Norwegian University of
Science and Technology

Cyclic Same-Well Enhanced Oil Recovery

Gas EOR Assessment for a Horizontal
Multi-Fractured Well

Reizky Azhar

Petroleum Engineering
Submission date: July 2016
Supervisor: Curtis Hays Whitson, IPT

Norwegian University of Science and Technology
Department of Petroleum Engineering and Applied Geophysics

Abstract

Demand for petroleum hydrocarbons in the world is still high but the proven conventional reserves has decreased progressively. Alternative power resource in the form of unconventional reservoir eventually has become a prime target for development. Since numerous challenges are encountered in this resource's development, many ways to improve oil recovery have also been performed. Nowadays, shale oil reservoir as one of unconventional resource gains much attention in the oil industry.

The common technique used to produce shale oil reservoir is by occupying stimulated horizontal well with hydraulic fracturing optimization. However, the capability produced by this existing method is still very small, with final oil recovery factor generally in the range of only a few percent. However, many challenges are faced associated with more expensive development costs and low world oil prices which make the exploitation and learning curve for this reservoir was very slow. Optimum development strategy by maximizing oil recovery and minimizing the investment cost is urgently required to overcome current challenges.

This study addressed the key importance of implementing EOR gas injection to shale oil reservoir. Then, its implementation by using the principles "make the most out of what we have" was proposed and studied in every aspect. Cyclic gas injection strategy is the development idea for improving recovery and minimizing operating cost, which are the main focus to be evaluated in this study. In the research, this study used numerical simulator to predict the performance. The analysis showed that the cyclic gas injection EOR has very good potential to be applied in shale oil reservoir.

Sensitivity study against various uncertain environment as well as technical operating scenario were examined in this study. Range of typical value of these parameters were systematically investigated. It is shown that the higher improvement to connectivity between fractures result in higher oil recovery factor. It was also observed that the lower cycle time gives promising oil recovery, although there is an optimum value which will incorporate the needs of improving oil recovery as well as controlling the gas injection efficiency. Monte Carlo simulation to assess the more reliable probability was also performed to further assure this strategy. From all simulated cases, it was shown that the implementation of same-well cyclic injection strategy is highly recommended to be applied in shale oil reservoir.

Preface

This report is written as the final part of Master Thesis Course (TPG4920) for my master degree program at the Department of Petroleum Engineering and Applied Geophysics in NTNU, Trondheim, Norway.

First and foremost, I would like to thank my supervisor, Professor Curtis Hays Whitson for the opportunity given to me as his mentee. All his guidance, support and encouragement were very valuable and becoming extraordinary experience for my academic activity here at NTNU. It was really nice being supervised and working with him.

I would also like to acknowledge Petrostreamz A/S for the Pipe-It license and Coats Engineering, Inc. for the Sensor license.

My appreciation also goes to Wynda Astutik M.Sc (reservoir engineer from Petrostreamz A/S) and Dr. Silvya Dewi Rahmawati (petroleum engineering professor from ITB Indonesia), for not only helping me to understand Pipe-It and Sensor software, but also always have discussion time for me to consult any problem that I have met.

Finally, I am truly indebted to my family, Mega, Kiana, Mama, Papa and Arin for all their great support and endless pray.

Trondheim, July 2016

Reizky Azhar

Table of Contents

List of Tables

List of Figures

1 Introduction

The depletion of petroleum proven reserves in the world has now become significant issue in the last decades. Consequently, unconventional resources become very attractive for energy providers in this industry to make them able to meet the demand for energy resources. Since this type of reservoir has very small recovery factor, many experts in oil industry massively go into a research and study to improve this oil recovery.

Unconventional resources refers to ultra-low permeability reservoirs that is challenging to be produced at economic rates without assistance from technologically advanced stimulation. Based on Clarkson and Pedersen (2011), there are three categories of unconventional light oil resources; halo oil, tight oil and shale oil. Halo oil is found within a considerably low permeability reservoirs with permeability around 0.1 md, tight oil is found in tight reservoirs with permeability even lower than 0.1 md, and shale oil is found in extremely tight reservoirs with permeability less than 0.01 md. **Figure 1.1** shows typical permeability distribution based on reservoir type. This study explored about field development potential by modeling the reservoir into typical characteristic of shale oil reservoir.

Like unconventional resources in general, shale oil reservoir is produced using an advanced drilling and well completion technique to support the production. Combination of a horizontal well with hydraulic fracturing stimulation has been proven to be an effective technique for shale oil reservoir. **Figure 1.2** is Bakken tight reservoir’s well completion schematic, which describes typical well completion for unconventional reservoirs. Even after applying this advance techniques, the final oil recovery factors are nevertheless still very low, in order of several percentages. Production rate and reservoir pressure drop very quickly, leaving huge amount of potential oil reserves behind. Engineers as well as scientists have been working on several studies that come up with the conclusion that utilization of enhanced oil recovery (EOR) method is considered essential to increase the oil production in this type of resource.

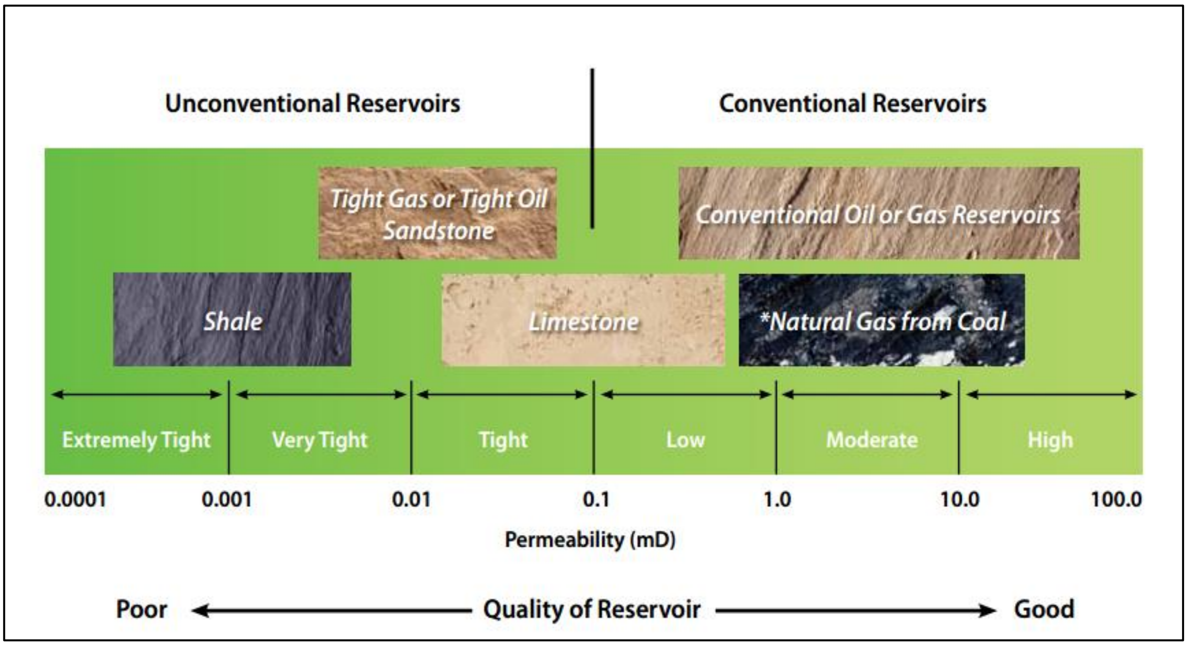

Figure 1.1 Permeability distribution based on reservoir and rock type (Canadian Society for Unconventional Resources, 2014)[1]

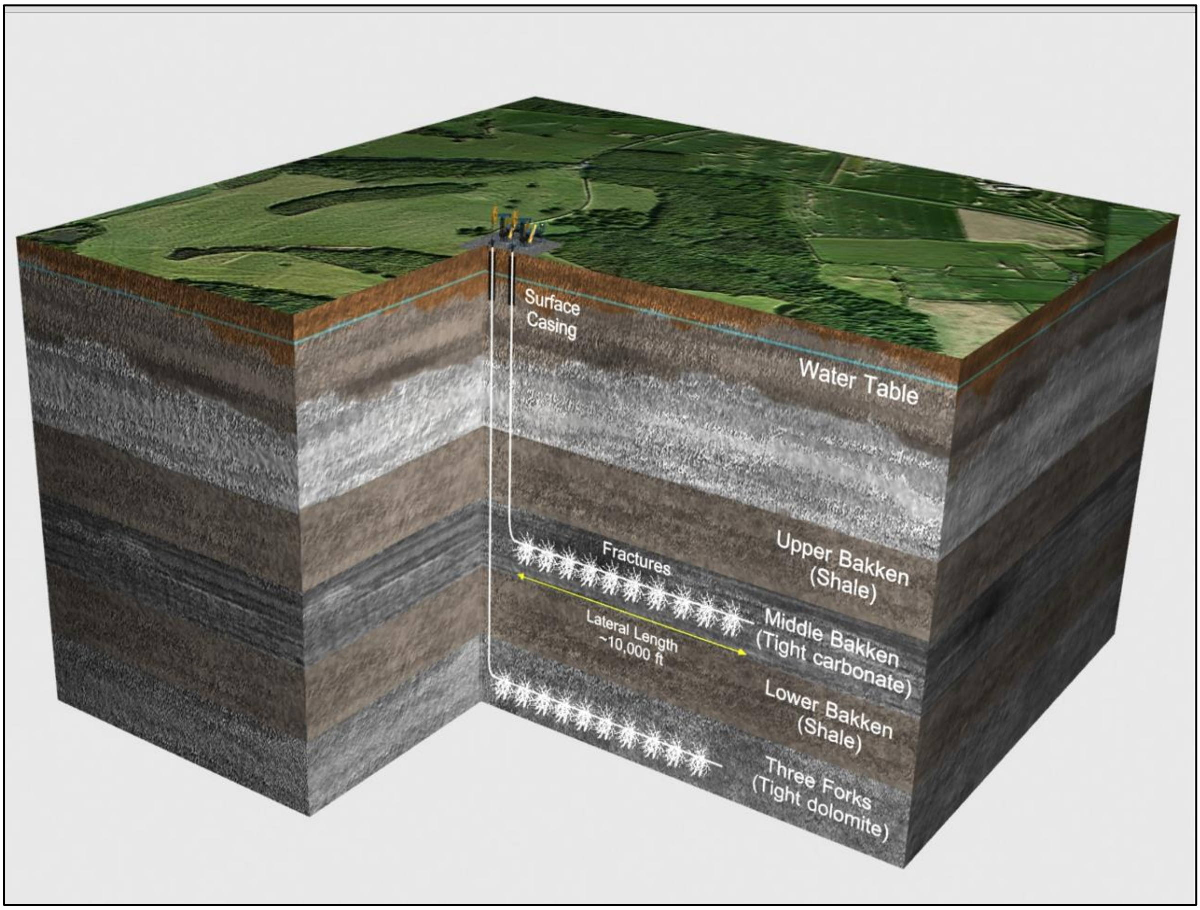

Figure 1.2 Tight reservoir well completion schematic of Bakken field (Statoil, 2013)

1.1 Problem Statement

The development of shale oil faces a lot of issues especially in the increasing capital costs of implementing advanced production techniques such as hydraulic fracturing and EOR. It is even worse due to relatively low oil price in recent years, requiring the field development cost reduction but still targeting optimal production to maintain industry sustainability. Therefore, it takes the best development strategy that integrates optimum EOR method as well as strategic well completion and production scenarios to maximize oil recovery and generate higher revenue. However, immature learning development and lack of experience are still the major challenge in this particular field.

Based on several EOR analysis that has been carried out by many researchers, the most optimum EOR method is through gas injection that supports pressure maintenance as well as recovers oil from the reservoir pores. Combination of EOR gas injection and maximized resource usage is expected to produce best production strategy for shale oil reservoir development.

1.2 Study Objective

The purpose of this study is to evaluate the potential of the same-well enhanced oil recovery by cyclic gas injection scenario method in shale oil reservoir. The evaluation will consider the advantages of this method to the natural flow production scenario generally and the simultaneous injection-production specifically. There will also be discussion regarding the effects of uncertain variables on the performance of this strategy and assess possibility to implement some other alternative methods using the same well completion set.

The main desired outcome of this study is to present the potential development strategy to help addressing the technical challenge, investment cost as well as project economy of the development of shale oil reservoirs by making the most out of what we have.

1.3 Scope of Work

The scope of work for this study is divided into two main groups:

1. Study preparation
    - Modelling the reservoir with its variables within SENSOR simulator format
        a. Reservoir grid model initialization
        b. Rock properties initialization

    c. Fluid properties initialization
- Develop Pipe-It project template
    a. Integrate SENSOR and GAWK with Pipe-It
    b. Build complete work process from user input until end point result
    c. Prepare project template's graphical user interface
- Default reservoir model adjustment
    a. Grid length
    b. Perforation grid
    c. Calculation solver
    d. Cyclic simulation start day
    e. Gas injection pressure
    f. Fractures configuration

2. Reservoir Simulation
    - Analyze same well EOR Potential
        a. Prepare base case scenario
        b. Study cyclic production-injection potential
    - Reservoir uncertain variables sensitivity
        a. Communication behind casing
        b. Uncertain fracture properties
        c. Uncertain reservoir half width
        d. Secondary fracture
        e. Uncertain matrix properties
        f. Reservoir lateral heterogeneity
    - Simulation controllable variables sensitivity
        a. Cyclic interval
        b. Injection rate target
        c. Production rate target
        d. Inter-fractures distance
    - Analyze other possible EOR scenario alternatives
        a. Shut in between cycles
        b. Huff and Puff
    - Compare the cases in terms of reservoir key performance indicator
    - Study the uncertainty analysis

2 Shale Oil Reservoirs

Shale oil reservoir well's productivity depends basically on stimulated reservoir volume (SRV). SRV is the amount of volume surrounding reservoir fracture system that occurs either naturally or artificially as stimulation result, which provide highly conductive conduits to communicate reservoir matrix and the wellbore. The complexity is critical to the well production performance since this is what provides an avenue for injected fluids to displace the oils.

Oil produced from shale reservoirs are basically the same with oil produced from conventional reservoir, but many challenges and requirement of advanced technology in its application makes the development become unconventional. Stimulation by adding connectivity between reservoir's interest zones with the wellbore is required to make a better production. The most common stimulation used nowadays are the combination of horizontal drilling and multi-stage hydraulic fracturing.

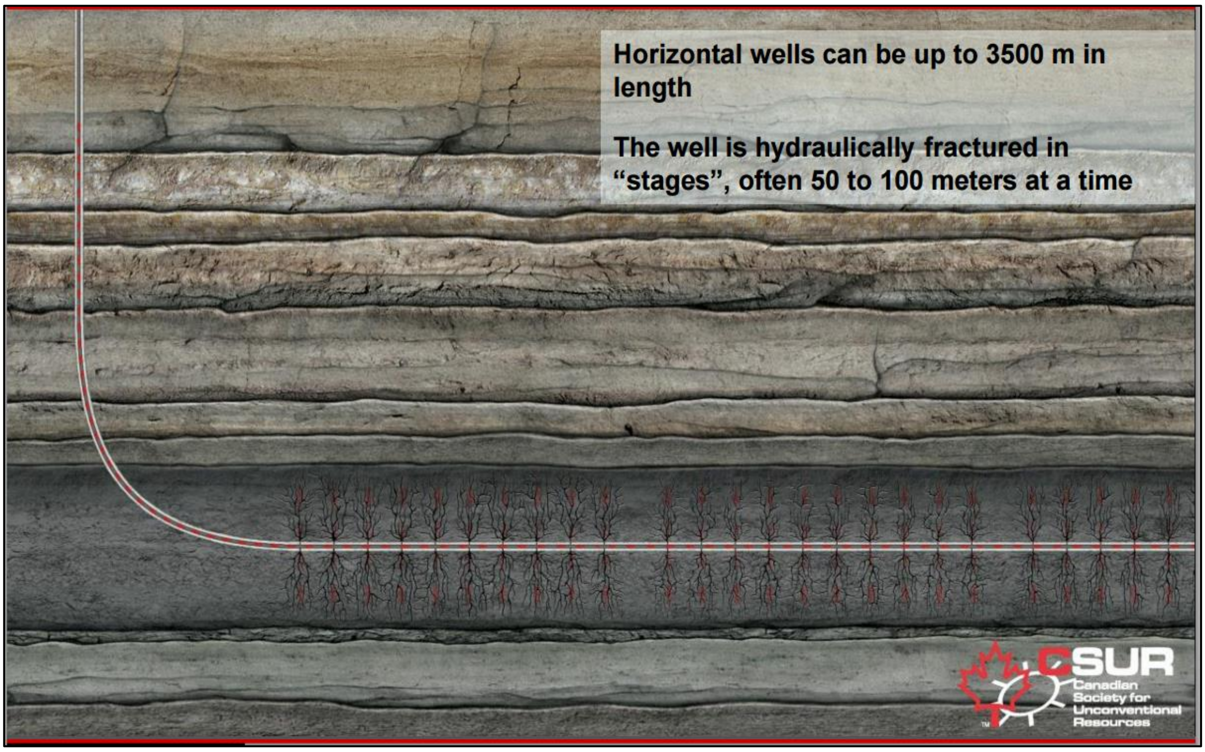

Figure 2.1 Combination of horizontal drilling and hydraulic fracturing in a well (Canadian Society for Unconventional Resources, 2014)[1]

Figure 2.1 shows an illustration of a combination of horizontal and hydraulic fracturing method in a well. Hydraulic fracturing stimulation that is performed by pumping fluid to the reservoir to create new fracture in the reservoir. It is done in several stages with the expected length and spacing between fractures can be determined based on needs. The combinations is expected to enable this ultra-tight reservoir to produce more hydrocarbons.

2.1 Typical Reservoir Characteristics

Shale reservoir is usually formed from clastic-dominated sedimentary rocks consisting of quartz and lime. It contains considerably high total organic content (TOC) as the source of hydrocarbon. Grain size from this rocks is usually below 60 microns, very small thus it results in tiny pore throat size from micro to nanometers. That's why these rocks have relatively much smaller porosity and permeability compared to conventional reservoir. Despite the poor rock quality, it commonly contains sweet light oil with average 42^{o} API with very low H_2S contents.

One examples of shale reservoir is the Bakken formation, located in the Williston Basin, around the border of the United States and Canada. It was deposited in a deep anoxic marine environment around 360 million years ago during the Devonian-Mississippian period. This formation consists of three parts; the upper, middle and lower members. The upper and lower parts have TOC that reaches 36%, with a thickness ranging from 0.5 to 46 feet spread toward the center of formation. The permeability is ranging from 0.01 to 0.03 md and its average porosity is 5% .[2]

Another example is the Eagle Ford formation that is located in the southwest Texas. This hydrocarbon-bearing shale is the Late Cretaceous formation that was deposited in marine continental shelf environment. It consists of organic-rich calcareous-mudrock containing the following mineralogy; 40-90% carbonate minerals, 15-30% clay and 25-20% silica. Total organic carbon ranges from 2-12% with API gravity of 28-62 degrees. Average porosity is about 8-12% and average permeability is about 0.0001 md with thickness ranges from 50 to 400 feet.[3]

2.2 Enhanced Oil Recovery Options

Although it is equipped with a sophisticated advanced technology stimulation, the recovery of this reservoir is generally still very small with a range of 5-15%. Therefore, the implementation of EOR is beneficial to increase of the well efficiency.

