Project report

TPG4560 – Petroleum Engineering,
Specialization Project

D Aqnan Marusaha Matthew

CO_2 EOR, Optimization Study and Smart Proxy Model Development

Trondheim, December 1, 2020

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

NTNU

Abstract

CO_2 injection is one of the most common EOR methods employed in mature fields. When injecting CO_2 into the reservoir, it will reduce the oil viscosity, add an acidic effect on carbonate and shaley rock, swells oil, and reduces IFT between oil and water-based on the miscibility condition. When planning a CO_2 injection project, usually it needs to go through a conceptual design study. Several stages and design variables need to be passed before the project gets approved. One of them is a reservoir engineering design study. When coming to the design, optimization needs to be performed to find the optimum flow design. Doing this takes a lot of time, as conventional reservoir simulation takes many times to generate one case, not to forget that the optimization problem is a complex problem.

Employing Smart Proxy Model (SPM) to substitutes our reservoir model and coupling with an optimization algorithm helps to tackle this problem. An optimization study to maximize the total oil produced on CO_2 flooding synthetic case was performed in this study. The SPM that we built using Artificial Neural Network (ANN) reduces the run time from 4.5 minutes to less than 10 seconds for running one case. Genetic Algorithm (GA) then coupled with our SPM to solve the optimization problem. The optimum condition that reached with the optimization algorithm then tested in Eclipse. It shows that our SPM has a relative error of 1.63% when compared with the Eclipse run result.

Keywords: Smart Proxy Model, Genetic Algorithm, Artificial Neural Network, CO_2 flooding

Table of Contents

List of Tables

List of Figures

Abbreviations

AF	=	Acentric Factor
AI	=	Artificial Intelligence
ANN	=	Artificial Neural Network
BHP	=	Bottomhole Pressure
BIP	=	Binary Interaction Parameters
Bo	=	Oil Formation Volume Factor
CCE	=	Constant Composition Expansion
CCS	=	Carbon Capture and Storage
CI	=	Computational Intelligence
DFF	=	Deep Feed Forward
DL	=	Differential Liberation
DOE	=	Design of Experiment
EOR	=	Enhanced Oil Recovery
EOS	=	Equation of State
FOPR	=	Field Oil Production Rate
FOPT	=	Field Oil Production Total
FORE	=	Field Oil Recovery Efficiency
FPR	=	Field Average Pressure
FVF	=	Formation Volume Factor
GA	=	Genetic Algorithm
GOC	=	Gas Oil Contact
GOR	=	Gas Oil Ratio
HSE	=	Health, Safety and Environment
LHS	=	Latin Hypercube Sampling
LU	=	Lower Upper
MCN	=	Multiple Carbon Number
ML	=	Machine Learning
MMP	=	Minimum Miscibility Pressure
MRST	=	Matlab Reservoir Simulation Toolbox
NCS	=	Norwegian Continental Shelf
Pc	=	Critical pressure
PCA	=	Principal Component Analysis
PR	=	Peng-Robinson
PS	=	Particle Swarm
Psat	=	Saturation Pressure
PVT	=	Pressure, Volume, Temperature
QC	=	Quality Control
ROR	=	Rate of Return
SAGD	=	Steam-assisted Gravity Drainage

SCN	=	Single Carbon Number
SPM	=	Smart Proxy Model
SRK	=	Soave-Redlich-Kwong
SRM	=	Surrogate Reservoir Model
STOOIP	=	Stock Tank Oil Original in Place
Tc	=	Critical temperature
Tres	=	Reservoir Temperature
Vc	=	Critical Volume
WAG	=	Water Alternating Gas
WOC	=	Water Oil Contact

Chapter 1

Introduction

1.1 Background

The decline in oil production and fewer discoveries force engineers to think out of the box. Even the shifts of renewable energy are surging, it still can't fulfill the increment of the worldwide energy demand. Tertiary oil recovery becoming more promising, especially CO_2 injection. More countries started to study the feasibility of this EOR method.

Other than that, the mitigation to decelerate the impact on climate change is being highlighted by most of the countries in the world. Norway starts to consider and apply Carbon Capture and Storage (CCS) in the Norwegian Continental Shelf (NCS). This can be a chance for oil field development which yields a decent increment of profit if being coupled together.

Coming up with the final decision to start the CO_2 injection into the reservoir itself is not easy. A lot of things need to be designed, such as the surface facilities, wells, transportation process, storage system, recycling plant, and other parameters. Deciding whether it is profitable is such an exhaustive process, where one of the components to be done is to optimize the CO_2 injection EOR based on selected optimization parameters, which usually is done using reservoir simulator at first before moving to the small coverage area then to the field-scale project.

Conventional reservoir simulation nowadays is sufficient enough to model our reservoir behavior. But, the current limitation is its prohibitive running time, especially for large, fine-gridded reservoir models. Considerable memories also needed for result reporting, which will computationally heavy to be stored. And not to forget that we need a considerable amount of runs to reach the optimal results.

In parallel with that, Artificial Intelligence nowadays becoming a hotspot. Mohaghegh et al. (2000) initiated the study of coupling Artificial Intelligence with reservoir simu-

lation. In 2006, the first smart proxy model was built. This advanced application is a combination of reservoir engineering, reservoir modeling, and reservoir management in the art of Artificial Intelligence and Machine Learning. The coupling of these expertises can yield a highly accurate and fast proxy model which breaks the limit of conventional numerical modeling.

Based on that, this specialization project was made to develop a new strategy for establishing Smart Proxy Model (SPM) by coupling optimization, statistics, and data-driven techniques with an in-depth numerical and physical formulation to facilitate optimization of the processes. The simple case created as a study base which the improvement will be discussed in the latter part.

1.2 Objective

The main research objective is to develop new strategies that will allow a significant reduction of runtime associated with the commercial simulators without sacrificing accuracy. This will be collaborated with CO_2 as one of the flooding techniques for Enhanced Oil Recovery (EOR), which can be applied as an optimization method to maximize the oil recovery.

This study is focusing on:

1. Construction synthetic model for proxy model base data.
2. Formulation the optimization problems, which maximize oil recovery and choosing the optimization parameter for the study.
3. Design of experiments, data sampling, and analysis for proxy model input dataset.
4. Developing a proxy which represents a dynamic process of the used synthetic model.
5. Perform the optimization with generated proxy to solve the optimization problem.

Chapter 2

Theory

All basic concepts, theories, and frameworks for the current study are presented in this chapter.

2.1 Field Production Phases

When it comes to field recovery methods, or commonly mentioned as production phases, generally those methods grouped in three recovery methods or production phases (Carcoana, 1992).

Primary Oil Recovery

Hydrocarbon produced in this phase is produced by natural energy from the reservoir. Driving forces that present may differ for each reservoir, such as an active aquifer, gravity drainage, gas cap expansion, or any combination of them.

Secondary Oil Recovery

When natural energy is not able to flow hydrocarbon economically anymore, to improve the recovery, external forces can be given to the reservoir. When injection wells introduced to the reservoir to increase or maintain the reservoir pressure, secondary oil recovery is applied to the field. Injection fluids that are introduced in this phase do not change the oil or rock properties at this phase.

Tertiary Oil Recovery

It is common to align both primary and secondary oil recovery at the beginning of the production phases. The purpose is to prolong the field life. Usually, they recover 1/3 of original oil in place (Lee et al., 2020). But, when the secondary recovery reaches its

end time, when no longer economically attractive, tertiary recovery can be introduced. This method introduces injection fluid which does not originally exist in the reservoir. The difference with the secondary recovery is the injected fluids now designed to tweak our reservoir oil properties. It is more common to mention it as Enhanced Oil Recovery (EOR).

2.1.1 Enhanced Oil Recovery

As more and more discovered fields reach their decline phase, with lower discoveries throughout the world, EOR starts to glance. Despite the high capital cost when implementing it, many fields start to implement EOR. Satter and Iqbal (2016) studied average cumulative recovery for each production phases, showing that 30-60% oil can be recovered under the favorable condition when doing EOR. This sounds promising, but several studies need to be done before implementing EOR in a real field case.

There are a lot of methods in EOR. Some classified as thermal and nonthermal. Chemical, miscible, and other advanced EOR methods (such as microbial, electromagnetic, etc.) can be classified as nonthermal methods. Every method has its own characteristics, principles, and type of fields which it fits to. Al Adasani and Bai (2011) classifies the fluid and reservoir condition which EOR methods proven to work based on a collection of data where EOR methods have been applied. In summary, three big groups of EOR methods and their principle are tabulated in **Table 2.1**.

Table 2.1: Methods of EOR

Group	Methods Used[1]	Basic Principle[2]
Chemical	Polymer, surfactant, alkaline, emulsion, micellar-polymer	Sweep or displacement efficiency improvement
Miscible	CO_2 flooding, N_2 flooding, enriched gas drive, vaporizing gas drive, water alternating gas (WAG)	Displacement efficiency improvement
Thermal	Cyclic steam injection, steam flooding, SAGD, hot water flood, huff and puff	Sweep and displacement efficiency improvement

[1] Satter and Iqbal (2016)

[2] Carcoana (1992)

The investment cost for moving from secondary to enhanced oil recovery is really high since not all wells and production facilities are designed for EOR implementation during its first design. Different production problems may occur during the process, yet need to be mitigated. This differs for different EOR methods. A sketch of oil cost for some extraction methods tabulated in **Table 2.2**.

Based on Kokal and Al-Kaabi (2010), thermal EOR is the most applied method worldwide. Several countries applied this, where unconventional oil deposited such as Venezuela, Canada, Indonesia, US, and many other countries. While in the US, CO_2 related EOR are surging, more than 100 projects already started and/or finished. Other methods already started to be implemented, with smaller scale and fewer projects, such as hydrocarbon injection (Venezuela, Canada, Libya), chemical injection (China), and any other methods.

Table 2.2: Cost of oil based on extraction method (Satter and Iqbal, 2016)

Method of production	Cost ($/bbl)
Conventional oil	10-30
Enhanced oil recovery	
CO_2 injection	20-70
Thermal	40-80
Others	30-80
Unconventional oil	50-80

2.1.2 Carbon Dioxide Enhanced Oil Recovery

The surge of CO_2 related projects is one of the attractiveness of this study. And also the other reason is the current issue which is the increase in global temperature. NASA reported that the latest annual average anomaly is 0.99°C (2019). As one of the main greenhouse gases, development in CO_2 utilities will be yield tremendous effect.

Kamali et al. (2017) mentioned CO_2 EOR as the most promising method. This is considering CO_2 and oil interaction, availability, and technology development. The various way for method implementation also became a main interest, such as continuous flooding, cyclic (huff-n-puff), and water alternating gas (WAG). But before moving into this, a brief explanation of why CO_2 is one of the injection agents and what mechanism occurs during the mixing process with hydrocarbon.

Here, the process will be described based on CO_2 continuous injection. The process illustrated by **Figure 2.1**. In execution, the processed CO_2 (either bought or recycled) transported to the injection well. Production facilities designed for the injection process such as compressor then help the injection process.

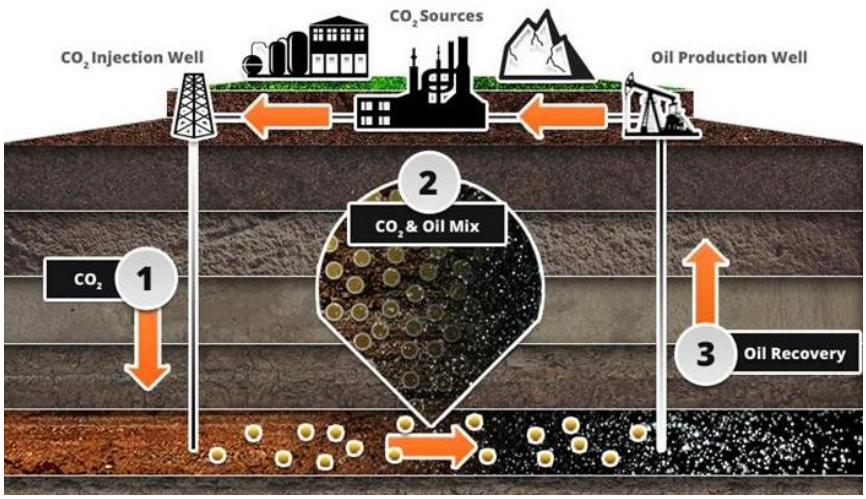


Figure 2.1: CO_2 continuous injection (LLC, 2020).

In the injection depth target, when CO_2 meets the reservoir fluid, it will mix. CO_2 is highly soluble in oil and less soluble in water, which will yield these effects to our reservoir fluid:

- Reduction in oil viscosity
  CO_2 reduces the oil viscosity to which it mixes to. The study performed by Simon and Graue (1965) shows the reduction of viscosity to the saturation pressure. This will yield to increment of mobility ratio, which can be noticeable in more viscous crude oil
- Swelling of oil
  As CO_2 mixes, it will increase the oil volume. This will increases the recovery as the mass of the remaining oil is lower than the abandoned oil which was CO_2 free.
- Increment in oil density
  Other than to the oil, CO_2 effect can't be ignored. Expansion of water when CO_2 being injected will make the densities for both fluid closer, which reduces gravity segregation effect (Gholami, 2014).
- Acidic effect on carbonate and shaley rock
  CO_2 forms carbonic acid with water, which dissolves calcium and magnesium carbonates. As more carbonates being dissolved, this increases injectivity and permeability. CO_2 also has a stabilizing effect on shaley rocks, which prevents clay swelling (Carcoana, 1992).
- IFT reduction
  Xing et al. (2013) studies shows the interfacial tension between water and oil. Studies show an increment of pressure and salinity increases IFT while increasing molar fraction of CO_2 decreases it.

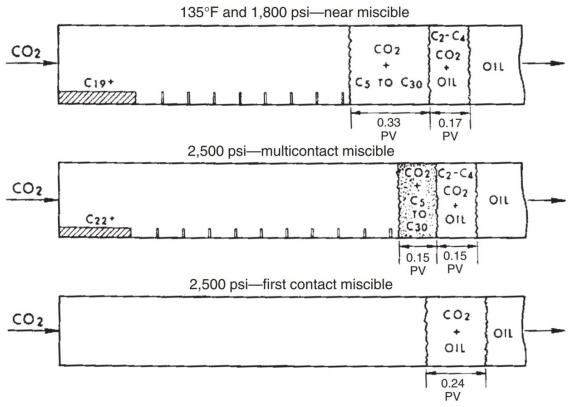


Figure 2.2: Schematic of distribution of components in CO_2 displacement (Whitson et al., 2000).

Those effects happen based on the mixing pressure and temperature, which will yield whether the flooding process is miscible or immiscible. **Figure 2.2** shows the mixing result for different miscible condition and the component mixture for each part of the fronts. Miscible flooding will yield better recovery than immiscible.

The oil and CO_2 mixture produced on the surface then can be separated. Separated CO_2 then can be reinjected while the produced oil can be transported to be processed by the downstream section. Some CO_2 might be not produced, as it will be deposited in the reservoir. This is one of the efforts for CO_2 sequestration.

When it comes to determine the CO_2 design study, Jarrell et al. (2002) shows that there are several points which needs to be done, which are in sequence listed below (illustrated in **Figure 2.3**).

