significant value. Adding to the overall challenge of identifying the optimal abandonment time is the uncertainty associated with many of the value-determining parameters, such as price, cost, and production. The best method to identify the optimal time is to apply a consistent decision analytic approach, which includes uncertainties and the operator's flexibility in choosing when to initiate the abandonment.

In this thesis we develop a case study representative of an oil field with declining production. We then implement three different approaches to estimate value and determine the optimal abandonment time. The three methods are: (i) a negative cashflow approach that abandons the field at the first negative cashflow, (ii) a greedy optimization approach that abandons the field after an already determined set waiting time criterion is fulfilled, and (iii) a real option valuation method that abandons the field when the economic outlook becomes unfavorable. The results were compared and evaluated, after which the greedy and real option valuation approaches were further assessed using sensitivity analysis.

We conclude that value is almost always gained by not abandoning the field at the first negative cashflow. Using a waiting criterion for the abandonment decision significantly improved the timing, resulting in a higher net present value. The best method for our case is real option valuation, which creates value through the combined effects of uncertainty and flexibility.

Content

List of figures

List of tables

1 Introduction

The oil and gas sectors of Norway and the UK currently face the same challenge, as numerous mature fields will need to be abandoned in the coming decade. This will cause the stakeholders of the fields to incur large costs, as wells need to be plugged and sites restored to their original condition. The timing of the decision has large effects on the fields net present value (NPV) as abandoning to early will result in lost revenues and too late by accumulating costs that exceeds the revenues. Hence, it is important to create models that can capture the value-creating potential from dynamical abandonment decision combined with uncertainty.

This thesis first builds a case study of an oil and gas field with similar characteristics to those of Veslefrikk, located on the Norwegian Continental Shelf (NCS). This late-life field is facing declining production. The uncertain variables in the case study are the variable operative expenses and oil price, which were calibrated with historical data to determine prospective ranges and volatility. We then used three methods to determine the best time for abandonment. The first method is the first negative cashflow approach (FNC), whereby the field is abandoned after the first month of negative cashflow. The second is a greedy optimization approach (GO), in which abandonment follows the fulfillment of a predetermined continuous negative cashflow criterion. The last method is a real option valuation (ROV) implemented using the least-squares Monte Carlo (LSM) algorithm that abandons the field when the economic outlook turns unfavorable. The results of the methods were compared, and a sensitivity analysis was conducted between the GO and LSM to assess the robustness of these results.

We conclude that FNC is the worst strategy to find the optimal time of abandonment and maximize the net present value (NPV) of the field, because it does not capture possible values created by postponing the abandonment cost after the first negative cashflow has occur (time-value-money) or the possibility that the field could become cash-flow positive in the future. While the GO approach significantly improves the timing and the expected NPV, it does not reflect realistic decision-making, as it fails to consider the change in economic outlook of the field with time. The optimal method is the ROV, which mimics how decisions are really made. Implemented using the LSM method, this last approach quantifies uncertainties in future values from current information (taken at any time) to determine if a given time point is a favorable time to execute the abandonment. The sensitivity analysis of the LSM method indicates that this approach yields a $1.02-2.55 Million (MM) higher NPV than that from the GO approach and a $38.67 MM improvement over the FNC method in the base case.

1.1 Veslefrikk

The model is based on historical and estimated data representing Veslefrikk. Located in the northern part of the North Sea, Veslefrikk was discovered in 1981 (The NPD, 2021). Production started in late 1989 using a hybrid solution consisting of a fixed platform connected to a semisubmersible facility. The platform supports the wellheads and drilling system, while the semisubmersible accommodates a processing plant and living quarters. The field is connected to pipelines for oil and gas transportation and uses water-alternating-gas injection as pressure support for the reservoir. The production rate has been in decline for several years, and the field is currently producing without pressure support and will continue to do so until it is abandoned. The field decommissioning plan was submitted to the relevant authorities in autumn 2020. One of the field's stakeholders reported that a plug and abandonment campaign was launched for its 24 wells in January 2021 (Equinor, 2021). The report further states that the plan is to tow away the semisubmersible in autumn 2021 and remove the fixed platform in 2025/26.

2 Literature review

Fields that have been brought to production can produce for decades, and the life of mature fields is continually extended through enhanced oil recovery. Moreover, abandonment decisions do not yield immediate returns, unlike areas such as oil recovery enhancement, drilling, and exploration. Consequently, there has been limited research on the topic, resulting in a scarcity of published papers.

In 2004, Begg, Bratvold, and Campbell (2004) conducted a case study on abandonment timing. Having stated that one common method is to abandon fields at first negative cashflow, they later confirmed that there is value in not automatically doing so. They supported their claim by introducing a criterion that the field requires continuous negative cashflow for a predetermined amount of time before it is abandoned. In most scenarios, this results in an increase in NPV, as the discounting factor of the abandonment cost offsets small negative cashflows, and the field has the potential to start generating positive cashflows.

The use of ROV approaches has since refined the methods for determining the time of abandonment. Jafarizadeh and Bratvold (2012) applied the LSM to solve an ROV formulation of the abandonment problem; they concluded that applying ROV to determine the optimal abandonment time yields a greater increase in value compared to Begg et al.'s (2004) approach.

3 Theory

This chapter discusses the theory behind the models, methods, and programs used in this thesis

3.1 Python

The models used in this work were implemented with Python, an open-source programming language. As one of the most popular languages (Jetbrains, 2020), Python offers a wide variety of modules, which can be explained as tools that help facilitate coding, graphical interfacing, connecting to databases, and other functions.

3.2 Net present value

The time-value-of-money concept reflects the fact that money has a higher value today than compared to the same amount in the future. This is because money today can be invested and start to yield returns immediately. In consideration of this difference, future cashflow is discounted back to today's value through the NPV function as follows:

$$NPV = \sum_{t=1}^{n} \frac{Cf_t}{(1+i)^t} \tag{1}$$

Cf_t = Cashflow from period t
i = Discount rate
t = Number of time periods

The discount rate is set by the company and reflects its expected return on capital. This rate varies depending on company and sector. NPD used a rate between 4 to 7% when estimating oil and gas exploration profitability on NCS (NPD, 2020). As our field is located on the NCS was a rate of 7% used.

3.3 Oil price model

In a study of 127 years of oil price data, Pindyck (1999) concludes that the price follows a mean-reverting stochastic process. The first mean-reverting model – and one of the most straightforward to implement – was introduced by Ornstein and Uhlenbeck (1930) and is commonly called the Ornstein-Uhlenbeck (OU) process. The OU model has a lognormal distribution that increases with time while moving closer to its long-term mean (Figure 1). After a specific time, the size and movement of the distribution plateau and become constant.

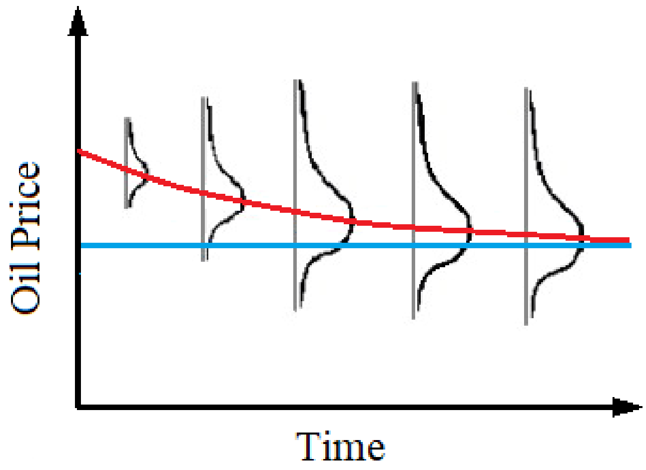

Figure 1. Example of a stochastic mean-reverting process with lognormal probability distribution over time (Dias, 2004). The red line represents the mean oil price and the blue line the long-term mean.

The OU model can be expressed as a differential equation:

$$dX(t) = \theta[\mu - X(t)]dt + \sigma dW(t) \tag{2}$$

$dX(t) = Increment\ in\ oil\ price$
$\theta = Speed$ of $reversion\ towards\ the\ long\ term\ mean$
$\mu = Mean$
$X(t) = Oil\ price$
$dt = Increment\ in\ time$
$\sigma = Volatility$
$dW(t) = Increment\ of\ a\ standard\ brown\ motion$

To prevent the OU model from giving negative values, the logarithmic value of all oil prices can be used (Vega, 2018). The final equation using discrete time steps adopted in our model:

$$ln(X_t) = ln(X_{t-1})e^{-\theta\Delta t} + \left(\mu - \frac{\sigma^2}{2\theta}\right)\left(1 - e^{-\theta\Delta t}\right) + \sigma\sqrt{\frac{1}{2\theta}\left(1 - e^{-2\theta\Delta t}\right)}\ \varepsilon \tag{3}$$

$X = Oil\ price$
$t = Time\ from\ start\ of\ simulation$
$\Delta t = Time\ step, month$
$\mu = Mean$
$\sigma = Volatility$
$\theta = Speed$ of $reversion\ towards\ the\ long\ term\ mean$
$\varepsilon = Random\ normal\ distribution$, N[0,1]

3.4 Monte Carlo simulation

The Monte Carlo simulation is a method for obtaining numerical results by creating a model over the problem and then sampling input values from the appropriate distributions (Figure 2). This process is repeated *n* times until a stable distribution of the output is achieved. This output distribution is then analyzed to determine the range of possible outcomes and their respective likelihood of occurrence.

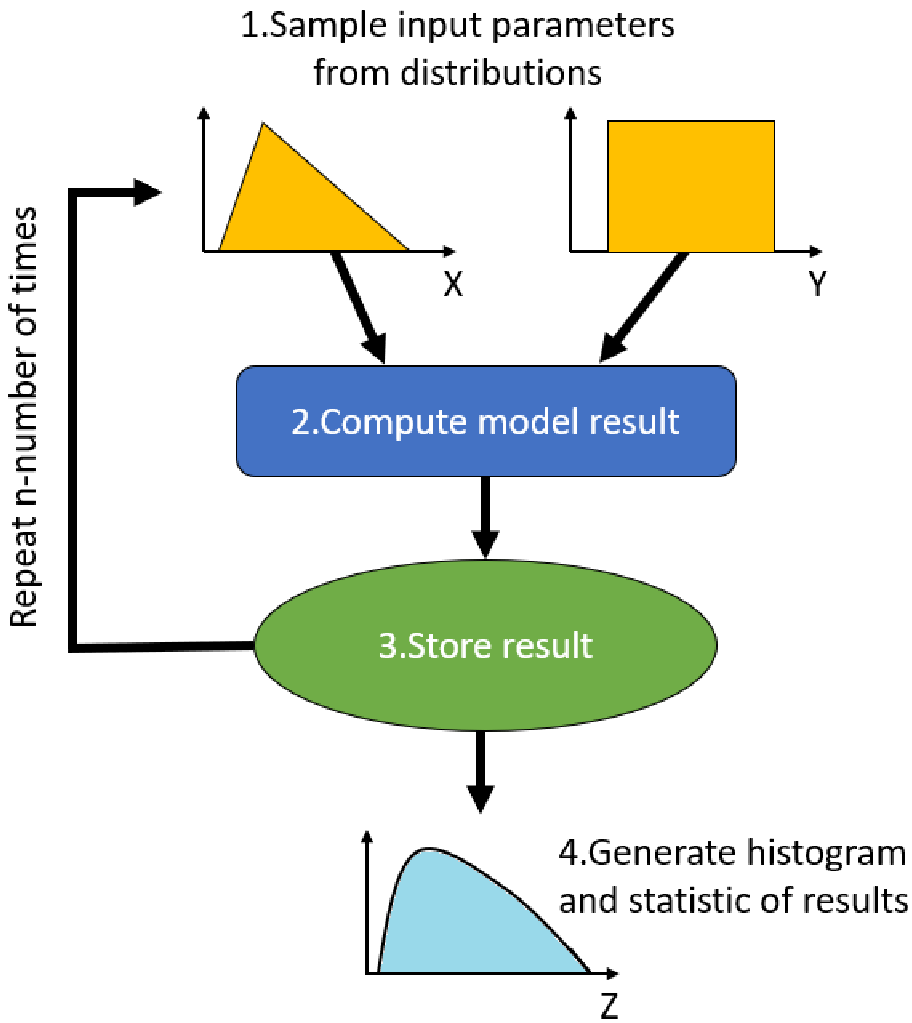

Figure 2. Schematic of Monte Carlo simulation (Bratvold & Begg, 2010)

3.5 Real option valuations

The ROV methods applies option valuation techniques to a business decision. Real option refers to the possession of the right but not the obligation to make decisions regarding real (or non-financial) assets (Gilbert, 2004). This means that the ROV accounts for the stakeholder's flexibility in terms of responding to changes during the project. In this thesis, ROV was applied to implement the stakeholder's option to abandon the field immediately if the economic outlook becomes unfavorable. For example, a field which permanently loses production and starts to generate significant losses due to technical challenges could be abandoned immediately under this approach. Real option valuation can be implemented using various stochastic dynamic programming frameworks, such as binomial lattice and the finite difference method. The drawback of these methods is that they cannot readily accommodate multiple sources of uncertainty as the number of nodes grows exponentially with each time-step. A more appropriate technique in such a case is the LSM, which was therefore adopted to find the optimal time of abandonment in this case study.

3.5.1 Least-squares Monte Carlo

The LSM method was first introduced by Jacques Carrier (1996) to value Bermuda or American options. Multiple papers have since been published using the LSM. One of the best-known papers was written by Longstaff and Schwartz (2001), who improved the method and provided examples for its practical implementation. The technique has also been modified for the oil and gas industry in publications by Smith (2005), Willigers and Bratvold (2008), Jafarizadeh and Bratvold (2012), and Thomas and Bratvold (2015).

4 Methodology and procedure

This chapter first states the assumptions and limitations of the abandonment model in question, followed by an explanation of how the data for the model was obtained.

4.1 Assumptions and limitations

To refine the scope of the thesis and obtain the critical data needed for the simulation, certain limitations and assumptions were introduced into the model. Table 1 summarizes and comments on the main points.

Table 1. Assumptions and limitations with comments

Assumptions and limitations	Comments
The only uncertainties included in the model are oil price and variable operative expenditures.	There are numerous uncertainties such as production profiles, fixed operative expenditures, and abandonment costs.
The model excludes tax and royalties.	These variables need to be considered in real cases.
Inputs to simulate the field were estimated or calculated using empirical evidence.	These variables should be based on actual data that the company has obtained.
Oil and gas production was combined into oil equivalents (OE) and sold for the price of oil.	These should be treated individually, with the respective production profiles and price forecasts.
Inputs and comparisons were taken from both UK and Norwegian oil and gas industries.	Different taxes and rules affect the data, depending on where it was obtained. However, as both nations operate in the North Sea, these amounts were assumed to be equivalent.

4.2 Collecting the data for the model

The model is based on the Veslefrikk field on the NCS, and the data collection process was divided into four parts: calibrating the oil price model, abandonment cost, production profile, and operational cost. Publicly available sources from both UK and Norwegian oil and gas sectors were used to gather or estimate the needed data. Abandonment in the UK sector has been more common than on the NCS, as the former consists of more mature fields. Consequently, a large amount of cost data has been shared that can be applied to the NCS. The Norwegian Petroleum Directorate (NPD) has also generated considerable public information on the NCS, which was used to determine the production profiles and operational costs. The production cost was compared to other fields in the UK sector due to its similarity to the NCS.

4.2.1 Calibrating the Ornstein-Uhlenbeck process

The least-squares method was used to find the parameters for the OU model. The natural logarithm of historical oil prices was regressed against its lag using linear regression. The output of this regression was then entered into equations 5, 6, and 7 to obtain the parameters for the OU model (Berg, 2011).

$$\sigma = Std(\eta_t)\sqrt{-\frac{2ln(a)}{\Delta t(1-a^2)}} \tag{5}$$

$\sigma = Volatility$
$\eta_t = Residual$
$a = Slope\ coefficient$
$\Delta t = Time\ increment$

$$\theta = -\frac{ln(a)}{\Delta t} \tag{6}$$

$\theta = Speed$ of $reversion\ towards\ the\ long\ term\ mean$
$a = Slope\ coefficient$
$\Delta t = Time\ increment$

$$\mu = \frac{b}{(1-a)} \tag{7}$$

$\mu = Mean$
$b = Interception$
$a = Slope\ coefficient$

The oil calibration time window used for the base case in the thesis extends from January 1, 2015 to February 1, 2020. The average price for every 30 days was used for the calibration and several different time windows were taken to a sensitivity analysis. Table 2 presents the length and parameters of the calibration.