The classical methods of enhance oil recovery are categorized into thermal, chemical, miscible and microbial methods (Lake, Johns, Rossen & Pope, 2014).[4]

Thermal recovery is generally used to reduce the viscosity of heavy oil. Since shale oil reservoirs viscosities are considered low, the effort to reduce oil viscosity does not that significantly help. Moreover, there is also an economic issue behind using thermal recovery as it requires relatively more energy to produce the heat.

Considering that most shale oil rocks are relatively oil wet, surfactant becomes the most potential chemical EOR method among the other possibilities, as it will change rock wettability and enhance water imbibition. But in practice, if the fracture density is low, the recovery rate by spontaneous imbibition process will be slow and uneconomic. Furthermore, the low injectivity issue of shale oil reservoir also makes this type of EOR not attractive.

Microbial EOR method is based on biological technology to enhance the reservoir recovery. However, since it will require to inject bio-polymer and bio-surfactant, the low matrix injectivity of shale reservoir and spontaneous imbibition issue also hamper the efficiency of this method.

Some scientists has tried to do simulation studies in gas and water injection method. Sheng et al. (2015)[5] performed reservoir studies to evaluate potential of gas and water injection by both flooding and huff-n-puff methods. The result shows that huff-n-puff gas gave highest recovery, followed by gas flooding, water huff-n-puff and water flooding. Nguyen et al. (2015)[6] reinforces this conclusion by saying that gas injection is promising method to improve oil recovery in shale oil reservoirs due to its more favorable injectivity compared to water flooding. Other researchers, Wan et al. (2013)[7]; Gamadi et al. (2014)[8] also confirms that cyclic gas injection is the best options since flooding method may cause the injected fluids easily breakthrough to producer well via the fracture network and not sufficiently enhanced the oil recovery.

2.3 Gas Injection Enhanced Oil Recovery Opportunities

There are several ways to apply gas injection EOR in oil shale reservoir. In its application, we can occupy a dedicated injector and producer separately or one well as both injector and producer. If there is only one well set in the injection and production process, it will also require a special well completion design.

2.3.1 Conventional Well-to-Well

The first implementation alternative is through conventional well-to-well method, in which dedicated injector and producer is required in its operation. **Figure 2.2** shows an illustration of how the process is going on inside the reservoir.

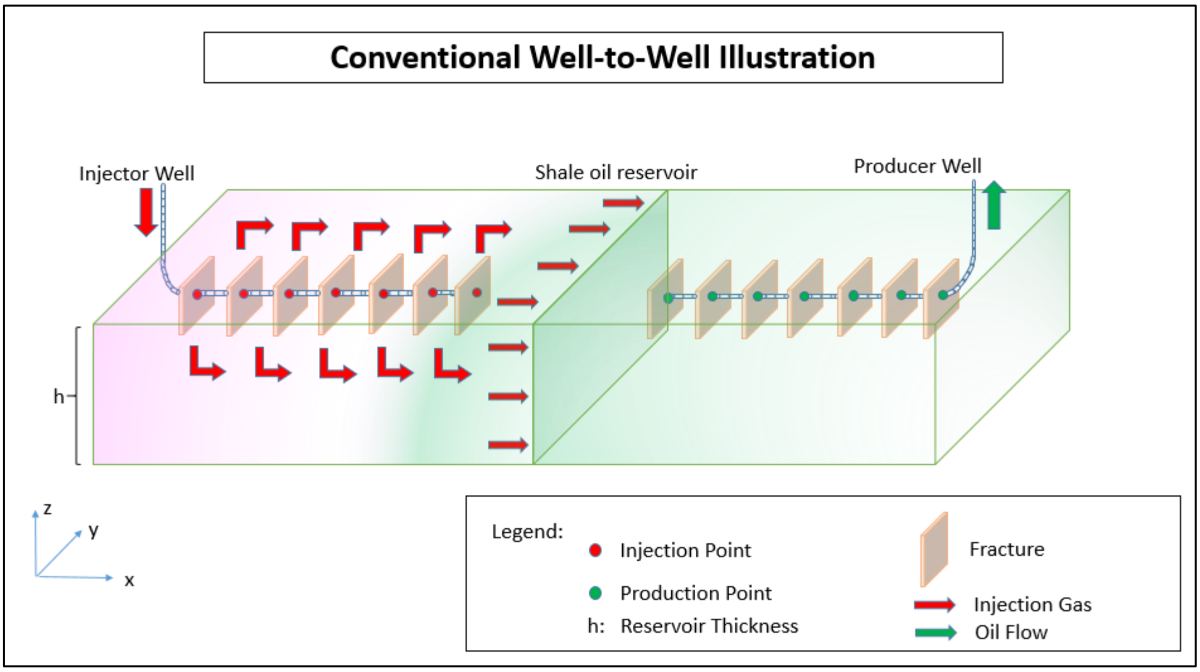

Figure 2.2 Conventional well-to-well enhanced oil recovery schematic processes

In this method, the injection wells will aid gas flow from the wellbore into the reservoir through all perforations and fracture networks. Gas is injected concurrently with oil production by the producer. The injected gas is expected to drive oil towards producer fracture network and finally producing. Continuous injection and production scenario and cyclic injection can be performed using this method, because each well has separate well head flowing control.

2.3.2 Same-Well Continuous (Dual-Tubing String)

The next possible implementation is by utilizing one well for injecting gas as well as producing oil from the reservoir. This method requires two dedicated tubing strings with each different function. **Figure 2.3** shows an illustration of how the process takes place in each tubing string in the well.

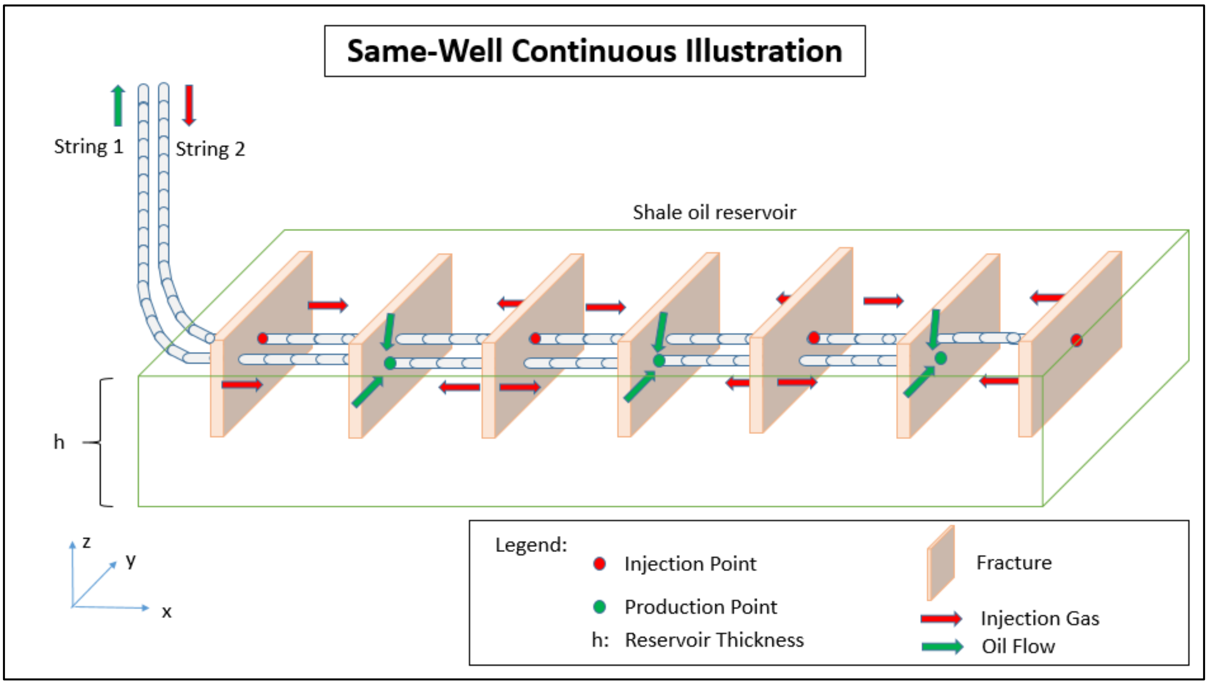

Figure 2.3 Same-well dual-tubing string enhanced oil recovery schematic processes

In this method, gas injection will be channeled through the injection tubing into the reservoir and the oil produced will flow through the production tubing from the reservoir. To produce more optimal sweeping efficiency, the perforation configurations in alternating intervals is preferred. **Figure 2.4** shows an illustration of how the process of oil displacement by gas occurring in the reservoir.

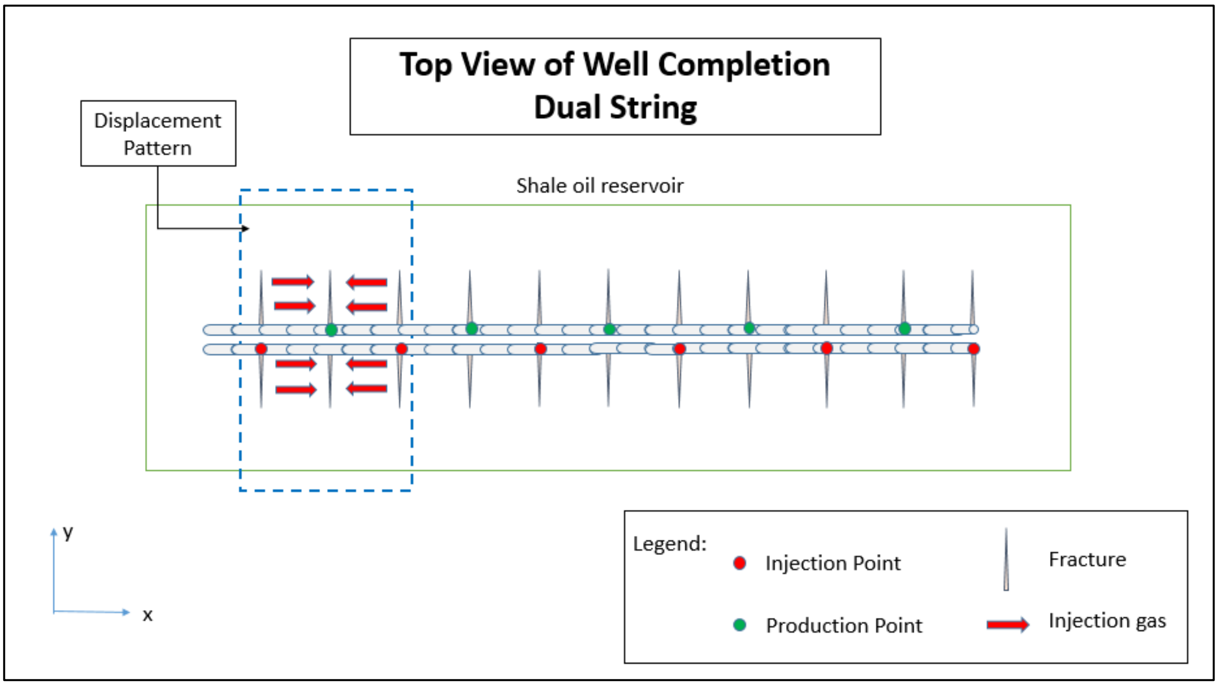

Figure 2.4 Same-well dual-tubing string gas injection sweeping movement illustration

For easier analysis, the author mentions the three adjoining fractures as a displacement pattern. An efficient displacement pattern is the one which has a fracture production connected with producer perforation, which is surrounded by two injection fracture connected with injector perforation. By setting this configuration, it is expected that oil which lies in between will be better displaced because it receives the same support from both directions. Further description about the process which occurs in the reservoir will be explained in sub chapter same-well cyclic (alternating injection-production fractures) because of their similarity in the process.

To implement this configuration, it needs packer which is placed between two adjacent fractures so it can separate the gas injection flow with oil production flow. **Figure 2.5** shows the well completion illustration of this method, in which a packer is set between fractures. Each tubing string is equipped with a valve which can only be opened in fracture zones in accordance with a predetermined function. Author calls the injection valve as the injection point and it can only be found in the injection tubing string and placed according to the position of injection fracture zone. Continuous injection and production scenario as well as cyclic injection scenario can be applied to this method because each tubing string has separate wellhead flowing control.

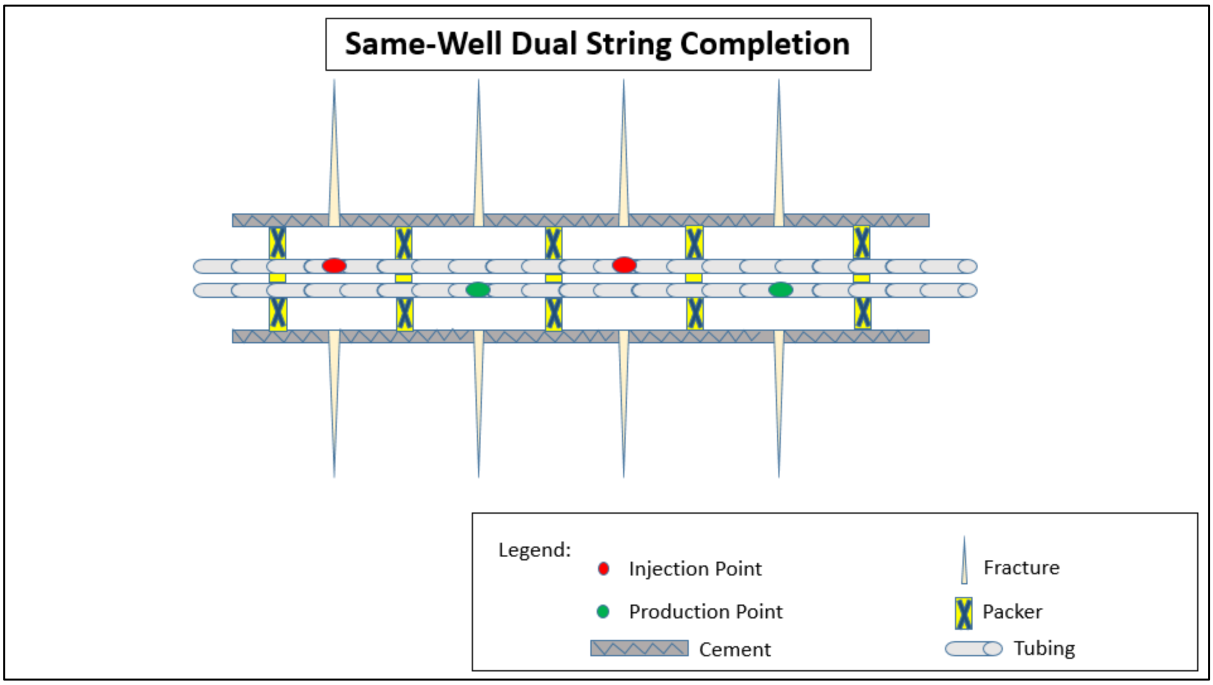

Figure 2.5 Same-well dual-tubing string well completion illustration

2.3.3 Same-Well Cyclic (Huff-n-Puff)

The other method is to utilize one well as a dedicated injector as well as dedicated producer alternately or it is commonly referred as huff and puff. **Figure 2.6** shows an illustration of huff process in which well only injects gas into the reservoir for a certain duration. The injected gas is expected to be soluble in oil around the stimulated reservoir volume and increase its mobility. Tubing and all fracture networks are used by the gas to flow into the reservoir.

Huff process is then followed by oil production process, called puff. In this process, stimulated oil will be produced during the same duration as huff. **Figure 2.7** shows the illustration of puff process within the well. In this process, the tubing and the entire fracture networks will be used by oil to flow from reservoir to the surface.

To implement this method, it doesn't require certain well completion design since all perforations has the same function in either huff or puff process. The well function as injector and producer wells can be replaced easily because the wellhead flow direction can be modified based on the needs in each process.

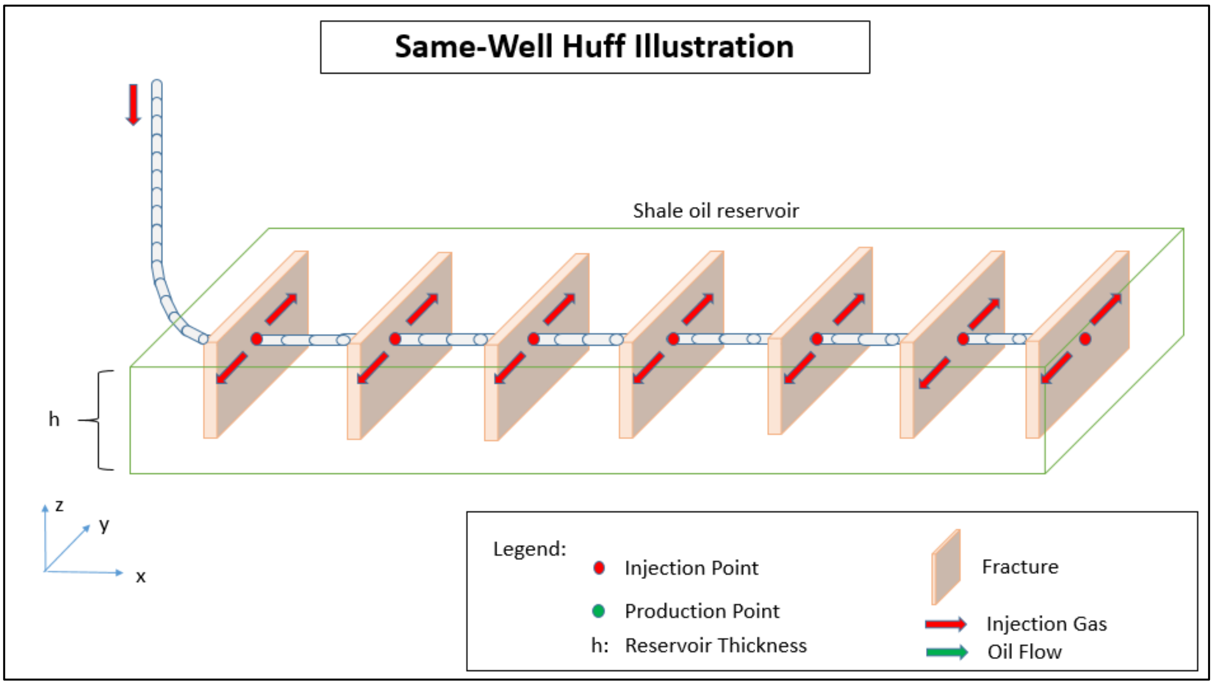

Figure 2.6 Same-well "Huff" enhanced oil recovery schematic processes

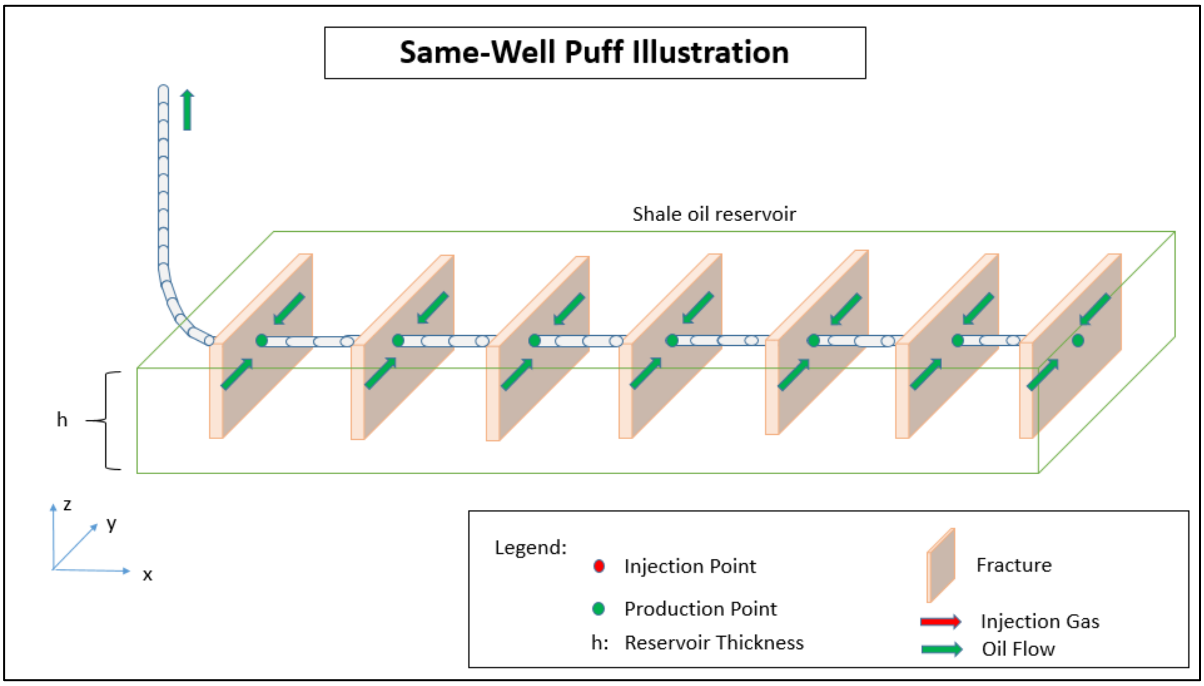

Figure 2.7 Same-well "Puff" enhanced oil recovery schematic processes

2.3.4 Same-Well Cyclic (Alternating Injection-Production Fractures)

The next method which becomes the method used in this study is utilizing one tubing string for injecting gas as well as producing oil from the reservoir. The process of assigning fracture's function is similar with dual tubing string method that has been explained before. The difference in this method is on how it only used single tubing string as a medium for both gas injection and oil production.