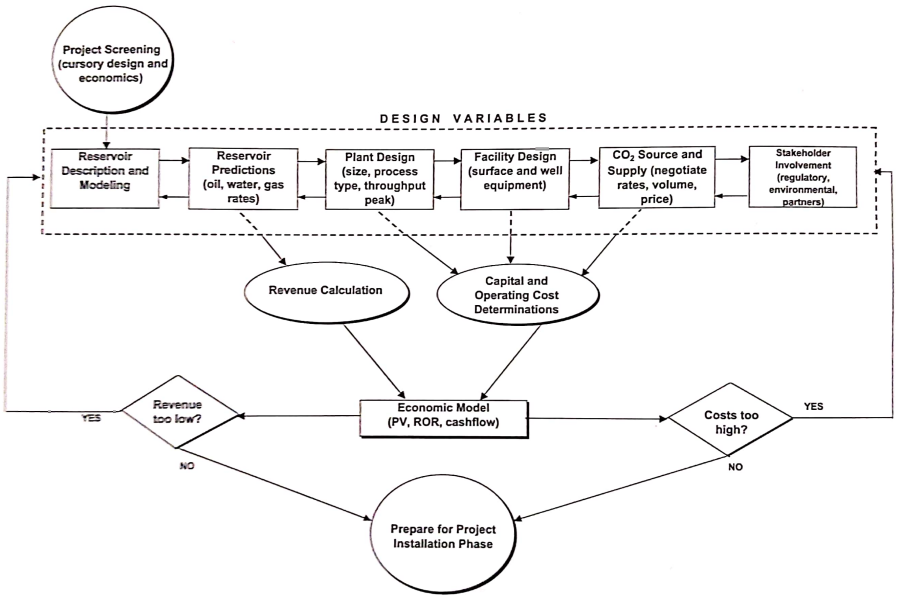


Figure 2.3: Conceptual design process (Jarrell et al., 2002).

1. Screening
   Pre-study related to the project feasibility. It is a must to be technically and economically feasible before starting a project.

2. Reservoir engineering design
   This phase is one of the important phases, which required data gathering and advanced PVT tests, reservoir model modification, prediction of flood performance, and optimization for the real case.

3. Surface facilities design
   All tools, from injection facilities, CO_2 transport, and gathering, and recycle plant need to be designed. Oil mixed with CO_2 also needs extra treatment when compared with conventional oil as sellable oil itself has an upper limit of CO_2 presence.

4. Well design
   Both segment, production and injection wells, need to be redesigned. Production

problems that might happen such as corrosion, asphaltene deposition need to be prevented beforehand.

5. Implementation
   This comes with a huge decision, such as planning, preinjection data gathering, permits, and monitoring. Not to neglect HSE planning throughout the project.

This study will look more into depth to the reservoir engineering design part for CO_2 injection design.

2.2 Reservoir Engineering Design for CO_2 Injection

Several points are needed to be taken care of engineering design purposes for this EOR method as described by Jarrell et al. (2002). Some of them are:

- Collecting valid input data
  This itself includes the fluid PVT data, which representative and CO_2 behavior at the studied reservoir condition. Relative permeability, advanced CO_2-related PVT tests (swelling test, multi-contact test, and slim-tube test) will hugely help the study.
- History matching
  This process, which usually performed throughout the production life is crucial to refining the reservoir model. This process can be done by tuning the reservoir description parameters to the obtained production data.
- Predicting CO_2 performance
  The refined reservoir model coupled with a good fluid model then can be used for CO_2 injection performance prediction. The results obtained then can be used as a reflection of the real-field performance.
- Determining the optimum flow design
  A lot of parameters need to be taken care of, such as the injection pressure limit, CO_2 availability, surface facilities limitation, and any other parameter. This usually coupled with economic analysis to select the optimal result.

For study purposes, as no field data nor case available to be used, a synthetic model can be built which designed to reflects the field case. Below will be explained some points that need to be taken care of when building the model, especially for the CO_2 injection case.

2.2.1 Fluid Modeling

The fluid model is one of the reservoir modeling components. There is two way to model the fluid for CO_2 flooding.

1. Modified Black-Oil model
   This model itself the improvement from the standard black-oil model. As at first, it uses three components, which are stock-tank oil, hydrocarbon gas, and water. Here, CO_2 will be added as the fourth component. As no EOS performed, it assumes the first-contact miscibility when it reaches thermodynamic MMP.

This model itself developed as a simplified approach to a complex problem. Jarrell et al. (2002) shows that usage in this problem may lead to several false predictions. One of them is a high grid size effect on the result. It shows that at least 6x6 grid size will be the best representation, while the common case is larger than that size. The smaller size of the grid itself increase the computation cost, which should be considered.

2. Fully Compositional Here, EOS models are used, the most common EOS are Peng-Robinson (PR) and Soave-Redlich-Kwong (SRK). This model itself able to solve the miscibility problem encountered when using the modified black-oil model. Yet, the main problem is to do the EOS tuning, which a lot of test data needed to perform.

   As the goal is to model the fluid behavior in CO_2 injection, fine-tuning main goal is to predict the development of miscibility. Another intake is to group the components (lumping). This needs to be done correctly to not sacrifice accuracy.

Best practices in the field are using a fully compositional model (Jarrell et al., 2002). As miscibility development is crucial in CO_2 EOR, this accuracy can't be sacrificed.

Performing tuning in the EOS model itself needs to be taken care of. Ali et al. (2015) studied that the best method for EOS tuning approach is Al-Meshari (2005), which fits nicely for oil samples. Another note from that study is splitting C_{7+} to 2 pseudo component will be enough, with the lumping idea can be matched to the model purpose.

Below are the steps of EOS Tuning as explained by Al-Meshari (2005).

1. Split the laboratory plus fraction to SCN (single carbon number) groups, usually up to SCN 44; the last component will be C_{45+}.
2. Use correlations which are usually functions of normal boiling point temperature and specific gravity, to estimate the critical properties and acentric factor for each SCN group.
3. Match the saturation pressure at reservoir temperature using the extended composition.
4. Group SCN to MCN groups
5. Assign critical properties and acentric factor for each MCN group
6. Match the saturation pressure at reservoir temperature using the grouped composition
7. Match the volumetric data by regression of the EOS parameter

This step itself only a guide, which in the end the implementation is based on the available data and the simulator that is used.

2.2.2 Reservoir Modeling

Building reservoir model also needs to be concerned, as the reservoir that will be represented needs to be a good CO_2 EOR candidate. For this, screening criteria introduced by

Al Adasani and Bai (2011) are used. **Table 2.3** listed the screening criteria for miscible CO_2 injection.

Table 2.3: CO_2 EOR screening criteria (Al Adasani and Bai, 2011).

Reservoir Characteristics	Range values	Average
Gravity (°API)	28 - 45	37
Viscosity (cp)	0 - 35	2.1
Porosity (%)	3 - 37	14.8
Oil Saturation (%)	15 - 89	46
Permeability (md)	1.5 - 4,500	201.1
Depth	1,500 - 13,365	6,171.2
Temperature (°F)	82 - 250	136.2
Thickness	Wide range	
Formation Type	Sandstone/Carbonate	

Available open-source or public dataset can be considered if the model that will be built wants to represent the real field case. Datasets that Sintef published for the MRST toolbox in MATLAB are available. The geological realization shown in **Figure 2.4**. Each model has its own properties and fluid inside the datasets.

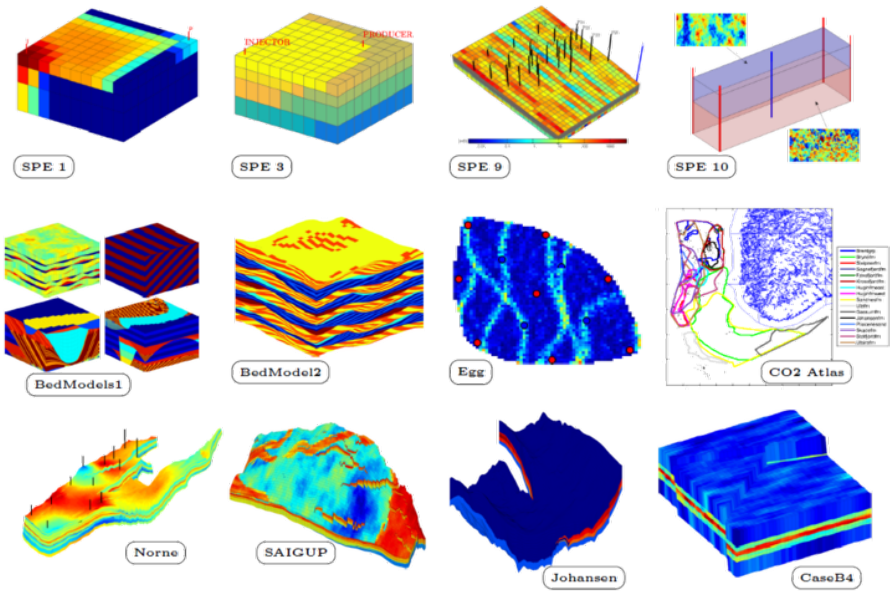


Figure 2.4: Available model datasets (Sintef, 2020).

Creating a new static model without any geology, geophysical and geostatistics background will yield an unrepresentative geological model. The best shot to be done for making a synthetic case is by using pre-made mode. Choosing the right model to be used,

considering the study goal and time factor is crucial.

2.3 Smart Proxy Model

Reflecting on the design steps in the previous section, we know that we ought to do optimization. Doing this, even using the available advanced optimization algorithm will yield numerous amount of runs to reach the objective. Solving a complex reservoir model for this will definitely cost a huge amount of time. To overcome this, Smart Proxy Model produced to tackle the limitation. Smart Proxy Model (SPM), or commonly called Surrogate Reservoir Model (SRM) is a prototype of the full fluid models with high accuracy yet shorter amount of time needed compared to reservoir simulation (Mohaghegh et al., 2009).

SPM studies already started in 2000, where some successful results already published, either in form of journals, papers, or dissertations. **Figure 2.5** shows one of the results of SPM. This fine-scale grid-based SPM models the CO_2 injection performance for one layer of field-scale reservoir. It can be seen that SPM shows good performance to simulate the trained dynamic model.

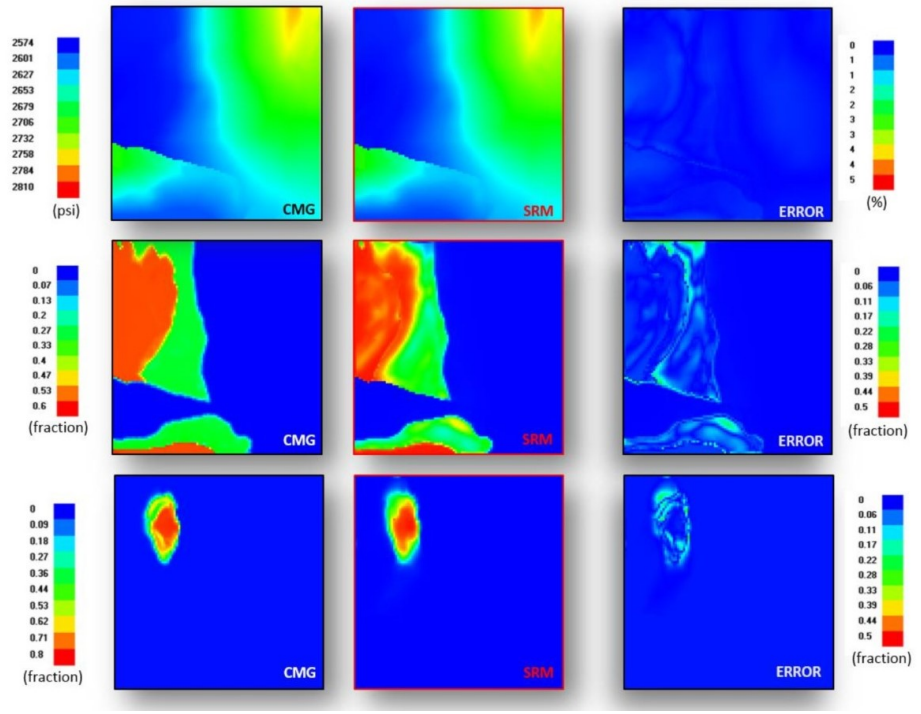


Figure 2.5: SPM results for grid-based SPM (Amini, 2015).

Why and when SPM development started? The time when the boom in artificial intelli-

gence (AI) study is the answer. The main idea to build an SPM is by using generated data obtained from conventional reservoir modeling as input data. Then, with help of machine learning, it will create a model which trained based on our input to predict the preferred output.

When will SPM be handy? The answer to this is when simultaneous runs needed, especially on a high complexity model. High time and computational load will be needed for that goal. We know that reservoir simulations are based on efficient algorithms for solving partial differential equations, and yet it still takes a lot of time. The next question is how to build SPM? The development of SPM can be partitioned into four big stages, listed below.

1. Objective of the study
2. Data extraction, both static and dynamic data which then stored in a temporal dataset for neural network data training
3. Neural network design and training
4. Validation and blind testing of neural network generated

The first stage is the study objective. This will yield to what type of SPM needs to be built. If we try to differentiate types of SPM based on the proxy scale, we can differentiate them into three SPM:

1. Field-based SPM
   This SPM is the simplest SPM, which reflects the current study being done. Usually, this model performed only to predict the production profile from one field. One geological realization used to generate the dataset, while all synthetic data doesn't need to be used as input parameters.

   The usage of this model itself quite specific, where the position of the well is not allowed to be changed as this resulting in different behavior of dynamic results. When it comes to helping the optimization problem, parameters that can be optimized using this model is only the external-related parameters (field level), such as injection rate, injection duration, and start time.

2. Well-based SPM
   When different reservoir parameters or operational constraints on the well level, this SPM will be enough to reflect the reservoir model. Here more data introduced to the reservoir, and more output types can be extracted from the SPM.

   More studies can be done using this SPM, such as individual well injection rate, bottom-hole pressure, and well positioning. This SPM also unlocks the possibility for sensitivity analysis, history matching, and uncertainty optimization.

3. Grid-based SPM
   Able to mimic the smallest scale of reservoir model which are grids, this SPM helps to estimate the properties in space and time domain. All properties that want to be modeled can be estimated using this SPM, such as pressure, saturation, or even composition. All studies that can be performed in the previous SPM types can be performed in this model also.

The cost to build this model is quite high, especially for the dataset memory. The high amount of input parameters, with countless grid data generated for every timestep, may be exhaustive. But this can be tackled with good temporal dataset management. The output of the SPM is very useful when flood front is important to be known (especially for risk assessment) (Mohaghegh, 2007).

Different SPM needs different input which, again, based on the purpose of the SPM. **Table 2.4** describes some inputs that can be used as baseline when building CO_2 SPM.

Table 2.4: Needed input for each type of SPM.

Data	Grid-based*		Well-based*		Field-based	
	Property	**Domain**	**Property**	**Domain**	**Property**	**Domain**
Static	Grid Type	Grid	Drainage Area	Well	No input needed here (constant geological/ static condition)	
	Location (i, j, k, Long, Lat)	Grid/Tier	Location (i, j, k, Long, Lat)	Well		
	Thickness	Grid	Thickness	Tier		
	Porosity	Grid	Porosity	Tier		
	Permeability (x,y,z)	Grid	Permeability	Tier		
	Grid top	Grid/Tier	Grid Top	Tier		
	Distance to boundary	Grid/Tier	Distance to boundary	Well		
Dynamic	Time		Time		Time	
	Pressure	Grid/Tier	Pressure	Tier		
	Saturation	Grid/Tier	Saturation	Tier		
	CO_2 Mole Fraction	Grid/Tier	CO_2 Mole Fraction	Tier		
	COW BHP	Well	COW BHP	Well		
	COW Amount of Prod/Inj	Well	COW Amount of Prod/Inj	Well		
	Amount of Prod/Inj	Field	Amount of Prod/Inj	Field	Amount of Prod/Inj	Field

*Gholami (2014)

As shown in the table above, the same type of input data even needs to be extracted from different domains. There are four 'levels' of the domain as mentioned. The field is the value we obtained at the field level, such as production rate. Well is the value we obtained or determined for each well we used, such as BHP, injection, and production rate. Tier, as illustrated in **Figure 2.6**, is an upscaling system that predefined to consider the impact of the surrounding grid blocks. The smallest level is the grid, which is the value from each grid block. All values are extracted for each timestep.