Table 2. Oil calibration window and its parameters for the UB model

Calibrated to Feb 1, 2020			
Mean (μ)	Speed (θ)	Volatility (σ)	Calibrated from
53.99	1.69	0.34	Jan 1, 2015
54.75	1.38	0.32	Jun 28, 2015
57.42	2.37	0.30	Dec 23, 2015
58.34	2.73	0.30	Jun 18, 2016
58.51	2.73	0.32	Dec 13, 2016
60.41	3.85	0.33	Jun 9, 2017

Figure 3 illustrates an example of the base-case calibration window and the range of future fluctuation using the UB model, as well as a single realization.

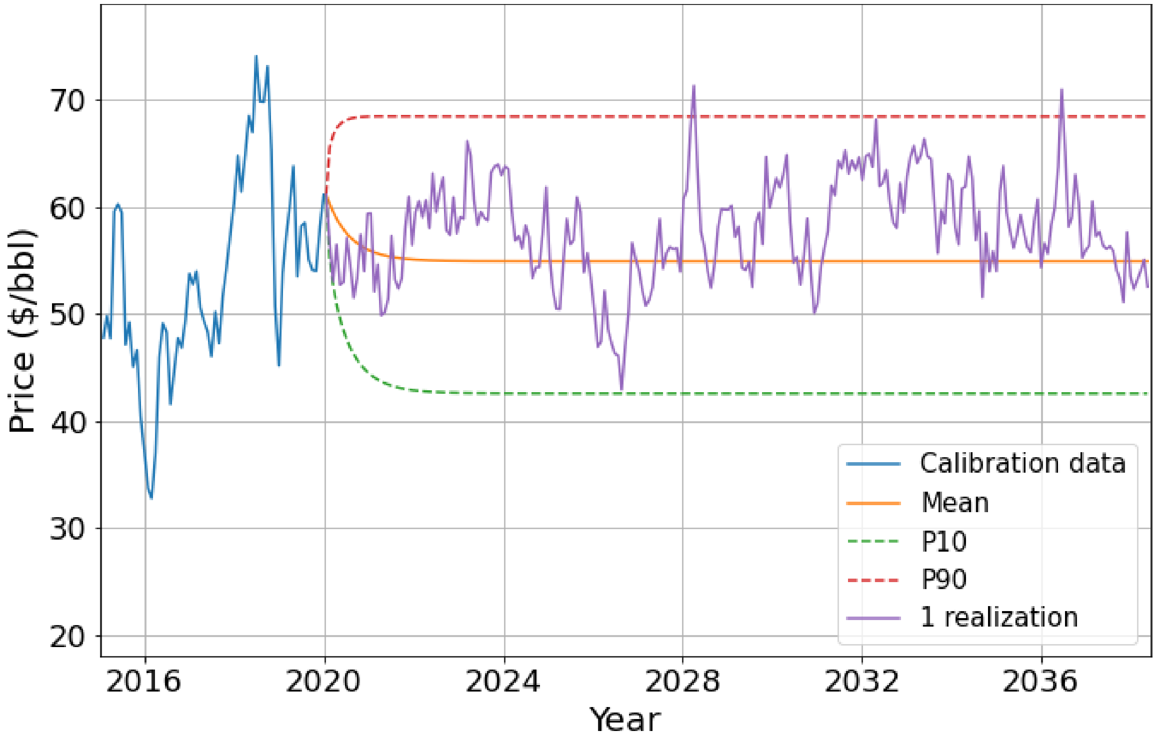

Figure 3. Historical and forecasted oil price

4.2.2 Estimating the abandonment cost

The UK oil and gas authority decommissioning survey reports and tracks the decommissioning costs to the country. It reports a median well decommissioning cost of £3.7 MM in the Northern and Central North Sea from a fixed platform (Oil & Gas Authority, 2020), which is equivalent to $4.76 MM using the 2020 average exchange rate of the Bank of England (Bank of England, 2020).

Veslefrikk has to plug 24 wells for the abandonment process (Equinor, 2021), resulting in a cost of $114.30 MM using the cost per well noted in the previous paragraph. A breakdown of the forecasted decommissioning expenditures in the Northern North Sea and West of Shetland reveals that 46% of the total costs derive from well decommissioning (Figure 4).

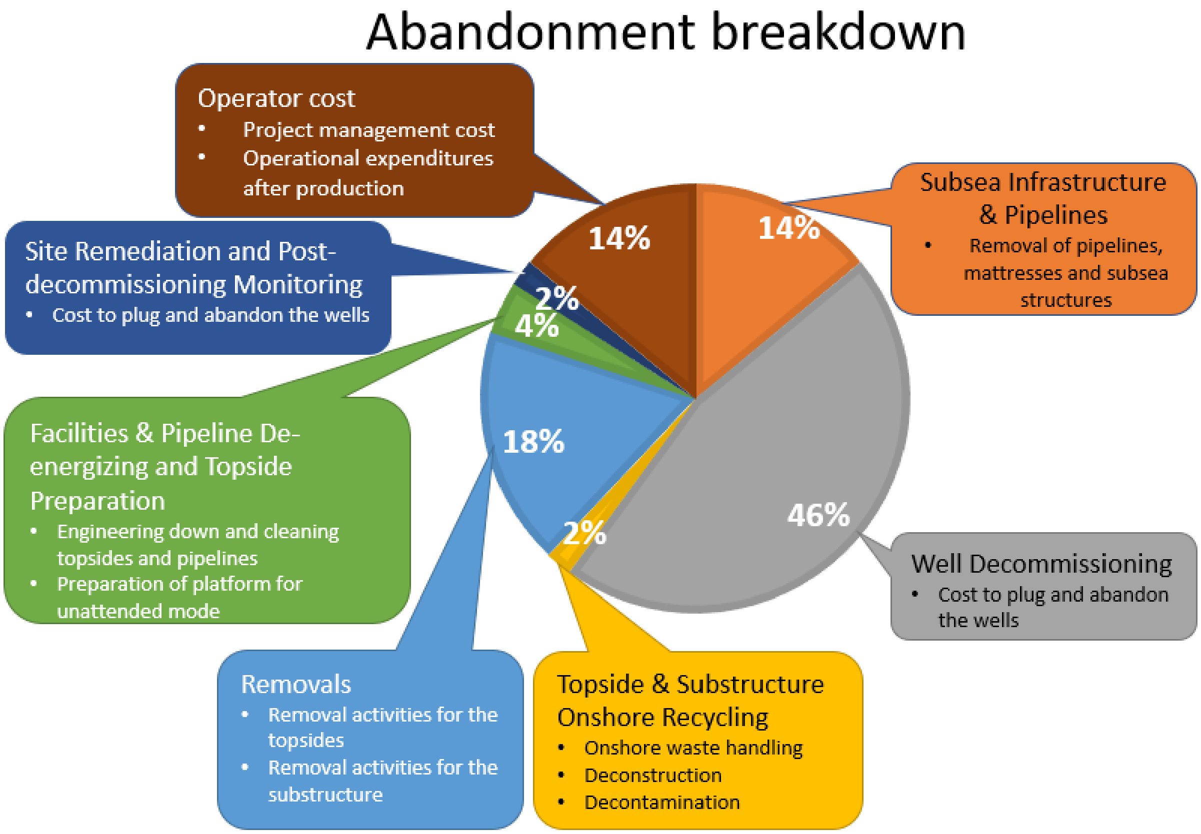

Figure 4.Breakdown of the abandonment costs in the Northern North Sea and West of Shetland (Oil & Gas UK, 2018)

This implies a total decommissioning cost of $233.27 MM for Veslefrikk. Table 3 presents all the values used to calculate the abandonment cost.

Table 3. Values used to calculate the abandonment cost

Wells	24 (pc)
Exchange rate	1.2837 ($/£)
Well decommissioning cost	3.71 (£MM)
	4.76 ($MM)
Well decommissioning fraction of the total cost	0.46
Total well decommissioning cost	114.30 ($MM)
Total field abandonment cost	233.27 ($MM)

4.2.3 Modeling the production decline

The production decline is modeled on data from Veslefrikk (The NPD, 2021); gas and oil are combined into OEs. A second-degree polynomial is fitted on the decline curve to estimate a function representing future production. Figure 5 presents the historical production and forecast data, along with the function for the curve.

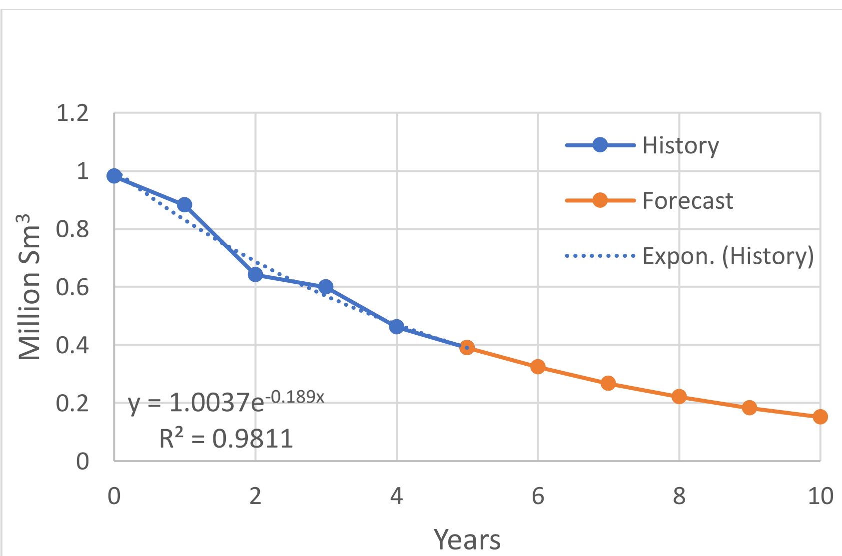

Figure 5. Historical and forecasted production of OEs, function for the forecast, and the R-squared fit

The method used to determine the function for production was also applied to the produced water (Figure 6).

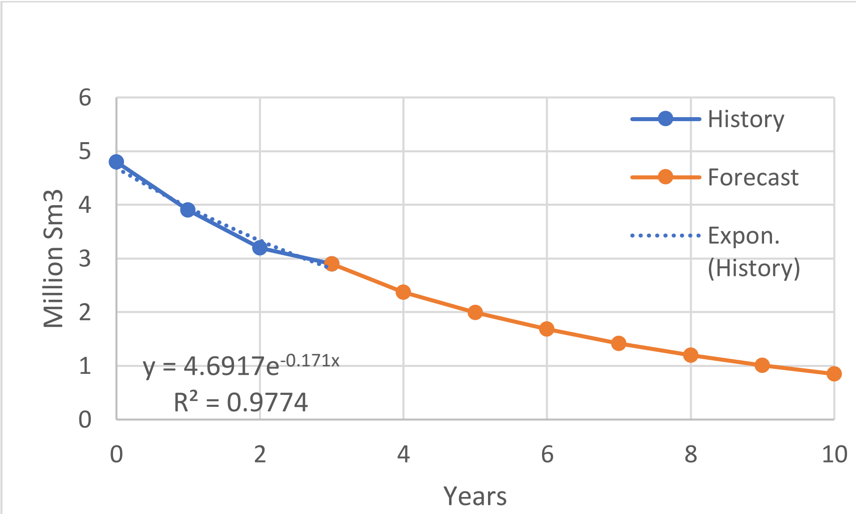

Figure 6. Historical and forecasted production of water, function for the forecast, and the R-squared fit

Ideally, more data points for the declining water and oil levels should have been obtained to estimate a suitable decline for this field. However, since the focus of this study is on comparing different abandonment approaches, the above is an adequate estimation for this purpose.

4.2.4 Operating expenses

The operating expenses (Opex) were divided into two categories: variable Opex, which is applied for each barrel (bbl) produced, and fixed Opex, which is a monthly cost. To obtain the variable Opex, the yearly operating expenses from 2008 to 2020 (Norskpetroleum, 2020) were divided by yearly OEs produced (Norskpetroleum, 2021) on the NCS. This yields the average cost of producing one OE on the NCS, which is treated as the past variable Opex from the field. The data and conversion are presented in Appendix 1, from where the exchange rate is taken (Norges Bank, 2021). To replicate future fluctuations in variable Opex, the OU model was calibrated to the past variable Opex. The calibration was performed in the same manner as the oil price in Section 4.2.1, and the final parameters are presented in Table 4.

Table 4. Final parameters used to simulate variable Opex with the OU model

Mean (μ)	Speed (θ)	Volatility (σ)
2.01	0.1	0.189

According to Bradley and Wood (1993), oil price is one factor that affects the long-term Opex of an oilfield. To determine any correlations on the NCS, the historical Brent price (EIA, 2021) and variable Opex were plotted against each other, as illustrated in Figure 7. Appendix 1 contains the data used for this plot.

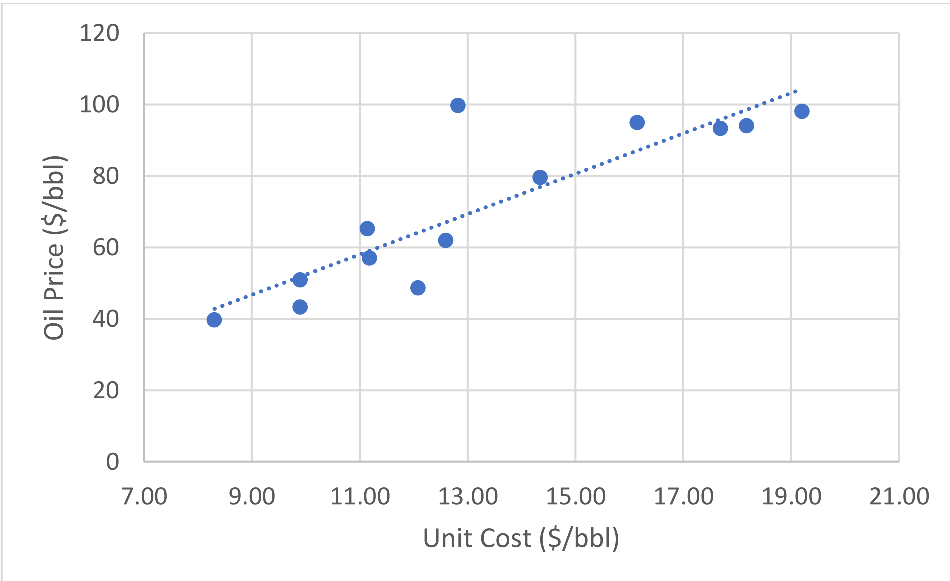

Figure 7. Average annual oil prices against average annual unit costs with a trend line

The plot indicates a linear relationship between the oil price and the variable Opex. The Pearson's correlation coefficient for this linear correlation was then used to determine the correlation factor:

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X * \sigma_Y} \tag{8}$$

$\rho_{X,Y} = Correlation\ factor$
$\sigma_X = Standard\ deviation\ of\ oil\ price$
$\sigma_Y = Standard\ deviation\ of\ unit\ cost$
$cov(X,Y) = Covariance\ of\ oil\ price\ and\ unit\ cost$

The correlation was determined to be 0.896, which suggests that the oil price and Opex are highly correlated. To replicate this finding, the Python module Scipy.stat.multivariate_normal was used to create a pair of correlated random numbers with a factor of 0.896 sampled from a $standard\ normal\ distribution$, $N[0,1]$ (ε). The pair was divided for the two OU models that simulate oil price and variable Opex, respectively. The OU models used the individual parameters obtained earlier, along with the correlated ε. In this manner, the amplitude of change would differ for the two models, but the direction of change would be almost identical. To prevent the variable Opex from changing instantly with the oil price, a lag time of 12 months was included. Figure 8 provides an example of the simulated price and correlated variable Opex.