Figure 2.8 shows displacement process that occurs from each injection fracture in the reservoir. Single well is functioned as the injector and producer by placing and assigning the tubing valve into those two functions. To simplify it, the author identifies the valve tubing as either injection or production point according to its placement. With this method, the gas injected from the injection point, denoted by red circles, can flow into fractures associated with it, and push the oil towards the production fracture that is also associated with the production point, symbolized by green circles.

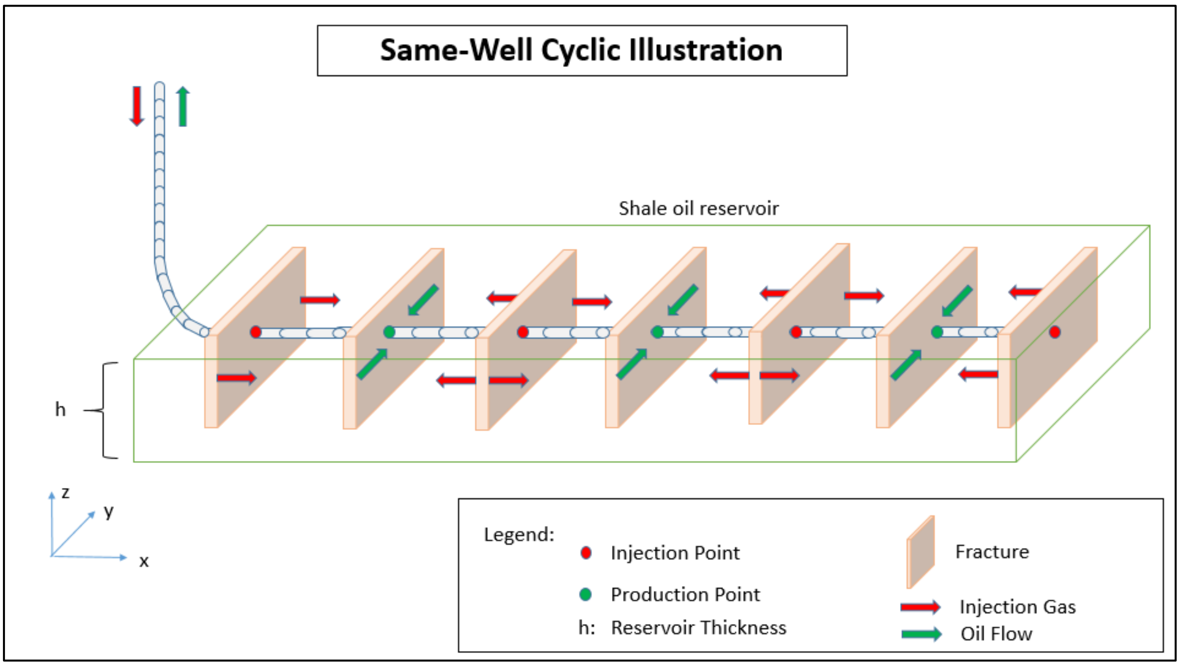

Figure 2.8 Same-well single-tubing string enhanced oil recovery schematic processes

Figure 2.9 shows the displacement process that is expected from this method in the reservoir. Three adjoining fractures are referred as displacement pattern. This configuration is called displacement pattern since the gas injected to the injection fracture push the oil contained in the matrix towards the production fracture and then flow into the well. Each injection fracture gives ½ of its total gas injection to each production fracture nearby so every production fracture gets the full support from all injection fractures from both directions. This process occurs in every pattern in the reservoir.

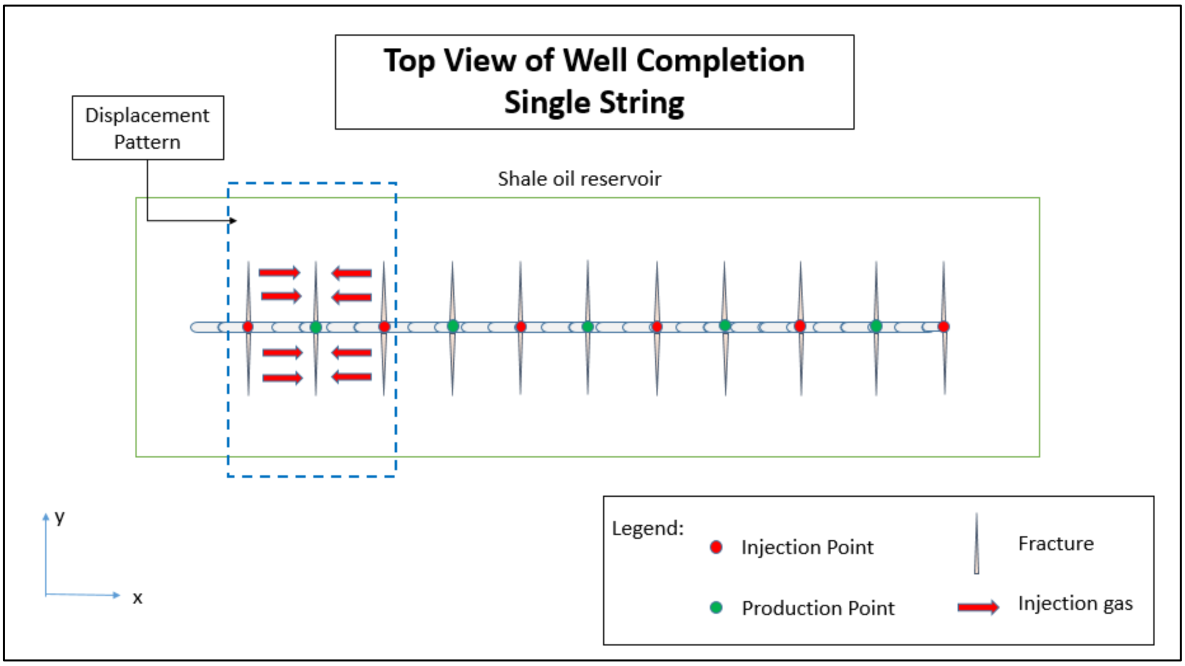

Figure 2.9 Illustration of studied hydraulically fractured horizontal well completion in top view

One pattern consists of four symmetrical blocks with same drainage area, geometric shape and conductivity. Each quarter pattern has formed two half-fracture configurations, one as a production fracture and another one as an injection fracture which are opposite to one another. The similarity of each pattern is advantageous because it becomes easier for modelling the reservoir. This study will create simulation of this quarter pattern's grid models by representing the process of displacement by ¼ of the total gas injection for one pattern that comes out from the injection point into the reservoir and produces ¼ of the total production of one pattern into the production point. The simulation result will then be converted to full-scale well result to see the well performances.

Figure 2.10 shows the well completion illustration in this method. To separate each perforation point in this well according to its function, it requires packer between each fracture. Tubing is equipped with one-way valve at each perforation point, which would distinguish the function of fracture associated with it. One-way valve lets the fluid flow but only to one direction, either out of or into the tubing.

At each injection point, the valve is set to only be able to let the fluid flowing from the tubing into the reservoir, not either way. While at the production point, the valve is set to only allow the fluid flowing from the reservoir into the tubing, not to the opposite direction. Hence, when the well is used for gas injection, only the injection point will be opened and when the well is used for oil production, only the production point that will be opened.

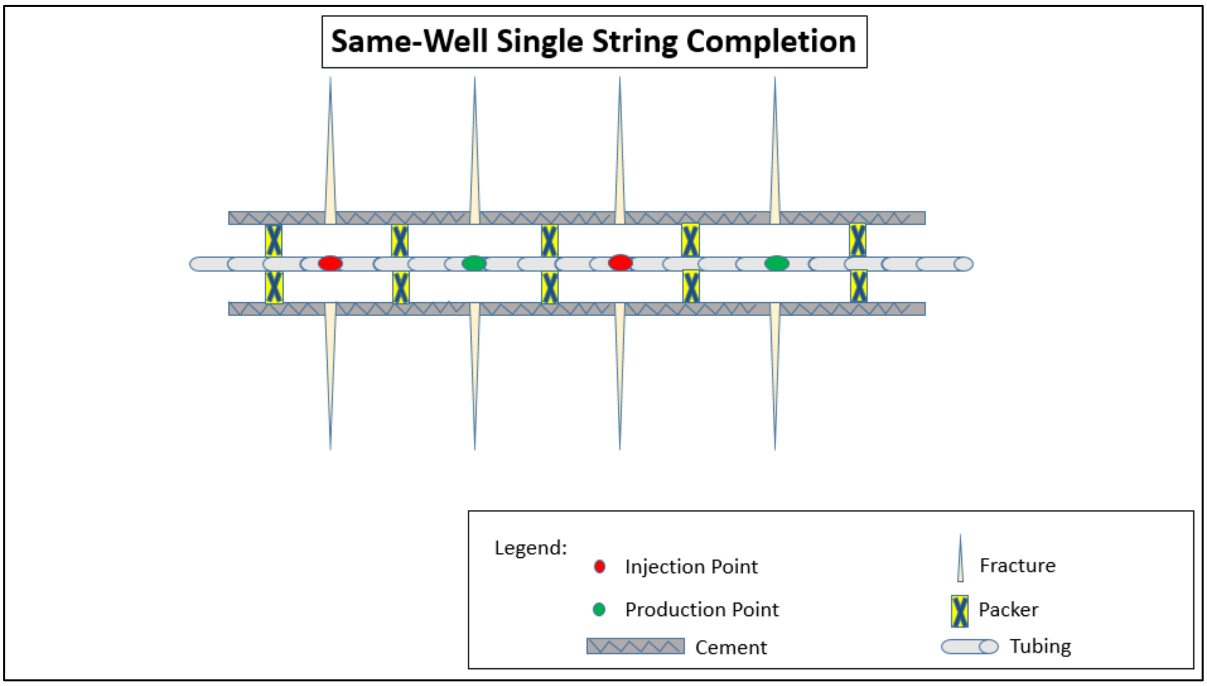

Figure 2.10 Same-well single-tubing string well completion illustration

Scenario applied to this method is cyclic injection and production by using the same well during a particular interval cycle. At the time of injection, gas is expected to put pressure into the reservoir through the connected fractures. In this process, the production valve is not open so given pressure will continue to build-up until the end of duration. By the time oil production, production point will be opened but injection point will be closed. It will let the oil flow through the production fracture proportionally as high as the built-up reservoir pressure from gas injection.

This process is beneficial to minimize operating cost since this cyclic process uses the same horizontal well for injection as well as production, compared to continuous gas injection which requires more capital in the development thus giving longer payback period.

3 Software Used in Same-Well Cyclic EOR Modeling

3.1 Sensor Reservoir Simulator

This simulation study utilized SENSOR reservoir simulator, which stands for System for Efficient Numerical Simulation of Oil Recovery, the compositional and black oil reservoir simulation software that was developed by Coats Engineering, Inc. *This software is a generalized 3D numerical model used by engineers to optimize oil and gas recovery processes through simulation of compositional and black oil fluid flow in single porosity, dual porosity, and dual permeability petroleum reservoir.*[9] This numerical simulation is used since the author believes that SENSOR is the fastest and most cost-effective simulator in the industry.

Impes and Implicit formulations are included in SENSOR. There are three linear solvers available; reduced bandwith direct (D4), Orthomin preconditioned by Nested Factorization (NF) and Orthomin preconditioned by ILU (red black and residual constraint options). Sensor is flexible in reservoir gridding as it can handle any grid types and it uses Equations of State (EOS) of the Peng-Robinson (PR) and Soave-Redlich-Kwong (SRK) for compositional simulation.

In this study, SENSOR was used to make the shale oil reservoir model with a hydraulically-fractured horizontal well, run the model, and extract the simulation results to understand the proposed EOR scenario potential together with its uncertainties.

3.2 Pipe-It

Pipe-It is a unique IAM (Integrated Asset Management) software to integrate models and optimize petroleum assets. Pipe-It is an application generated by Petrostreamz A/S, a software company developed at PERA A/S. Pipe-It allows user to chain several applications in series and parallel, and also launch any software on any operating system. Pipe-It provides a framework to connect together the array of software, integrate projects which is still incorporating interaction between one to another, automate workflow and perform optimization.

It also has a function to connect and compile data from different resources, thus it gives better working efficiency and helps to automate a process. It is very useful to reduce manual edit, copy and paste work which is time consuming.

Pipe-It has powerful features that are very beneficial in this study, which are Maplinkz, Linkz dan Optimizer. These are used to connect data from different resources and run a bunch of simulation case matrix automatically.

In this study, Pipe-It was used for integrating GAWK and SENSOR so it provide easiness in conducting simulation and scenario set up. It also can automatically run any number of simulation, tabulate output file, easily locate required data to be analyzed and then plot the simulation result from SENSOR reservoir simulator. The developed Pipe-It project template makes the work much easier since all program is run at the same window and it automatically follows the workflow that has been made.

3.3 GAWK

Gawk is a programming tool that can be used to create or select particular records in a file and perform operations upon them. It also can be used to perform simple programming to generate a written file to be further used in subsequent applications.

In this study, gawk is used to automatically generate include files to be simulated in SENSOR, based on input variables that are defined by the user. It is also used to perform simple calculations to get additional information from the simulation results for further analysis such as unit conversion, the calculation of scale factor multiplier, the calculation of quality-control relative error, and reporting output message from several sub processes.

4 Reservoir Model

The objectives of this study are to see and compare the potential of the Same Well Cyclic Gas Injection EOR with other development strategies and the effect of uncertain variables to the performance of hydraulically fractured shale oil reservoir. To begin with, a base case model was generated to represent the well in the reservoir.

4.1 Reservoir Problem Set

This study worked with single hydraulically fractured horizontal well in the shale oil reservoir. In the horizontal section, fracture formed will be used for both injection and production based on the chosen EOR. **Figure 4.1** best illustrates the completion schematic of this well in a 3D view.

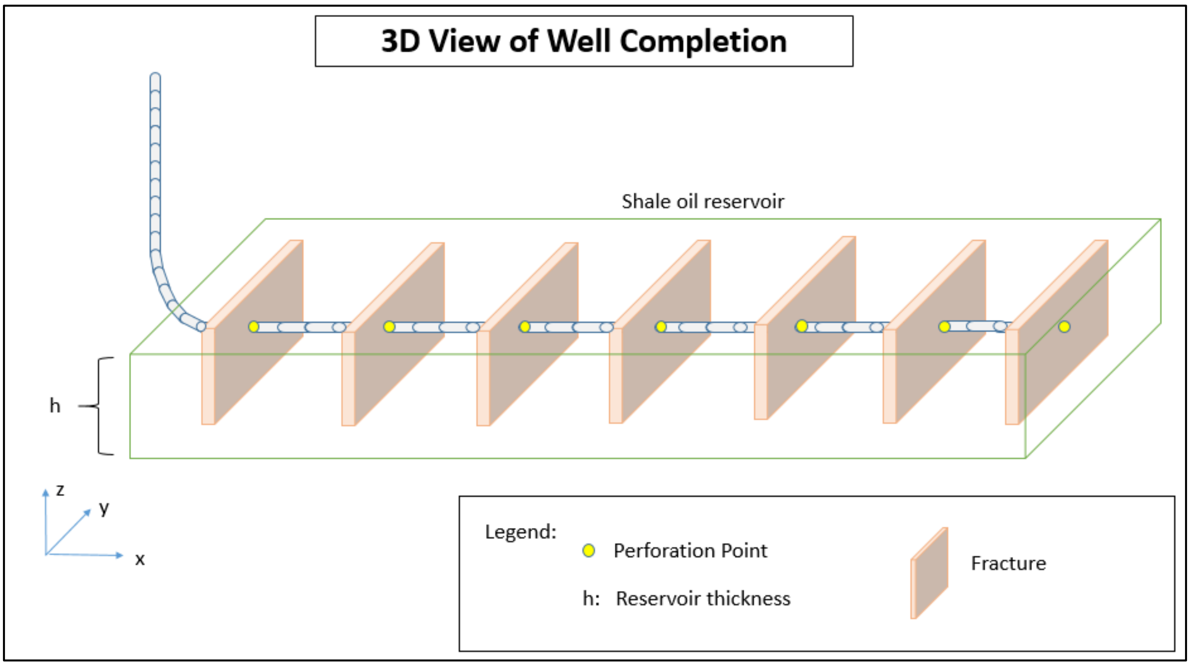

Figure 4.1 Illustration of studied hydraulically fractured horizontal well completion in 3D view

As previously described, the author mentions three adjacent perforation points as a displacement pattern. For simplicity reason, the models will only represent a small scale of the fracture pattern. Assuming the hydraulic fracture spacing and reservoir dimensions on each pattern are identical and symmetrical, the simulated reservoir model will only cover specifically a quarter of a pattern as shown in **figure 4.2**, and will finally be converted to well-scale result in accordance with the data given. This modeling method is possible since each fracture behaves independently.