After knowing which data need to be collected for a preferred type of SPM, data extraction can be done. This process can be done using an available conventional reservoir simulator. The question that will be popping up here is picking run samples for neural network training. For this part and building and validating neural network will be discussed in the next section since it overlaps each other.

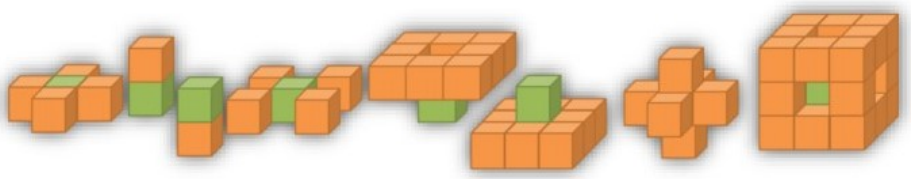


Figure 2.6: Tiering system (Gholami, 2014).

2.4 Artificial Intelligence

Before moving up to the details of building the proxy model, first, the definition should be clear. There are a lot of definitions and terms used for this. Terms that usually come up related to this are artificial intelligence (AI), computational intelligence (CI), machine learning (ML), deep learning, and many other terms. But this is not important for this study.

Until now, this definition used for the definition for explaining artificial intelligence:

> *"the study of intelligent agents, any device that perceives its environment and takes actions that maximize its chance of successfully achieving its goals"*
> *- Poole et al. (1998)*

Other than this, the term machine learning also can be introduced. Machine learning is the learning process done by machines on their own without being programmed explicitly. Two scopes of machine learning based on its learning algorithm is supervised and unsupervised learning.

- Supervised Learning
  Machine trained using well-labeled, correct data sets. Given based on labeled input data, machine trained to produces correct output defined in the dataset. Two common algorithms used in this method are classification and regression.

- Unsupervised Learning
  Machine learning that trained using undetected patterns of the dataset. Unlike supervised, no pre-existing labels attached in the input, yet machine trained to act on the information with minimum guidance. Two common categories for this algorithm are clustering and association.

As explained in the previous section, SPM development can be categorized as supervised learning. The data labeling as input value (such as porosity, permeability, and any other input) proves this. Mainly influenced from ML, **Figure 2.7** explains the most common workflow for developing SPM. This workflow has almost the same approach when compared to most proxy development study.

A lot of methods can be performed to build a proxy model. The simplest one is the polynomial regression. Other than that, multivariate kriging models, thin-pate splines models, or artificial neural networks can be used. But, for problems in which interactions between input parameters and the output are unknown or undefined, the first best shot is using a neural network.

2.4.1 Design of Experiments

Knowing the type of machine learning that needs to be built and where and how to extract the needed input-output datasets barely copes half of SRM. As we moved on, the question popping up now will be 'how many runs will be representative of the proxy?', or 'must I run as much as possible?'.

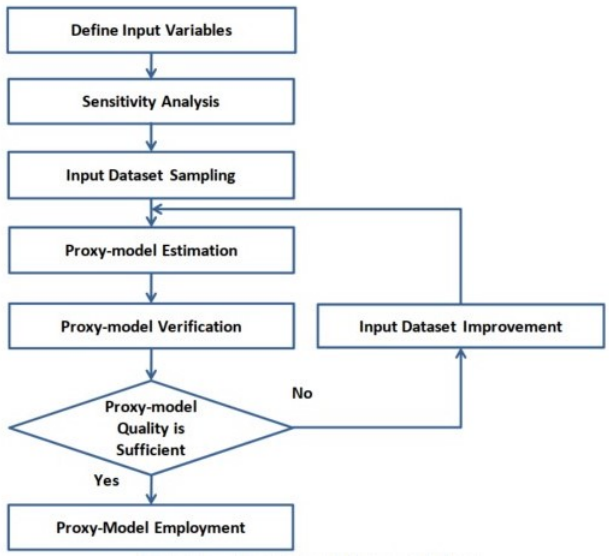


Figure 2.7: SPM workflow (Zubarev et al., 2009)

There is no correct answer to this. But, this problem can be approached using several statistical approaches, called Design of Experiments (DoE). This branch of statistics deals with planning and analyzing to evaluate factors that controlling groups or parameters. We use DoE when we suspect several parameters affecting the outcome/result of output(s). Several types of DoE that can be used are:

- Full Factorial Designs
- Fractional Factorial Designs
- Response Surface Designs
- D-Optimal Designs
- Latin Hypercube Designs
- Quasi-Random Designs

Each method has its own way to do data sampling. **Figure 2.8** illustrated how the data sampling performed for different type of DoE. Again, the time cost to generate the data sampling itself needs to be considered. Doing full factorial seems to be possible if working with a small amount of study parameter, but as the dimension increases, it will become a problem.

After choosing the DoE to be performed, the generated input-output datasets then can be analyzed. Working with a lot of parameters will be hard later when moving to network construction. Training inputs with less influence on the output will burden the proxy and resulting in a reduction of the proxy's accuracy. One of the methods, if needed, is principal component analysis (PCA).

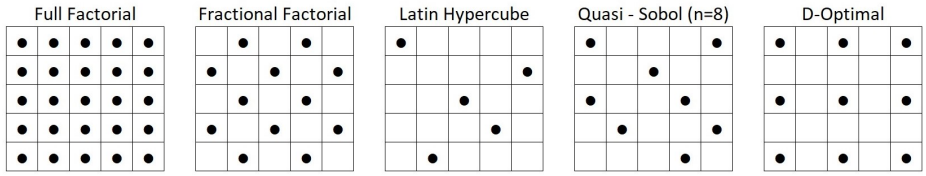


Figure 2.8: Data sampling for different DoE

2.4.2 Artificial Neural Networks

Artificial Neural Networks (ANN) is a computational model inspired by the human brain and nervous system. This model able to learn from a given combination of input and output and produces a model that reflects the relationship between them. The difference between other proxy development methods is due to the complexity of ANN, it can yield a better model compared to the other methods.

Several terms that commonly used in ANN are architecture, topology, nodes, layers, and weights. Architecture is the whole system of ANN, which constructed from layers structured parallelly, which referred as a topology. Inside each layer, there are nodes which are the calculation units of ANN. Weights assigned for each layer from each node. Inside the nodes, an activation function is assigned after each node being calculated to generate the output. Illustration of ANN shown in **Figure 2.9**.

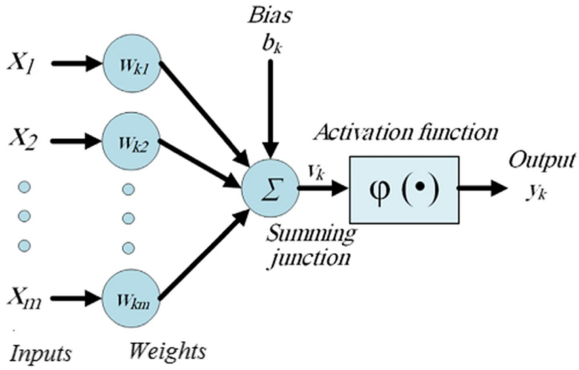


Figure 2.9: ANN structure (Mohamed, 2019)

Using the data obtained with the pre-defined architecture, the learning process can be started. Two processes that will be introduced here are forward propagation and back-propagation. **Figure 2.10** illustrate both learning method. Before the learning process started, the activation function and the weight initial value also need to be defined.

First, forward propagation occurs. The network moves from input to output, calculating based on the given weights. The values then processed and calculated on the receiving neurons. While passing through neurons, the activation functions activated to transform the calculated value obtained by the neuron. This transformation then being sent to the

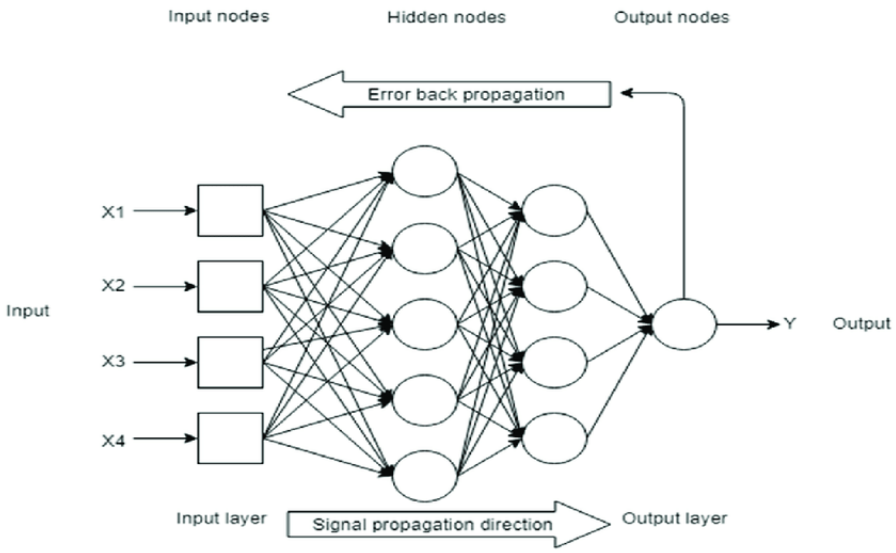


Figure 2.10: ANN learning method (Azlah et al., 2019)

next layer. When all the layers passed, the final layer, with activation function defined inside the neurons will perform the final calculation. The result obtained in the output layer then noted as the result of neural network prediction.

Of course, it won't instantly resulting in the exact output we have in the dataset. Here, the loss function introduced. The loss function will estimate the error or loss and compare it to the correct result we have on our training data. It is expected to have zero loss, which means that no difference between ANN result with correct output, if we want to make a perfect model. To minimize the error, backpropagation starts.

From the output layer, the loss will be sent back through all neurons in the previous layer (hidden layer). This information will only be received as a fraction of the total signal of the loss. This process will be based on the relative contribution of each neuron to the output layer. The same process then will be continued to the next previous layer, until it reaches back the input layer. Then, as the process is done, from these two processes, the weights for all connections of neurons will be updated. The process ends until the error satisfies the limit.

A lot of backpropagation algorithm can be used. Some of them are Levenberg-Marquardt algorithm, Polak-Ribiere conjugate gradient, Powell-Beale conjugate graduate, and quasi-Newton backpropagation. Activation functions also have different forms, the most commonly used are sigmoid, while the other are threshold, Gaussian, linear, and ReLU function.

2.4.3 Optimization Algorithm

The proxy model usually developed for doing sensitivity analysis which involves several variables, probabilistic forecasting, history matching, risk analysis, field development planning, and optimization. What will be studied in this project is optimization, as one of the engineering design. This needs to be performed as one of the main decision tools for 'go or no go' project decisions.

Optimization itself is an act of making the most effective use of a situation or resource. In mathematical modeling, optimization is a method of finding preferred extreme values (maxima or minima) of a function based on a set of input values which then computed to the function. Mathematically approached, it can be written as shown in **Equation** (2.1).

$$max_{x \in \mathbb{R}} \, f(x) \tag{2.1}$$

For a simple optimization problem as written above, it might be as simple as applying a common bracketed search, golden ratio, or Fibonacci search. But, the problem that we have in field development and production optimization might be a mix of nonlinear functions, with defined constraints and high dimensional space.

It will be hard for those gradient-based search to solve the problem, especially its behavior that easily converged to local optima. Hence a lot of global optimization algorithms developed to conquer this limitation. There is a lot of group of optimization algorithm already, such as swarm-based optimization, evolutionary algorithm, reactive search algorithm, etc. In this study, two of the most common optimization algorithm will be introduced and used, which are Genetic Algorithm (GA) and Particle Swarm (PS) optimization algorithm.

Genetic Algorithm

Genetic algorithm (GA) is a stochastic optimization algorithm in which a population of candidate solutions of an optimization problem is evolved to the better solution. This algorithm was developed by biology-inspired behavior such as crossover, mutation and selection.

At first, a population of solutions randomly scattered throughout the solution space. Each candidate has properties that later can be altered. These properties can be defined in binary, permutation, or real number encoding. After that, each individual being evaluated to the objective function. The values which each individual then being ranked from best to worst. Here, the selection, crossover, and mutation process will occur, yielding a new generation which properties are selected from those high-rank generation's properties through selection and crossover. Mutation helps to prevent the solution converges to local optima.

The process itself then being iterated. Each iteration then yield a new generation. When it reaches the targeted fitness level or reached the maximum generations, the algorithm will stop. The final individual with the best solution in the solution space then being selected as the solution of the optimization problem. This algorithm workflow illustrated in **Figure 2.11**.

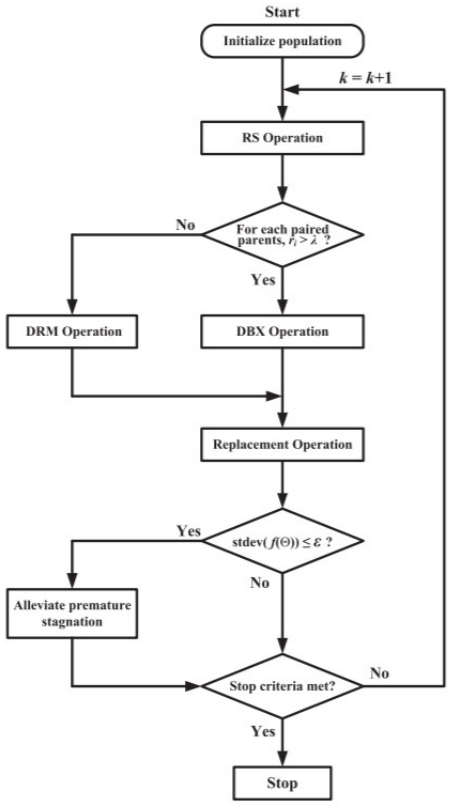


Figure 2.11: Genetic algorithm workflow (Chuang et al., 2015)

When coming to the algorithm design, those processes are controlled with several parameters. Some parameters that affect the performance of this optimization algorithm are listed below.

1. Chromosomes
2. Population size
3. Number of generations
4. Type of selection
5. Crossover's probability
6. Mutation's probability
7. Type of mutation and crossover
8. Type of replacement

Particle Swarm Optimization

Particle swarm optimization algorithm (PS) is one of swarm-based optimization methods. Other algorithms that belong to the same group are social cognitive optimization, multi-swarm optimization, and ant colony optimization. This algorithm was developed based on the insects' social behavior when they're trying to find foods for their swarm. Swarm

then changes their behavior based on their own performance and the other members experiences.

The method works by scattering particles throughout the function space, then moving to find the optimum solution for each iteration step. For each iteration, learning factors captures each particle's movement when trying to find the optimum solution. The movement then being lead by each particles' best-known position and entire particles' best-known position. Every iteration will improve this position which will move the swarms to the improved position. The iteration is done in hope that the solution gets discovered, although it is not guaranteed. The workflow of this experiment shown in **Figure 2.12**.