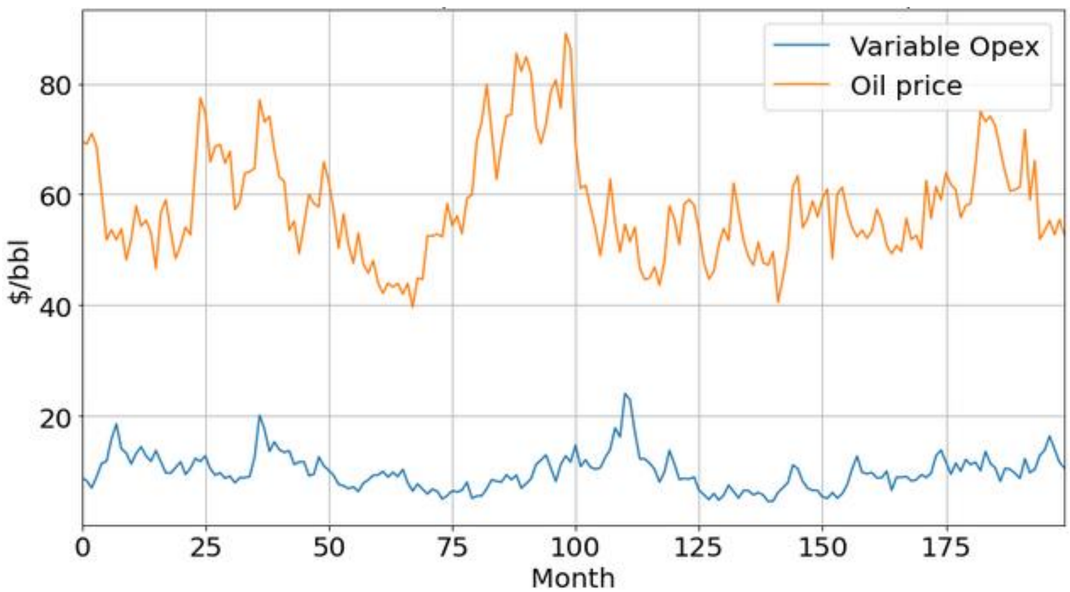

Figure 8. Example of a simulated oil price with correlated variable Opex

Two expenditures not affected by the oil price was added, water treatment cost of $ 0.5/bbl and a fixed yearly operative cost of $42 MM. A comparison was made to determine if the expenses were in proportion to other fields in the UK sector in 2017. With the estimated costs from 2017, the historical production data from Veslefrikk was used to calculate the total cost and unit operating cost (UOC) (Table 5). The exchange rate used to convert dollars ($) into pounds (£) was taken from (Bank of England, 2020).

Table 5. Parameters used to calculate total cost and unit operating cost for 2017

OE production (bbl)	Variable Opex ($/bbl)	Oil production cost ($MM)	Fixed Opex ($MM)	Exchange rate ($/£)
3 957 599	9.89	39.15	42.00	1.29
Water production (bbl)	**Water cost ($/bbl)**	**Water production cost ($MM)**	**Total cost (£MM)**	**Total UOC (£/bbl)**
30 191 088	0.50	15.10	74.68	18.87

A comparison of Veslefrikk with fields in the same age group indicates that it is among the more expensive of such fields (see Figure 9), although not exceptionally so.

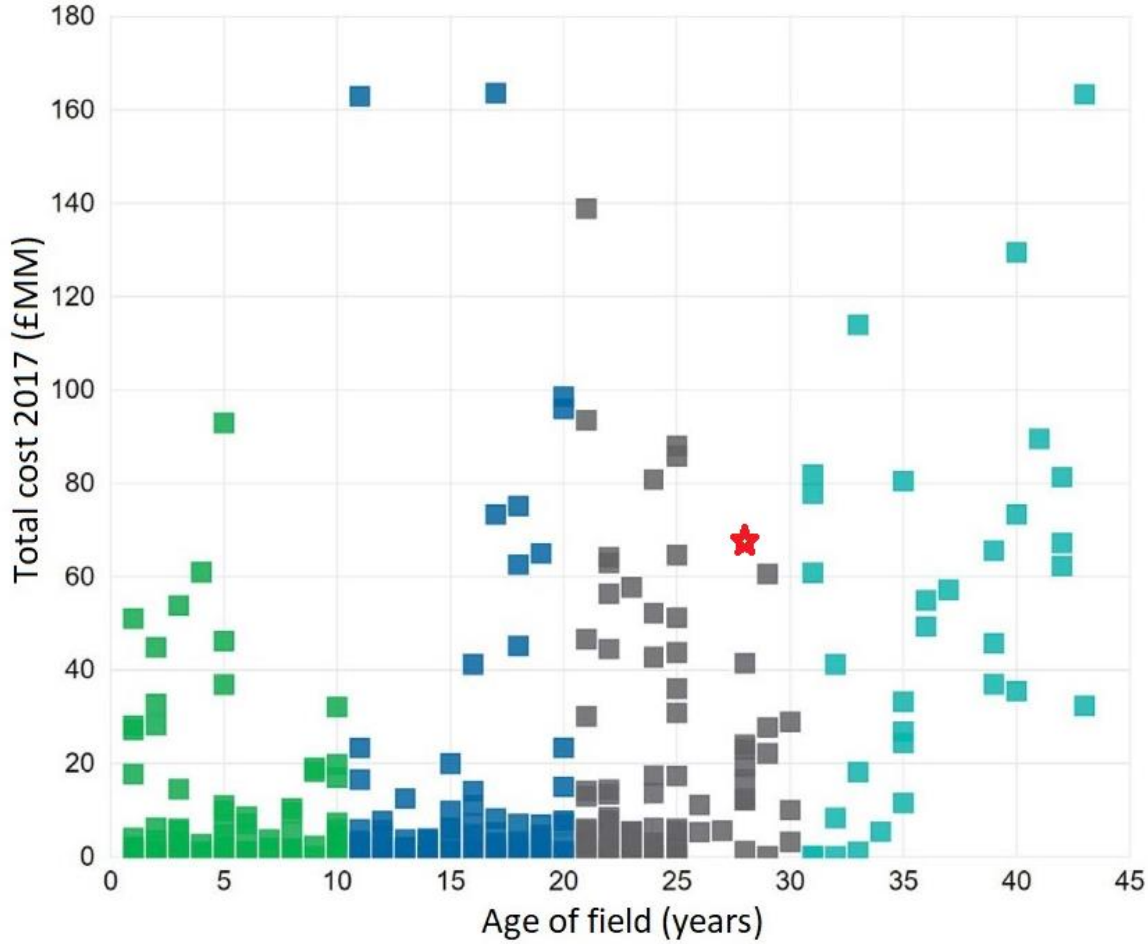

Figure 9. Age of field versus its total cost on the UK continental shelf for 2017. The red star indicates Vesslefrikk's position with our estimated costs and 2017 production (Oil & Gas Authority, 2017)

A comparison of the fields total production and cost indicate that it has higher UOC compared to the average field producing over 2 MM bbl annually (Figure 10).

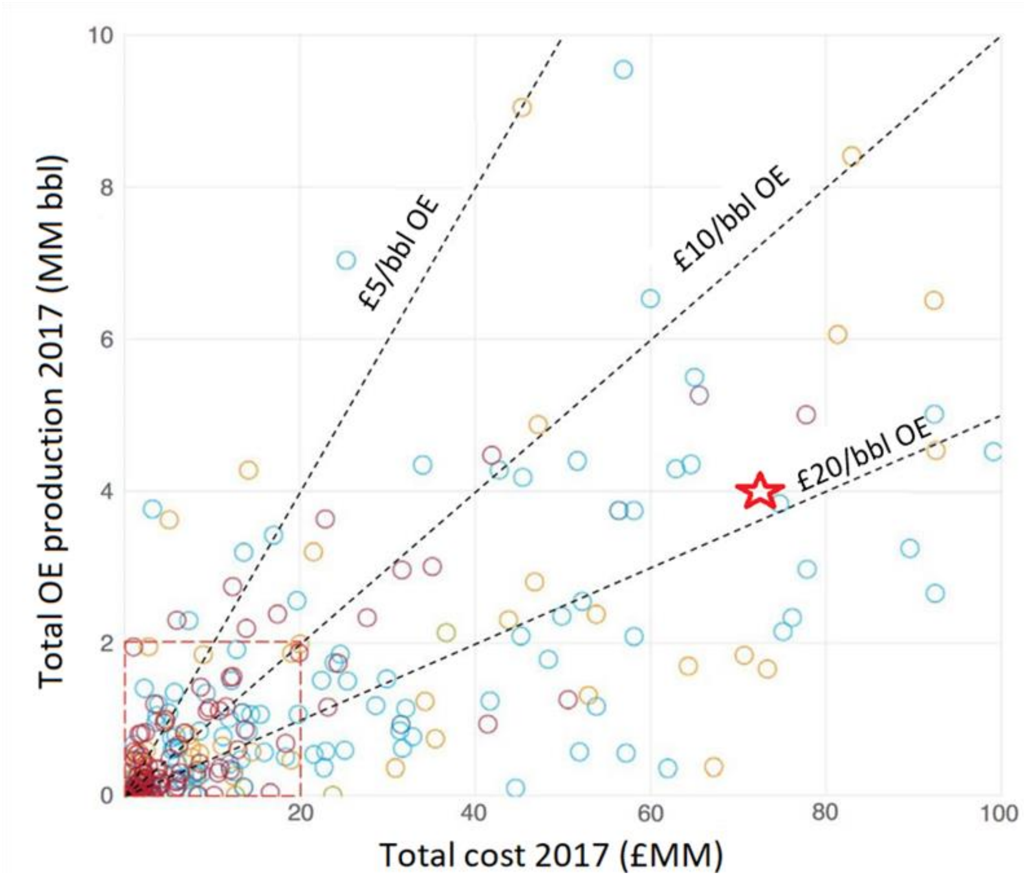

Figure 10. Unit operating cost for each field on the UK continental shelf. Each circle represents a field, and the red star denotes Veslefrikk (Oil & Gas Authority, 2017)

5 Approaches to determining the optimal time of abandonment

This chapter explains each approach used to determine the time of abandonment and how they were implemented into the abandonment model.

5.1 First negative cashflow approach

The FNC abandons the fields one month after the occurrence of the first negative cashflow. This is the simplest approach to implement and understand.

5.2 Greedy optimization approach

The GO introduces a predetermined waiting criterion for which the field needs to generate continuous negative cashflow. If the field starts to generate positive cashflow, the continuity is broken, and the period starts over again.

5.2.1 Finding the optimal waiting criterion

The GO waiting criterion is required to determine the optimal time of abandonment. It is obtained by running the GO approach with different waiting times until the highest NPV is achieved, as illustrated in Figure 11, where the average NPV peaks at a waiting period of 22 months. The 22-month criterion was therefore used in the model. Since the curve is quite flat between 16 and 30 months, using any value within this range would result in very similar end results.

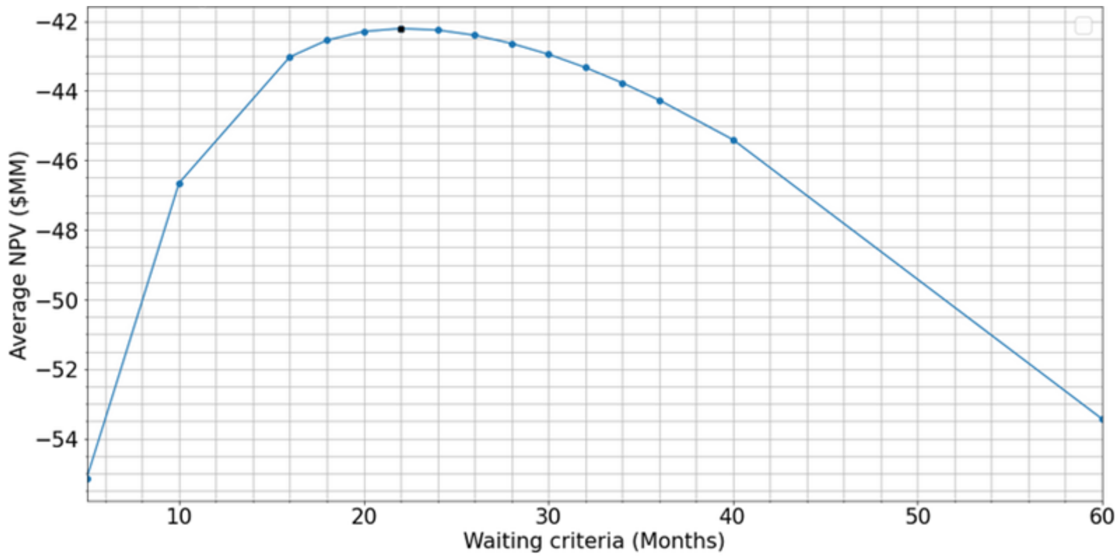

Figure 11. Average NPV under GO using different critical waiting criteria

5.3 Least-squares Monte Carlo approach

The LSM approach uses backward induction to identify the optimal abandonment time, which means that it starts from the last decision point in time, then walks backwards for every decision made until it reaches the start of the simulation. There are two alternatives for every decision point: continue to produce or abandon. The LSM approach makes the decision by approximating the NPV using continued production (conditional continuation value) and comparing this value to the abandonment cost (exercise value) for each point in time. When the continuation value is less than the exercise value will the field be abandoned. Repeating this procedure for all the decision points, the LSM will make a near-optimal decision for each path. Figure 12 is a visualization of the LSM decisions at each decision step for five paths, while Figure 13 illustrates the final decision for the same five paths. The continuation value is only used for the decision. Subsections

5.3.1 ("Continuation value") and 5.3.2 ("Obtaining the continuation value") provide a more detailed explanation of the continuation value and how it was obtained.

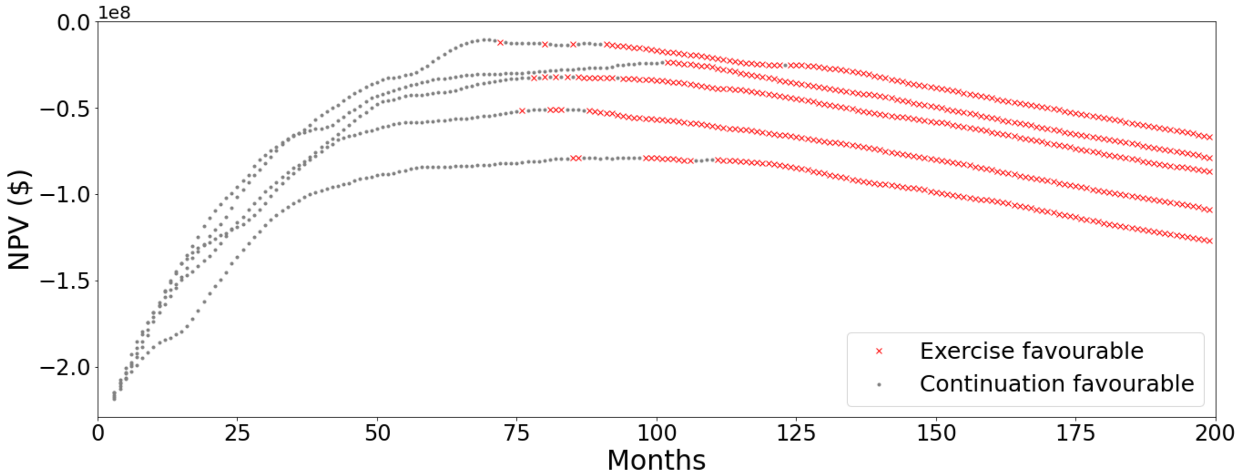

Figure 12. Five NPV paths for a field with LSM continuation or exercise decision for each time step

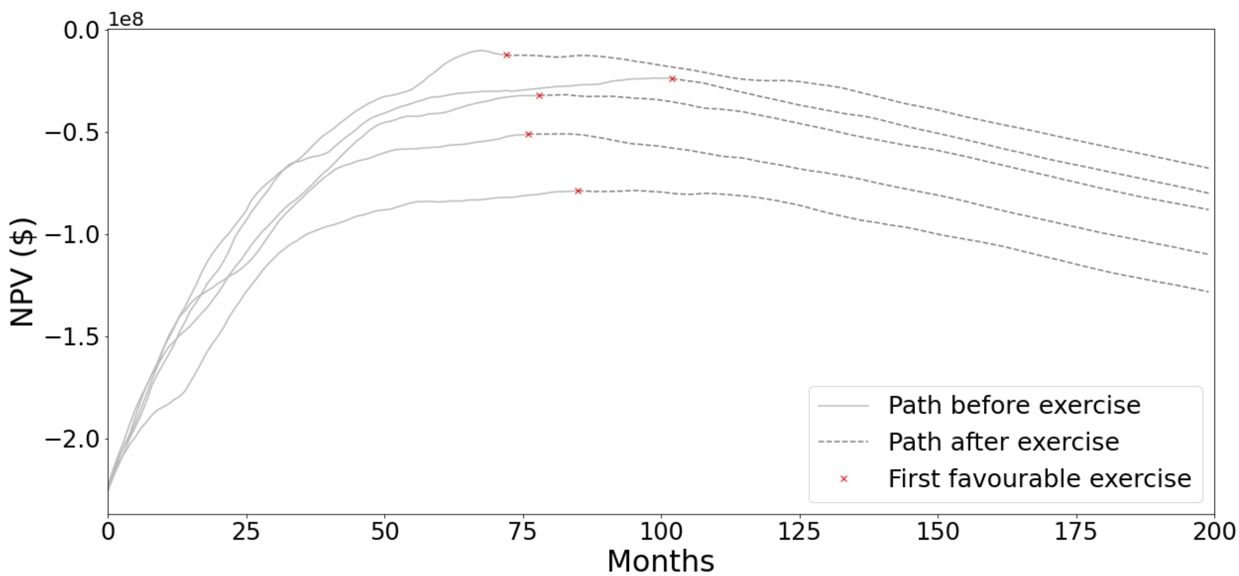

Figure 13. Five NPV paths for a field with LSM final decision to abandon

5.3.1 Continuation value

The continuation value is the approximate NPV that the field would obtain by continuing production from the decision point to abandonment. It is only used to decide when to abandon and not to calculate the resulting value of abandoning at that time. The recursive process starts with the calculation of the abandonment value at the last time step and gradually moves towards the first time period so as to identify the time at which abandonment first became favorable. In this manner, future price and cost paths – and hence the NPV – are conditioned on the current (at any time) price and cost.