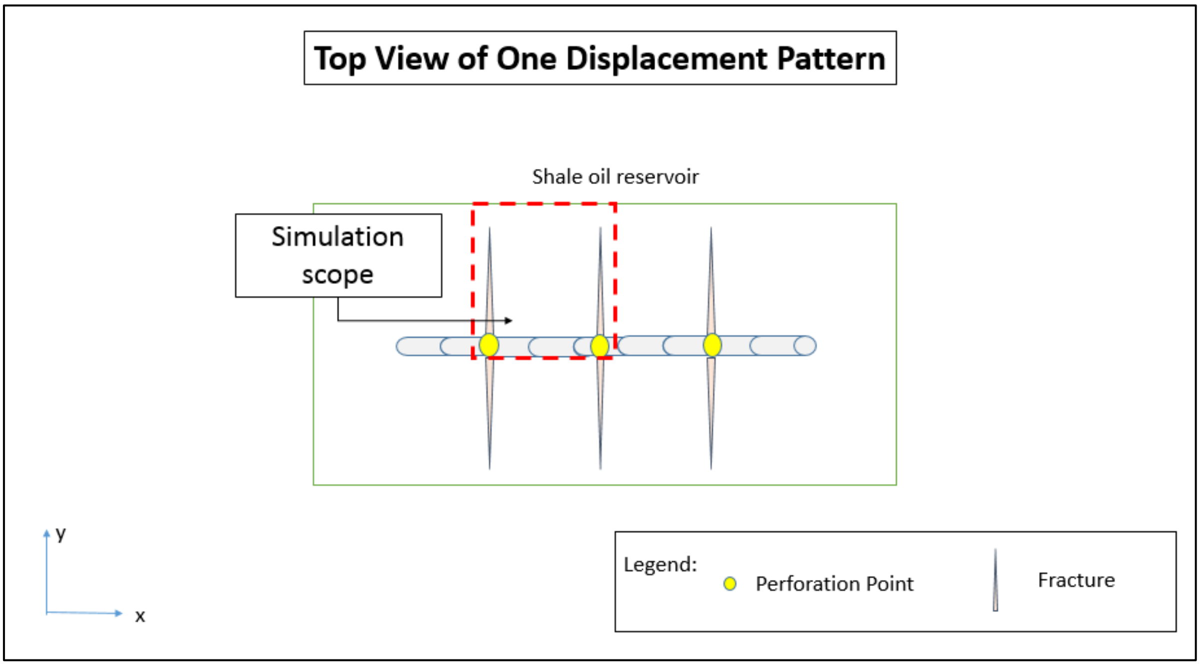

Figure 4.2 Illustration of one displacement pattern in top view

For the initial model to begin with, author indirectly make an assumption that each block is homogeneous, isotropic, and has same rock and fluid properties. Wan et al . (2013) compared the results between simplified and the whole model in their research, and the results are well matched. The properties can be modified later on as per needed every time we perform the simulation of certain case.

4.2 Sensor Reservoir Grid Model

The reservoir model is built as closely as possible to the typical unconventional shale oil reservoirs. By setting up 200 ft length in horizontal direction, 500 ft width and 200 ft layer thickness, it has two fractures perpendicular to the horizontal well section with a length of 250 ft and a width of 1 inch on both ends. It also has 1 inch width cement layer alongside the first row of horizontal well section to accommodate the presence of cement behind the casing. **Figure 4.3** shows the schematic reservoir model in top view.

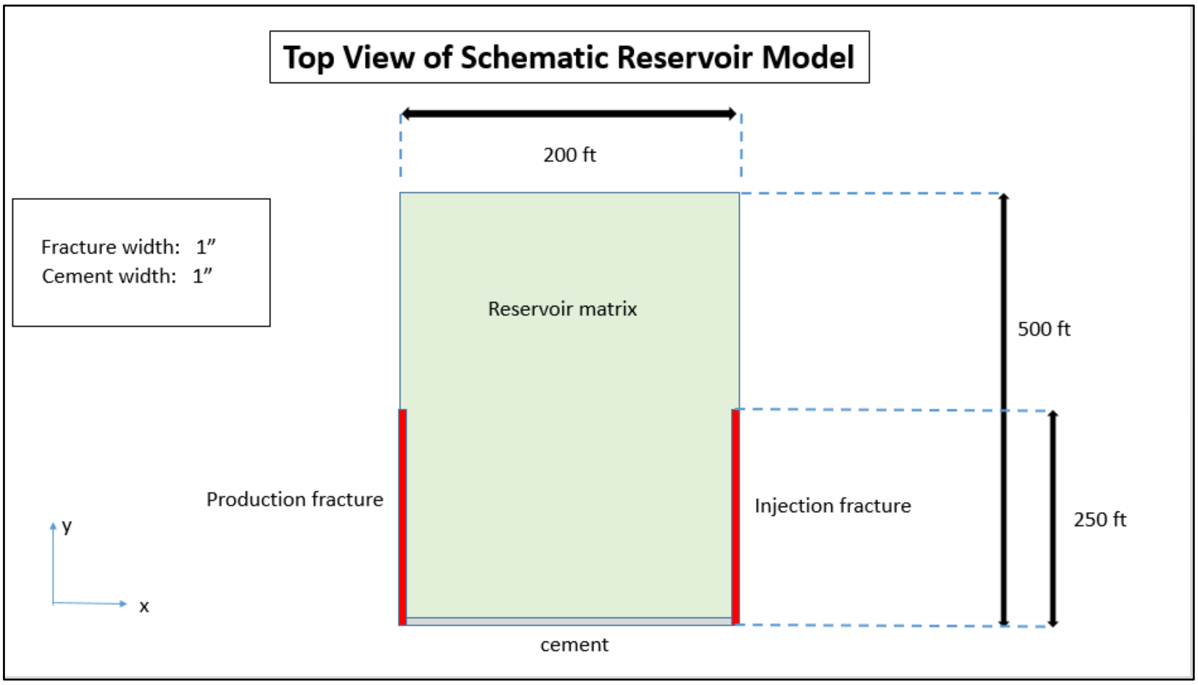

Figure 4.3 Schematic Reservoir Model

Regular cartesian xyz fully implicits grid model is used for this study. The initial model was set to have 2,000 total number of grids (Nx=200, Ny=10, Nz=1). **Figure 4.4** shows the top view of reservoir grid model in terms of oil saturation map, run from SensorMap feature in SENSOR.

As previously mentioned illustration, this model was adjusted to have fracture grid in both ends of the x axis and a cement grid line at the beginning of its y-axis. For its z direction, this model only utilized one layer with thickness of 200 ft for simplification purposes. **Table 4.1** summarizes the initial reservoir grid model's dimension.

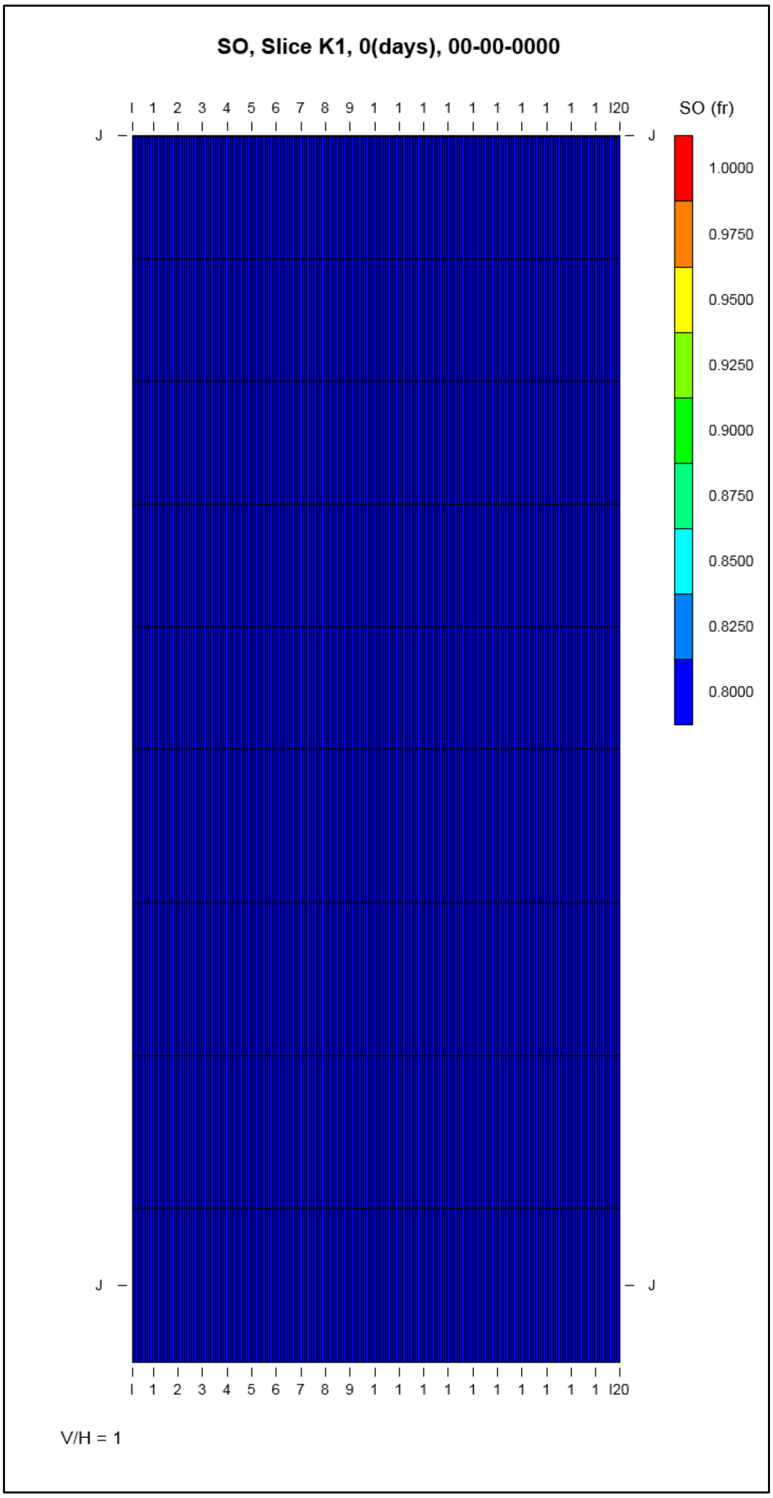

Figure 4.4 SENSOR reservoir grid model in terms of oil saturation map in top view

Table 4.1 Reservoir Dimension Summary

Reservoir Grid Dimension and Geometry	
Total Grid, N	2,000
Number of x grid, Nx	200
Number of y grid, Ny	10
Number of z grid, Nz	1
<u>Matrix grid length</u>	
Between Fractures in x-direction, $(\Delta x)_m$	1 ft
Between Fractures in y-direction, $(\Delta y)_{m\,1}$	50 ft
Between Fracture and Reservoir Boundary in y-direction, $(\Delta y)_{m\,2}$	62.5 ft
<u>Fracture grid length</u>	
Fracture in x-direction, $(\Delta x)_f$	1 inch
Align with cement in y-direction, $(\Delta y)_{f\,1}$	1 inch
Beyond cement in y-direction, $(\Delta y)_{f\,2}$	50 ft
<u>Cement grid length</u>	
Cement in x-direction, $(\Delta x)_c$	1 ft
Cement in y-direction, $(\Delta y)_c$	1 inch
Layer thickness, Δz	200 ft

The reservoir properties used to build this model are based on typical shale oil reservoir characteristic. The depth to the top of the reservoir is set to 10,000 ft with 6,500 psi and 240^o Fahrenheit initial reservoir condition. Rock compressibility is 4x10-6 1/psi, permeability is 1,000 nano-Darcy, porosity is 5% and fracture conductivity is 833 md-ft. For cement grid, the default values are 0% porosity and 0 mD permeability. **Table 4.2** summarizes the initial reservoir properties that are set in this model.

Author uses 1.0 fracture porosity and 0.0833 ft fracture width instead of 1 ft since based on Whitson et al. (2012)[10], as long as fracture conductivity and volume are honored, the numerical fracture width may be 0.01 or even 1 ft without having any real impact on the results.

Table 4.2 Reservoir Properties Summary

Reservoir Model Properties	
Initial Reservoir Pressure, P_R	6,500 psi
Top Reservoir Depth	10,000 ft
Reservoir Temperature, T_R	240 ^oF
Formation Compressibility, c_f	$4x10^{-6}$ 1/psi
Fracture Spacing	200 ft
Reservoir Thickness, h	200 ft
Horizontal Well length, L_h	200 ft
<u>Matrix</u>	
Permeability, k_m	$1000x10^{-6}$ md
Porosity, $ø_m$	0.05
<u>Fracture</u>	
Width	0.0833 ft
Conductivity	833 md-ft
Porosity, $ø_f$	1
Half Length, x_f	250 ft
<u>Cement</u>	
Width	0.0833 ft
Permeability, k_c	0 md
Porosity, $ø_c$	0

4.3 Fluid and Rock Properties

The fluid contained in the model is under saturated oil with 2,364 psi saturated pressure. Its water properties are set as follows; 63.1 lbs / cu-ft density, 3x10-6 1/psi compressibility and 0.6 cp viscosity. Soave - Redlich - Kwong equation of state (EOS SRK) are used with nine components that represent the reservoir crude oil composition. **Table 4.3** shows the summary of reservoir fluid composition and Soave - Redlich - Kwong fluid description used in the model.

Table 4.4 shows relative permeability curve summary used in the model. Traditional rock relative permeability is assumed applicable to shale reservoir. Saturation exponent of 3 for water and 2 for gas in matrix are used to represents typical saturation exponent of more compact rocks and low pore connectivity from the shale, while 1 and 2 are used for fracture and cement respectively for both water and gas. Saturation exponent 1 represents fracture, which has a higher pore connectivity, while 2 represents cement that is analogous to typical conventional reservoir rock in general such as sandstone. The typical SENSOR simulator master data-file which has all the geometry and properties mentioned in this chapter can be seen in Appendix A.

Table 4.3 Reservoir Fluid Properties and SRK EOS Fluid Description in SENSOR format

```
C =======================================================================
C Fluid Properties
C =======================================================================
PVTEOS SRK
240    ! Reservoir temperature (deg F)

CPT         COMP(MOLE.FRAC)     MW          TC          PC          ZCRIT       SHIFT       AC          PCHOR
  H2S         0.00000000        34.082      672.12      1300        0.28292     0.10153     0.09        80.1
  CO2         0.00305556        44.01       547.42      1069.5      0.27433     0.21749     0.225       80
  N2C1        0.31403421        17.135      331.17      650.14      0.28666     -0.00234    0.01266     69.91
  C2C3        0.17744379        36.69       606.72      659.03      0.27761     0.07656     0.12462     129.88
  C4C5        0.10705390        63.881      793.27      520.66      0.27282     0.11088     0.21674     206.63
  C6C9        0.17600376        104.93      1012.5      438.12      0.30723     0.105       0.31764     286.83
  C10C17      0.14159629        177.2       1216.1      303.63      0.29442     0.14617     0.62464     460.28
  C18C29      0.05844783        308.78      1428.7      199.32      0.29041     0.1885      1.1802      776.07
  C30+        0.02236467        548.58      1634.8      136.8       0.32159     0.16213     2.2561      1351.6
BIN
0.12        0.08125     0.07        0.06        0.03326     0.03        0.03        0.03
0.11246     0.15        0.15        0.10814     0.1         0.1         0.1
0.00288     0.00337     0.00337     0.00337     0.00337     0.00337
0           0           0           0           0
0           0           0           0
0           0           0
0           0
0

C   Lohrenz, Bray, Clark Viscosity  Coefficients
CVISO  0.1023    0.023364    0.058533   -0.0333815    0.00599457

SEP  1
100   150
14.7  60

INITIAL 1
DEPTH
10000    2364.02    ! depth  psat
```

Table 4.4 Relative Permeability Analytical Data

Parameter	Matrix	Fracture	Cement
Connate water saturation, Swc	0.2	0	0
Critical gas saturation, sgc	0.05	0	0
Residual oil saturation to water, Sorw	0.2	0	0
Residual oil saturation to gas, Sorg	0.38	0	0
Rel. perm. of oil, kro	1	1	1
Rel. perm. of gas, krg	1	1	1
Rel. perm. of water, krw	1	1	1
Rel. perm. exponents of oil, now = nog	3	1	2
Rel. perm. exponents of water, nw	3	1	2
Rel. perm. exponents of gas, ng	2	1	2

5 Pipe-It Project

The initial stage of this project is to build Pipe-It project templates to make the running of numerous number of simulation cases easier. Pipe-It is used as a medium for users to easily input data, generate an input file to SENSOR, run SENSOR, tabulate the results and post process the result to get the required calculated data or chart. This chapter will discuss about the workflow process and features of the Pipe-It project templates used during the simulation.

To make the process in this Pipe-It project template easily understood, we will first discuss about graphical elements in Pipe-It that will be covered in every sub-chapter. Each element has a unique symbol so it can be easily recognized.

1. Resources

   Resources are representations of files that we working with. It is associated and linked to appropriate file on computer disk. The shape is rounded rectangle with a pale blue color.
2. Processes (Programs)

   Processes are operations that modifies an upstream resource as input to create a new downstream resource as output. Process consists of executable command line in the form of generic script specified by user. The shape is an ellipse with a pale green color.
3. Connector

   Connectors are graphical lines with single arrowhead denoting computational flow direction. They are used to connect Pipe-It resources and processes. Connectors indicate the order of dependencies of the various resources and processes. The color is black.
4. Composites

   Composites are used to organize large projects with many resources and programs. It can be thought of as folders in the system. It composes parts of a project together into one box for a less cluttered appearance. The shape is rounded pale brown rectangle with double black outline.

All the elements mentioned were renamed to appropriate term in the context of the project. The purpose is to make the workflow become more intuitive and understandable.

5.1 User Interface

The interface is built to fulfill the purpose of helping users to easily sense the process that will be done, the dependencies, key input data and key outcomes from the simulation. **Figure 5.1** shows the graphical user interface display of this template project. At the upper side there is a box called "Key Input Variables" which will make it easier to determine the value of the primary variable in each simulation performed. The value is displayed in red color that symbolizes user input. The next box, called "Key Outcomes" and marked with blue color, symbolizes the main outcomes generated by each simulation. Result displayed in a box outcome is the result of a full well-scale. The value color in each box is intentionally made different to allow the user to identify the origin of these values. This section serves as a place for user to have a quick look on input and results of the simulations being performed.

At the lower side there is a top-level workflow that is formed from all projects' resources that has been simplified into several composites making it easier for users to understand and examine each part of the performed process. This interface is also equipped with the "Instructions" resources containing instructions on how to use this template. In short, here the user can see and modifies each value of the simulation variables by opening the resource "User Input" and look at the chart of simulation results by opening the "Main Output Results (Plots)" resource. The tabulated simulation result can be found in composite "Output Table and Plot" for full simulation time step and composite "Depletion Phase" for the simulation time during the depletion phase. Key result of these two processes are what is shown in the box "Key Outcomes". While the value shown in box "Key Input Variables" is taken from the resource "User Input".

This Pipe-It project template can be utilized for simulating cyclic injection-production and huff and puff injection-production scenario. User can define it in the resource “User Input” by inputting number “1” for cyclic or “2” for huff and puff within variable “EOR Type”.