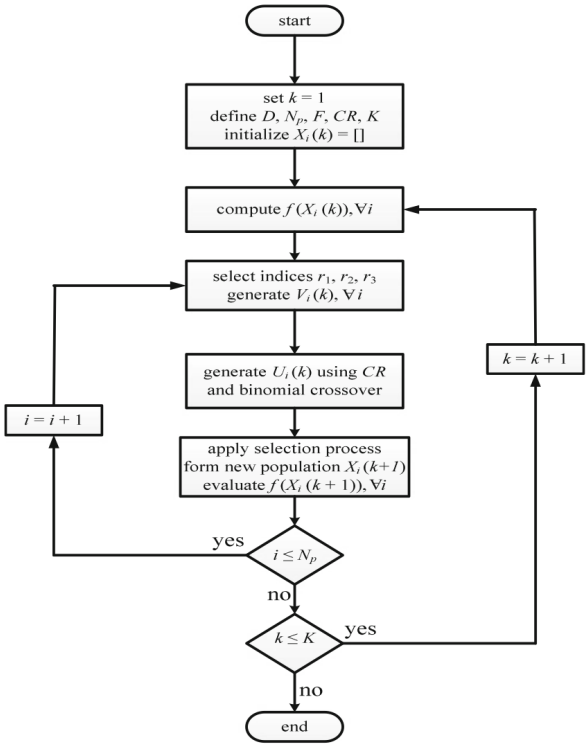


Figure 2.12: Particle swarm workflow (Nwankwor et al., 2013)

Chapter 3

Methodology, Problem and Model Description

3.1 Workflow

This project workflow is shown in **Figure 3.1**. The workflow is mainly inspired from Gholami (2014), a previous study related to surrogate reservoir model development. Available fluid and static model data collected from references are used for reservoir model data. A depletion study then performed to see the field performance.

As optimization objective already defined, design of experiments then performed. When needed, data sampling can be performed. Dataset obtained then analyzed to see the impacting input for the neural network (if needed). Then, after all the data sampling process, construction of the neural network then can be done. After training and validation, then it can be tested with blind data, to see the performance.

Right after the network performance is validated and approved, optimization can be done using that network. Results obtained then can be concluded as reservoir model optimization result, as model represented with the neural network already.

3.2 Software

There are softwares used to develop the proxy and to do the optimization. All needed licenses are provided by NTNU.

CMG - Winprop

CMG (Computer Modelling Group) is one of the commercial software which widely used in the industry. CMG itself covers all aspects of reservoir flow and advanced processes

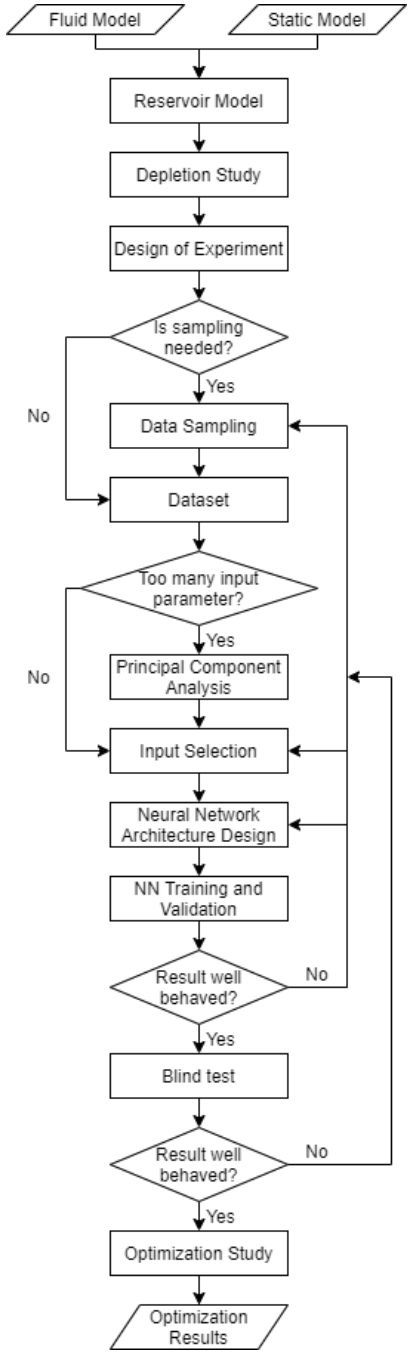


Figure 3.1: Project workflow.

modeling (CMG, 2020). Winprop is one of the packages inside CMG software. This package helps to model the fluid property and characterization tool.

Winprop itself used as fluid model creation for this study, which later combined with PVTp to confirm the generated model. As Eclipse used as the main simulator for this model, the result later being converted to Eclipse compositional fluid model. Several functions used in this study are PVT tests, regression, splitting, and lumping. In this study, version 2019 was used.

PVTp

PVTp as one of the available fluid modeling software are also used. PVT analysis and EOS creation in this software are based on lab experiments. PVTp enables their user to perform these tasks by matching compositions to PVT data available (PETEX, 2020). Using PVTp helps to convert CMG results to Eclipse format and extra fitting, as different scripting used on both. IPM 11 (2019) version used in this study.

ECLIPSE - E300

ECLIPSE is used in this study as it's one of the most commercially used reservoir simulators in the industry. ECLIPSE integrates static and dynamic model process into a seamless workflow (Schlumberger, 2020a). Developing using Eclipse itself rather than CMG-Builder is based on data availability which in Eclipse format. Converting might results in glitches and unexpected errors while processing.

E300 environment used for the study since the changes in the fluid composition will be important for EOR study, especially gas injection. Changes in permeability curve and endpoints can also be implemented in this environment. Data extracted for the proxy model in this study all generated using this. ECLIPSE 2019 used for this study.

MATLAB

MATLAB combines a desktop environment tuned for iterative analysis and design processes with a programming language that expresses matrix and array mathematics directly (MATLAB, 2020). Design of experiments, neural network construction, and optimization developed using this. Other than common toolbox like statistical and mathematical packages, some advanced packages are used, Deep Learning Toolbox and Optimization Toolbox are one of them. MATLAB R2020b used in this study.

3.3 Reservoir Model Description

Limited amount of compositional model data encountered in this study. Therefore, a synthetic model constructed using the available static model and fluid model found in books and papers.

3.3.1 Fluid Model

This PVT model was constructed from data provided in Phase Behavior book (Whitson et al., 2000) named Good Oil Co. Well 4 Oil (Chapter 6). It is synthetic PVT data which contains common PVT tests results. The one which available are fluid composition, CCE test, DL test, and separator test. Other needed tests that need to be used as CO_2 injection data later created numerically, such as slim tube test.

As the overview, this fluid sample itself is a light oil with API around 40. The viscosity itself is low, varies around 1 cp. It fulfills the common screening criteria for CO_2-based EOR (Al Adasani and Bai, 2011), which values are within its ranges (28-45°API and 0-35 cp). The reservoir condition of this fluid sample is 4,100 psig at 8,692 ft with reservoir temperature 220°F. The saturation pressure is 2,620 psig.

3.3.2 Static Model

Free-distributed reservoir models mostly made with black oil for their fluid model. As this study is a CO_2 injection study, a representative model needed as a study material. Therefore, Egg Model will be used for this study. Egg Model is a synthetic reservoir model with three-dimensional realizations of channeled oil reservoir (Jansen et al., 2014). The model itself was constructed to study water flooding but will be used here to model the CO_2 flooding.

The reservoir modelled with 60 x 60 x 7 grid cells which 18,553 cells are active. The one which is used in this study is the first permeability realization. There are several modifications done to make the model reflects a small field case. Detailed information about the reservoir is written below. Other than the porosity (constant 0.2 throughout the grids), permeability (as shown in **Figure 3.2**), and active grids, other values are being modified to fit with the fluid model condition. The model modification and construction will be explained in the next chapter.

3.4 Optimization Problem

Different optimization problems will yield different types of smart proxy models that should be constructed in a study. As there is no available geological realization other than the current model (even though there is other permeability realization, but no for the other static model parameter), data will only be sufficient for creating a field-based proxy model for now. No economical constraints used for this study.

3.4.1 Objectives

The current study is expected to maximize the FORE (Field Oil Recovery Efficiency). Parameters to be optimized in this study itself defined below. All optimization parameters are determined after the reservoir model constructed, which the model development process explained in the next chapter.

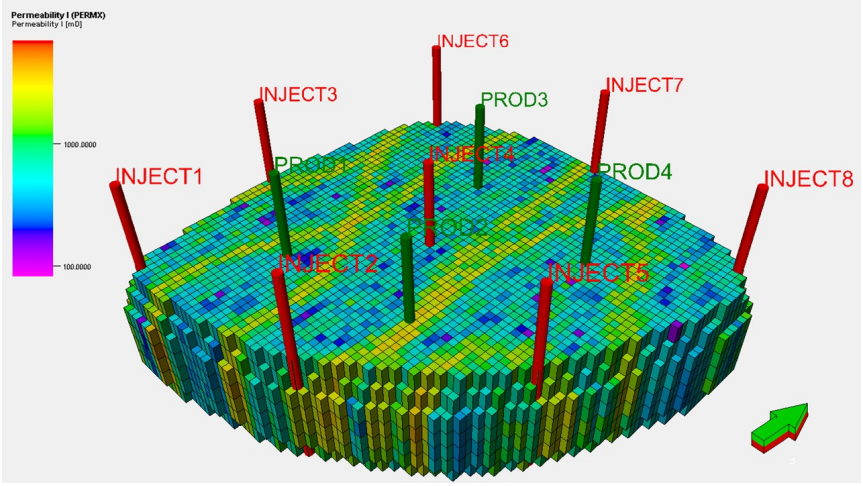


Figure 3.2: Static model permeability distribution and initial well placement.

- Injection rate
  Injection rate of our injection fluid affects a lot in the EOR study, which mainly will affect the miscibility process in our case. The range which will be studied is 1 to 2 million sm^3/day. The rate itself is set as field injection rate, which later being divided equally for all wells. Higher injection rate being ignored to make it more realistic with CO_2 availability as economic parameters being ignored.

- Injection fluid composition
  As we know that CO_2 is usually limited and far from the field, methane and ethane are commonly mixed with it. They are available as being produced from the reservoir. In this study, the injection fluid composition is limited to those three. The sampling itself is ranged between 80-100% of CO_2, 0-20% of methane, and 0-20% of ethane.

- Pore volume injected
  Rather than common flooding study results in CO_2 flooding study (1.2 PV), here larger interval used. We studied the best amount of pore volume injected in intervals of 1 and 2 pore volumes. Higher larger pore volumes will yield a larger injection time. As there are no economic limits in this study, higher amount of pore volume injected is neglected.

- Injection start time
  The start time chosen based on the depletion performance. As explained in the previous section, three points chosen are when the reservoir pressure reaches saturation pressure (6 months), when the plateau ends (3.75 years), and when the well dies (5 years, all based on production constraints). All deductions shown on **Figure 3.3**. This parameter itself studied as discrete points, which later can be concluded as the best time to start the injection.

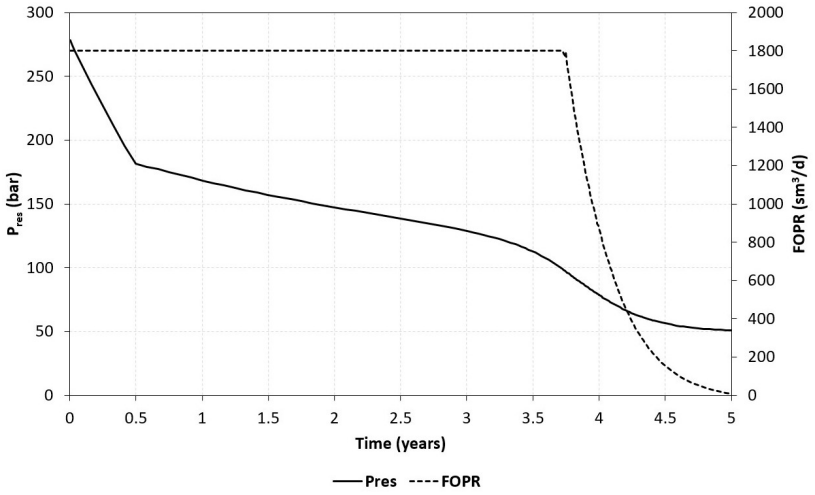


Figure 3.3: Depletion study results.

3.4.2 Constraints

Constraints below also determined based on the field condition after the reservoir model constructed. Common technical constraints are used for this. which are BHP of producer wells (50 bar), BHP of injection wells (300 bar), and maximum field liquid production rate (1800 sm^3/day).

3.5 SPM and Artificial Neural Network

Based on the determined optimization objective, field-based SPM will be enough to perform the task. There are a lot of neural network structures and modifications inside our ANN. But, for the base of this study, several templates that is given will be used at first. The limitation for this SPM study is no deep neural network structure and architecture will be studied here as the focus only to build a smart proxy model only.

The structure that will be used in this study is deep feed-forward (DFF) ANN. **Figure 2.10** shows the structure of this type of ANN. Other than that, the activation function will be limited also, which only using logsig function. This function one of the activation function available and used in ANN. Amar et al. (2018) success in making ANN for WAG case will be used as the base for our ANN settings.

3.6 Optimization Algorithm

The optimization will be performed either using GA or PS due to its availability in MATLAB. The study of which algorithm to be used and their parameters are not being focused for this study, which resulting in room for improvement.

Chapter 4

Results

In this chapter, the chronological process of how the study performed based on workflow provided in section 3.1 is described.

4.1 Reservoir Model Development

As mentioned before, there is a lack of a reservoir model with a compositional fluid model available during the study. Here then described how the synthetic reservoir model being developed by combining Good Oil fluid data with Egg Model.

4.1.1 Fluid Modeling

The fluid itself is modeled using PVTp at first, where all the available data inputted to the software. The sequence in which the modeling being performed, simplifying what Al-Meshari (2005) explains are:

QC — Split — Regress — Lump — Regress — Additional PVT Test — Slim Tube Test

As no additional nor advanced PVT test data available to be matched, they will be generated by the simulator. The generated results itself obtained after the EOS able to fit the available data and already lumped beforehand. This was done to reduce the amount of computation needed later in the dynamic model.

PVT Data Quality Check

Based on available data, several quality checks can be performed. This will help later during the EOS tuning, so outliers/non-physical data that counts as noise won't interfere with the modeling results. Basic QC performed in this study, which are physical/logical behavior of PVT data, material balance, and thermodynamic consistency. Detailed results shown in **Appendix A.1** while the summary are tabulated in **Table 4.1**.

Table 4.1: PVT QC summary

Data	Comments	Acceptability
Recorded Sample	Thermodynamic consistency achieved Acceptable composition	Yes
CCE Test	Linearity of Y function Linearity of isothermal compressibility	Yes
DL	No sufficient data for material balance	Yes*
Separator	Material balance satisfied	Yes
Viscosity	Logical behavior	Yes

*can't be left out since limited data available

EOS Tuning

The process of pseudo component splitting, regression, and lumping will be covered here. The process itself is using PVTp at first because it can converge way faster than Winprop. EOS that being used are Peng-Robinson (PR). Peng-Robinson and Soave-Redlich-Kwong (SRK) both provide the same accuracy, and volume translation helps PR to perform well to predict liquid volumes (Whitson et al., 2000).

After reaching a somewhat satisfying result, Winprop used to do more matching, as more parameters can be regressed in Winprop. Pseudocomponent (C_{7+}) being split into three pseudo components, because after trial, splitting to 44 SCN are insufficient, yet three yields satisfying result. This was done to help the matching to be faster. The parameters that are regressed for the EOS tuning are critical pressure (P_c), critical temperature (T_c), critical volume (V_c), volume shift, Acentric Factor (AF), and Binary Interaction Parameters (BIP).