5.3.2 Obtaining the continuation value

The continuation value was regressed on the NPV to fit our purpose. This was accomplished using multiple linear regression, whereby the oil price and variable Opex are the independent variables (Eq. 9) and NPV the dependent variables (Eq. 10).

$$X = \begin{bmatrix} OP_t^1 & VO_t^1 \\ OP_t^2 & VO_t^2 \\ \vdots & \vdots \\ OP_t^{i-1} & VO_t^{i-1} \\ OP_t^i & VOP_t^i \end{bmatrix} \tag{9}$$

$i = Path$
$t = Time\ step$
$OP = Oil\ price$
$VO = Variable\ Opex$
$X = Independent\ variables$

$$Y = \begin{bmatrix} ENPV_{t+1}^1 \\ ENPV_{t+1}^2 \\ \vdots \\ ENPV_{t+1}^{i-1} \\ ENPV_{t+1}^i \end{bmatrix} \tag{10}$$

$i = Path$
$t = Time\ step$
$ENPV = Remaining\ NPV$
$Y = Dependent\ variables$

The oil price and variable Opex were chosen as the independent variable as they are uncertain. If there are any additional uncertain variables can they easily be added to Eq. 9. Longstaff and Schwartz (2001) suggest using linear regression to find the continuation value, which is the preferred method in this thesis. More advanced regression models combined with cross terms can be used to find the continuation value but were deemed unnecessary for this task. This was determined by visually inspecting plots like figure 13 containing more paths where the continuation value was based on linear regression. The figure indicated that the preferred method made the LSM abandon the fields close enough to their individual NPV peak. The multiple linear regression function was solved using the Scikit-learn module in Python, which integrates a range of state-of-the-art machine learning algorithms for mathematical problems (Pedregosa et al., 2011). The coefficients for the function were obtained through Eq. 11 and the constant through Eq. 12 as follows:

$$\beta_1, \beta_2 = Sklear.linear\_model.LinearRegression.fit(X,Y).coef\_ \tag{11}$$

$\beta_1 = Coefficient\ for\ oilprice$
$\beta_2 = Coefficient\ for\ variable\ Opex$
$Y = Dependent\ variables$
$X = Independent\ variables$

$$\beta_0 = Sklear.linear\_model.LinearRegression.fit(X,Y).intercept\_ \tag{12}$$

$\beta_0 = Constant$
$Y = Dependent\ variables$
$X = Independent\ variables$

The final function to determine the continuation value is shown below.

$$\begin{bmatrix} ANPV_{t+1}^{1} \\ ANPV_{t+1}^{2} \\ \vdots \\ ANPV_{t+1}^{i-1} \\ ANPV_{t+1}^{i} \end{bmatrix} = \beta_0 + \beta_1 * \begin{bmatrix} OP_t^1 \\ OP_t^2 \\ \vdots \\ OP_t^{i-1} \\ OP_t^i \end{bmatrix} + \beta_2 * \begin{bmatrix} VO_t^1 \\ VO_t^2 \\ \vdots \\ VO_t^{i-1} \\ VOP_t^i \end{bmatrix} \tag{13}$$

$i = Path$
$t = Time\ step$
$\beta_0 = Constant$
$\beta_1 = Constant\ for\ oil\ price$
$\beta_2 = Constant\ for\ Opex$
$OP = Oil\ price$
$VO = Variable\ Opex$
$ANPV = Continuation\ value$

Figure 14 provides an example of the approximation with five paths.

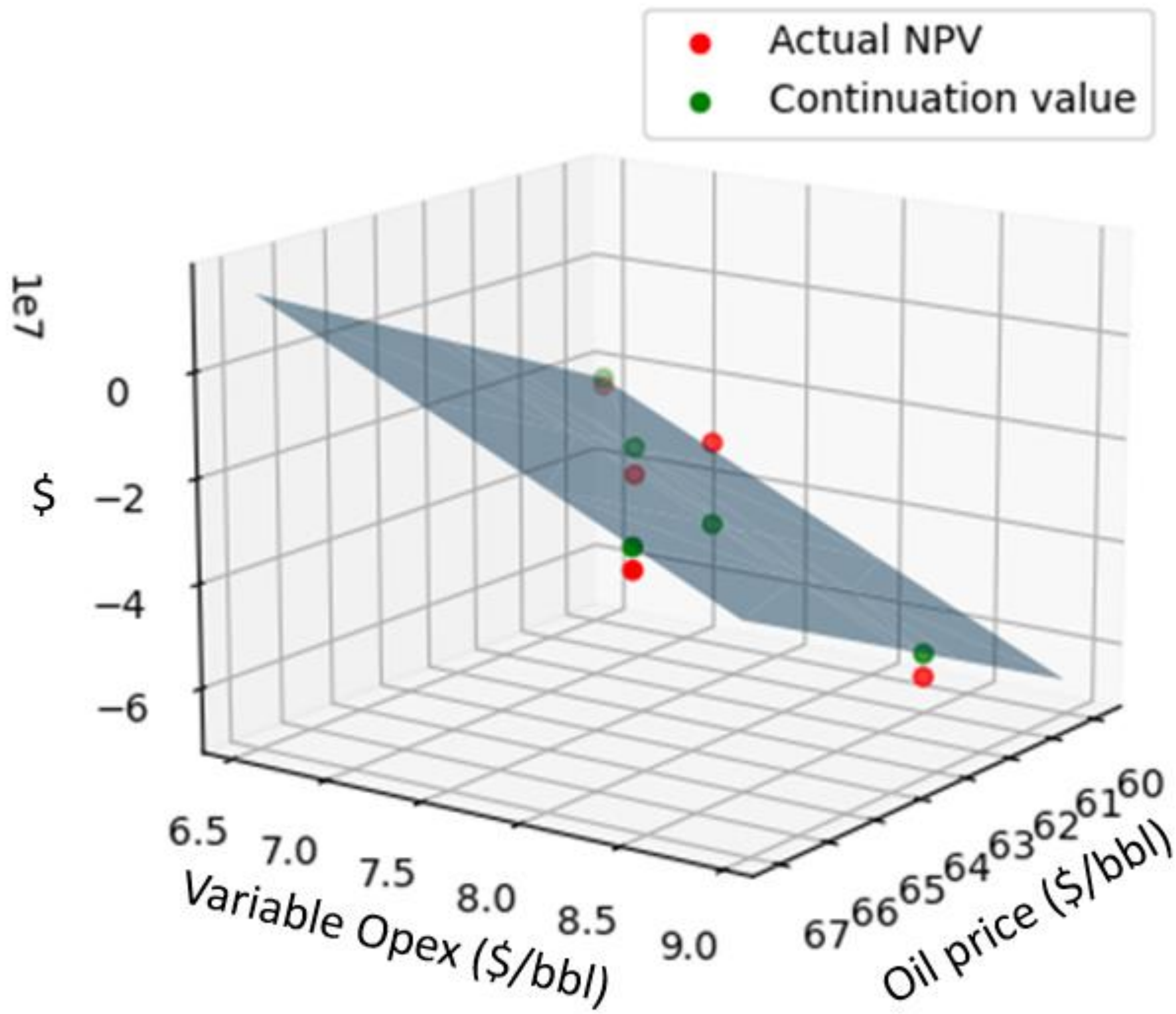

Figure 14. The regressed NPV is represented by red dots, the continuation function by the gray surface, and the continuation value by the green dots

6 Case simulation, results, and analysis

For the abandonment model, we decided to use 180 000 paths, as the output was fairly stable between runs and it did not take too long to process. The model started to create random oil price scenarios with correlated variable Opex, which are the uncertain inputs in the Monte Carlo model. Along with the estimated and assumed parameters from Chapter 4, it generated paths with different cashflows, which represent the cashflows that the field would generate if it was never abandoned. Three approaches were then used to determine the optimal time to abandon the field for each path. All three modified the paths in an identical manner when they found the optimal abandonment time, namely by adding the abandonment cost and removing future cashflow. The NPV was then calculated by discounting back all values to the beginning of the simulation. The resulting NPVs from each approach were then compared and evaluated. The results of this method are presented in the order from the lowest- to the highest-yielding NPV. Appendix 2 presents the Python code for this simulation.

6.1 Results of the first negative cashflow approach

Using the FNC approach, the distribution over the abandonment time ranges from 3 to 80 months, with a mean of 38 months, as illustrated in Figure 15.

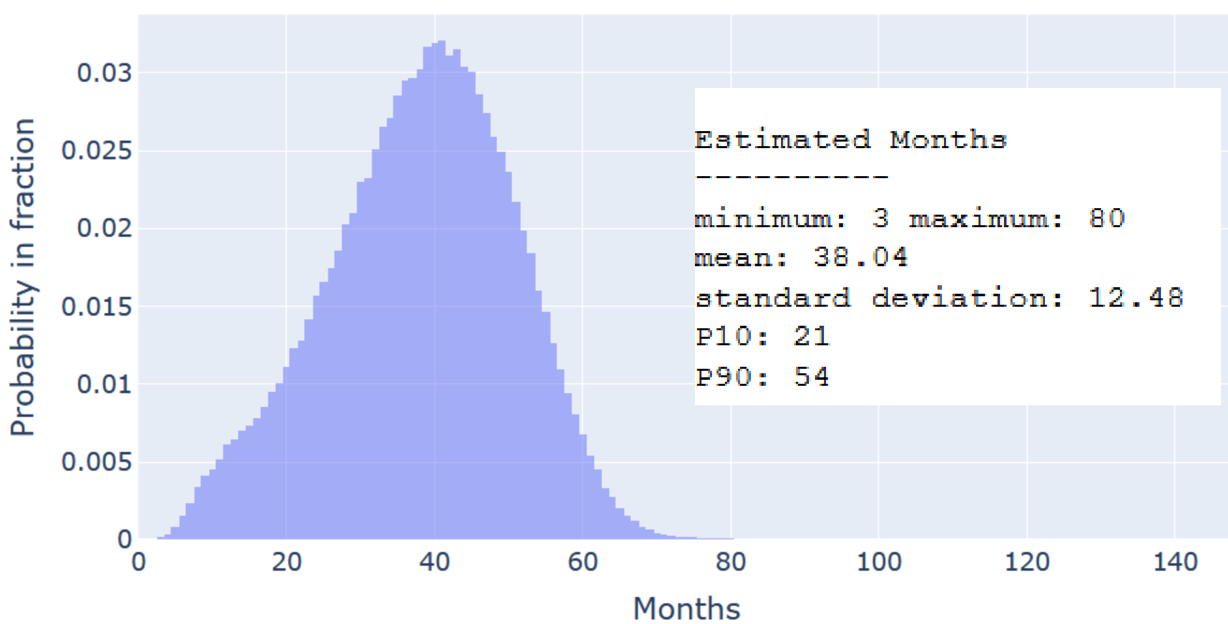

Figure 15. Distribution of abandonment time using the first negative cash flow approach

This strategy would result in an average NPV of -$78.90 MM (Table 6), with the NPV distribution presented in Figure 16.

Table 6. Results obtained using the negative cashflow approach

	NCA (NPV)
Min ($MM)	-228.37
P10 NPV ($MM)	-133.82
Mean NPV ($MM)	-78.90
P90 NPV ($MM)	-30.07
Max ($MM)	74.90
P(NPV<0)	0.986

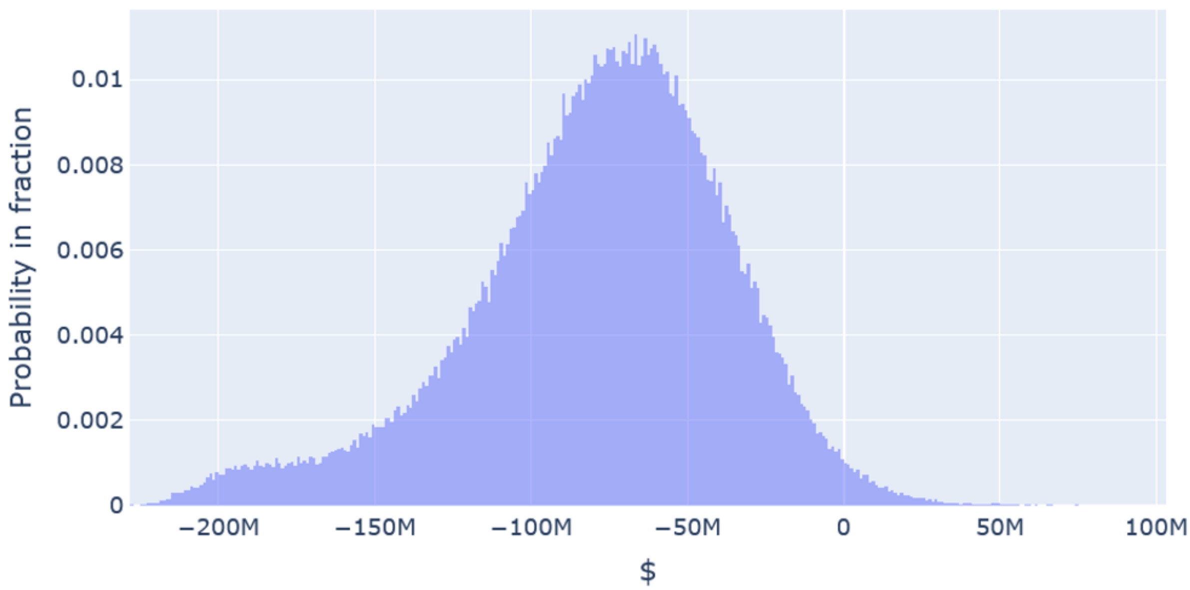

Figure 16. Net present value distribution from using the first negative cashflow approach

Figure 17 presents a scatter plot of the time of abandonment versus the corresponding NPV. The shape of the plot demonstrates that the longer the field can produce before a negative cashflow occurs, the higher its NPV becomes.