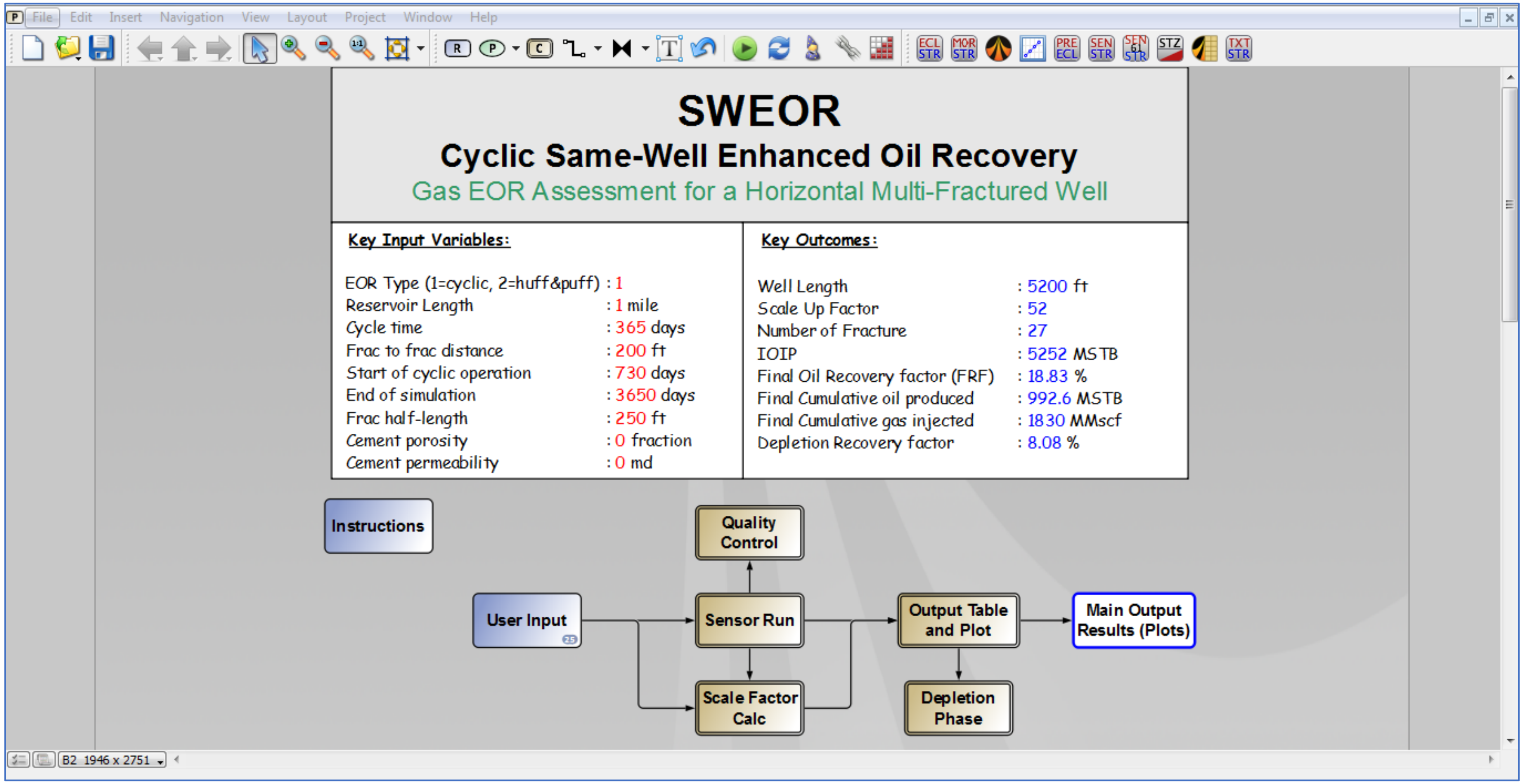

Figure 5.1 SWEOR Pipe-It Project User Interface

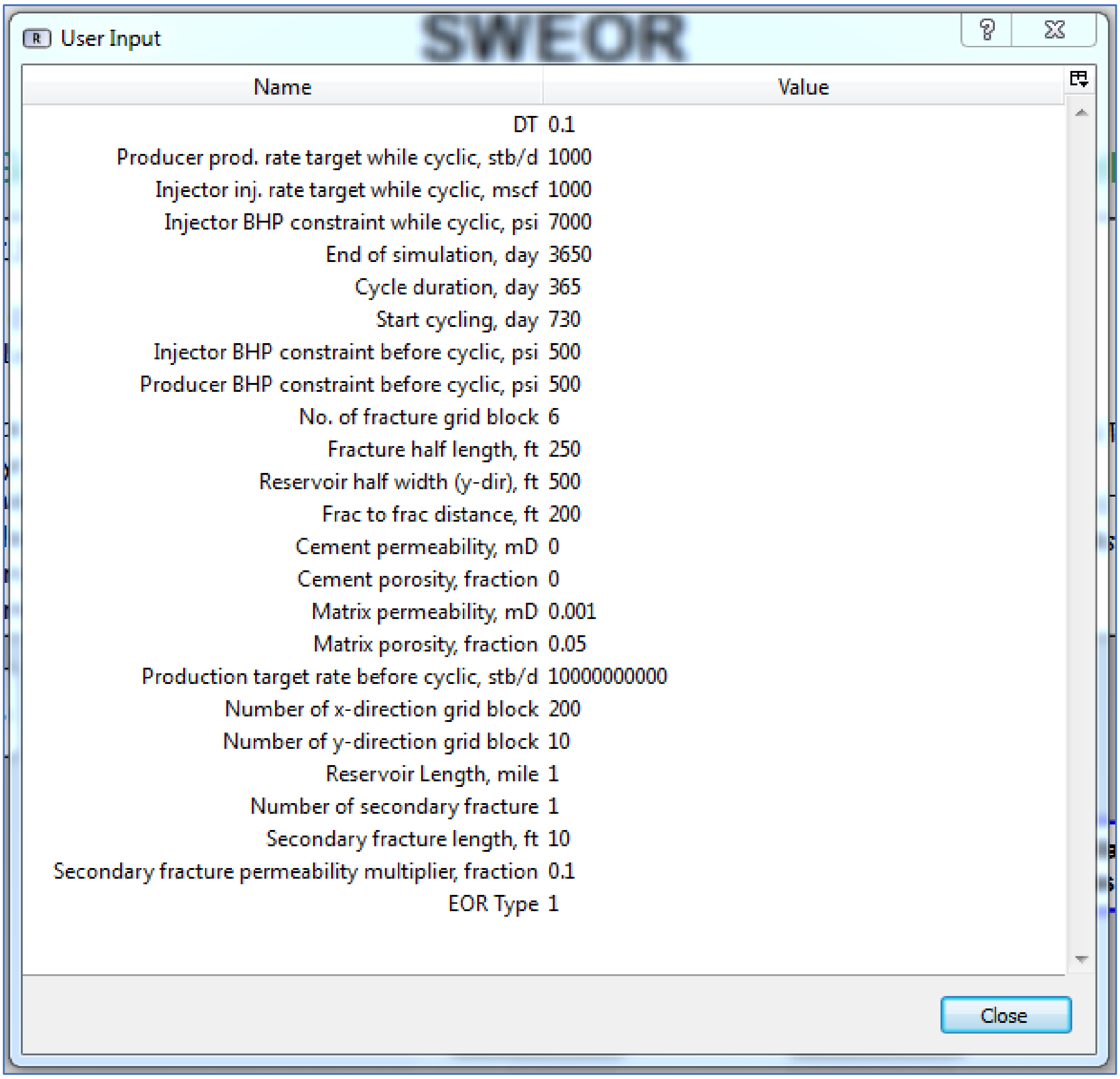

Figure 5.2 SWEOR Pipe-It Project User Input

Figure 5.2 shows all variables that can be modified by the user in the form of reservoir models and simulation scenarios as the input for SENSOR reservoir simulation process later on. All the reservoir geometry variables were initially set to an initial value based on the reservoir base case models as the default form. Except for variables defining simulation scenario, the values are defaulted to cyclic injection-production scenario with 365 days cycle duration because of the author want to express the sense that this template is mainly aim to simulate same well EOR strategy and 365 days simulation will give faster simulation time with clearer plots of the result to help first-time users understand the template. However, the user can freely modify the geometry and properties of reservoir models as well as determining the desired injection and production simulation scenario to be simulated. Here the user can also define secondary fracture as well as its property to be included in the input reservoir geometry for the simulation.

Figure 5.3 shows the schematic illustration of the location and geometry of secondary fractures in the reservoir model. In brief explanation, the user can determine whether there is a secondary fracture or not. If the user wants secondary fracture in the reservoir model, the user can also specify its quantity and length. For the sake of simplification, any value that the user enter will be generated to both hydraulic fracture grids system making it easy to know the effect of adding the quantity and the length, as well as to maintain the assumption of symmetric and mirroring effect that reservoir model used to justify that any pattern simulation geometry and behavior are the same.

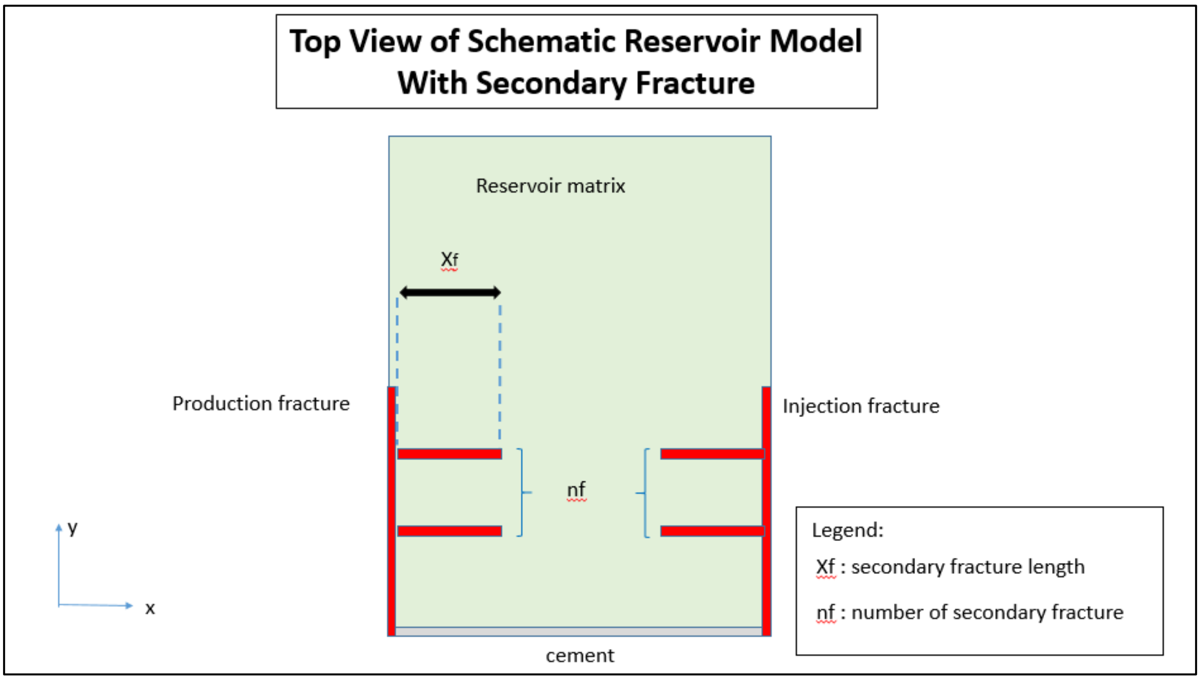

Figure 5.3 Secondary fracture in reservoir model

Figure 5.4 exhibits the display of "Main Output Results (Plots)" resource that is generated from each simulation. The examples shown in figure are typical plots that are commonly generated from cyclic injection scenario.

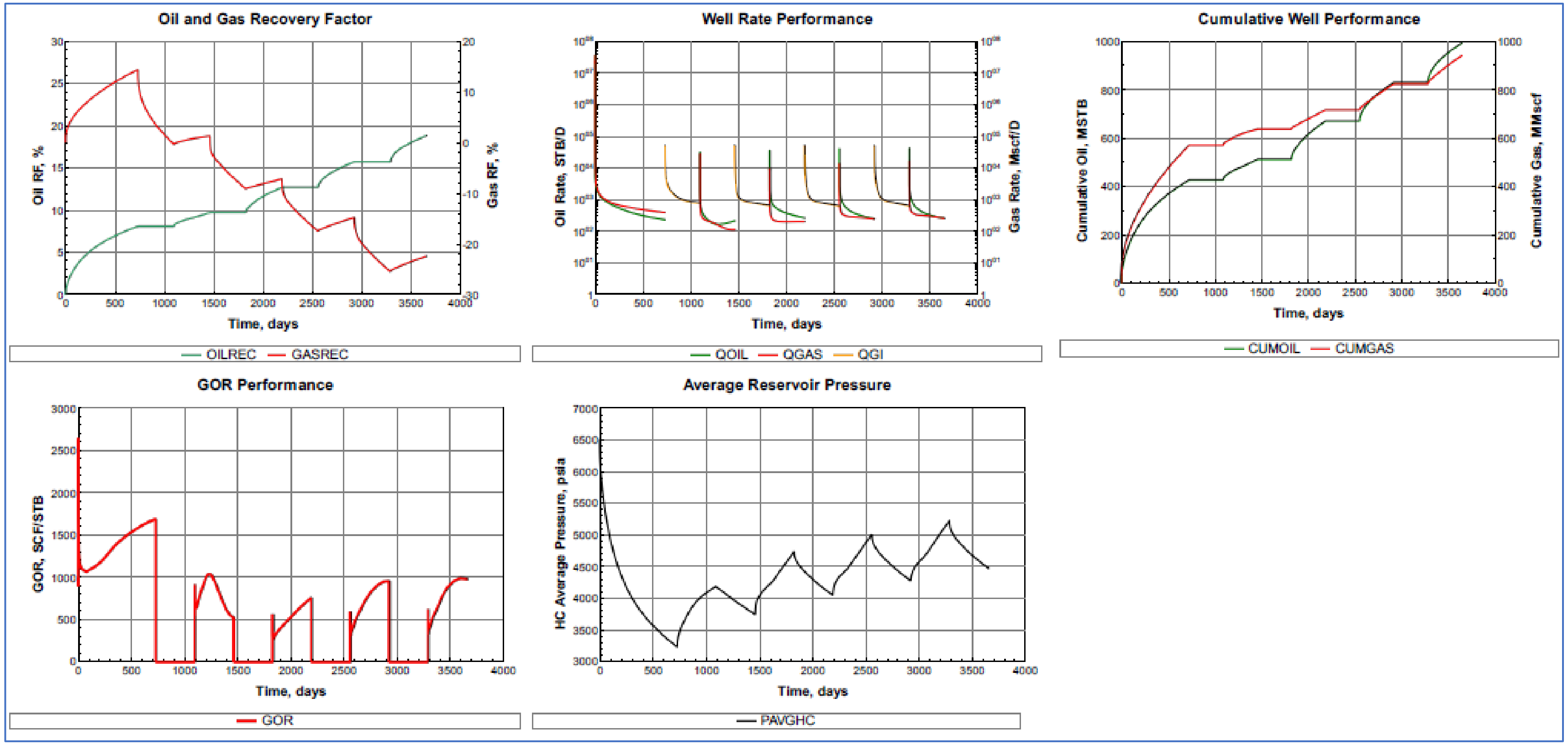

Figure 5.4 Cyclic SWEOR Pipe-It Project Main Output Plots

Charts shown is a key output plots that are considered as the main result that the user look for. On this group of charts, there are a set of recovery factor, well daily rate, well cumulative rate, gas oil ratio and reservoir pressure plot. Charts shown here are the result of a relatively longer cycle duration time which is 365 days, so it clearly exhibits the difference from injection phase to production phase. Gas oil ratio plot is the easiest way to identify each process, in which the value of gas oil ratio will be zero when the well is injecting gas. The longer the duration, the longer the separation and otherwise.

5.2 Sensor Run Composite

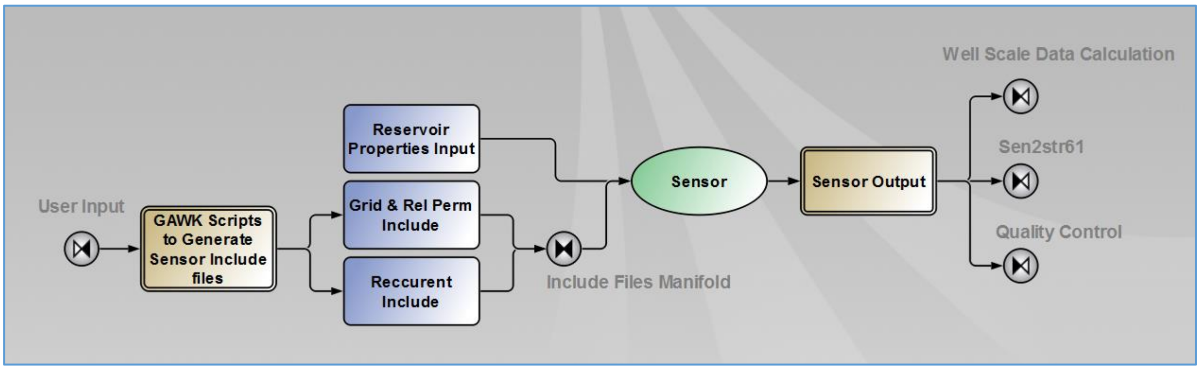

Figure 5.5 Sensor run composite workflow

Figure 5.5 shows the high level workflow of the project's process that is packaged in composite "Sensor Run" and is the main composite of this Pipe-It project template. Within the composite, there are several processes taking place such as process to generate include files which are then used in conjunction with the main data file as simulation SENSOR input, SENSOR simulator process itself, and grouping of simulation output to be finally assigned to each appropriate composite and continue to post processing calculation and tabulation. Used main data file script can be seen in appendix B.

5.2.1 Generating Include Files Composite

Figure 5.6 is the content of the composite "GAWK Scripts to Generate Sensor Include Files" which shows the two separated GAWK operation processes to generate both include files, "Grid & Rel. Perm. Include "and" Recurrent Include" which will be set as the input to the SENSOR simulator. Each GAWK process will receive the input that has been processed in advance in "GAWK Script".

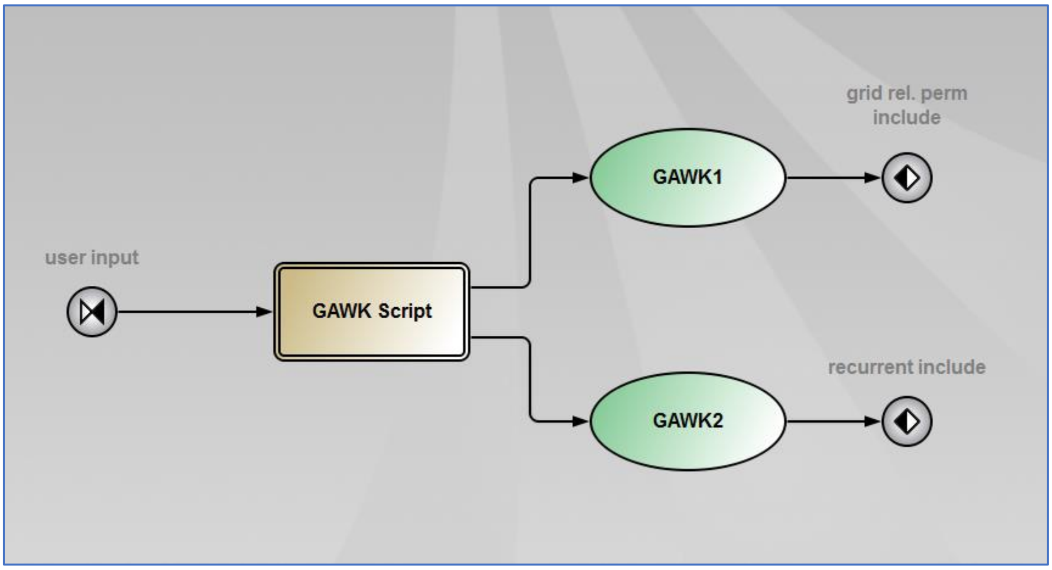

Figure 5.6 Generate sensor include files composite workflow

Figure 5.7 is the content of the composite "GAWK script" which collects scripts to be processed to run GAWK application to generate include files. It points out that in this composite, there is a process called "MapLinkz" to copy values derived from the "User Input" resource to its appropriate script. Each script becomes the input along with an assisting file named "Dummy" to be later run in GAWK process.