Splitting then resulting in 15 components in the system. Lumping then done after the regression. Final fluid components then reduced to seven, because five to eight should be sufficient to simulate reservoir process (Whitson et al., 2000). After lumping, then regression was done to re-fit the deflection due to the lumping process. Phase envelope acquired after EOS tuning are shown in **Figure 4.1** with final fluid composition tabulated in **Table 4.2**. Complete tuning results which contains all PVT Tests attached in **Appendix A.2**.

Table 4.2: Final fluid composition

Composition	mol%
N_2	0.16
CO_2	0.91
CH_4	36.47
C_2H_6	9.67
C_3H_8	6.95
C_4-C_6	12.55
C_{7+}	33.29
Sum	100.00

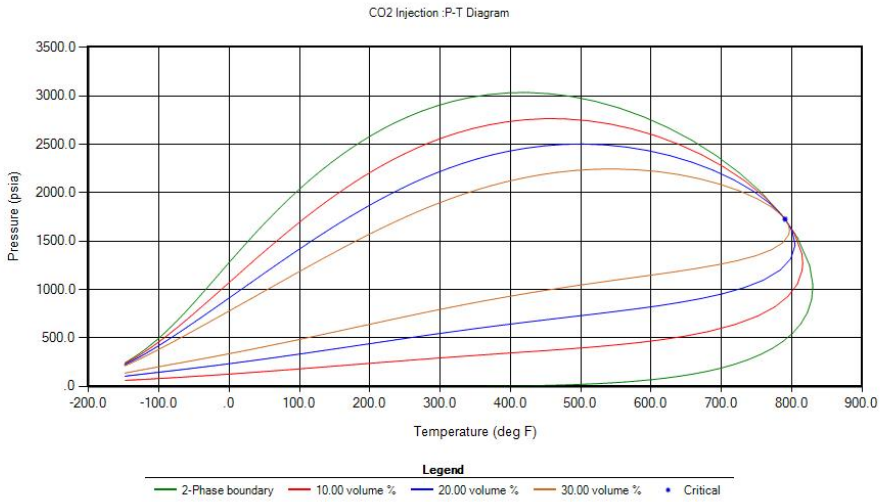


Figure 4.1: Phase envelope after tuning.

Additional PVT Test

Common tests that being done as a preliminary study in CO_2 injection study other than slim tube test are swelling test and asphaltene onset precipitation. Those two itself quite crucial especially when the flooding started as CO_2 mixes with reservoir fluids, swells the fluid and asphaltene precipitation might occurs soon after the breakthrough (Jarrell et al., 2002). This might lead to production tools problem and lowers the well and reservoir performance.

Good PVT data needed when those two will be studied. As the available data insufficient to be utilized for these additional PVT tests, hence those tests are ignored. Even though it can be built using the simulators, verification needed for its performance by match the results with laboratory tests, as tuning might change the real fluid behavior.

Slim Tube Test

The data generated in our PVT model then being used for slim-tube numerical modeling. Since no Minimum Miscibility Pressure (MMP) data provided as a reference, then the result obtained here can't be compared with any available information. For making the numerical model, the configuration provided by Eclipse is used as a reference. Grid size, injection rate, and timesteps are reduced to minimize the numerical dispersion as explained by Vulin et al. (2018).

The definition of MMP used here itself is the same as real slim tube test, where numerically obtained cumulative oil recovery factors at 1.2 cumulative pore volumes injected are plotted against the injection pressure for obtaining the MMP from the intersection point of two trend lines on the cumulative oil recovery versus injection pressure plot (Saini, 2019).

A unique feature observed after the fluid reaches its saturation pressure. This small deflection observed but neglected for now when building the trendline. The details of the injection configuration shown in **Table 4.3**, while the injection performance illustrated in **Figure 4.2**. It can be concluded that our reservoir fluid MMP, when injected with pure CO_2, is 192.3 atma (194.8 bar).

Table 4.3: Slim tube model configuration

Parameter	Values
Grid	200 x 1 x 1
Grid dimension	5 cm x 1 cm x 1 cm
Porosity	0.1
Permeability	2000 mD
Temperature	104.4°C
EOS	PR
Total PV	100 cc
Injection Rate	0.1 PV/hour

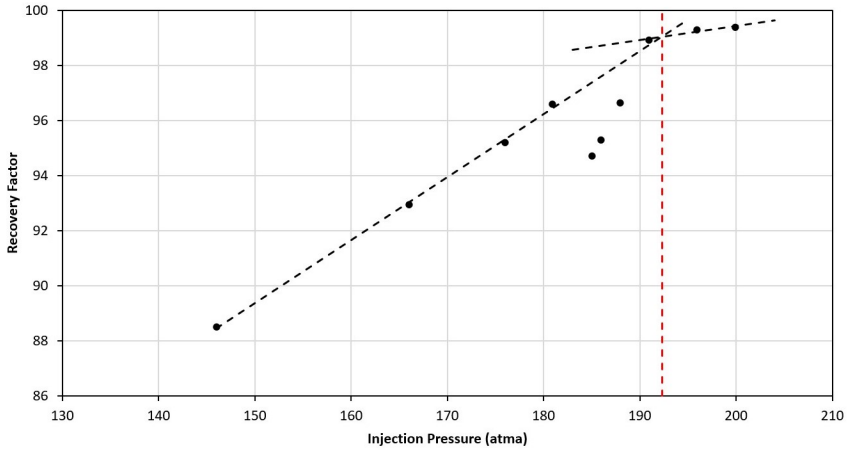


Figure 4.2: Numerical slim tube test result.

Overall, **Table 4.4** shows the details of the fluid model produced after the tuning and several additional PVT tests. The obtained model here then converted to E300 format for the study purpose.

4.1.2 Static Model Modification

After the fluid model finalized, then the fluid model used in the static model. To convey the real field situation, the grid size modified from 8m x 8m x 4m to 32m x 32m x 5m. This increases the STOOIP from 512,000 sm^3 to 10.2 million sm^3. The increase itself to make field-like production and injection scenarios such as injection rate, production rate, and plateau duration.

Table 4.4: Fluid model overview

Parameter	Values	Units
Sampling datum	2649	m
Datum pressure	283.7	bar
EOS	PR	
Components	7	
P_{sat}	182	bar
T_{res}	104.4	°C
API	41.28	°API
B_o	1.465	
GOR	750	sm^3/sm^3

More to be added, the amount of injector well is decreased. Only four injector wells used, which are INJECT1, INJECT4, INJECT6, INJECT8 (refer to **Figure 3.2**). The well selection itself is based on the position, which on the edges and the center of the reservoir. This reduction is based on logical reasoning which it is not that common to have many injector wells in such a small field. We tried to portray the real field condition in the study.

As Egg Model doesn't consist of oil-gas relative permeability curve, the relative permeability curve then all modified for this study. Sand preset available in PETREL is used (Schlumberger, 2020b). The fluid compressibility also adjusted to the compressibility data provided in the PVT data.

4.1.3 Dynamic Model Construction

The constructed model then being initialized. Based on the data shown in **Table 4.4**, the sampling datum and pressure being used for the model equilibrium. The gas-oil contact (GOC) and oil-water contact (WOC) placed way higher and way lower than the reservoir datum. Doing this results in a fully saturated oil reservoir with no gas cap with its initial water saturation as initial saturation. No aquifer applied in the model as the purpose of the model is to see the flooding performance without any external energy support applied except the injection fluid. All data after being modified are listed in **Table 4.5**.

4.1.4 Depletion Study

After initialized, then the depletion behavior of the reservoir being studied. As there are no external forces such as aquifer in the reservoir, the pressure sharply decreases right after the well being opened. To make a realistic condition, four existing producer wells then being open and the production rate and average reservoir pressure. But, the first thing to be determined is how the field being exploited before the tertiary recovery performed.

There are two ways in which the field can be customized to represents a real case. The first one is being depleted naturally, while the second one is depleted with water injection as pressure maintenance. The second one is the most commonly applied in the field, but for this study's simplicity, the first one will be used for our depletion study.

Table 4.5: Reservoir model properties

Parameter	Value
Grid length (x and y direction)	32 m
Grid height	5 m
Porosity	0.2
Relative permeability model	Sand preset
Permeability	81 - 7000 mD
STOOIP	10.2 mil. sm^3
Initial water saturation	0.2
Top depth	2649 m
Initial reservoir pressure	283.7 bar
EOS	PR
Temperature	104.4°C
Start of simulations	1 Jan 2020
Number of injection wells	4
Number of production wells	4

The next point is to find the production constraints for the depletion study which later will be optimized with CO_2 injection. After several trial and error, 1800 sm^3/day field production rate limit is used with BHP of 50 bar as production constraints. These constraints are chosen based on its production profile throughout the year, shown in **Figure 3.3**. As we want to know when is the time which injection best to be performed, using these limit give good time points to be coded later. Also the value itself well-picked to represents a small field case. The chosen production constraints here will be used throughout the study.

4.2 Design of Experiment and Data Analysis

To make a representative proxy model that represents our reservoir model, several runs needed as input data. Design of Experiments (DoE) helps to determine how many and which data samples to be run.

4.2.1 DoE and Data Sampling

We will look into the optimization parameters defined before. As taking a high float form of number might yield computation problem (such as script making and increase of error due to rounding up values), the interval then being discretized. As there is no ground of unique probability distribution for all optimization parameters, uniform probability distribution will be used for this study.

Special case for injection gas composition. As we know that the goal is CO_2 injection, CO_2 will be the main composition, varies between 80-100%, with methane and ethane combination between 0-20%. Partitioning the change of each composition with 5% interval will yield 15 possibilities for injection gas composition parameters. Detailed parameter partitioning tabulated in **Table 4.6**.

Table 4.6: Parameter partitioning

Parameter	Code		Data Points								Probability Distribution
Field gas injection rate	q_{inj}		1	1.2	1.4	1.6	1.8	2			Uniform
Pore volume to be injected	PV_{inj}		1	1.2	1.4	1.6	1.8	2			Uniform
Injection gas composition	Comp	$\%CO_2$	100	95	95	90	90	90	85	85	Uniform
		$\%C_1$	0	5	0	10	0	5	15	0	
		$\%C_2$	0	0	5	0	10	5	0	15	
		$\%CO_2$	85	85	80	80	80	80	80		
		$\%C_1$	10	5	20	0	5	15	10		
		$\%C_2$	5	10	0	20	15	5	10		
Start time	t_{start}		0.5	3.75	5						Uniform

As the optimization problem is based on a specific synthetic field, and also as there is no other geological realization available, the proxy that can be made works only for this model. With a case defined as CO_2 continuous injection (CO_2 flooding), with trials of sensitivity analysis for each parameter, it is verified that all parameters affect the total produced oil, which is the optimization objective.

As the impact of the studied parameters confirmed to affect the study objectives, sampling can be done. Here, we definitely need a sampling strategy, since all parameters affect the production performance in unknown behavior. For this, Latin Hypercube Sampling (LHS) then used. The reason itself is the random samples generated fill the hyperplane equally while remembering samples that already taken before.

Available LHS function in MATLAB itself based on 0 to 1, which not the same range as what we have here in our optimization parameter. Then, denormalization needed to translate the results to our distribution points. The generated points then allocated to the probability tray we have in for each parameter. By doing this, no extra functions needed for our LHS design.

A code to generate the data sampling and denormalize it to our case was created using MATLAB. Ten run samples are designed for each start time, as time is the main indicator of time series data. This resulting in 30 run samples. Another three run samples also generated here, yet being random to be used as blind test data. Run samples generated tabulated in **Table 4.7**.

As the dataset already made for this study, comes to the challenge to make a way to run simultaneously based on the defined run scenarios. Here, again with MATLAB, a code made to do those task. The generated parameter then being translated to the schedule part of the main case data. The objective parameters from our DoE then being translated to:

- Injection Rate : Divided equally to all injection wells.
- Pore Volume Injected : Adjustment from surface injection rate using gas FVF of 0.02 rm^3/sm^3 (pure CO_2 mixed with our reservoir fluid has gas FVF of 0.22 rm^3/sm^3).

Table 4.7: LHS generated run samples

Run	q_{inj}	PV_{inj}	Comp ID	t_{start}	Run	q_{inj}	PV_{inj}	Comp ID	t_{start}
1	1.60	1.40	12	0.50	16	1.40	1.00	6	3.75
2	1.80	1.40	1	0.50	17	1.80	1.20	7	3.75
3	1.60	1.20	7	0.50	18	2.00	2.00	10	3.75
4	2.00	1.60	9	0.50	19	1.00	1.40	13	3.75
5	1.40	1.60	15	0.50	20	1.60	1.80	15	3.75
6	2.00	1.80	4	0.50	21	1.00	1.00	9	5.00
7	1.40	2.00	3	0.50	22	1.00	1.00	4	5.00
8	1.00	2.00	10	0.50	23	1.60	1.20	6	5.00
9	1.20	1.00	13	0.50	24	2.00	1.40	1	5.00
10	1.00	1.00	6	0.50	25	2.00	1.80	15	5.00
11	1.60	1.60	12	3.75	26	1.60	1.60	13	5.00
12	1.00	2.00	4	3.75	27	1.20	1.40	10	5.00
13	2.00	1.40	3	3.75	28	1.80	1.60	7	5.00
14	1.40	1.00	9	3.75	29	1.40	2.00	3	5.00
15	1.20	1.60	1	3.75	30	1.40	2.00	12	5.00

Then this converted to injection time using the current injection rate where float numbers ceiled to a month.

- Injection Fluid Composition : Approached based on Composition ID. (ID described in sequential as **Table 4.6**).

The approach itself is in sake of easier case generation for model running purposes. Errors might occur, especially in pore volume injected, as the gas FVF being approached and the result being ceiled. This will yield longer injection time for all cases, resulting in a higher amount of pore volume injected.

The run output then being stored in separate files for each timestep and Field Oil Production Rate (FOPR). Field Average Pressure (FPR) and Field Total Oil Produced (FOPT) also stored just in case needed. Total 7939 data points obtained from 30 run scenarios. This is way higher than the expected amount (30 cases with 1 month timestep) as we need to decrease timesteps, especially during the flooding to reach the convergence.

For information, the running time spent on running all 30 run samples is 134 minutes. Intel(R) Core(TM) i7-8565U with 8GB of RAM used for this study. For each scenario, it takes around 4.5 minutes on average.

4.2.2 Data Analysis

Obtained data then analyzed for our proxy model input. As for now, the input data still less than ten inputs, it won't be that computationally heavy for our network to be trained base on those inputs. Hence, no analysis needed to pinpoint which parameters affect greatly the output of our network and reduce the insignificant input parameter.

4.3 Proxy Model Construction

The database obtained from the data then being used as the data for our proxy building. For reminder, the current available data is all optimization parameters, timesteps, and FOPR, FOPT, and FPR for each timestep. Timesteps here are the generated timesteps from Eclipse, not what designed beforehand.

4.3.1 ANN Settings and Architecture Design

MATLAB used for building the neural network. Before moving to the architecture and settings, first data needs to be stored and managed. Generated run results from 30 run samples merged as one, tabulated in 9 columns consisting timestep, injection rate, PV injected, start time, CO_2 composition, C_1 composition, C_2 composition, FOPR before the current timestep, and FOPR in sequence. For now, FPR not being used as modeling the reservoir pressure is not the current goal of this study.