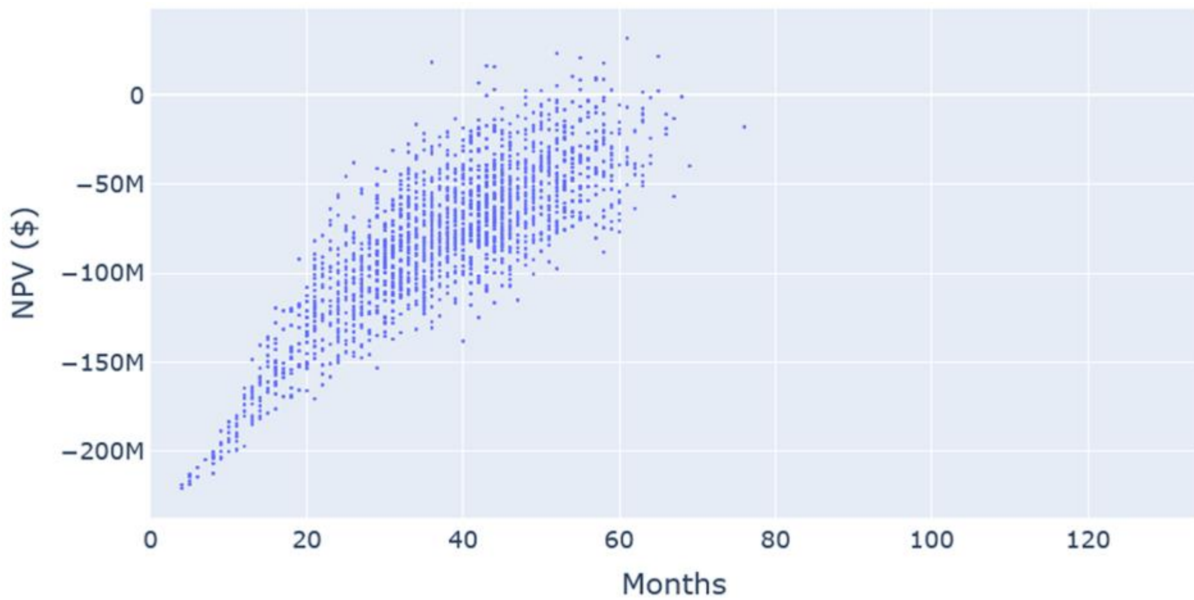

Figure 17. Time of abandonment versus net present value. This plot presents a random 1% selection of the 180,000 simulated paths obtained using the first negative cash flow approach

6.2 Results of the greedy optimization approach

As stated in Section 5.2.1, "Finding the optimal waiting criterion," a criterion value of 22 months was used for the GO approach. On average, this method abandoning the field at 87 months with a range between 36 to 129 months (see Figure 18), significantly later than the FNC average of 38 months.

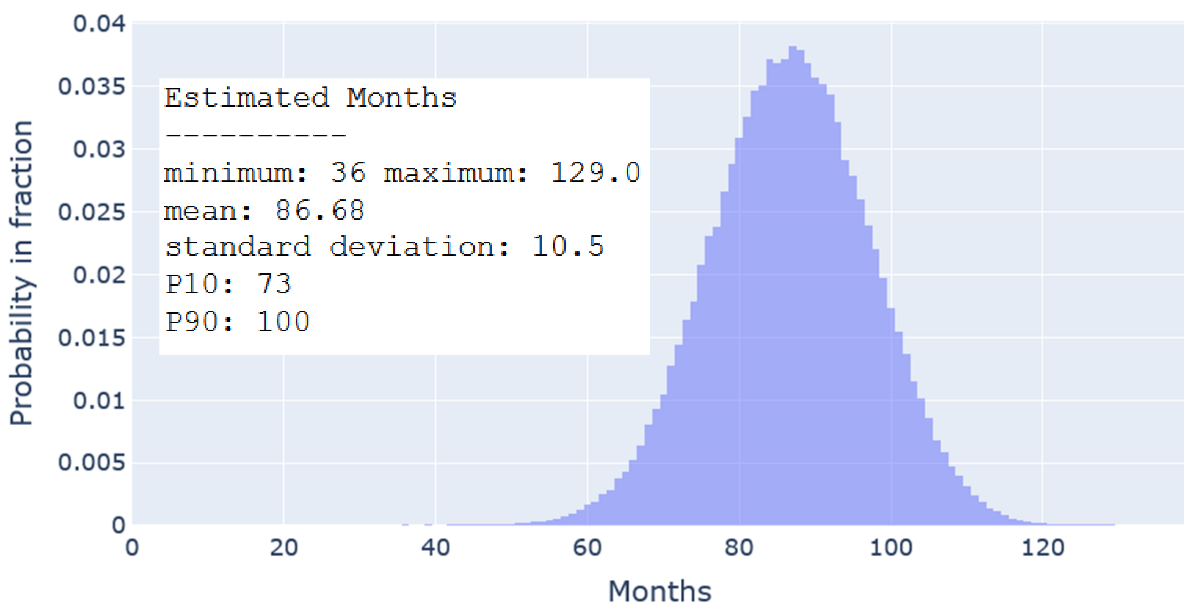

Figure 18. Abandonment time distribution obtained using greedy optimization

As Table 7 illustrates, this method results in a mean NPV of -$42.21 MM. The total NPV distribution is presented in Figure 19.

Table 7. Results of the greedy optimization

	GO (NPV)
Min ($MM)	-162.51
P10 NPV ($MM)	-75.35
Mean NPV ($MM)	-42.21
P90 NPV ($MM)	-8.30
Max ($MM)	102.99
P(NPV<0)	0.942

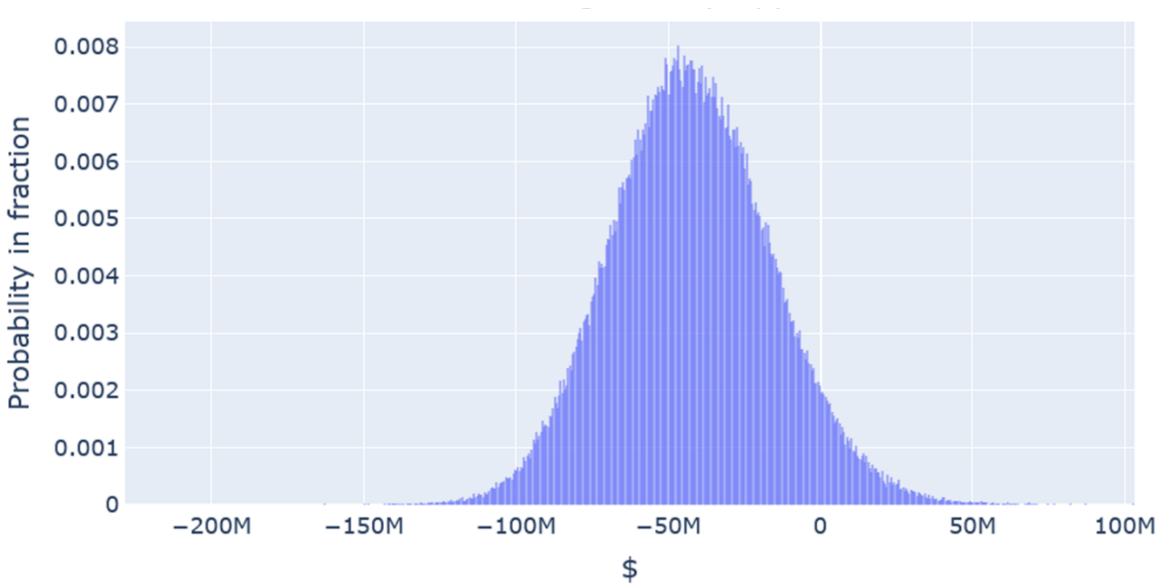

Figure 19. Net present value distribution from the greedy optimization

The GO scatter plot in Figure 20 reveals that the abandonment occurs much later and yields a higher NPV than the NCA approach.

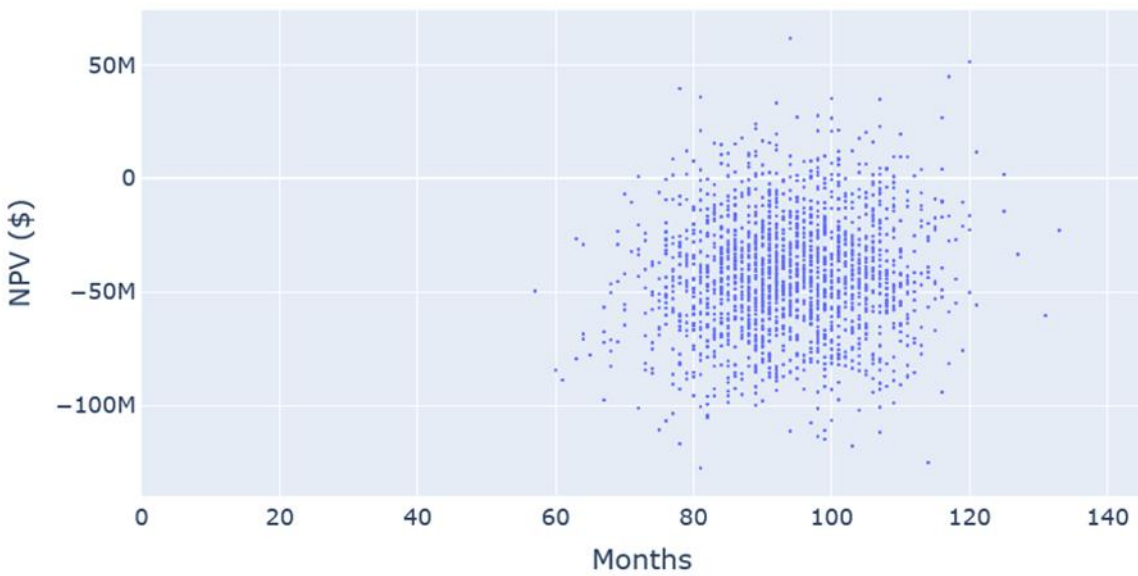

Figure 20. Abandonment time versus net present value. This diagram presents a random 1% selection of the 180,000 simulated paths for the greedy optimization

The GO approach resulted in an average increase in NPV of $36.69 MM compared with the FNC approach. This can be explained by its waiting criterion, which increases the NPV in two ways. First, it offers a chance for the field to start generating positive cashflow after the first negative cash flow. As Figure 21 illustrates, less than 8% of the paths that become cashflow-negative never turn positive again, i.e., 92% of the paths yield positive cash flows after the first negative cash flow. Second, the NPV would increase as long as the discounting factor for the abandonment cost exceeds the negative cash flow.

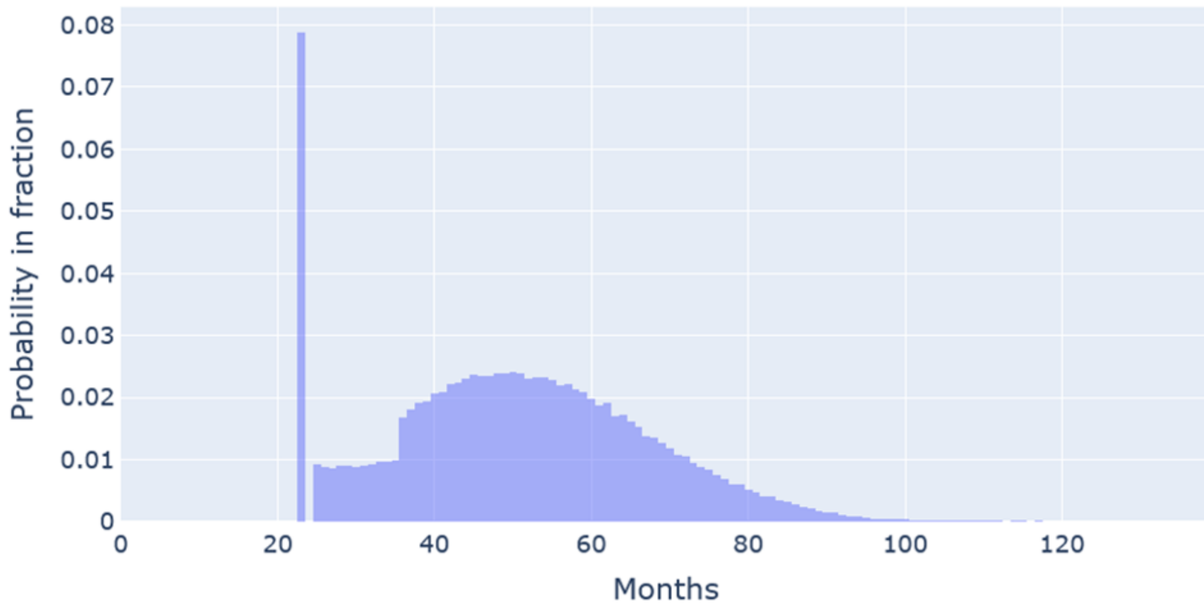

Figure 21. Time (in months) between first negative cashflow and abandonment according to the greedy optimization

6.3 Results of least-squares Monte Carlo approach

Finally, the ROV method using LSM, on average, abandons the field after 84 months with a range between 42 to 111 months (see Figure 22). Compared to the Go approach is it slightly earlier and the distribution of when the abandonment occurs gets more concentrated.

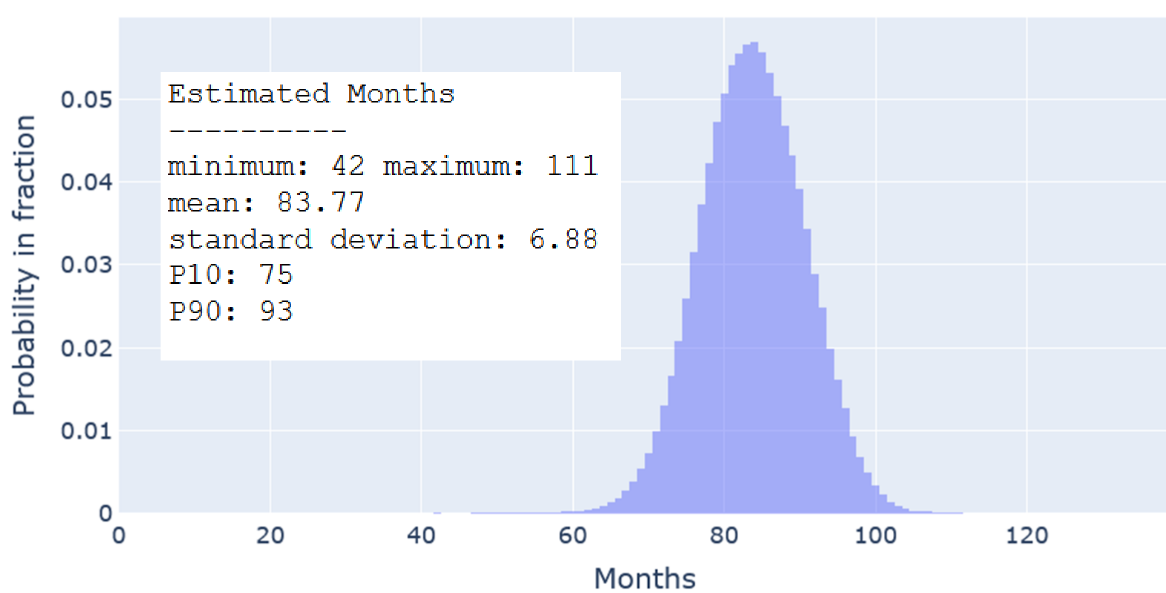

Figure 22. Abandonment time distribution using the least-squares Monte Carlo method

As Table 8 indicates, the LSM results in a mean NPV of -$40.21 MM, which is $ 2.0 MM more than the GO and $38.69 MM better than the FNC approach. This is because the ROV method makes flexible abandonment decisions that respond to uncertainty and learnings gained over time. Figure 23 presents the total NPV distribution obtained using the LSM approach.