Script herein contains algorithms to process raw input data from user input, to produce a set of data in SENSOR simulator data-file format to be simulated. In this script, screening to user input is performed, and type of calculation that should be done is decided based on the reservoir condition or certain simulation scenario from the input data.

Dummy file is an assisting file that becomes a requirement of GAWK command line. Every time GAWK application is running, it will first read and print the contents of this dummy file before it executes the main script. All the GAWK process in this template will be accompanied by its respective dummy file. Some of dummy files in this template contain a header that will be printed together with the desired output from the main script, and some are left blank to produce certain files that are required to follow the desired unique format. The complete

GAWK scripts from this composite can be seen in appendix C entitled GAWK1 Grid & Rel.Perm Script and GAWK2 Schedule Script.

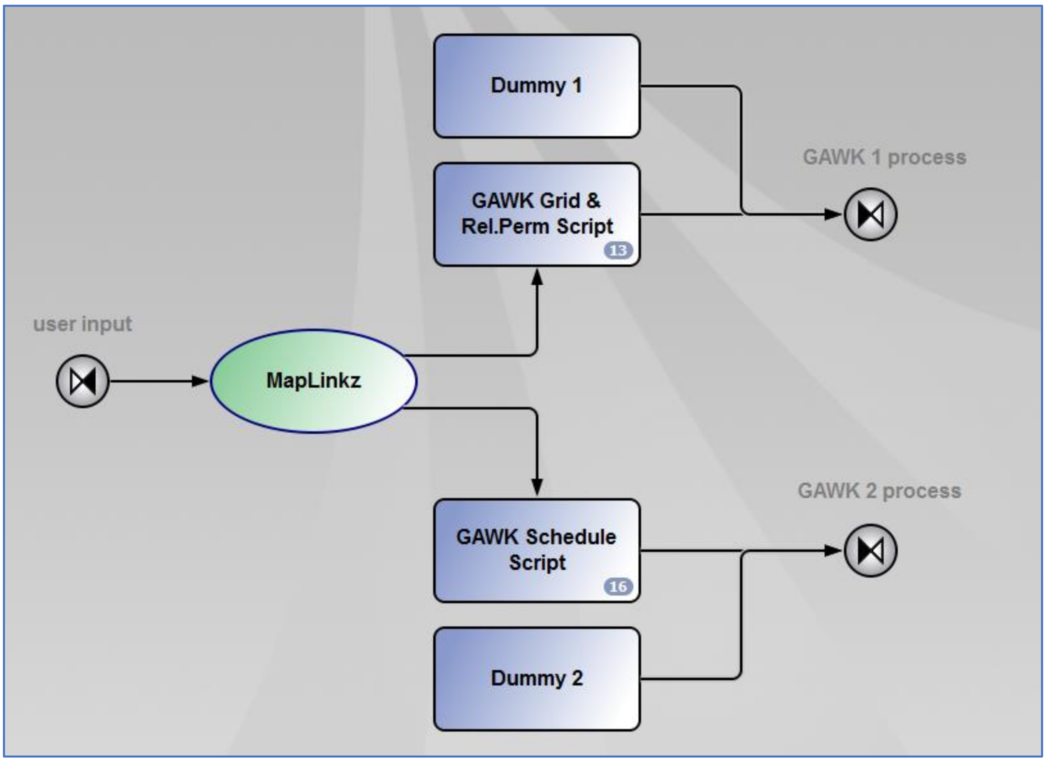

Figure 5.7 GAWK script composite (to generate sensor include files) workflow

5.2.1.1 Reservoir Grid Properties Section Include File

In this template, there are two include files generated based on the input that the user defines. One of them is called "Grid & Rel. Perm. Include ". This file contains reservoir model geometry data and properties in the SENSOR simulator format. Geometry data contained in this file are reservoir gridding number and grid size, with also the matrix, cement and fracture system gridding details. While the grid properties contained in this file are matrix, cement and fracture system grid's rock type, permeability and porosity.

5.2.1.2 Recurrent Section Include File

Another include file generated based on user input is called “Recurrent Include”. This file contains data of well completion, well control, well constraint, simulation scenario and simulation scheduling in SENSOR simulator format. It covers such as fracture grid position that is assigned as injection and production point, both well injection and production target rate and bottom hole pressure constraint, and also injection and production schedule.

5.2.2 Sensor Output Composite

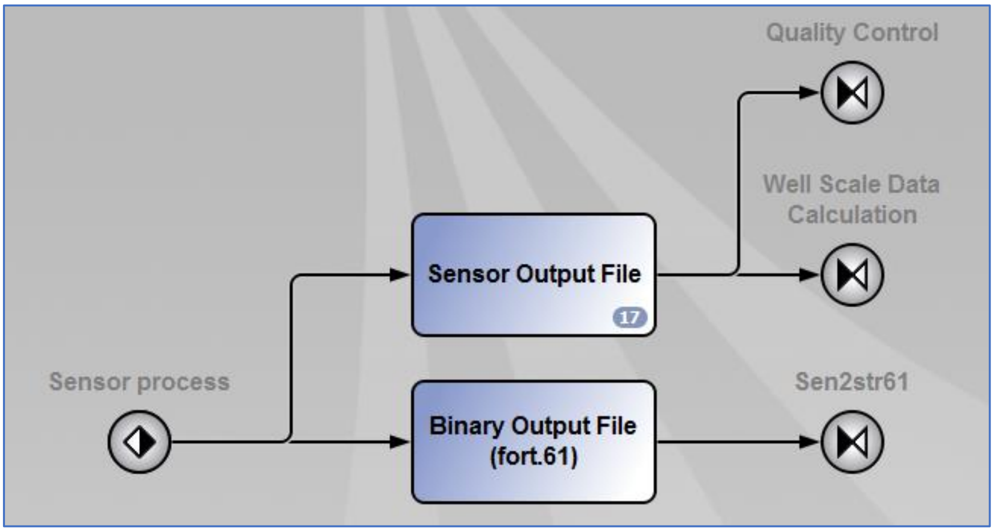

Figure 5.8 Sensor output composite workflow

Figure 5.8 is the content of “Sensor Output”, the place which collects simulator output files needed by the template to post processing the result. SENSOR main output file (.out) will be used to calculate well-scale data while SENSOR binary output file (fort.61) will be processed in generating result’s table plots. Sensor main output file can also be used to perform quality checking on the project template if there is any modification on the template, shown by the connector moving to the composite “Quality Control”.

5.3 Scale Factor Calculation Composite

Figure 5.9 displays the workflow of "Well Scale Calculation" composite. The main process in this composite is to calculate well segmentation based on user input. Well segment data generated will be used in conjunction with the main SENSOR output file to calculate the well-scale result in "Well Scale Data Calculation" composite. In addition, the well segment generated will also be used as input to calculate the well-scale basis simulation result in the "Output Table and Plot" composite.

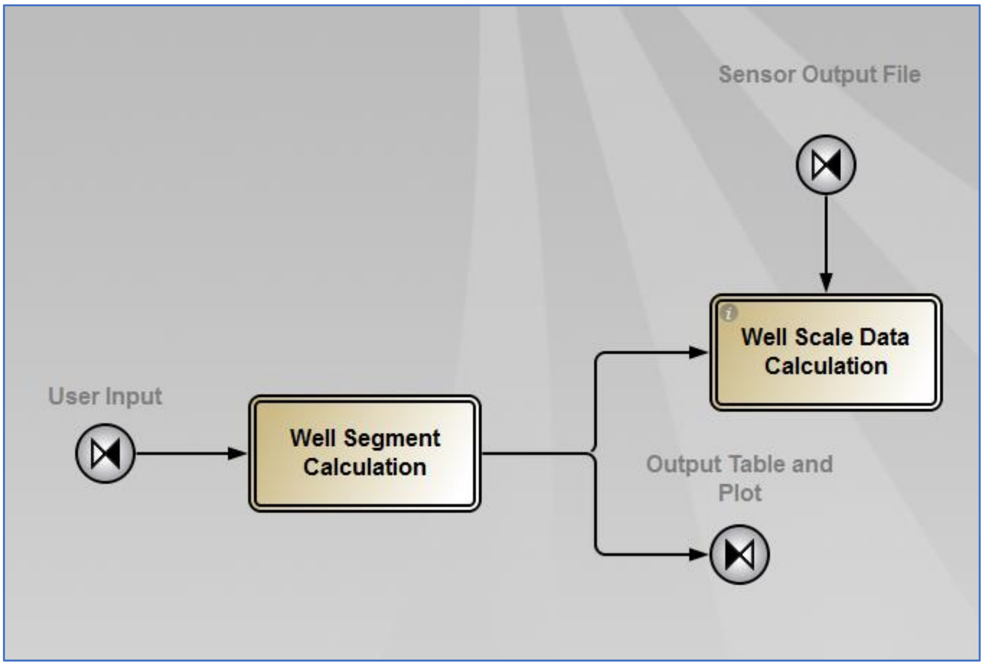

Figure 5.9 Scale factor calculation composite workflow

Figure 5.10 is the content of the composite "Well Segment Calculation" which portrays the workflow of well segmentation calculation process based on user input. In short, after reading the user input on reservoir length, this process will calculate maximum pattern quantity that can be generated based on the inter-fractures distance defined in user input. GAWK process in this composite will convert the unit and rounding the maximum pattern value that was previously generated, then produce well-scale factor to be copied to composite "Output Table and Plot".

This well-scale factor will also be copied to another GAWK script to be transferred to composite “Well Scale Data Calculation” for calculating full well-scale simulation result along with input from SENSOR main output file, like what is shown in **figure 5.11**. The complete GAWK scripts from this composite can be seen in appendix C entitled GAWK4 Scale Up Factor Script, GAWK5 Well Data Smoothing Script and GAWK6 Well Scale Calculation Script.

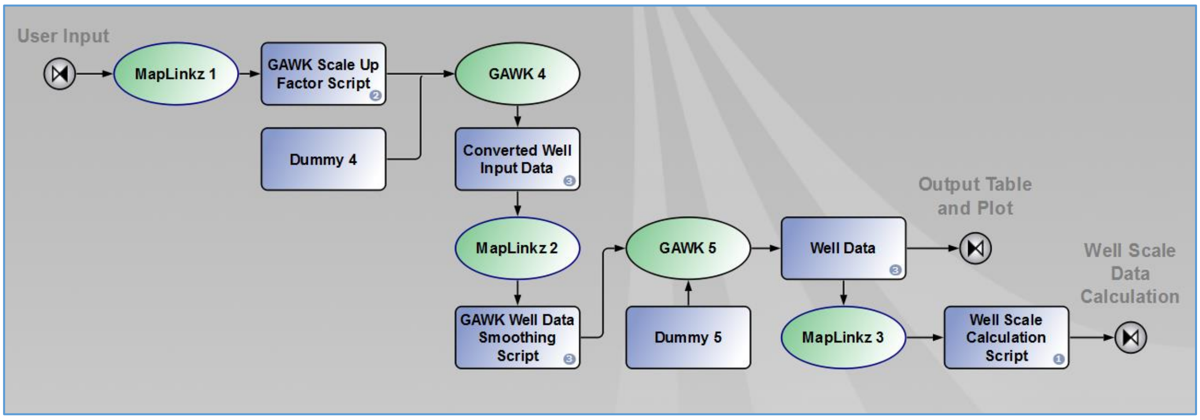

Figure 5.10 Well segment calculation (to generate scale factor) composite workflow

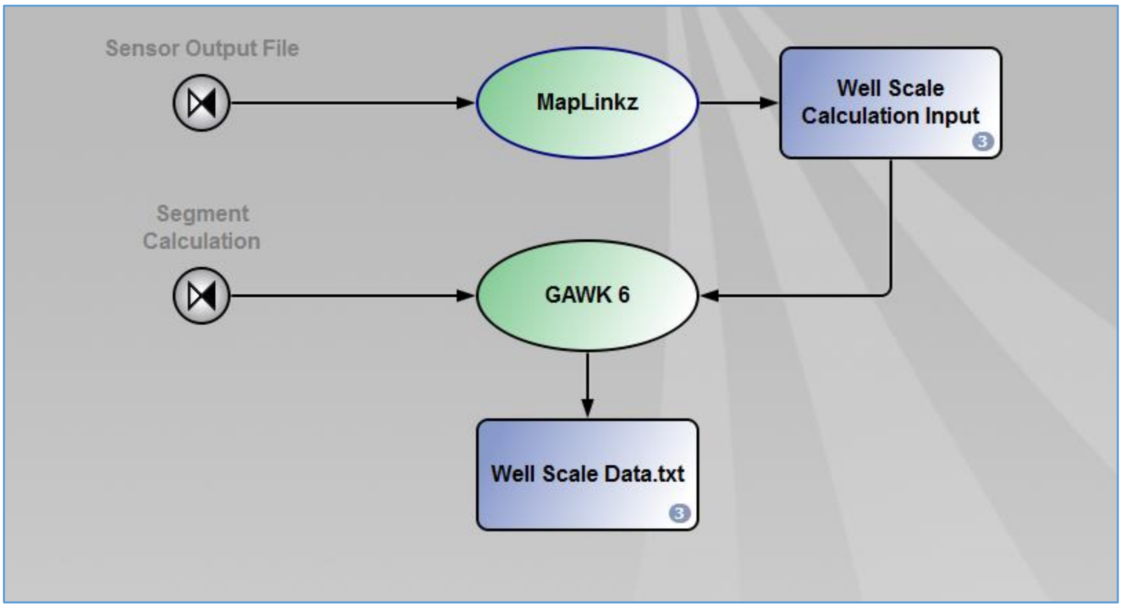

Figure 5.11 Well-scale data calculation composite workflow

5.4 Output Table and Plot Composite

Figure 5.12 is the content of the composite "Output Table and Plot". In this composite, SENSOR main output file will be tabulated using the process "Sen2Str61". The process is a powerful Pipe-It feature, in which output binary file (fort.61) of sensor simulator will automatically be converted to the table form. Then by using the scale up factor that calculated in composite "Scale Factor Calculation", it will calculate and tabulate the simulation result using a process called "Strexzel" in a full well-scale basis that becomes the source for "Plotz" process. Plotz is a program that will generate the chart as the end point output of this Pipe-It project template, and it will be stored in resource "Main Output Results (Plots)".

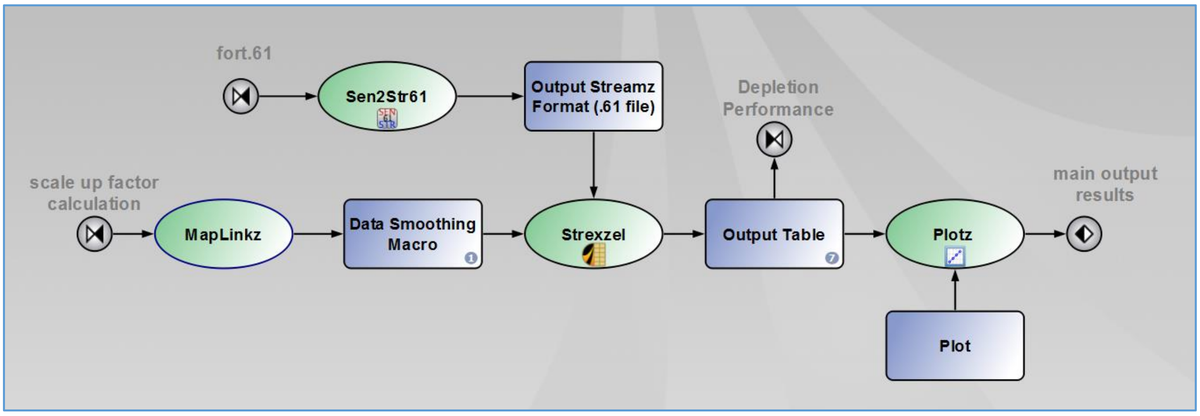

Figure 5.12 Output table and plot composite workflow

5.5 Depletion Composite

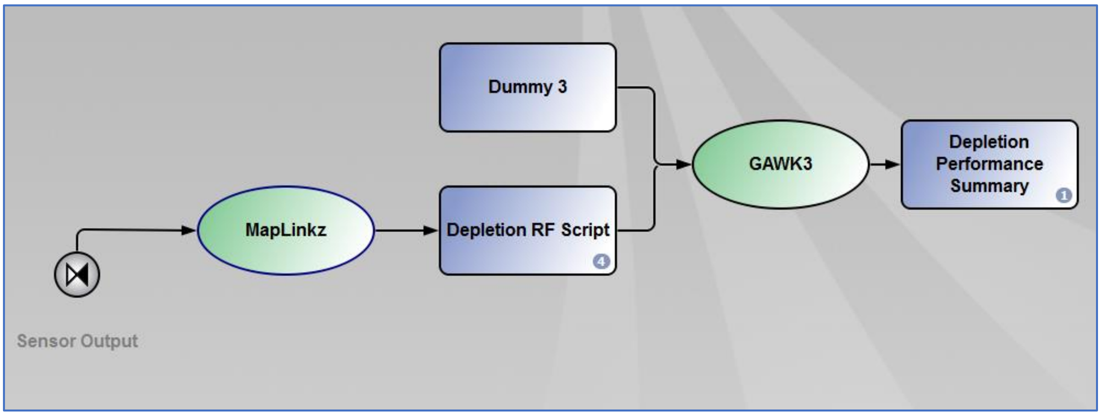

Figure 5.13 Depletion phase calculation composite workflow

Figure 5.13 is the content of the composite "Depletion Phase" which describes the workflow of depletion-phase performance calculation. In this composite, calculation of simulation performance for depletion periods takes place that covers the natural flow production period before EOR begins. In this composite, the depletion phase performance result in the main SENSOR output will be copied using "Maplinkz", and then entered into GAWK script to calculate full-well scale result. The complete GAWK scripts from this composite can be seen in appendix C entitled GAWK3 Depletion RF Script.

5.6 Quality Control Composite

Figure 5.14 is the content of the composite "Quality Control", where the process of checking the reliability of used Pipe-It project is performed. Author has prepared three cases as the reference to check simulation results. Users can easily drag and move connector to the case that they want to check. Each case contains some simulation results from SENSOR reservoir simulator that is previously run without Pipe-It. Simulation result generated from the run using Pipe-It will be automatically compared to the selected simulation result reference, and relative error of each displayed result's variable will be calculated.