Now, the structure of our ANN will be decided. For this study, it will be based on deep feed-forward neural network only (DFF-ANN), as one of the most common structures used for the first trial. There are a lot of structure such as recurrent, radial basis function, long/short term memory, and others. Those can be the chance for improvement. A code, completed with how data managed, neural network creation, and verification was made, while the first settings are listed in **Table 4.8**.

Table 4.8: ANN initial training settings

ANN Settings	Value
Training function	Levenberg-Marquardt
Performance function	Msereg
Epoch	500
Validation check	30
Min grad	1E-10
Transfer function	Logsig
Training ratio	70
Validation ratio	15
Test ratio	15

Then, now we move to how the architecture needs to be designed. For all activation functions, logsig function used. The learning process for building ANN usually comes from trial and error. The common practice is using a number of layers based on what ANN represents listed in **Table 4.9**. AS no Based on this, two hidden layers chosen for this study.

Coming up with the number of neurons, Heaton (2020) mentioned several rules of thumb that can be used for neural network number of neurons decision. Based on this, the configuration of 8-8-5-1 used. This notation means 8 input layers with 8 nodes in the first hidden layer, 5 nodes in the second hidden layer, and 1 output layer. As the structure, architecture, and settings, the neural network then trained.

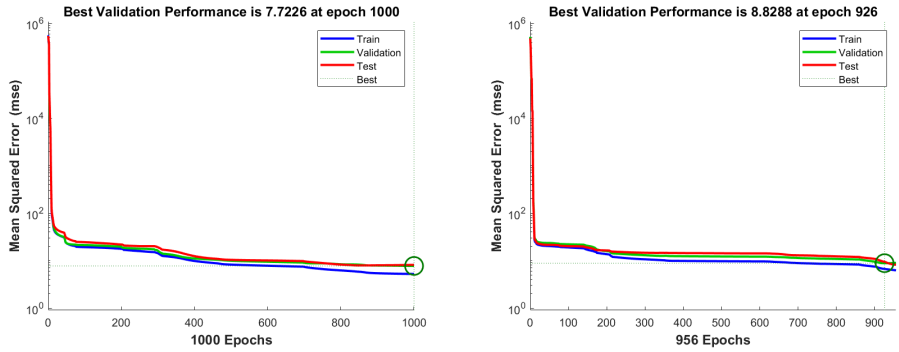
Table 4.9: ANN common number of layers rule of thumb (Heaton, 2020)

Num Hidden Layers	Result
none	Capable to represents linear functions or decisions.
1	Can represent functions with continuous mapping from one finite space to another.
2	Able to represents arbitrary decision boundaries to with rational activation functions and any smooth mapping.
>2	Learn more complex representations (i.e.automatic feature engineering).

4.3.2 Network Training, Validation and Testing

Using the proposed structure, ANN fails to learn based on the given data. The first guess is the number of epoch used. Epoch is a term that indicates the number of passes of the entire training dataset that our algorithm has completed. Two trials are done, increasing epoch to 1000 and 2500, yet the network still fails to learn the model (no network can be built based on the current proposed architecture). After looking back, the main problem might be the number of nodes, and it was decided to increase the number of neurons.

MATLAB itself recommends its users to start with ten neurons. It was decided that we need to change this configuration based on that recommendation. 8-10-8-1 then used for the network building, using 500 epochs. Again, ANN fails to be built with this setting. Then, epochs increased to 1000. And now ANN able to be generated, but based on its performance it's optimum is still at the maximum value of 1000 epochs **Figure 4.3a**.



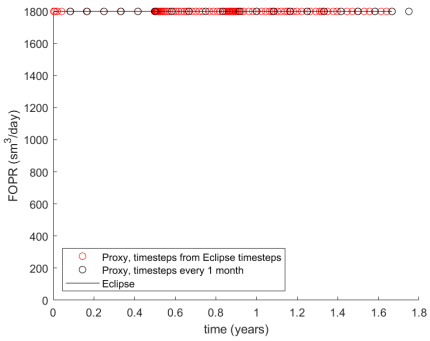
(a) 1000 epochs

(b) 2500 epochs

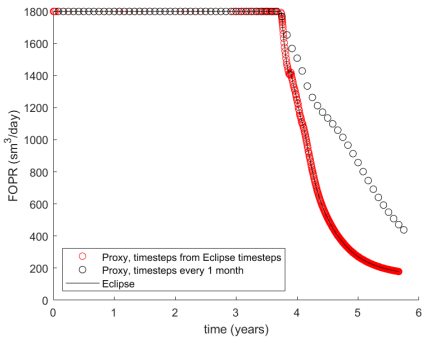
Figure 4.3: First ANN performance plot

To make sure the ANN learns and reaches its optimum performance, the epoch then increased to 2500. Now, the optimum performance is at epoch 926, shown in **Figure 4.3b**. Then the regression result studied, it seems that the network able to learn properly based on their average R square value reached 0.9999.

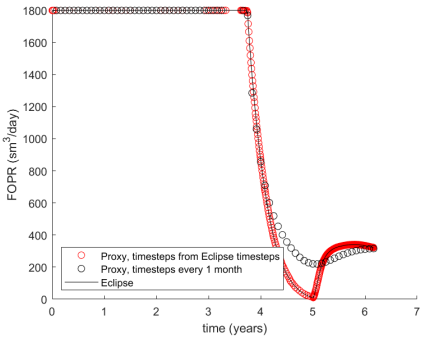
Is this a good performance or overfitting? The best way to see it is by plotting the results with the training data. We try to check the performance based on the obtained model. **Figure 4.4** shows three samples represent three different start time. The generated ANN able to model both the plateau and the decline behavior (red bullets). When we try to think later when we use the model as our SPM, the timesteps we used here will be a little bit off. The generated timesteps from Eclipse are based on the convergence when solving the partial differential iteration. This makes the timesteps generated way more than the preferred amount (preferred timestep = 1 month). For this, we check how the proxy performed with 1 month timestep. The result (black) shows that it shows deflection during the decline phase.



(a) Case 1



(b) Case 19



(c) Case 28

Figure 4.4: ANN results

To solve this problem, RPTONLY keyword added to the Eclipse file. The run results for training data (still using LHS generated run samples written at **Table 4.7**) reduced to 1615 data points. Then, using the same settings, ANN trained to learn this data with structure 8-8-5-1. The performance using this architecture is 0.9999 for both training, validation, and test data, shown in **Figure 4.5**.

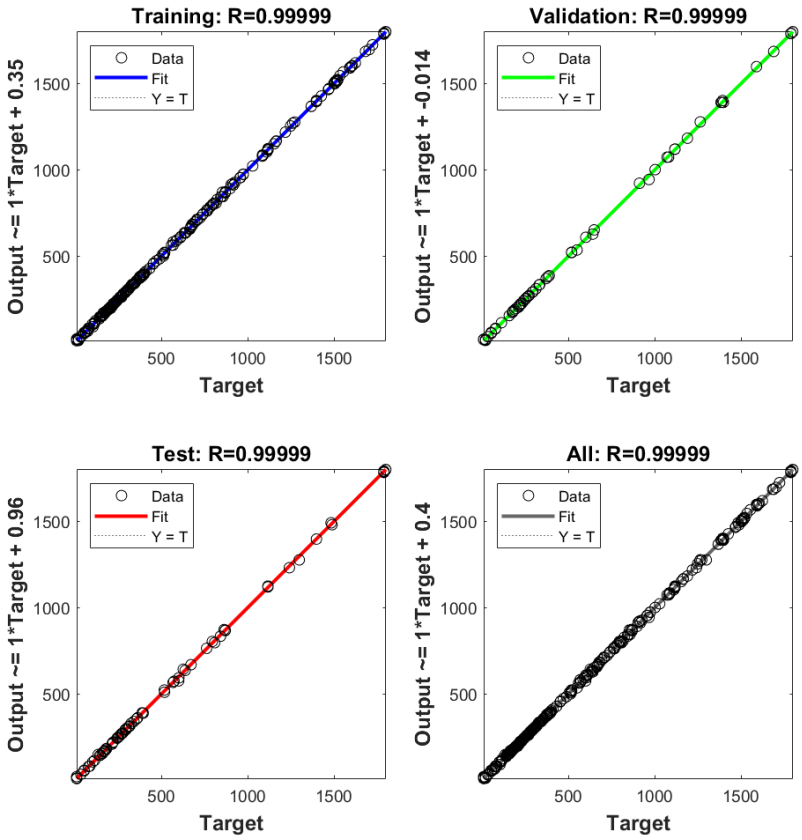


Figure 4.5: Generated ANN performance results.

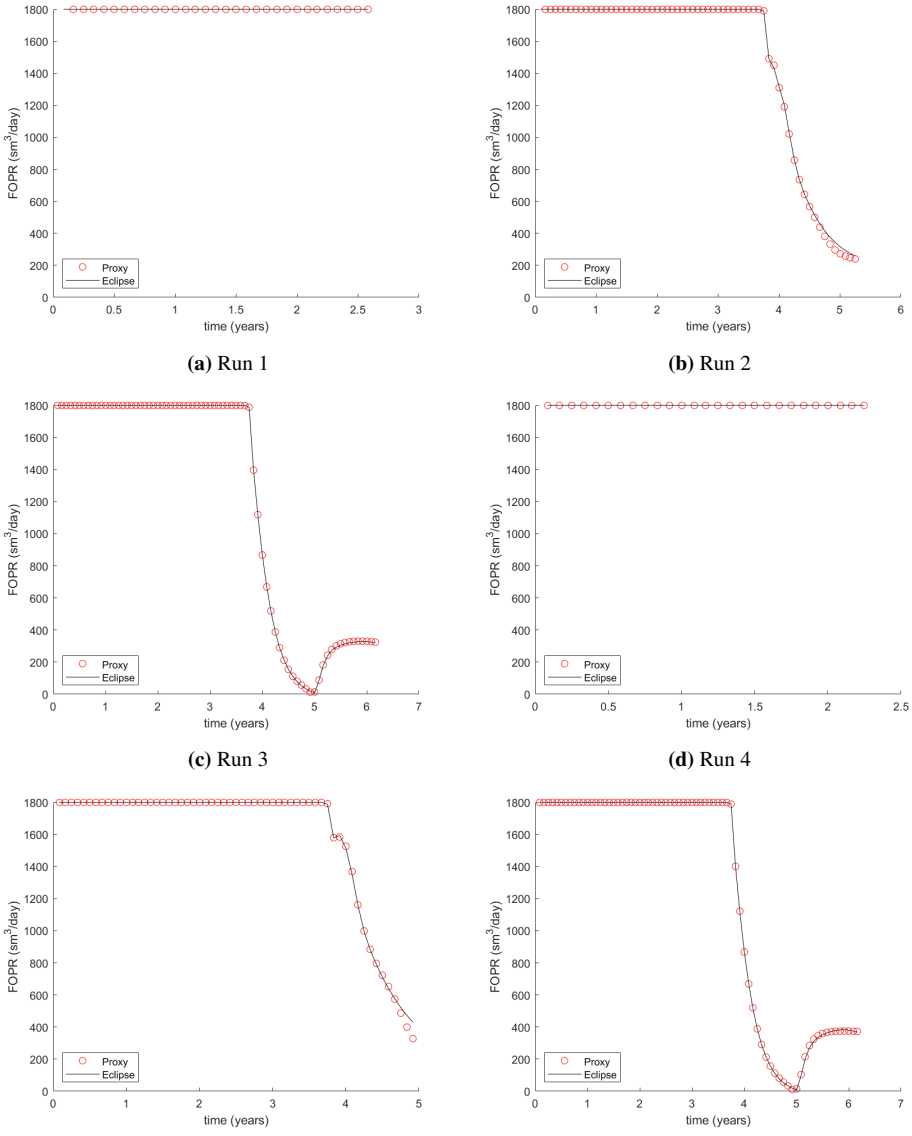
4.3.3 Network Blind Test

We know a high R square value can mean two things, either ANN performs terrific or overfitting. To check this, a blind test performed. Random points generated in between the uniform range for injection rate and pore volume injected, while the other still based on defined as before. All cases listed in **Table 4.10**, with the results and comparison shown in **Figure 4.6**.

Table 4.10: Blind test scenarios.

Run	q_{inj}	PV_{inj}	Comp ID	t_{start}	Run	q_{inj}	PV_{inj}	Comp ID	t_{start}
1	1.10	1.70	13	0.50	4	1.50	1.90	1	0.50
2	1.20	1.30	10	3.75	5	1.70	1.45	7	3.75
3	1.68	1.50	3	5.00	6	1.90	1.65	5	5.00

Based on the result, it can be said that the proxy performs well to represents our dynamic model behavior. The plateau perfectly learned, but there is a little deflection in the tail for cases with a start time of 3.75. Other than that, all works fine. Now, the ANN is ready to be used to perform optimization.



(a) Run 1 **(b)** Run 2

(c) Run 3 **(d)** Run 4

(e) Run 5 **(f)** Run 6

Figure 4.6: Blind test results

4.4 Optimization Study

Here optimization performed with MATLAB. Before that, the objective function will be formulated first for easier computation. As explained before, the objective function is to find the maximum total oil production. The production constraints are production well BHP of 50 bar, injection well BHP of 300 bar. Other than that, the maximum field production rate is limited to 1800 sm^3/day.

Those constraints are defined in the Eclipse file, hence the produced data for ANN learning already obeys those constraints. In that sense, those constraints do not need to be applied in our mathematical optimization problem. Now we need to observe the input data of our ANN. **Table 4.11** listed the input of our generated ANN and the range of parameter values.

Table 4.11: ANN input parameter

Parameter	Min	Max	Description
Injection Rate (sm^3/day)	1	2	-
PV Injected (PV)	1	2	-
Start Time (years)	0.5	5	3 values only (0.5, 3.75 and 5)
CO_2 Composition (%)	0.8	1	-
C_1 Composition (%)	0	0.2	-
C_2 Composition (%)	0	0.2	-

From this information, we need to add a constrain to make it right. We need to make the total composition to 1 to make the system right. Other than that, we have no more constraints. Based on this, we can define the optimization problem in a mathematical formulation, as shown in **Equation** (4.1).

$$P = \begin{cases} maxFOPT(q_{inj}, PV_{inj}, t_{start}, CO_{2-comp}, C_{1-comp}, C_{2-comp}) \\ CO_{2-comp} + C_{1-comp} + C_{2-comp} = 1 \end{cases} \tag{4.1}$$

This notation will be used for the algorithm formulation. In this study, FOPT measured by summing production rate by the timestep. Here, we assume a constant production rate throughout the month, as we partitioned our rate into months. The month itself defined in the real amount of days per month, not assuming 30 days per month.

In addition to that, FOPT measured from the first production date (1 Jan 2020), rather than from the injection date. This performed as no economic parameter nor constraints used. If we only measure from the time where injection starts, the results will surely go towards early time injection (0.5 years). Several adjustment made for this function, which coded with MATLAB.

Genetic Algorithm

Here, the GA function installed in ANN applied for study. The adjustment made was the objective function. As MATLAB GA function works for finding minimum function, we

need to tweak our objective function into a negative system, so the obtained optimization solution later works as our system solution.

After the adjustment, now coming to the algorithm adjustment. **Table 4.12** shows the settings of our optimization algorithm. The settings still using a high number of generations to make sure that the optimization converges due to the average change in the penalty fitness value less than options, rather than not enough iterations reached. **Figure 4.7** shows the result of our GA optimization. The results show that the result converges at iteration 24, with the best FOPT of 3,014,442 sm^3. The best injection condition listed at **Table 4.13**.