Table 8. Results of the least-squares Monte Carlo method

	LSM (NPV)
Min ($MM)	-148.74
P10 NPV ($MM)	-73.23
Mean NPV ($MM)	-40.21
P90 NPV ($MM)	-6.39
Max ($MM)	103.15
P(NPV<0)	0.935

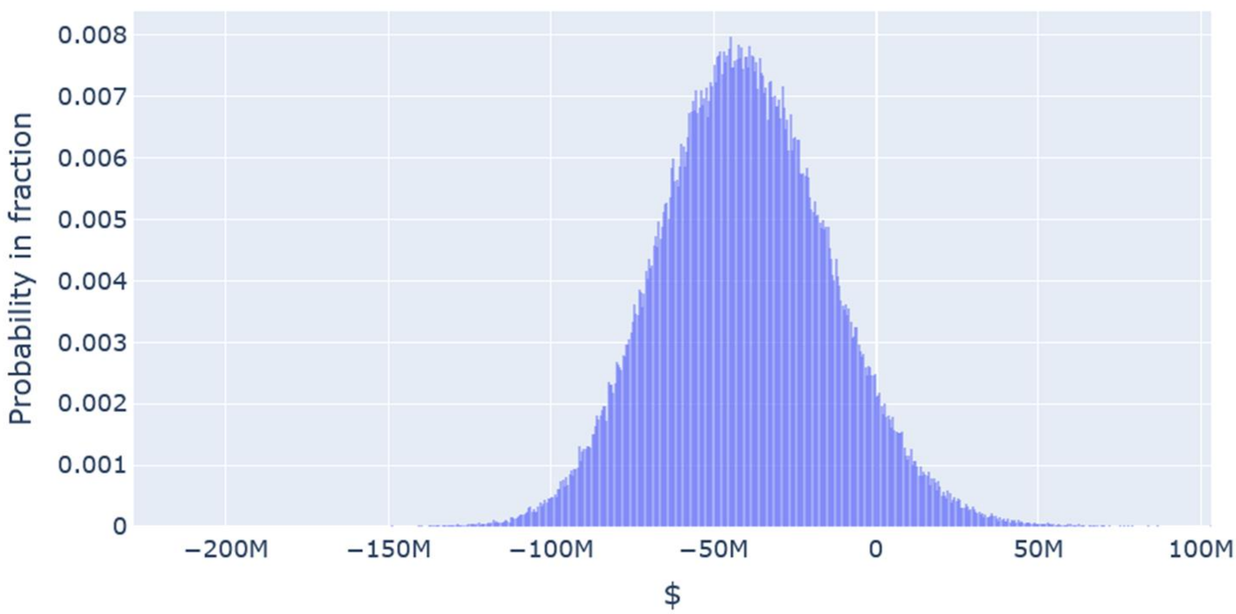

Figure 23. Net present value distribution obtained using the least-squares Monte Carlo approach

Figure 24 presents the LSM scatter plot for NPV against time of abandonment.

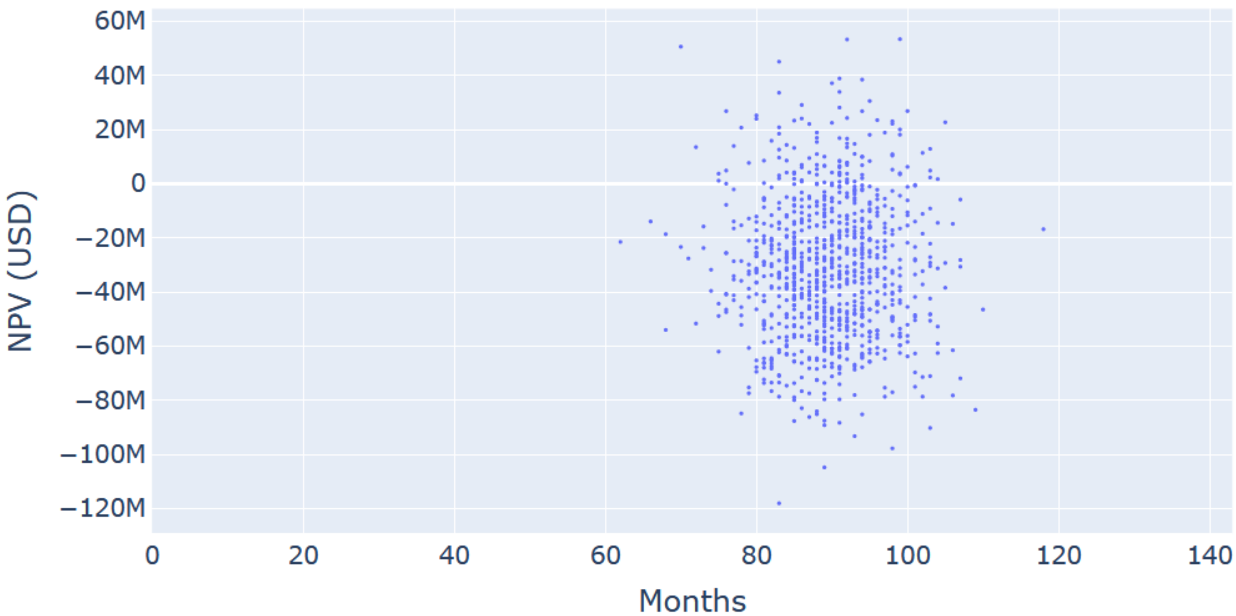

Figure 24. Time of abandonment versus net present value. This diagram presents a random 1% selection of the 180,000 simulated paths for the least-squares Monte Carlo approach

6.4 Sensitivity

A sensitivity study was conducted for the two best approaches, LSM and GO, to compare their performance in different scenarios.

6.4.1 Sensitivity of the greedy optimization approach to different oil price calibration windows

The GO was run with different calibration windows for the oil price, using the final parameters provided in Table 6. Figure 25 suggests that the best waiting criterion for all calibrations is between 22 and 24 months making the criterion quite insensitive for this change. The later the start of the calibration window, the higher the average NPV, most probably due to an increase in mean oil price.

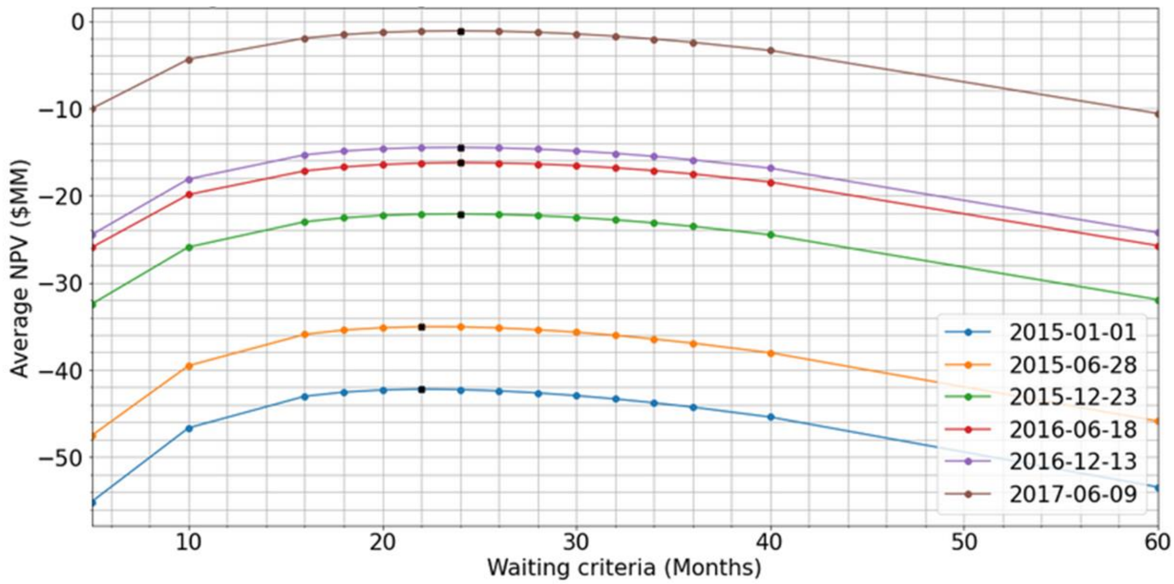

Figure 25. Waiting criterion versus net present value for different oil calibration windows that ends on February 1, 2020. The black dot represents the best criterion for each oil calibration window and the legend show the start of the calibration

6.4.2 Sensitivity of the least-squares Monte Carlo approach to different oil price calibration windows

The LSM was performed with the same oil calibration windows as the GO shown in Figure 26. As mentioned earlier, the mean oil price increases with a later start date for the oil calibration. This is most probably increasing the expected NPV and lowers the probability of obtaining a negative value.

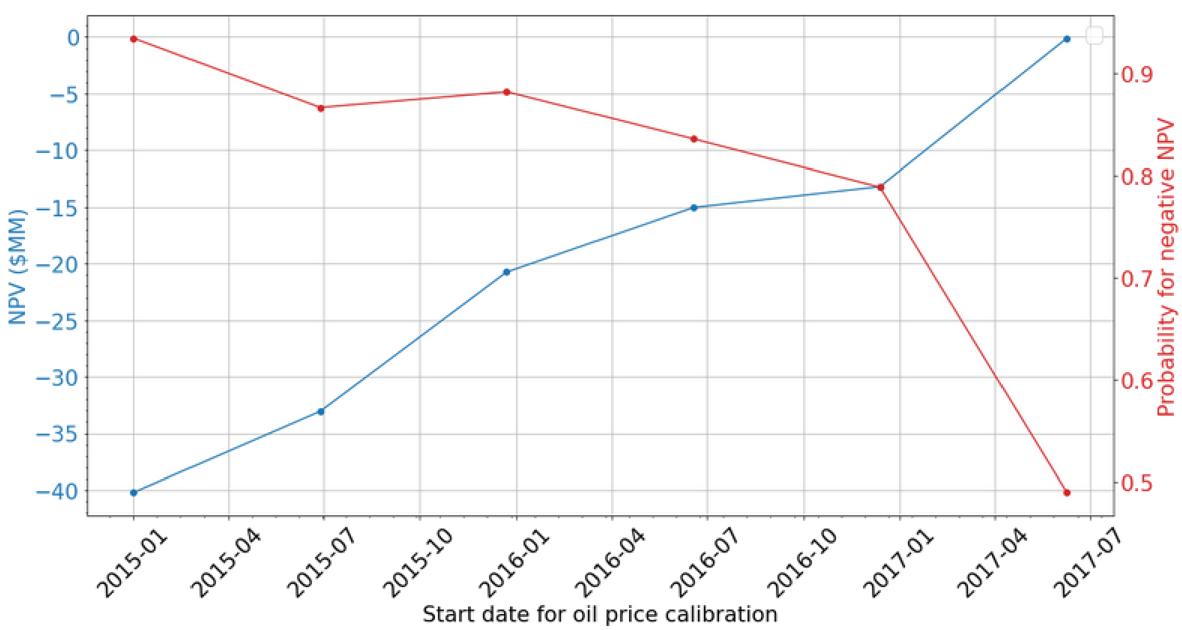

Figure 26. Net present value and probability of obtaining a negative NPV using different oil calibration windows that ends February 1, 2020

6.4.3 Least-squares Monte Carlo versus greedy optimization

This chapter analyses whether the LSM always exhibits better performance than the GO by comparing the two sensitivity studies. To this end, we use the studies with different oil calibration windows obtained earlier. One more sensitivity analysis was performed on each approach using different fixt Opex. The result of comparing those two is also presented in this chapter.

6.4.3.1 *Different oil calibration windows*

As Figure 27 depicts, the LSM approach results in an increase NPV between $1.02 MM and $2.00 MM depending on the oil calibration window. It can be noted that the NPV difference decreases between the two the later our windows start.

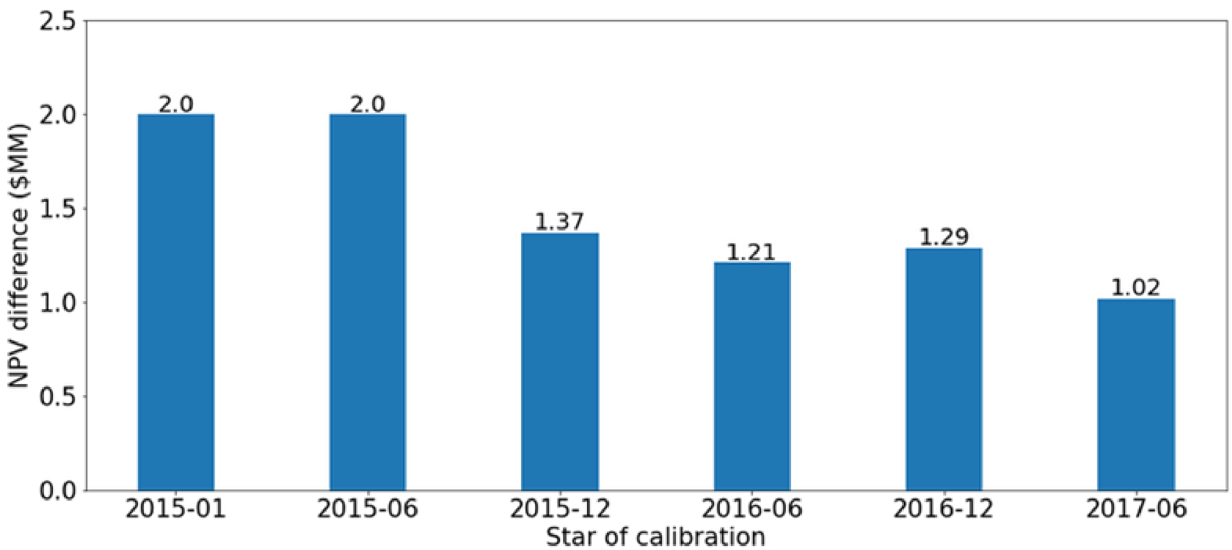

Figure 27. Sensitivity analysis displaying the net present value increase obtained using the LSM over the GO approach for different oil price calibration windows. All the calibration windows end on February 1, 2021

6.4.3.2 *Different fixed Opex*

The LSM and GO were compared across different fixed Opex values. Figure 28 indicates that the former approach yields a better NPV by $1.44 MM to $2.55 MM. It is also worth noting that increasing the fixed Opex increases the NPV difference.

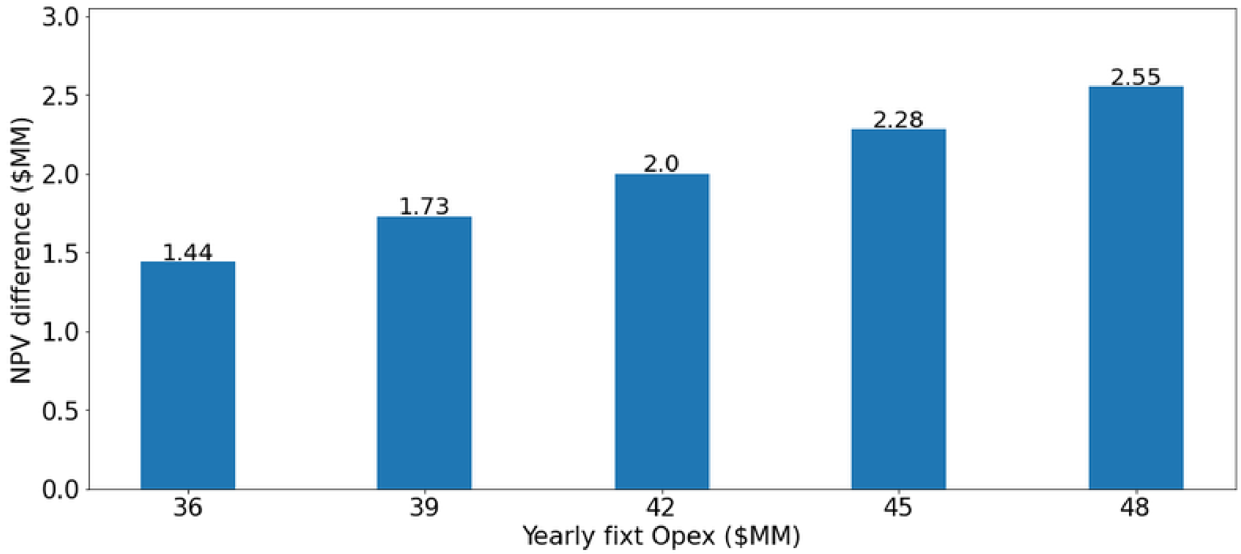

Figure 28. Sensitivity analysis displaying the net present value obtained using the LSM over using the GO approach for different fixed Opex values

7 Discussion

Earlier studies have shown that value can be created by not immediately abandoning the field at the first negative cashflow (Begg, Bratvold, & Campbell, 2004). This thesis supports these findings and arrives at the same conclusion. As both the GO and LSM approaches resulted in significant value creation over the FNC method, we focused the sensitivity analysis on the former two.

A comparison of the sensitivity results revealed that the LSM performed better across all the scenarios tested. One explanation for these results is that the LSM is a ROV method that can adapt its decisions dynamically based on learnings over time, compared to the GO, which bases its decision on a static rule. This make in terms of how decisions are made, the former process more realistic compared to the latter. These findings cannot be compared to those from other publications due to a lack of research comparing abandonment methods.

A disadvantage of the LSM and GO approaches may be that they are computationally heavy compared to the NCF. Reducing the number of paths and time steps can speed up the processing time, but this will also result in less accurate outputs from the model. One way to reduce processing time without sacrificing accuracy is to implement efficient code. A powerful method is to use array programming. This enable the model to process whole arrays of data in one step compared to use loops that processes one scalar at the time.