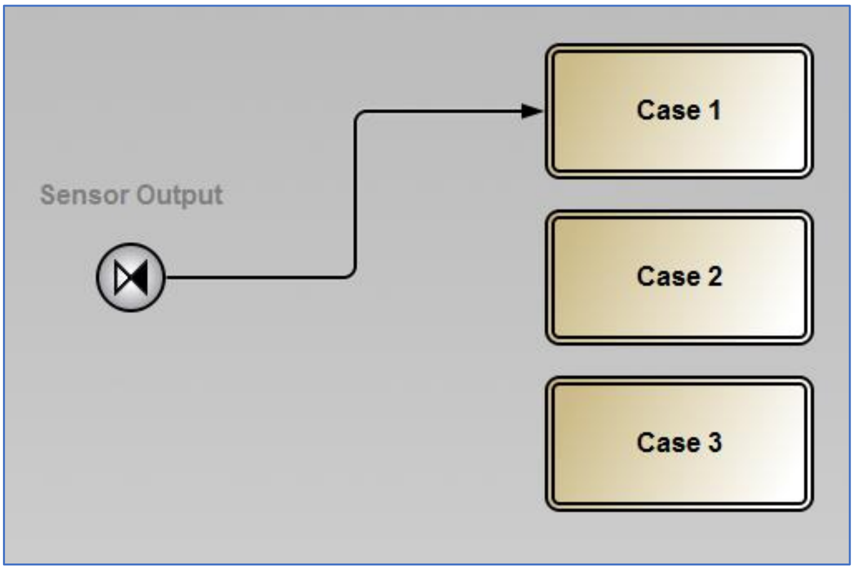

Figure 5.14 Quality control composite canvas

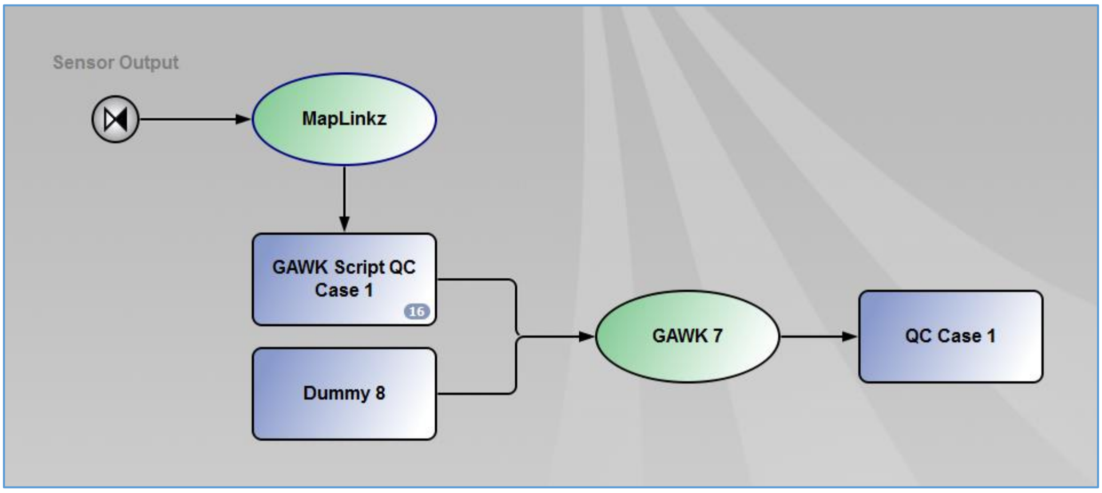

Figure 5.15 Case 1 (for quality control) composite workflow

Figure 5.15 is the typical workflow of each "Cases" composite. Briefly, SENSOR main output will be copied to GAWK script through "Maplinkz" process. The next process is calculating relative error and comparison between two cases, Pipe-It result against reference case result, that all are tabulated and stored in resource "QC Case". User can look at the comparison by opening the resource file. Since all GAWK scripts for all quality control checking are the same, the complete script is only displayed for one case which can be seen in appendix C entitled GAWK7 Script QC Case 1.

5.7 Pipe-It Project Template Features

This Pipe-It project template has some powerful key features that help user in doing a simulation for a case, such as:

- First run screening will read error or inappropriate values from user input and force the simulation to stop, resulting error message about the problem.
- There is case matrix template for three quality checking cases that ease user to automatically run the simulation in similar condition with the reference case and compare it in "Quality Control" composite.
- The generated simulation has been in the form of full well-scale result and displayed in key outcomes and plots.
- Depletion or natural flow period results before cyclic simulation start are also calculated and shown in key outcomes.
- Simulation is not limited to complete scheme (depletion and cyclic injection), but it will also accept simulation of only depletion or only cyclic injection scenario.
- This template can also simulate Huff and Puff scenario as an alternative.
- User can define the location of secondary fracture, as well as the length and the intensity.

Simulation performed by SENSOR reservoir simulator in this template is based on the model of a quarter reservoir displacement pattern that called a segment. However, the key feature of produced result is full well-scale results. This well-scale results are the outcomes from multiplying the simulation result with the scale factor. Scale factor here is a number that represents the maximum number of segment that can be generated from the given reservoir length. Consider that the length of a segment is equal to inter-fractures distance, the scale factor then can be calculated with the expression below:

Assume that,

a	: reservoir length, ft	c	: well length, ft
b	: inter-fractures distance, ft		

Then,

$$Scale\ Factor = \frac{c}{b}$$

Where c is the result of rounding down $\frac{a}{b}$, in order to also get the rounded number of segment.

Figure 5.16 can help to easily understand the process of how the full well-scale data is obtained. Reservoir length and inter-fractures variable distance can be specified in user input. When both value are known, the algorithm prepared in GAWK script will calculate the maximum length of the well section that can be modeled, which contains a number of uniform pattern throughout the well section. The total of these patterns is called scale factor, and it becomes a multiplier to convert pattern performance to well performance.

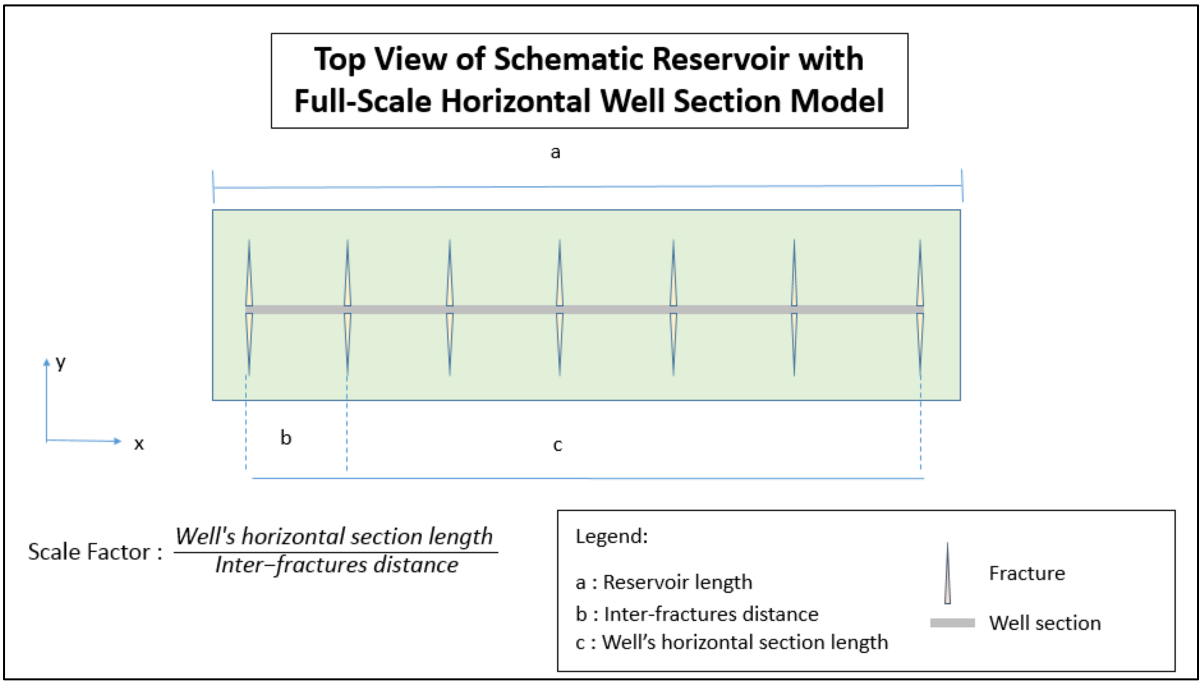

Figure 5.16 Well's horizontal section illustration

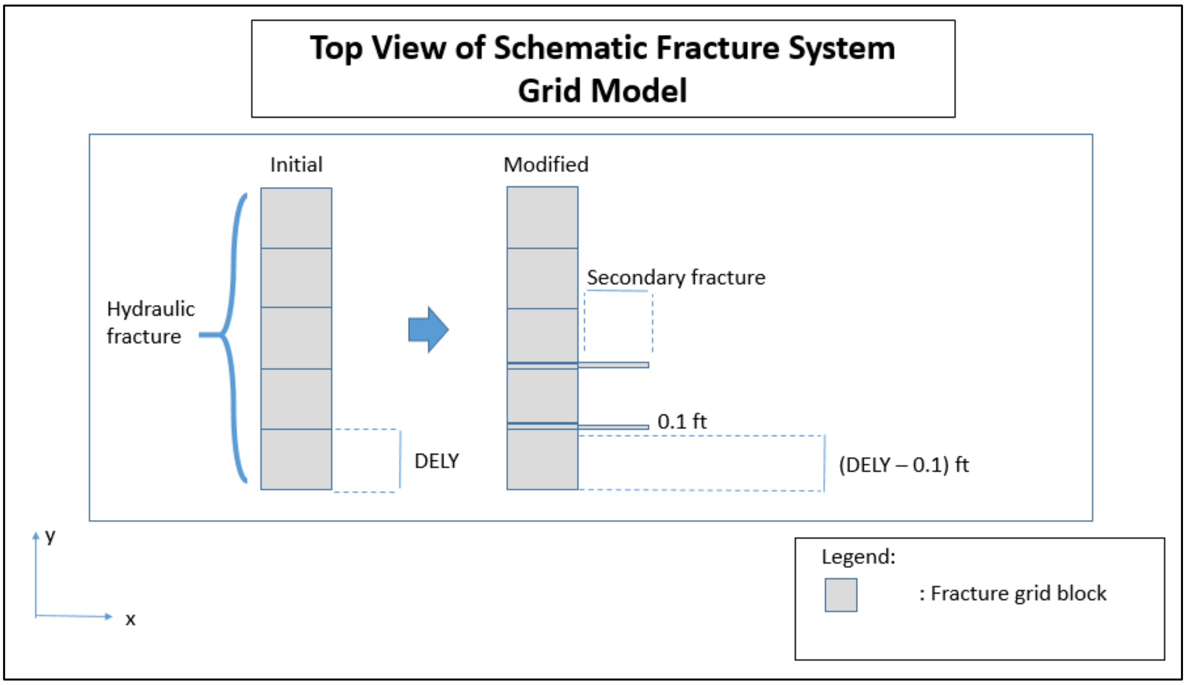

Figure 5.17 Fracture system illustration

Figure 5.17 illustrates how key process takes place when user input the secondary fracture. In the default case, a pattern had a pair of hydraulic fracture grid block with modifiable quantity and size. Secondary fracture is a grid block with fixed width of 0.1 ft toward the y direction, extending toward the x direction in accordance with the specified length. Maximum number of secondary fractures in a model cannot exceed the number of its main hydraulic fracture, besides the cement layer (y = 1).

By the existence of secondary fracture in reservoir, it will be followed by changes in the number of grid and the length in y-direction. In order to understand the process, the created grid model modification will be better explained through the following expressions:

Assume that,

Ny_1 : initial total number of grid in y-direction

Ny_2 : updated total number of grid in y-direction

Nyf_1 : initial total number of fracture grid in y-direction

Nym : initial total number of matrix grid (beyond fracture) in y-direction

Nyf_2 : updated total number of fracture grid in y-direction

$DELYF_1$: initial hydraulic fracture grid width (y-direction), ft

$DELYF_2$: updated hydraulic fracture grid width (y-direction), ft

x : number of secondary fracture

FL : hydraulic fracture half length, ft

Given that the secondary fracture length is set fixed at 0.1 ft then:

$$Ny_2 = Ny_1 + x = Nyf_2 + Nym$$

$$DELYF_2 = DELYF_1 - 0.1\ ft$$

Where,

$$Nyf_2 = Nyf_1 + x$$

$$FL = Nyf_1 * x = (Nyf_2 * DELYF_2) + (x * 0.1\ ft)$$

$$= constant$$

Each secondary fracture defined by the user would add a single grid block with a 0.1 ft width to represent small fractures that can occur in this type of fracture. So that the width (y-direction) of original hydraulic fracture block affected was reduced by 0.1 ft, to maintain the value of hydraulic fracture half-length on a previously defined reservoir model. Placement of secondary fracture moved from first main hydraulic fracture that was closest to the wellbore (y = 2), and increased in the direction of hydraulic fracture block that was away from the wellbore (y> 2) at each of the addition.

To easily understand the main process of this Pipe-It project, **Figure 5.18** features simplified template's complete-workflows that illustrates the key ideas and processes every time simulation is performed. This flowchart is the summary of discussion elaborated about the developed Pipe-It project template in this chapter.

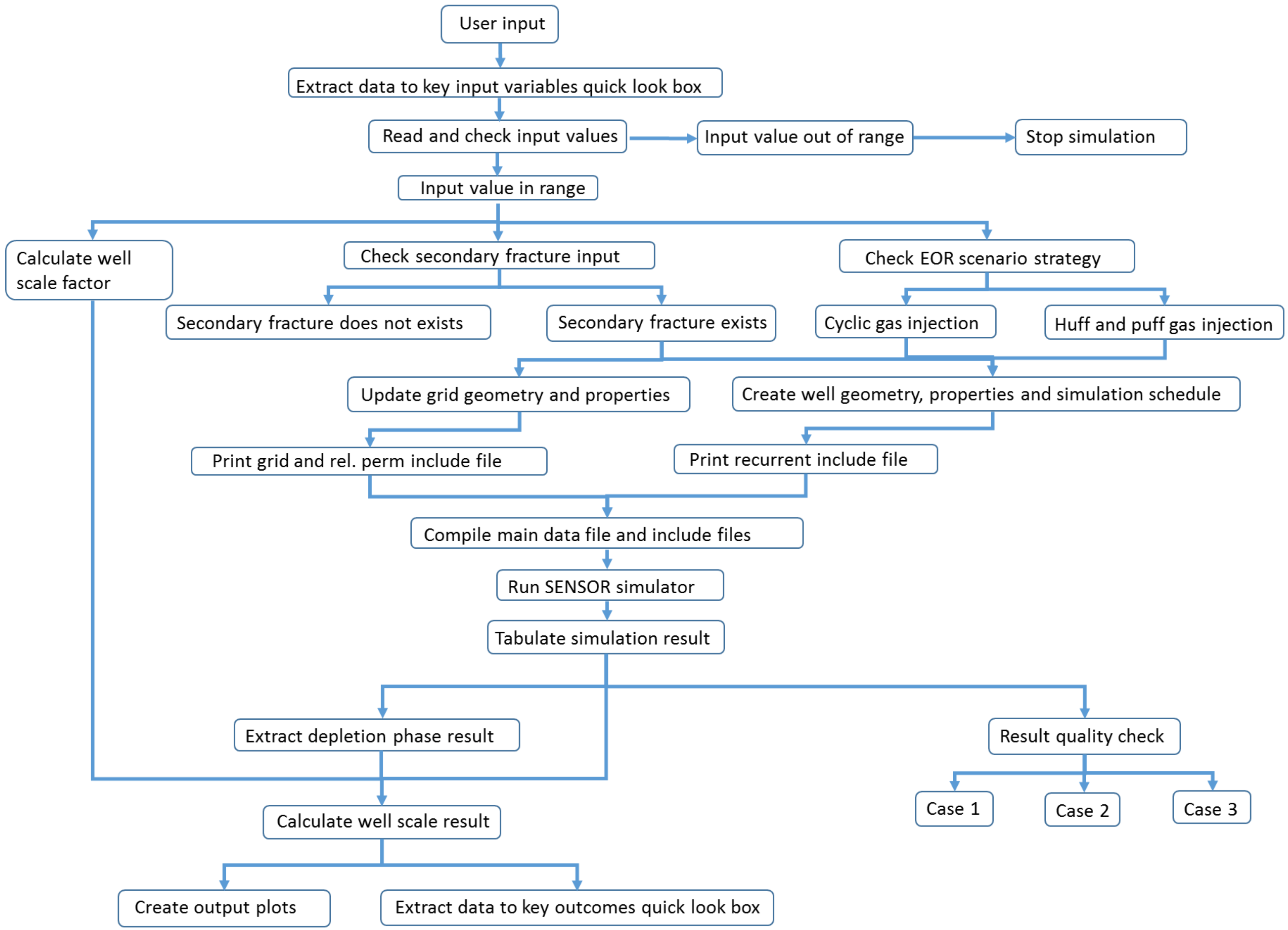

Figure 5.18 Pipe-It project template simplified full-process flowchart

6 Default Reservoir Model

Before starting the simulation study, adjustment was carried out to create appropriate reservoir default model to be used further. The default model will be set to become the one that was initially there in Pipe-It project template. The used scenario for the study in order to search the appropriate model was by producing the reservoir naturally for 2 years, continued by applying either continuous or cyclic gas injection EOR in the 3^{rd} to 10^{th} year. However, this section did not discuss about the physical phenomena behind the performance curve generated from the simulation, but only showed the comparison of several visible parameters during the simulation.

6.1 Perforation/ Connection Grid

Reservoir model employed had a pair of fracture system that is located opposite each other. Fracture was intended to represent a condition in which stimulation to the reservoir had been performed by improving productivity and connectivity between the wellbore and the reservoir.

Injection fracture aims to increase the reservoir injectivity. Based on that assumption, injector perforation was assigned to each injection fractures grid to better represent the injectivity improvement. For the production fracture, connectivity obtained was to facilitate the oil to flow into the wellbore. It cannot be directly concluded whether producer perforation can be assigned to all producer fractures or not.

Sensitivity study aims to see the difference in producer perforation assignment in production fractures grid. Two different reservoir model was prepared for this study. The first model just defined first production fracture grid as the injector, while second model assigned all production fracture grids as the injector.

As seen from the **figure 6.1** and **figure 6.2**, almost all generated curves overlay each other especially in oil recovery and average pressure which show exactly the same value. These graph indicate that the way we assign producer perforation in the reservoir model is not sensitive to the reservoir performance and the well ability to produce oil.

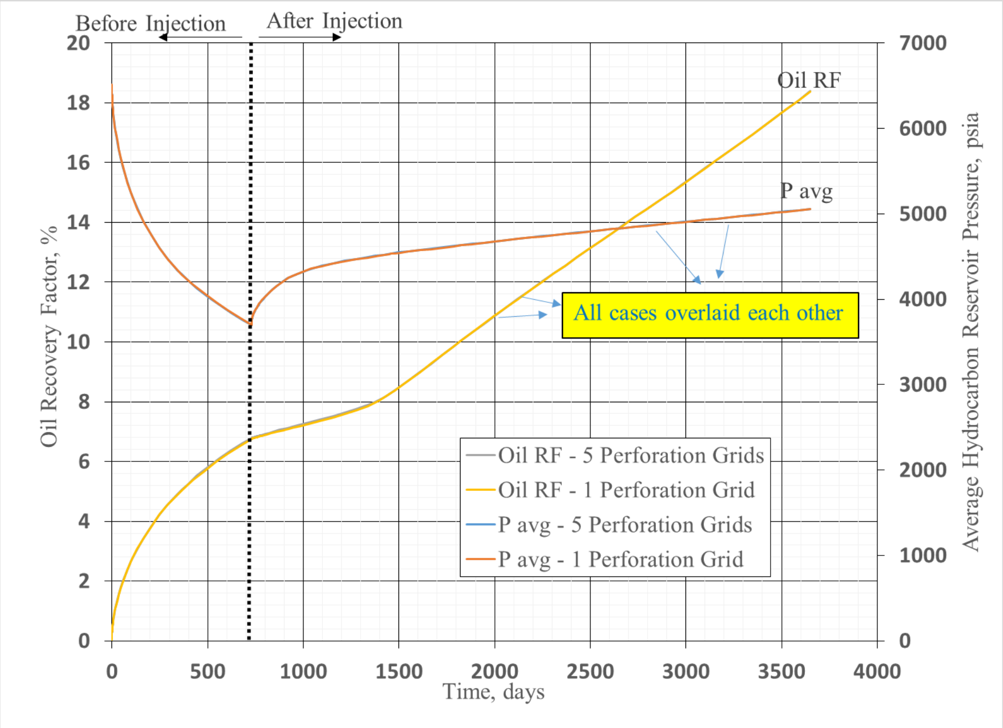

Figure 6.1 Sensitivity result of production fracture adjustment in terms of oil recovery factor and hydrocarbon pressure average (continuous injection and production scenario with 200 ft hydraulic fracture half length)

The differences are not so significantly seen in **figure 6.2**. In initial days of the simulation, it exhibited a little separation between these cases. Initial production rate was higher in the case with only one producer grid, but it experienced an extreme decline that made the rate equal to the other case within hours. Because this difference was very small and it happened in very short time, it could be considered negligible.