Table 4.12: GA parameter

Parameter	Value
Population Size	50
Max number of generation	250
Creation function	@gacreationuniform
Probability of mutation	@crossoverscattered
Mutation function	@mutationgaussian
Type of replacement	Elitism
Elite Count	10

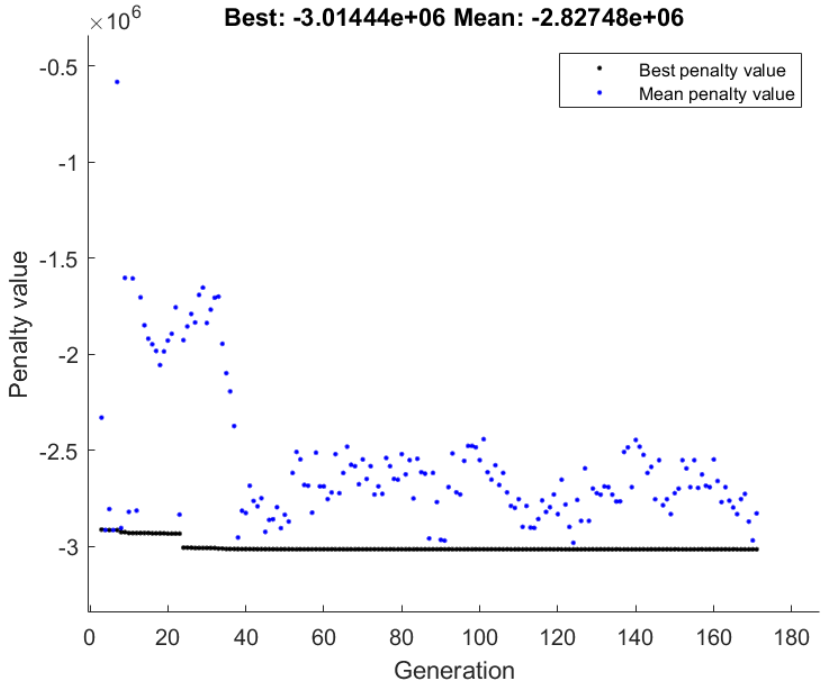


Figure 4.7: GA performance.

Table 4.13: GA optimization results

Parameter	Value
Injection Rate (sm^3/day)	2
PV Injected (PV)	1.9976
Start Time (years)	3.75
CO_2 Composition (%)	0.9977
C_1 Composition (%)	0.0008
C_2 Composition (%)	0.0015
FOPT (sm^3)	3,014,442

It is time to check whether the result reflects our Eclipse run. Using the same settings as obtained from the optimization, Eclipse run was done. The result are tabulated in **Table 4.14**. It can be seen that the relative error is 1.63%. This is expected as the blind test results show deflection on the tail of production.

Table 4.14: Optimum condition run result

	Eclipse (sm^3)	Proxy (sm^3)	Relative Error (%)
FOPT	2,965,879	3,014,442	1.63

Chapter 5

Discussion and Evaluation

5.1 Discussion

The discussions here are based on the obtained results. It is segmented based on the last chapter's sections.

Reservoir Model Development

The first thing that can be discussed is PVT modeling. Here, the model was created using both PVTp and CMG-Winprop. Moving back and forth through those PVT simulators are not that effective, especially in the end, PVT needs to be constructed in E300 format. Developing the PVT model using PVTp only is better, as the model constructed later can be converted either as CMG input or Eclipse input.

It was observed also that there are differences between the properties of pure components and functions available between both simulators. Again, sticking with one simulator will be better. Coming up to the EOS matching, the process is well adapted from available EOS tuning workflows and works fine with several adjustments. The workflow can be used if needed later in the future.

Moving up to the static model. This time, because the main objective yields to field-based SPM to be developed, constant values such as porosity are not problems. Later, when moving to a smaller scale of SPM, this might be a problem because the effect is masked due to the homogeneity. For now, other than this, no other evaluation can be taken. A better model might be needed for more field-like conditions, such as faults, regions, and other methods.

The depletion study itself performed for making logical field-like condition, yet there are more things that need to be taken care of, as choosing BHP and field constraints. Those values used in this study are only based on the production behavior, not really taking into

account how usually production facilities being designed in the field. Another depletion plan such as water injection is recommended as it is the most common plan applied in the field.

The final one is the pit hole of this study. Miscibility, the most important parameter in CO_2 flooding is not really highlighted here. The depletion plan itself if being observed on its reservoir pressure shown in **Figure 3.3**, all chosen parameter are in the immiscible region, hence miscibility is not well developed in every injection plan in this study.

Design of Experiment and Data Analysis

The used DoE in this study is restricted only to LHS design. The determination of the amount of sample taken is not really focused on this study. The process when developing this is if the sample not enough, we just add extra run samples later. But luckily the samples seem enough as the blind test shows good results already when using 30 run samples as training data.

As our current ANN input parameter is not that high, no complex data analysis should be performed in this study such as principle component analysis. This might not be the case if a more complex SPM needed. A comprehensive study needs to be performed when moving to data analysis so the model can still represent our dynamic model even though the input parameter is reduced.

Error during the design in this study needs to be highlighted. The same constant gas FVF are used, which are 0.02 rm^3/sm^3. This might be a good simplification, but for more precision, this must be avoided. The round-up of timesteps also increase the error of $PV_{injected}$. This can be improved by decreasing the timesteps. But again, it will increase the amount of data sampling which means more complex data management and ANN design. Running time for data sampling from Eclipse will be another main concern to solve this.

Proxy Model Construction

A comprehensive study for each ANN setting is not performed here, as the objective is still trying to develop an SPM to help the optimization study. This can be performed if needed later on, as those settings affect greatly on the ANN performance. The amount of training run samples are enough for this study, and the amount of blind test samples are enough to test the ANN.

But some things that need to be discussed are how the data being separated for training, validation, and test data set. Based on the observation, the data are separated randomly, not based on its run sample sets because all of them are merged into one. Not sure this is the right way to do or a modification is needed.

Optimization Study

The optimization study here is approached only using one optimization algorithm. No benchmark function performed assuming the algorithm working fine as it was developed by MATLAB. When modification of the algorithm later needed, benchmarks are needed

to check the algorithm performance. Other than that, the result obtained in this study is quite satisfying for an initial study.

PS is planned to be used also in this study, but it was hard to be developed as one of the optimization parameters are discrete. No constraints function also can be applied in the available PS algorithm in MATLAB. When doing optimization, especially for a proxy model, it is better to work with two or more optimization algorithm to confirm the obtained result.

Reflecting on the results obtained, it can be said that pure CO_2 with the highest injection rate, the largest pore volume injected, and being injected before plateau ends are the optimum condition. This is not a good optimization study example because the extremes are the optimum condition. As no economic considerations are applied in this study, these results can't be denied. But, in reality, it might be hard to be implemented as a high rate of pure CO_2 injected costs a lot of money.

5.2 Future Work Recommendations

There are definitely a lot of room for improvements in this study. Several things that can be improved based on what observed during this study are listed below.

Smart Proxy Model Scale

The current work still using the largest scale, which is field-based. Moving to a smaller scale will be nice, since more behavior can be seen, and more parameters can be optimized, even though complex input and SPM development await. But, this must be in alignment with the objective of the future study. No need to enforce a grid-based SPM model if the objective is only to optimize the amount of oil produced.

Reservoir Model

It will be better if there are available real field data that can be used for proxy development, as synthetic model results might not enough to represents the complexity we find in the real field. If there are no available data to be used, to capture more parameters that influence the flooding process, a more complex model can be used. For example, SNARKSIM model available in the Eclipse data set. The model has faults, different rock types, and heterogeneous porosity distribution.

For the fluid model itself, this model may be used again, but if there are available PVT data set that can be used, which contains advanced PVT tests for CO_2 flooding, it will be better. MMP can be one of the EOS tuning parameters, which may lead to a better PVT model for our reservoir model. But overall, the current study spends much time on this section, hence it will be better if more time can be spent on SPM development later.

Artificial Neural Network

The current study still limits the ANN structure to one type which is DFF-ANN. Changing the structure might be a new branch of study, to know which ANN structure to be used for SPM development. The study of ANN architecture also recommended for future study as the one that is used is still based on the common rule of thumb in ANN development.

When later, more complex problems defined as study objectives, more input data might be needed. This will yield more data analysis studies, and methods such as principal component analysis can be employed. Data reduction can be done also to lessen the proxy fitting weight. Different DoE, especially for data sampling is also recommended for further study.

Optimization Study

The optimization study, which is one of the objectives of the current study can be improved. Rather than optimizing only total oil produced, now it can be continued with NPV optimization (if there are economic models available or using a simplified economic model), or another production parameter optimization such as BHP of the wells. This will later affect which type of SPM to be built and another part of the study.

Sequestration of CO_2 also needs to be observed, as the current study is not focusing on that part. This is because the main project idea is CO_2 EOR and storage. The sequestration behavior can be seen in grid-based SPM if needed to be that complex, or just in well-based SPM by calculating the amount of CO_2 injected and produced.

Chapter 6

Conclusion

This study shows that SPM can be used as one of the strategies that allow the reduction of runtime without sacrificing accuracy. SPM constructed in this study able to mimic the dynamic model we developed, reducing run time from 4.5 minutes to less than 10 seconds for running one case. This SPM performs well to solve the optimization problem in the CO_2 flooding case study.

The developed synthetic model which was used for this study's SPM development is a CO_2 flooding study performed in a combination of the Good Oil fluid model and Egg Model. The optimization problem that was defined for this study is to maximize the total oil produced, with optimization parameters studied are injection rate, the start of injection, pore volume injected, and the composition of the injected fluid.

Using the developed SPM, it shows that almost pure CO_2 used as the injected fluid, flooding with the highest injection rate (2 million sm^3) with highest amount of pore volume injected (2 PV) after the plateau of normal depletion (3.75 years) is the optimum condition for our case. The relative error between the proxy model and the Eclipse run is 1.63%. No economic constraints might be the one which leads the solution to be in the extremes. This preliminary study of SPM viability can be improved more if planning to study this further.

Bibliography

Al Adasani, A., Bai, B., 2011. Analysis of eor projects and updated screening criteria. Journal of Petroleum Science and Engineering 79, 10–24.

Al-Meshari, A.A., 2005. New strategic method to tune equation-of-state to match experimental data for compositional simulation. Ph.D. thesis. Texas A&M University.

Ali, M.T., El-Banbi, A.H., et al., 2015. Eos tuning–comparison between several valid approaches and new recommendations, in: SPE North Africa Technical Conference and Exhibition, Society of Petroleum Engineers.

Amar, M.N., Zeraibi, N., Redouane, K., 2018. Optimization of wag process using dynamic proxy, genetic algorithm and ant colony optimization. Arabian Journal for Science and Engineering 43, 6399–6412.

Amini, S., 2015. Developing a Grid-Based Surrogate Reservoir Model Using Artificial Intelligence. Ph.D. thesis. West Virginia University.

Azlah, M., Chua, L.S., Rahmad, F., Abdullah, F., Wan Alwi, S.R., 2019. Review on techniques for plant leaf classification and recognition. Computers 8, 77. doi:10.3390/computers8040077.

Carcoana, A., 1992. Applied enhanced oil recovery. Richardson, TX (United States); Society of Petroleum Engineers.

Chuang, Y.C., Chen, C.T., Hwang, C., 2015. A real-coded genetic algorithm with a direction-based crossover operator. Information Sciences 305, 320–348.

CMG, 2020. Computer modelling group. URL: https://www.cmgl.ca/.

Gholami, V., 2014. On the optimization of CO2-EOR process using surrogate reservoir model. Ph.D. thesis. West Virginia University.

Heaton, J., 2020. The number of hidden layers. URL: https://www.heatonresearch.com/.

Jansen, J.D., Fonseca, R.M., Kahrobaei, S., Siraj, M., Van Essen, G., Van den Hof, P., 2014. The egg model–a geological ensemble for reservoir simulation. Geoscience Data Journal 1, 192–195.

Jarrell, P.M., Fox, C.E., Stein, M.H., Webb, S.L., 2002. Practical aspects of CO2 flooding. volume 22. Society of Petroleum Engineers Richardson, TX.

Kamali, F., Hussain, F., Cinar, Y., et al., 2017. An experimental and numerical analysis of water-alternating-gas and simultaneous-water-and-gas displacements for carbon dioxide enhanced oil recovery and storage. SPE Journal 22, 521–538.

Kokal, S., Al-Kaabi, A., 2010. Enhanced oil recovery: challenges & opportunities. World Petroleum Council: Official Publication 64.

Lee, K.S., Cho, J., Lee, J.H., 2020. CO2 Storage Coupled with Enhanced Oil Recovery. Springer.

LLC, C.E., 2020. Co2 enhanced oil recovery. URL: http://www.coreenergyllc.com/co2-enhanced-oil-recovery/.

MATLAB, 2020. Matlab. URL: https://se.mathworks.com/.

Mohaghegh, S., 2007. Smart completions, smart wells and now smart fields, challenges & potential solutions. SPE Foundation 2008.

Mohaghegh, S., et al., 2000. Virtual-intelligence applications in petroleum engineering: Part 1—artificial neural networks. Journal of Petroleum Technology 52, 64–73.

Mohaghegh, S.D., Modavi, A., Hafez, H., Haajizadeh, M., Guruswamy, S., 2009. Development of surrogate reservoir model (srm) for fast track analysis of a complex reservoir. International Journal of Oil, Gas and Coal Technology 2, 2–23.

Mohamed, Z.E., 2019. Using the artificial neural networks for prediction and validating solar radiation. Journal of the Egyptian Mathematical Society 27, 47.

Nwankwor, E., Nagar, A.K., Reid, D., 2013. Hybrid differential evolution and particle swarm optimization for optimal well placement. Computational Geosciences 17, 249–268.

PETEX, 2020. Petex - pvtp. URL: https://www.petex.com/products/ipm-suite/pvtp/.

Poole, D., Mackworth, A., Goebel, R., 1998. Computational intelligence: a logical approach. 1998. Google Scholar Google Scholar Digital Library Digital Library .

Saini, D., 2019. Fundamentals of co 2-reservoir oil miscibility, in: CO2-Reservoir Oil Miscibility. Springer, pp. 1–18.

Satter, A., Iqbal, G.M., 2016. 17 - enhanced oil recovery processes: thermal, chemical, and miscible floods, in: Satter, A., Iqbal, G.M. (Eds.), Reservoir Engineering. Gulf Professional Publishing, Boston, pp. 313 – 337. URL: http://www.sciencedirect.com/science/article/pii/B9780128002193000176, doi:https://doi.org/10.1016/B978-0-12-800219-3.00017-6.

Schlumberger, 2020a. Eclipse. URL: https://www.software.slb.com/.

Schlumberger, 2020b. Petrel. URL: https://www.software.slb.com/products/petrel.

Simon, R., Graue, D., 1965. Generalized corre-lations for predicting solubility, swelling and viscosity behavior of co2-crude systems. JPT,(Jan., 1965) , 102–106.

Sintef, 2020. Public open data set - mrst. URL: https://www.sintef.no/projectweb/mrst/modules/mrst-core/data-sets/.

Vulin, D., Gaćina, M., Biličić, V., 2018. Slim-tube simulation model for co2 injection eor. Rudarsko-geološko-naftni zbornik 33, 37–49.

Whitson, C.H., Brulé, M.R., et al., 2000. Phase behavior. volume 20. Henry L. Doherty Memorial Fund of AIME, Society of Petroleum Engineers ....

Xing, W., Song, Y., Zhang, Y., Nishio, M., Zhan, Y., Jian, W., Shen, Y., 2013. Research progress of the interfacial tension in supercritical co2-water/oil system. Energy Procedia 37, 6928–6935.