Although we chose the OU as our price model, this research can be easily extended to richer and more realistic price models such as Schwartz and Smith’s (2000) short-term–long-term model.

8 Conclusion and further research

Thousands of wells on the NCS must be plugged and abandoned in the next several decades. Although it is common procedure for many operators, this work demonstrates that value can be created by not abandoning a field at the first negative cash flow. Doing so ignores the uncertainty in the material variables, and the possibility that the cash flow may turn positive again at a later stage and, as a result, increase the field's profitability. We further demonstrate that the ROV model is optimal in assessing and extracting the value potential associated with the abandonment decision, It results in the highest NPV and thus illustrate the value created by not automatically abandoning a field at the first negative cashflow. Generally, the inherent combination of uncertainty and flexibility built into the ROV approach makes it superior in terms of creating value by identifying the optimal abandonment time. These two conclusions are addressed in the subsections below, the last of which contains recommendations for further research.

8.1 Value of using real option valuation

The ROV method, implemented through the LSM approach, is the best method to optimize the timing for abandonment and maximize the NPV of a field. It carries the ability to dynamically adapt its abandonment decision based on past learning as well as future expectations and decisions. This is also more realistic in terms of how decisions are made compared to the GO approach that statically make its decision based on a predetermined criterion. The sensitivity analysis indicates that the LSM results in a $1.02 MM – $2.55 MM higher NPV than the GO and a $38.69 MM improvement over the FNC method in the base case.

8.2 Value in not abandoning at the first negative cashflow

Abandoning a field at the first negative cashflow overlooks the possibility that the cashflow might become positive in the future. As Figure 20 illustrates, approximately 92% of all paths turn cashflow positive after a negative phase. This implies missed opportunities for creating value through an optimal timing of abandonment decision. The field may shift from cashflow negative to positive many times due to fluctuations in oil price and variable Opex. Another contributing factor to value creation is the fact that the field's NPV continue to increase as long as the discounting factor of the abandonment cost are larger than the negative cash flow.

8.3 Further research

We recommend a case study using a richer and more realistic oil price model, such as a two-factor mean-reverting model (Schwartz & Smith, 2000), calibrated with spot prices, oil futures and options, to evaluate the performance of the ROV model with more realistic oil price changes. Moreover, uncertainties into abandonment cost and production profiles could also be introduced to make the case more realistic.

References

Bank of England. (2020). *Data viewer.* Retrieved from https://www.bankofengland.co.uk/boeapps/database/fromshowcolumns.asp

Begg, S. H., Bratvold, R. B., & Campbell, J. M. (2004). Abandonment decisions and the value of flexibility. *Paper presented at the SPE Annual Technical Conference and Exhibition*. https://doi.org/10.2118/91131-MS.

Berg, T. V. (2011). *Calibrating the Ornstein-Uhlenbeck (Vasicek) model*. Retrieved from: https://www.statisticshowto.com/wp-content/uploads/2016/01/Calibrating-the-Ornstein.pdf

Bradley, M. E., & Wood, A. O. (1993). Forecasting oilfield economic performance. *Journal of Petroleum Technology, 46*, 965-971. https://doi-org.ezproxy.uis.no/10.2118/26054-PA.

Bratvold, R. B., & Begg, S. H. (2010). *Making good decisions.* Dallas, TX: Society of petroleum engineers.

Carriere, J. F. (1996). Valuation of the early-exercise price for options using simulations and nonparametric regression. *Insurance: Mathematics and Economics*, *19*(1), 19-30. https://doi.org/10.1016/S0167-6687(96)00004-2.

Dias, M. A. (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.

EIA. (2021). *Petroleum & other liquids*. Retrieved from https://www.eia.gov/dnav/pet/hist/rbrteA.htm

Equinor. (2021). *Planning for shutdown of the Veslefrikk field*. Retrieved from https://www.equinor.com/en/news/20210222-veslefrikk-shutdown-planning.html

Genesis Oil and Gas Consultants. (2019). *Decommissioning on the Norwegian continental shelf – cost effective and innovative solutions* (J75064A-A-RT-00001/D1). Retrieved from https://www.npd.no/globalassets/1-npd/publikasjoner/rapporter-en/decommissioning-on-the-norwegian-continental-shelf-cost-effective-and-innovative-solutions-2019.pdf.

Gilbert, E. (2004). An introduction to real options. *Investment Analysts Journal*, *33*(60), 49-52, https://doi.org/10.1080/10293523.2004.11082463.

Jafarizadeh, B., & Bratvold, R. B. (2012). Two factor oil price model and real option valuation: an example of oilfield abandonment. *SPE Economics and Management*, *4*(3), 158-170. https://doi.org/10.2118/162862-PA.

Longstaff, F. A., & Schwartz, E. S. (2001). Valuing American options by simulation: A simple least-squares approach. *The Review of Financial Studies*, *14*(1), 113-147. https://doi.org/10.1093/rfs/14.1.113.

Norges Bank. (2021). *Exchange rates*. Retrieved from https://www.norges-bank.no/en/topics/Statistics/exchange_rates/?tab=currency&id=USD

Norskpetroleum. (2020). *Investments and operating costs*. Retrieved from https://www.norskpetroleum.no/en/economy/investments-operating-costs/

Norskpetroleum. (2021). *Production forecasts*. Retrieved from https://www.norskpetroleum.no/en/production-and-exports/production-forecasts/

Oil & Gas Authority. (2017). *UKCS operating costs in 2017.* Retrieved from https://www.ogauthority.co.uk/media/5325/ukcs-operating-costs-a5-v2.pdf: Oil & Gas Authority.

Oil & Gas Authority. (2020). *UKCS decommissioning benchmarking report 2020.* Retrieved from https://www.ogauthority.co.uk/media/6942/oga_decommissioning_benchmarking-report-2020.pdf: Oil & Gas Authority.

Oil & Gas UK. (2018). *Decommissioning Insight Report.* (ISBN 978-1-9164677-4-3). Retrieved from https://oilandgasuk.co.uk/wp-content/uploads/2019/03/OGUK-Decommissioning-Insight-Report-2018.pdf: Oil & Gas UK.

Ornstein, E. G., & Uhlenbeck, L. S. (1930). On the theory of the Brownian motion. *Physical Review*, *36,* 823-841. https://doi.org/10.1103/PhysRev.36.823

Jetbrains. (2020). *The State of Developers Ecosystem 2020.* Retrieved from https://www.jetbrains.com/lp/devecosystem-2020/

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*(85), 2825-2830. https://www.jmlr.org/papers/v12/pedregosa11a.html

Pindyck, R. S. (1999). The long-run evolution of energy prices. *The Energy Journal, 20*(2), 1-27. https://doi.org/10.5547/ISSN 0195-6574-EJ-Vol 20-No 2-1, 1.

Schwartz, E. and Smith, J. E. (2000). Short-Term Variations and Long-Term Dynamic in Commodity Prices. *Management Science, 46*(7), 893-911. https://doi.org/10.1287/mnsc.46.7.893.12034

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.

The Norwegian Petroleum Directorate. (2021). *Veslefrikk*. Retrieved from https://factpages.npd.no/en/field/pageview/all/43618#

Thomas, P., & 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*. https://doi.org/10.2118/174868-MS.

Vega, C. A. (2018). Calibration of the exponential Ornstein–Uhlenbeck process when spot prices are visible through the maximum log-likelihood method. Example with gold prices. *Advances in Different Equations, 269*. https://doi.org/10.1186/s13662-018-1718-4

Willigers, B.J.A. & Bratvold R.B (2008). 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-MS.

Appendices

Appendix 1: Annular oil prices and data to estimate Variable Opex

Year	Total Opex on the NCS (bil NOK 2021)	Annual exchange rate (NOK/USD)	Total Opex on the NCS (mil USD)	Total production on NCS (mil Sm3 o.e)	Total production on NCS (mil bbl o.e)	Variable Opex /Unit cost (USD/bbl)	Annual Oil price (USD/bbl)
2008	64	5.6361	11 423	143.52	902.71	12.65	99.94
2009	68	6.2817	10 764	136.34	857.55	12.55	61.74
2010	68	6.0453	11 185	124.11	780.63	14.33	79.61
2011	67	5.6074	12 006	118.35	744.40	16.13	111.26
2012	74	5.821	12 747	111.58	701.82	18.16	111.63
2013	76	5.8768	12 872	106.65	670.81	19.19	108.56
2014	77	6.3019	12 153	109.56	689.11	17.64	98.97
2015	69	8.0739	8 554	112.92	710.25	12.04	52.32
2016	61	8.3987	7 218	116.01	729.68	9.89	43.64
2017	59	8.263	7 116	114.38	719.43	9.89	54.13
2018	61	8.1338	7 530	107.44	675.78	11.14	71.34
2019	62	8.8037	7 083	100.78	633.89	11.17	64.3
2020	57	9.4004	6 069	116.45	732.45	8.29	41.96

Appendix 2: Python code

Modules for calculations

```
import numpy as np
from numpy import sqrt
from numpy import exp
from numpy import log
from numpy import zeros
import pandas as pd
import math
from scipy.stats import *
import quandl
import random
from numpy.random import normal as nl
from statsmodels.stats.moment_helpers import cov2corr
from sklearn.linear_model import LinearRegression
from scipy.stats import multivariate_normal as mvn
from scipy.stats import pearsonr
```

Function for importing and calibrating oil price

```python
#Function for importing and calibrating Ornstein-Uhlenbeck oil price model (UB-Model)
#Modified code from Muhammad Waqar
dr_1 = datetime.strptime('01/01/2015', '%d/%m/%Y')#Start date of the imported data
dr_2 = datetime.strptime('01/01/2020', '%d/%m/%Y')#End date for the imported data

st_1 = datetime.strptime('01/01/2015', '%d/%m/%Y')#Calibrating from(Only for plot)
st_2 = datetime.strptime('01/02/2021', '%d/%m/%Y')#Calibrating to(Only for plot)

fc_1 = datetime.strptime("01/02/2020", '%d/%m/%Y')#Start forcast from date (Only for plot)
fc_2 = datetime.strptime("01/01/2040", '%d/%m/%Y')#Stop forcasting date  (Only fo plot)
oil = pandas.DataFrame(quandl.get("FRED/DCOILWTICO", start_date=dr_1, end_date=dr_2, collapse="monthly"))

pandas.options.mode.chained_assignment = None

def hist(FC: pandas.DataFrame):
    FC['hist'] = 0.0
    for indr in range(FC.shape[0]):
        try:
            FC['hist'][indr] = math.log(FC['Price'][indr]) - math.log(FC['Price'][indr - 1])
        except KeyError:
            FC['hist'][indr] = math.log(FC['Price'][indr])
    return FC

def mov_mean(data: pandas.DataFrame):
    data['avg'] = 0.0
    data['std'] = 0.0
    data['weeks'] = ''
    for indr in reversed(range(data.shape[0])):
        data['weeks'][indr] = str(data.shape[0] - indr)
        data['avg'][indr] = data['Price'][indr - data.shape[0]:].mean()
        data['std'][indr] = data['Price'][indr - data.shape[0]:].std()
    return data


class Demo:
    #Creating data frames
    def __init__(self, mode_: str, data: pandas.DataFrame, dr1: datetime, dr2: datetime,
                 st1: datetime, st2: datetime, fc1: datetime, fc2: datetime):
        self.forecast_data = pandas.DataFrame()
        self.mode = mode_
        self.data = data
        self.oil = data
        self.date = st1
        self.dr1 = dr1
        self.dr2 = dr2
        self.st1 = st1
        self.st2 = st2
        self.fc1 = fc1
        self.fc2 = fc2
        self.pre_process()
        self.diff =int((fc_2.year - self.date.year) + (fc_2.month - self.date.month) / 12 +
                       (fc_2.day - self.date.day) / 365) * 12
        self.delta_t = 1 / 12
```

```python
    def pre_process(self):
        self.data = self.data.reset_index()
        self.data['Price'] = self.data['Value']
        self.data = self.data.drop(['Value'], axis=1)
        self.data = self.data.loc[self.data['Date'] >= self.st1].reset_index().drop(['index'], axis=1)
        self.data = self.data.loc[self.data['Date'] <= self.st2]
        self.data['Years'] = self.data['Date'].index / 12
        self.date = self.data['Date'][-1:].reset_index()['Date'][0] + timedelta(weeks=-1)

    def auto_proc(self):
        lt_mean, speed, volatility = self.variables()
        self.mean_sd(volatility=volatility, speed=speed, lt_mean=lt_mean)
        self.forecast(volatility=volatility, speed=speed, lt_mean=lt_mean)
        self.data = hist(self.data)
        self.data = mov_mean(self.data)
        return self.data, self.forecast_data

    def variables(self):
        """
        Function used for calibration of the model
        Parameters (All parameters in this function are float type)
        ------------------------------------------------------------
        mean : Mean
        std : Standard Deviation
        pv : Percentage Volatility
        pd : Percentage Drift
        lt_mean : Long-term Mean
        :return:
        """
        self.data['x'] = 0.0
        self.data['y'] = 0.0
        num = 0
        for indr in range(self.data.shape[0]):
            num += 1
            self.data['x'][indr] = math.log(self.data['Price'][indr])
            try:
                self.data['y'][indr] = math.log(self.data['Price'][indr + 1])
            except KeyError:
                continue
        fitted_line = np.polyfit(self.data['x'][1:self.data.shape[0] - 1],
                                 self.data['y'][1:self.data.shape[0] - 1], deg=1)
        self.data['y_reg'] = 0.0
        self.data['residual'] = 0.0
        for indr in range(self.data.shape[0]):
            self.data['y_reg'][indr] = fitted_line[0] * self.data['y'][indr] + fitted_line[1]
            try:
                self.data['residual'][indr] = self.data['y'][indr + 1] - self.data['y_reg'][indr]
            except KeyError:
                continue
        sd_residual = tstd(self.data['residual'][1:self.data.shape[0] - 2])
        lt_mean = fitted_line[1] / (1 - fitted_line[0])
        speed = -math.log(fitted_line[0]) / self.delta_t
        volatility = sd_residual * math.sqrt(-2.0 * math.log(fitted_line[0]) /
                                             self.delta_t / (1 - fitted_line[0] ** 2))
        return lt_mean, speed, volatility
```

```python
def mean_sd(self, pd: float = 0.0, pv: float = 0.0, volatility: float = 0.0, speed:
            float = 0.0, lt_mean: float = 0.0):
    self.data['mean_lnP'] = 0.0
    self.data['sd_lnP'] = 0.0
    self.data['mean'] = 0.0
    self.data['sd'] = 0.0
    for indr in range(self.data.shape[0]):
        self.data['mean_lnP'][indr] = math.log(self.data['Price'][0]) *
        math.exp(-speed * self.data['Years'][indr]) + lt_mean *
        (1 - math.exp(-speed * self.data['Years'][indr]))
        self.data['sd_lnP'][indr] = math.sqrt(volatility ** 2 / 2 / speed *
                                      (1 - math.exp(-2 * speed * self.data['Years'][indr])))
        self.data['mean'][indr] = math.exp(self.data['mean_lnP'][indr] +
                                   (self.data['sd_lnP'][indr] ** 2) / 2.0)
        self.data['sd'][indr] = math.sqrt(math.exp(2 * self.data['mean_lnP'][indr] +
                                          self.data['sd_lnP'][indr] ** 2) *
                                  (math.exp(self.data['sd_lnP'][indr] ** 2) - 1))
def auto_proc2(self):
    lt_mean, speed, volatility = self.variables()
    self.forecast(volatility=volatility, speed=speed, lt_mean=lt_mean)
    self.data = hist(self.data)
    self.data = mov_mean(self.data)
    return self.data, self.forecast_data

def forecast(self, pd: float = 0, pv: float = 0, volatility: float = 0, speed: float = 0,
             lt_mean: float = 0):
    """
     Function used for forecast of the model
    Parameters (All parameters in this function are float type)
    ------------------------------------------------------------
    years_fc : forecast years
    years_fs : forecast years from start date
    mean_f : mean of Forecast price
    sd_f : standard deviation of forecast price
    mean_lnPt : mean of the natural log of oil price
    sd_lnPt : standard deviation of the natural log of oil price
    P10, P90 : P90 and P10 are low and high estimates respectively
    dt : time increment, most commonly a week
    wt, w : wiener process,  a normal distribution with zero mean and standard deviation
    Pt : oil price at Year t
    lnP : logarithm of Forecasted oil price
    :return:
    """
    date = self.date
    Price = self.data['Price'][self.data.shape[0] - 1]
    Year = self.data['Years'][self.data.shape[0] - 1]
    cols = ['years_fc', 'years_fs', 'mean_f', 'sd_f', 'mean_lnPt',
            'sd_lnPt', 'P10', 'P90', 'dt', 'w', 'lnP', 'Pt']
    FC = pandas.DataFrame(columns=cols)
    num = -1
    FC['Date_fc'] = date
```

```
        for indr in range(self.diff):
            FC = FC.append(pandas.Series([0.0] * len(cols), index=cols), ignore_index=True)
            date += timedelta(weeks=4)
            FC['Date_fc'][indr] = date
            rnd = random.random()
            FC['years_fc'][indr] = (num + 1) / 12
            FC['years_fs'][indr] = FC['years_fc'][indr] + Year
            FC['mean_lnPt'][indr] = math.log(Price) * math.exp(-speed * FC['years_fc'][indr]) +
            lt_mean * (1 - math.exp(-speed * FC['years_fc'][indr]))
            FC['sd_lnPt'][indr] = math.sqrt((volatility ** 2 / (2 * speed)) *
                                            (1 - math.exp(-speed * FC['years_fc'][indr])))
            FC['mean_f'][indr] = math.exp(FC['mean_lnPt'][indr] + FC['sd_lnPt'][indr] ** 2 / 2)
            FC['sd_f'][indr] = math.sqrt(math.exp(2 * FC['mean_lnPt'][indr] +
                                                  FC['sd_lnPt'][indr] ** 2) *
                                         (math.exp(FC['sd_lnPt'][indr] ** 2) - 1))
            FC['P10'][indr] = math.exp(FC['mean_lnPt'][indr] - 1.28 * FC['sd_lnPt'][indr])
            FC['P90'][indr] = math.exp(FC['mean_lnPt'][indr] + 1.28 * FC['sd_lnPt'][indr])
            if indr != 0:
                FC['dt'][indr] = FC['years_fc'][indr] - FC['years_fc'][indr - 1]
            else:
                FC['dt'][indr] = 0.0
            if indr != 0:
                FC['w'][indr] = norm.ppf(rnd, 0.0, math.sqrt(math.exp(2 * speed * FC['dt'][indr]) - 1))
            else:
                FC['w'][indr] = 0.0
            if indr != 0:
                FC['lnP'][indr] = FC['lnP'][indr - 1] * math.exp(-speed * FC['dt'][indr]) + lt_mean *
                (1 - math.exp(-speed * FC['dt'][indr])) + volatility / math.sqrt(2 * speed) * math.sqrt(
                    1-math.exp(-2*speed*FC['dt'][indr]))* FC['w'][indr]
            else:
                FC['lnP'][indr] = math.log(Price)
            FC['Pt'][indr] = math.exp(FC['lnP'][indr])
            num += 1
            self.forecast_data = FC.copy()
obj2 = Demo('GOU', oil, dr_1, dr_2, st_1, st_2, fc_1, fc_2)
```

Various functions

```
#Discounting factor
def Discount(y,Iterations,NetRevAd):
    DR=0.08 #Yearly discount rate
    DR=DR/12 #Monthly discount rate
    DCF=zeros((y, Iterations))
    DCF=NetRevAd/((1+DR)**np.arange(y).reshape((y,1)))
    return DCF
```

```
#Oil production
def Oilprod(Wells,y,Iterations):
    qw=np.full((y, Iterations),4.5)
    qt=np.zeros((y, Iterations))
    for j in range(Wells):
        for i in range(1,y+1):
            qw[i-1]=1.0039*np.exp(-0.189*(i/12+5))/12*6.28981
        qt=qt+qw
        qw*0
        return qw
```

```
#Water production
def Waterprod(Wells,y,Iterations):
    qww=np.full((y, Iterations),4.5)
    qtw=np.zeros((y, Iterations))
    for j in range(Wells):
        for i in range(1,y+1):
            qww[i-1]=4.6917*np.exp(-0.171*(i/12+3))/12*6.28981
    qtw=qtw+qww
    qww*0
    return qtw
```

```
#Create random correlated data with lag time
def Rand_gen(y,Iterations):
    lag=12#Months of lag
    Corr=0.836#Correlation fraction for Variable Opex
    A=np.zeros((y,Iterations))
    B=np.zeros((y,Iterations))
    A[0:lag,0:Iterations]=np.random.normal(loc=0.0, scale=1.0, size=(lag,Iterations))
    B[y-lag:y,0:Iterations]=np.random.normal(loc=0.0, scale=1.0, size=(lag,Iterations))
    v = np.array([1, Corr])#Correlation matrix
    cov = cov2corr(toeplitz(v))#Covariance matrix
    for i in range (Iterations):
        scores = mvn.rvs(mean = [0.,0.], cov=cov, size = y-lag)
        A[lag:y,i]=scores[:,0]
        B[0:y-lag,i]=scores[:,1]
    return B ,A
```

```
#The Ornstein-Uhlenbeck with logarithmic prices (For oil simulation)
def OneFactorMeanT(y,Iterations,StartPrice,lt_mean,speed,volatility,RA):
    dt = 1/12  # Time step.
    Start1=log(StartPrice)
    Oil_price=zeros((y, Iterations))
    A=exp(-speed * dt)
    B=lt_mean*(1-exp(-speed*dt))
    C=volatility/sqrt(2*speed)*sqrt(1-exp(-2*speed*dt))
    lnp=Start1
    for i in range(0,y):
        lnp=lnp*A+B+C*RA[i,:]
        Oil_price[i]=exp(lnp)
    return(Oil_price)
```

```
#The Ornstein-Uhlenbeck with logarithmic prices (For Variable Opex simulation)
def OneFactorMeanTO(y,Iterations,StartPrice,lt_mean,speed,volatility,RA):
    dt = 1/1  # Time step.
    Start1=log(StartPrice)
    Variable_Opex=zeros((y, Iterations))
    A=exp(-speed * dt)
    B=lt_mean*(1-exp(-speed*dt))
    C=volatility/sqrt(2*speed)*sqrt(1-exp(-2*speed*dt))
    lnp=Start1
    for i in range(0,y):
        lnp=lnp*A+B+C*RA[i,:]
        Variable_Opex[i]=exp(lnp)
    return(Variable_Opex)
```

```
#Finds first negative cash flow month
def accumulate_based(A2):
    A2[np.maximum.accumulate((A2 < 0), axis=1)[:, ::-1][:, ::-1]] = 0
    A2=np.sum(np.where(A2>0,1,0),axis=1)
    return A2
```

Generate random numbers for oil and variable opex simulations

```
y=200#Month of production
Iterations=180000#Number of paths
RA,RB=Rand_gen(y,Iterations)#Create correlated random numbers
```

Simulating production

```
#Variables
IterationsP=1
WellsOil=1#Oil producing units
WellsWater=1#Water producing units

#Simulating production
qt=Oilprod(WellsOil,y,IterationsP)#Oil production
qtw=Waterprod(WellsWater,y,IterationsP)#Water production
```

Calibrate and simulate oil prices

```
st_1 = datetime(2015, 1, 1)#Start date for calibration
obj2 = Demo('GOU', oil, dr_1, dr_2, st_1, st_2, fc_1, fc_2)#feeding the calibration model with data
lt_mean, speed, volatility=obj2.variables()#Calibrating with linear regression
StartPrice=61 #Initial oil price
Oil_price=OneFactorMeanT(y,Iterations,StartPrice,lt_mean,speed,volatility,RA)#Simulating oil prices
```

Persons correlation test

```
#Persons correlation test on variable opex and the oil price
OpexHistoryX=np.array([12.82, 12.60,14.35,16.15,18.18,19.20,17.69,12.08,9.89,
                       9.89,11.14,11.19,8.31 ])#Historical Variable Opex
oilhist=np.array([99.94, 61.74,79.61,111.26,111.63,108.56,98.97,52.32,43.64,
                  54.13,71.34,64.30,41.96 ])#Historical Brent prices
corr, _ = pearsonr(OpexHistoryX, oilhist)
print('Pearsons correlation: %.3f' % corr)
```

Calibrate the UB-model with historical variable opex values

```
#Calibrating the UB-model
OpexHistoryX=np.array([12.82, 12.60,14.35,16.15,18.18,19.20,
                       17.69,12.08,9.89,9.89,11.14,11.19,8.31 ])#Historical variable Opex
OpexHistoryY=np.array([ 12.60,14.35,16.15,18.18,19.20,17.69,
                       12.08,9.89,9.89,11.14,11.19,8.31])#Historical variable Opex with one offsets

#Calibrating the UB-model
delta_t=1
length=len(OpexHistoryX)
Opx=np.log(OpexHistoryX)
Opy=np.append(np.log(OpexHistoryY),np.array(0))
fitted_line = np.polyfit(Opx[1:length-1],Opy[1:length-1],deg=1)
y_reg=zeros(length)
residual=zeros(length)
for i in range(length):
    y_reg[i]=fitted_line[0]*Opy[i]+fitted_line[1]
for i in range (length-1):
    residual[i]=Opy[i+1]-y_reg[i]
sd_residual=tstd(residual[1:length-2])
lt_mean2=fitted_line[1]/(1-fitted_line[0])
speed2=-log(fitted_line[0])/delta_t
volatility2=sd_residual*sqrt(-2*log(fitted_line[0])/delta_t/(1-fitted_line[0]**2))

print("{:.3f}".format(volatility2),' Variable Opex Volatility')
print("{:.3f}".format(speed2),'Variable Opex Speed')
print("{:.3f}".format(exp(lt_mean2)),' Variable Opex $/bbl mean')
```

Simulating variable Opex

```
Start=OpexHistoryX[length-1]#First variable Opex is the last historical value
OpexF=OneFactorMeanTO(y,Iterations,Start,lt_mean2,speed2,volatility2,RB)#Simulating variable  opex
```

Simulating Net revenues

```
#Variables
qtwCost=0.5#Water production cost $/bbl
FixOpex=3500000#Monthly cost
Abandonment=233270000 #Abandonment costs
Rev=Oil_price*qt*1000000#Revenues
Expenses=qt*OpexF*1000000+qtwCost*qtw*1000000+FixOpex#Expenses
NetRev=Rev-Expenses#Net revenues
```

Greedy approach

```
#Variables
wai=28#Waiting strategy in months
AB=[0]*Iterations
AA=[0]*Iterations
NC=0
i=0
j=0
waiting=0
#Finding first negative cash flow and abandon after the waiting time is fullfilled
while j<Iterations:
    while i<y:
        if NetRev[i,j] <0:
            waiting=waiting+1
            if NC==0:
                AA[j]=i
                NC=1
            if waiting==wai:
                NetRev[i+1:y,j]=0#Remove cash flows after abandonment
                AB[j]=i
                NetRev[i+1,j]=-Abandonment#Adding abandonment cost
                i=y
        else:
            waiting=0
        i=i+1
    waiting=0
    i=0
    j=j+1
    NC=0
Diff=np.array(AB)+1# The abandonment time starts from zero
DCF=Discount(y,Iterations,NetRev)#Discounting cash flows
CumDCF = np.cumsum(DCF,axis=0)#Cumulative cash flow
NPV= np.sum(DCF,axis=0)#Net present value
maxi=np.max(AB)#Finding max abandonment time
print('Probabillity for negative NPV using GA', np.sum(np.where(NPV<0,1,0))/Iterations)
```

Negative cash flow approach

```
NPVFirst=np.zeros((y, Iterations))#Empty NPV maptrix
ABMatrix=np.ones((y, Iterations))*Abandonment#Create abandonment matrix
DCAB=Discount(y,Iterations,ABMatrix)#Discounted abandonment cost
NPVFirst=zeros(shape=(y,Iterations))

for i in range(Iterations):
    NPVFirst[0:AA[i]+1,i]=DCF[0:AA[i]+1,i]#Add cash flows until first negative cash flow
    NPVFirst[AA[i]+1,i]=-DCAB[AA[i]+1,i]#Add abandonment cost after first negative cash flow
NPVFirstAB=np.sum(NPVFirst,axis=0)
DiffFirst=np.array(AA)+1# The abandonment time starts from zero
print('Probabillity for negative NPV abandoning at first negative cash flow',
      np.sum(np.where(NPVFirstAB<0,1,0))/Iterations)
```

Least Square Monte Carlo

```
intermediate_results = []#Save data for plots
#Create data
lenN=y#
NetRevAdR=Rev-Expenses#Net revenue
DCFROVAd= zeros(shape=(y,Iterations))#Create Matrix
DCFROVAd=NetRevAdR
#Croped cash flow
DCFROVAdCrop=DCFROVAd[0:lenN,:]#Remove unused simulated lenght
DCFROVAdCrop=Discount(lenN,Iterations,DCFROVAdCrop)#Discount the value against time
CumDCFROV = np.cumsum(DCFROVAdCrop,axis=0)#Cummulative net pressent value
ROVNPV=CumDCFROV# NPV for each month
#For plot
ABMatrixErr=np.ones((lenN+2, Iterations))*Abandonment#Adjusted abandonment matrix
DCABErr1=Discount(lenN+2,Iterations,ABMatrixErr)#Adjusted discounted abandonment matrix
DCABErr=DCABErr1[1:-1]#Offset the matrix with one
Error=CumDCFROV-DCABErr#+Abandonment#NPV for each step if abandoning on the next time step
#Cashflow with abandonment cost in end
DCFROVAdCrop[-1,:]=-DCAB[-1,:]#Install abandonment cost in the end
ROVNPVNInv=np.flip(np.cumsum(np.flip(DCFROVAdCrop, 0), 0),0)#Reverse cumulative NPV
ROVNPVNInvC=ROVNPVNInv.copy()#
AbTime=DCAB[-1,:]*0#np.zeros((1,Iterations))#[0]*Iterations
#Prepare for LSM
exercise=zeros(shape=(lenN-1,Iterations))
continuation=zeros(shape=(lenN-1,Iterations))
intermediate_results = []
regression_model=LinearRegression()
cashflow=np.maximum(DCFROVAdCrop[-1,:],0.0)#
Rem=DCAB[-1,:]*0#Empty row
#Simulate LSM
for i in reversed(range(1,ROVNPVNInv.shape[0]-2)):#Start from the end
    X1=np.vstack([Oil_price[i-1,:],OpexF[i-1,:]])#Independent
    Y1=ROVNPVNInv[i,:]#Dependent
    fitted=regression_model.fit(X1.T,Y1)#Create function
    OilC,OpexC=fitted.coef_
    continuation=fitted.intercept_+OilC*Oil_price[i-1,:]+OpexC*OpexF[i-1,:]#Estimate continuation
    exercise=-DCAB[i,:]#Abandonment cost
    ex_idx=continuation<exercise#Test condition
    cashflow[ex_idx]=exercise[ex_idx]#Save data for the phat plot
    MovAB=-DCAB[i+1,:]#New abandonment cost
    DCFROVAdCrop[i+2,:][ex_idx]=Rem[ex_idx]#Move back the abandonment time if criteria is fulfilled
    MTF=np.tile(ex_idx, (lenN-i-1, 1))#Transform the 2D criteria array to a matrix
    Rem2=DCAB[i+1:,:]*0#Empty matrix to remove cash flows after abandonment
    DCFROVAdCrop[i+1:,:][MTF]=Rem2[MTF]#Remove cash flows after abandonment
    DCFROVAdCrop[i+1,:][ex_idx]=MovAB[ex_idx]#Install the new abandonment cost
    ROVNPVNInv=np.flip(np.cumsum(np.flip(DCFROVAdCrop, 0), 0),0)#New reverse cumulative NPV
    NowTime=DCAB[-1,:]*0+i#update the counter
    AbTime[ex_idx]=NowTime[ex_idx]#save abandonment time
    intermediate_results.append(( Y1,Error[i,:], ex_idx))#Used for plots
LSMNPV=np.sum(DCFROVAdCrop,axis=0)#NPV for each path using LSM
print('probabillity for negative NPV using LSM', np.sum(np.where(LSMNPV<0,1,0))/Iterations)
```