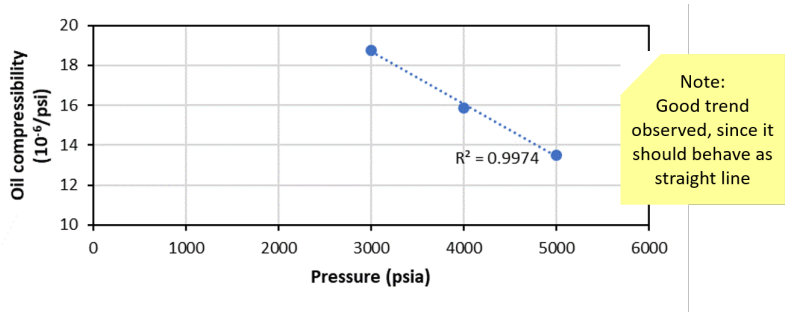
Zubarev, D.I., et al., 2009. Pros and cons of applying proxy-models as a substitute for full reservoir simulations, in: SPE Annual Technical Conference and Exhibition, Society of Petroleum Engineers.

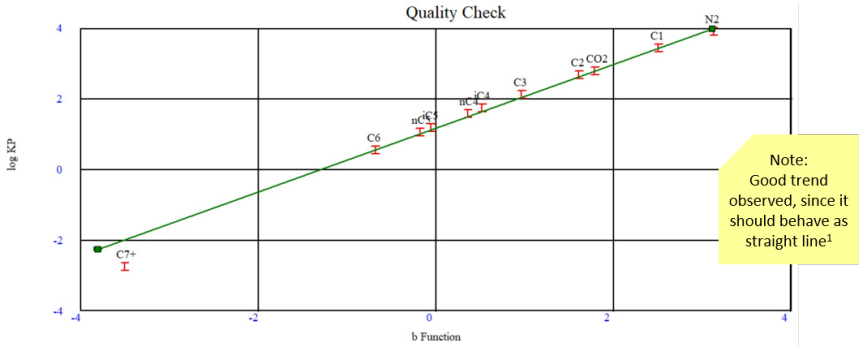
Appendix A

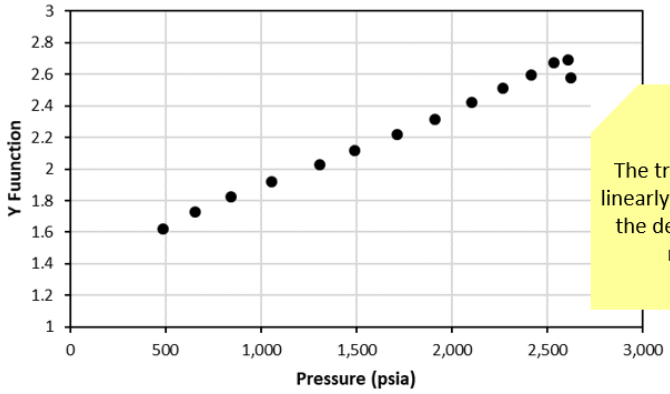
Fluid Model Construction

A.1 PVT QC Results

QC Results are attached in this appendix. In sequence are Y Function QC, Separator Material Balance QC, Oil Compressibility QC (from CCE), Hoffman Quality Check.







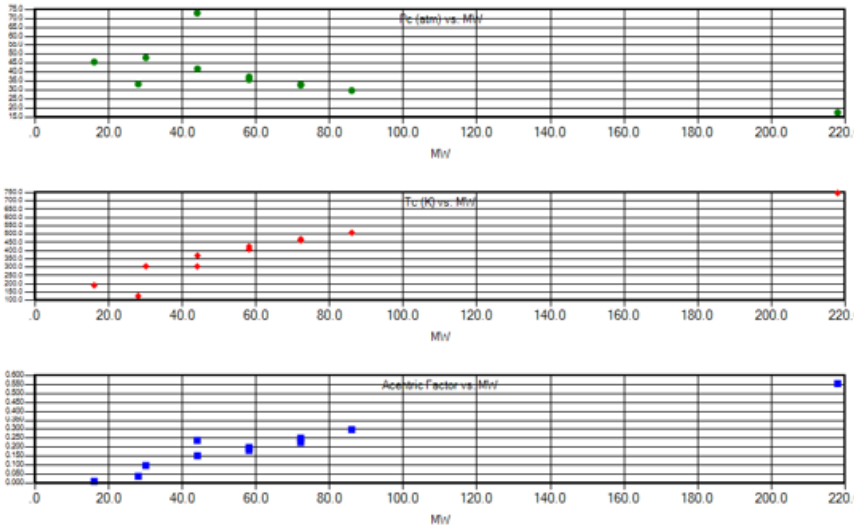
Note:
The trend should be linearly decreasing[3], so the deflected one is not used

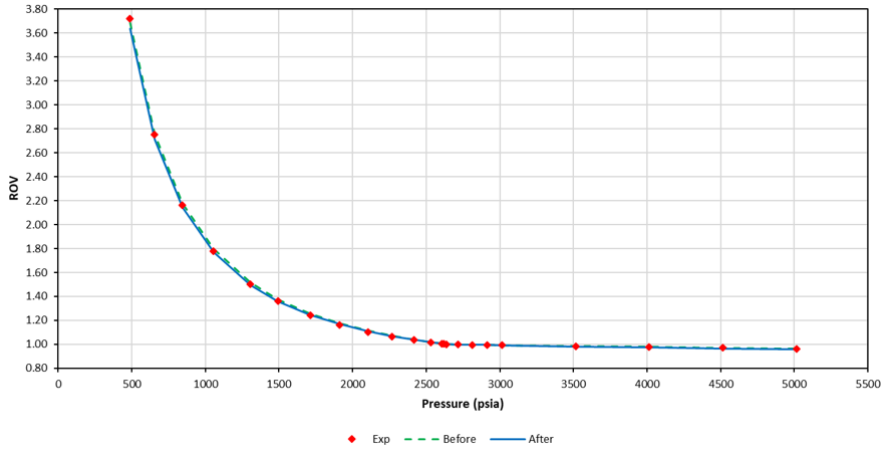
```
*************************************************
*                                               *
*                WINPROP  2019.10               *
*             2019-Aug-28   11:08:49            *
* Computer Modelling Group Ltd., Calgary, Canada *
*                       2                       *
*                                               *
*************************************************
Separator calculation

Summary of Separator Test Material Balance
Basis 1 barrel of stock tank oil

Overall material balance
------------------------

Initial in place =  153.778  kg
Residual oil at STP =  130.521  kg

Stage  GOR (scf/stb)     SG     Mass of gas removed (kg)
-------------------------------------------------------
 1         676.00      0.78600          18.339
 2          92.00      1.36300           4.328
 3           0.00      2.03900           0.000

Total gas removed =   22.666  kg
Material balance error =   0.3841  %
```

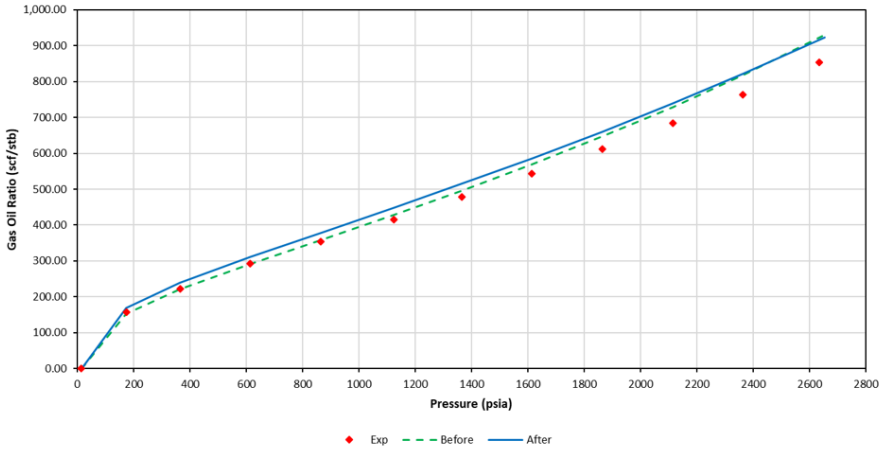
Note:
Only separator data sufficient to be MB QC-ed

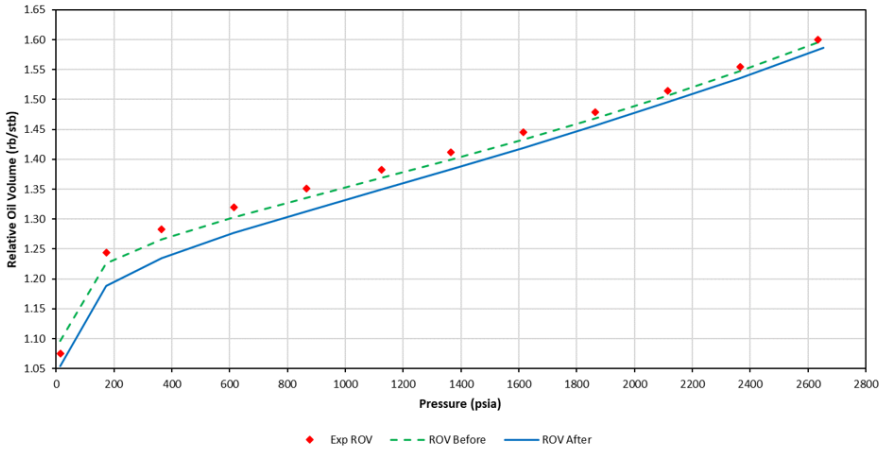
A.2 EOS Tuning

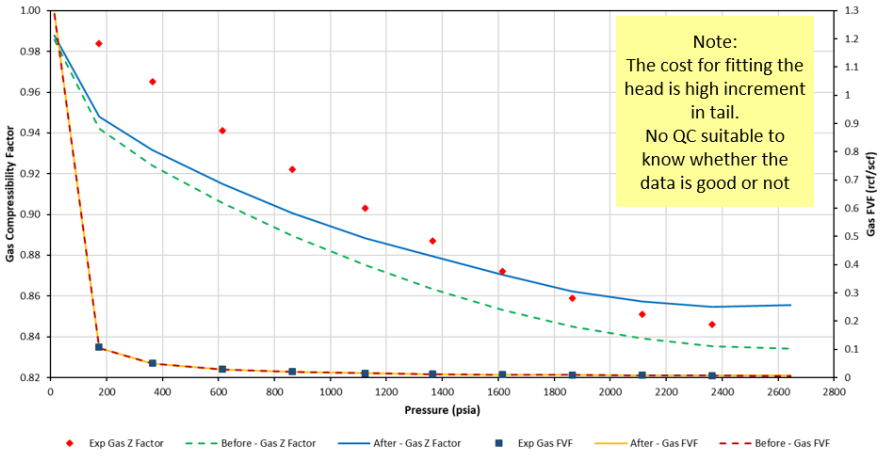
Matched results from EOS Tuning are attached in this appendix. Figures attached are relations between molecular weight (MW) with Pc, Tc and AF. The other picture shows the matching of CCE and DL results.

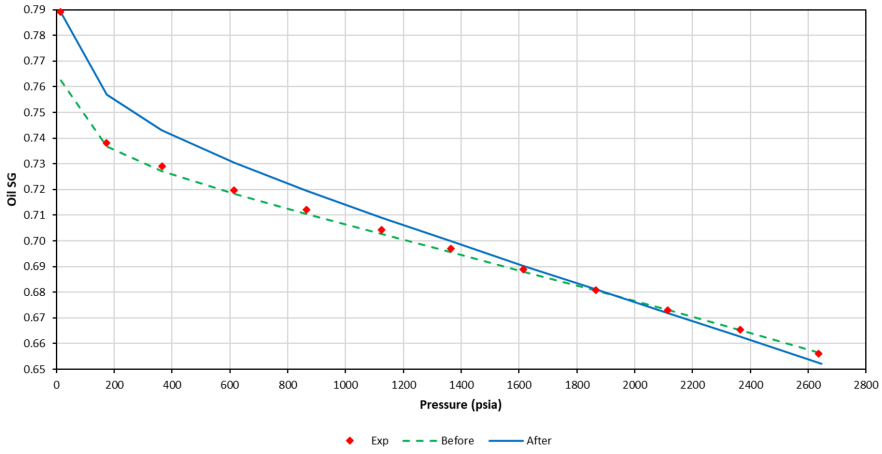


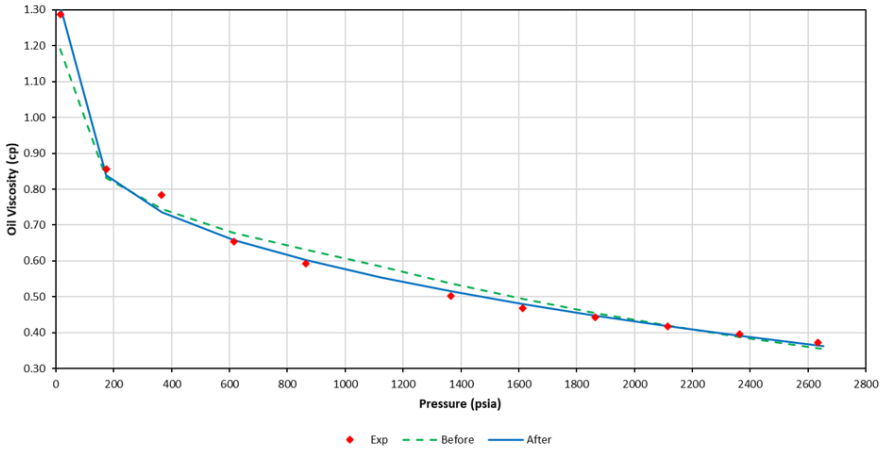



Gas Oil Ratio (scf/stb)
1,000.00
900.00
800.00
700.00
600.00
500.00
400.00
300.00
200.00
100.00
0.00
0
200
400
600
800
1000
1200
1400
1600
1800
2000
2200
2400
2600
2800
Pressure (psia)
Exp
Before
After


Relative Oil Volume (rb/stb)
1.65
1.60
1.55
1.50
1.45
1.40
1.35
1.30
1.25
1.20
1.15
1.10
1.05
0
200
400
600
800
1000
1200
1400
1600
1800
2000
2200
2400
2600
2800
Pressure (psia)
Exp ROV
ROV Before
ROV After


Gas Compressibility Factor
1.00
0.98
0.96
0.94
0.92
0.90
0.88
0.86
0.84
0.82
Gas FVF (rcf/scf)
1.3
1.2
1.1
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
200
400
600
800
1000
1200
1400
1600
1800
2000
2200
2400
2600
2800
Pressure (psia)
Note:
The cost for fitting the head is high increment in tail.
No QC suitable to know whether the data is good or not
Exp Gas Z Factor
Before - Gas Z Factor
After - Gas Z Factor
Exp Gas FVF
After - Gas FVF
Before - Gas FVF


Oil SG
0.79
0.78
0.77
0.76
0.75
0.74
0.73
0.72
0.71
0.70
0.69
0.68
0.67
0.66
0.65
0
200
400
600
800
1000
1200
1400
1600
1800
2000
2200
2400
2600
2800
Pressure (psia)
Exp
Before
After


Oil Viscosity (cp)
1.30
1.20
1.10
1.00
0.90
0.80
0.70
0.60
0.50
0.40
0.30
0
200
400
600
800
1000
1200
1400
1600
1800
2000
2200
2400
2600
2800
Pressure (psia)
Exp
Before
After

NTNU