This result made sense because oil suction process does actually only occur in the wellbore valve. Unlike the gas injection process, displacement occurs in every fracture grid because it is supported with huge conductivity, making similar gas rate and pressure to all fracture grids. Considering this reason, the default model was set by assigning producer perforation only in the first production fracture grid and injector perforation to all injection fracture grid.

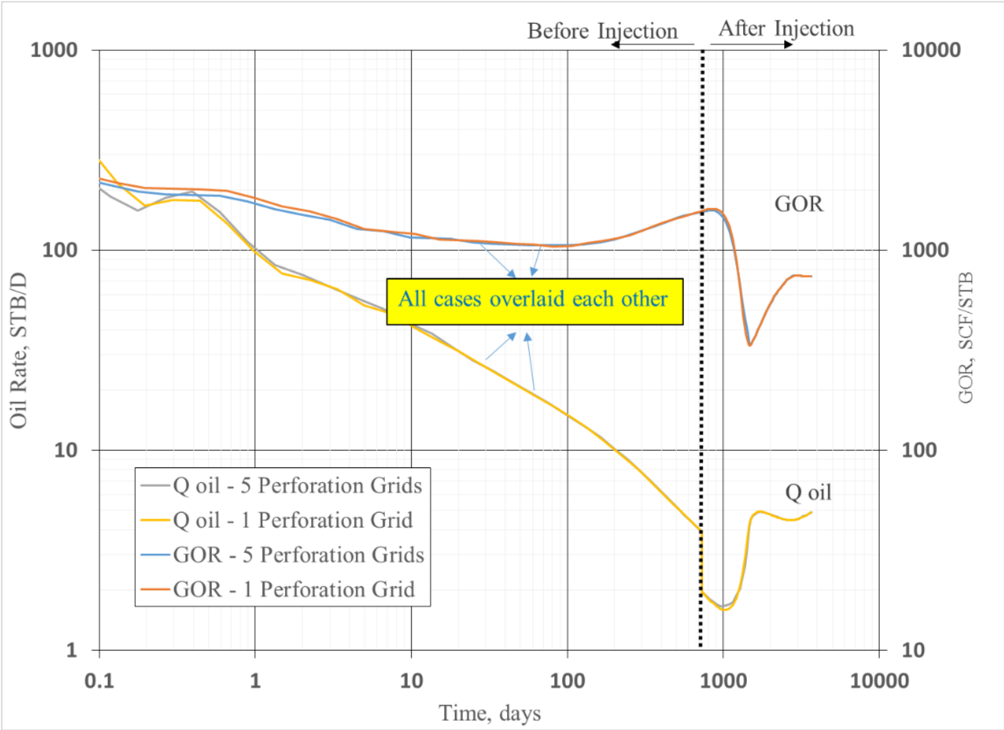

Figure 6.2 Sensitivity result of production fracture adjustment in terms of oil rate and GOR (continuous injection and production scenario with 200 ft hydraulic fracture half length)

6.2 Reservoir Gridding

Author did the gridding adjustment since based on previous researchers' conclusion, local grid refinement is required in fractured reservoir model to create more adequate representation of flow around the fracture. They said that shrinking the grid length around the fracture can help to achieve the objective.

Three models with different x-direction grid length $(\Delta x)m$ were prepared. The first model used $(\Delta x)m = 1$ ft, second model used $(\Delta x)m = 0.5$ ft, and the last model used $(\Delta x)m = 0.33$ ft. As seen from **figure 6.3** and **figure 6.4**, changing the $(\Delta x)m$ did not affect the reservoir simulation performance significantly. Clearly, there was no difference for all three cases in terms of recovery factor and reservoir pressure since all the curves exactly overlaid each other. There was only small difference in early time of GOR and production rate profile.

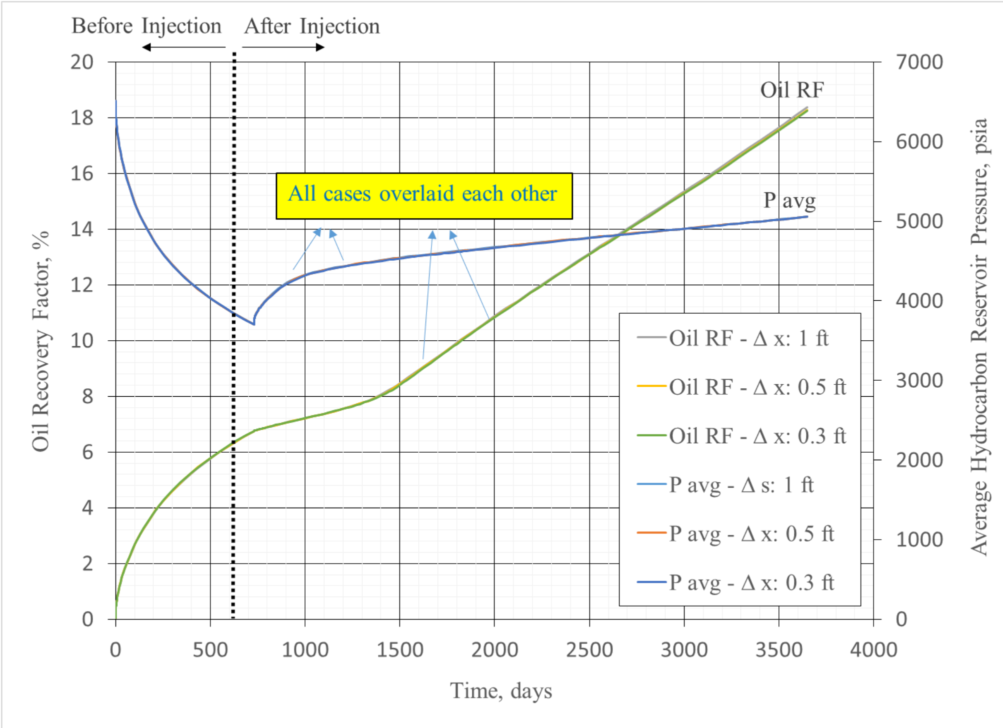

Figure 6.3 Sensitivity result of (Δx)m gridding adjustment in terms of oil recovery factor and hydrocarbon pressure average (continuous injection and production scenario with 200 ft hydraulic fracture half length)

Figure 6.4 shows that oil production and decline rate is higher on the first day for smaller gridding cases. This is possible because of the more detail modeling of pressure drop in the area around the fracture, causing the slightly faster depletion than the larger gridding case.

But since it only appears on the first day and does not make significant difference to the production accumulation, author chooses to use (Δx)m = 1 for simplicity reasons as well as considering the simulation time needed for doing the future simulation with layered model, which would double the number of total reservoir grid blocks in the model.

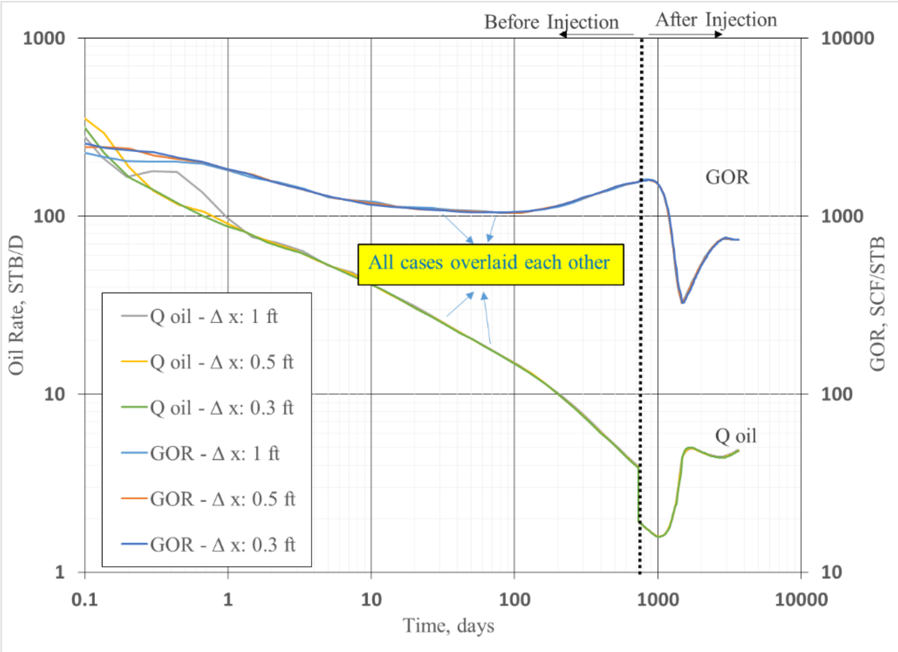

Figure 6.4 Sensitivity result of (Δx)m gridding adjustment in terms of oil rate and GOR (continuous injection and production scenario with 200 ft hydraulic fracture half length)

6.3 Calculation Solver

Considering that each solver has its own advantages and disadvantages to the numerical simulation, then determining the correct solver is very essential to maintain simulation efficiency and quality.

Four models with different calculation solver are prepared. The first model use the default solver RBILU 1, the second model use D4, the third model use nested Factorization (NF) and the last model use ILU 6. This sensitivity study only modifies the solver but still uses fully implicit method.

Figure 6.5 and **figure 6.6** shows that changing the calculation solver is insensitive to simulation performance. All curves overlaid each other. Like other sensitivity performed previously, the difference was only shown in early simulation time. Again, it was considered as negligible since almost all the lifespan profile looks overlaid each other.

Table 6.1 tabulates each solver's performance in terms of CPU time and numerical iterations. In overall, all solvers showed quite similar performance. Considerably small differences was only seen in CPU time, in which NF worked faster than others but this difference was insignificant.

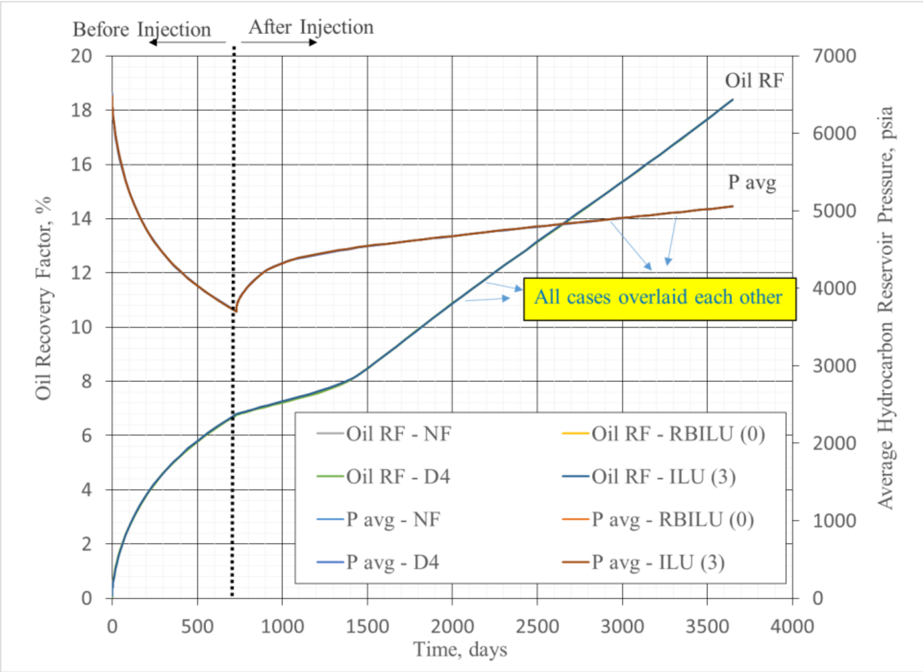

Figure 6.5 Sensitivity result of solver adjustment in terms of oil recovery factor and hydrocarbon pressure average (continuous injection and production scenario with 200 ft hydraulic fracture half length)

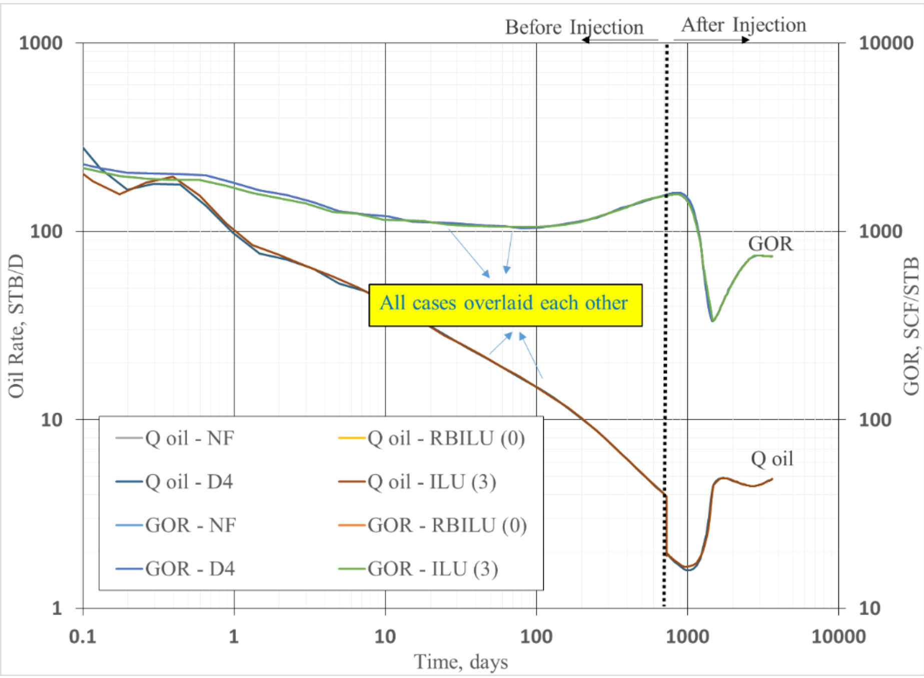

Figure 6.6 Sensitivity result of solver adjustment in terms of oil rate and GOR (continuous injection and production scenario with 200 ft hydraulic fracture half length)

Table 6.1 Solver Performance Comparison

Case	Solver	CPU Time (sec)	Time Step	Cuts	Newton Iteration	Solver Iteration
1	RBILU (0)	78.36	165	3	249	1090
2	NF	42.65	165	3	249	843
3	D4	83.74	165	3	249	0
4	ILU (3)	119.09	165	3	249	835

After considering the sensitivity result, author decided to use RBILU (0) as the default calculation solver for the simulation. This solver does not require longer CPU running time so it is efficient enough to simulate the heavier case and also more reliable than D4 and NF. ILU (3) was not selected since the reservoir model used is not so complex that default ILU is already good enough to be applied.

6.4 Gas Injection Start Day

Before gas injection EOR begins, the reservoir will be produced naturally at the first time, or called as the depletion phase. The timing of gas injection commencement is considerably critical because it will determine the final oil recovery factor, as well as the effectiveness of EOR implementation. Normally, the faster the gas injection begins, the greater oil recovery will be produced, while the later the gas injection started, the smaller final oil recovery will be resulted. However, if we see it in more detail, gas injection start day can begin after several years of natural production for operational savings.

Seven simulation cases which has different gas injection start are prepared for this sensitivity study. Simulations are performed to see the effect of delaying gas injection start from a year to four years. **Figure 6.7** shows the result in terms of final and depletion oil recovery factor. It is seen that the decline trend of final oil recovery factor is relatively low in four initial points, but then sharpen at the remaining points. In contrast, to the depletion oil recovery appears to be fairly stable from the first to the last point along with the delay in gas injection start time.

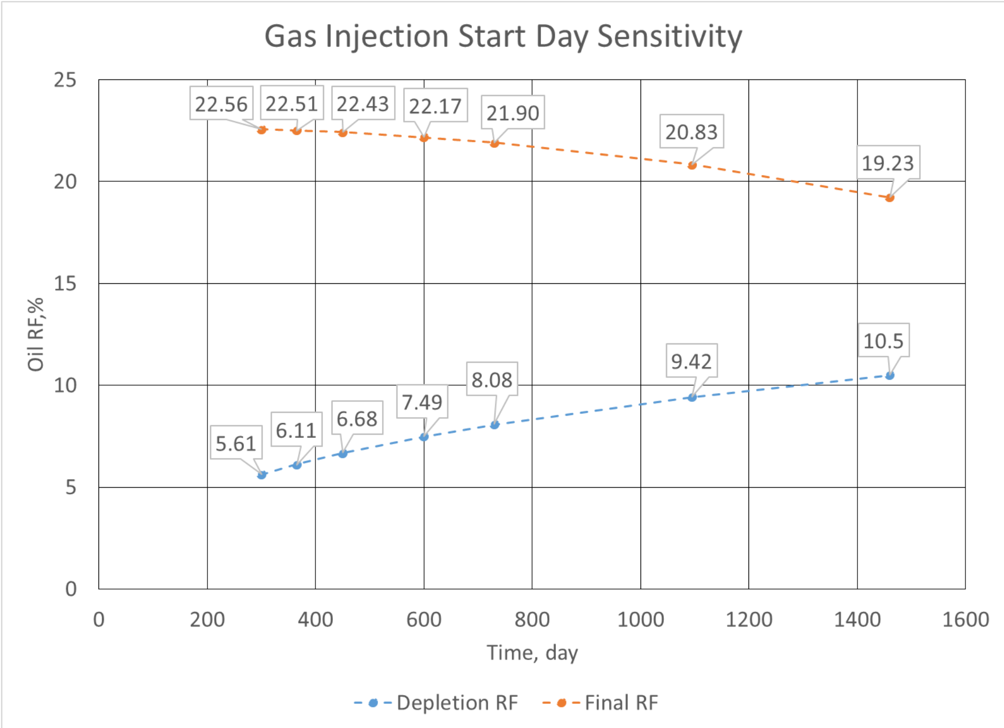

Figure 6.7 Sensitivity result of cyclic simulation start day (continuous injection and production scenario with 250 ft hydraulic fracture half length)

Consider the delta oil recovery result that was previously mentioned, the author concluded that postponing the time for 2 years is the most optimum time to begin the gas injection. It is since the reduction of final oil recovery factor of up to two years was relatively small, compared to the longer delay. Reduction of final oil recovery from one year to two years was only about 0.5%, while the delay until the third year result exceeded 1.5%. Meanwhile, the addition of depletion oil recovery looked fairly stable each year, in which each year delay would produce an increase of about 1.5% in depletion oil recovery. That's why the default simulation scenario applied was to start the gas injection at the beginning of the third year.

6.5 Injection Pressure

Sensitivity was performed to look at the optimum pressure gas injection. Nine case with different gas injection pressure control were prepared. Injection pressure started from the number that are considered suitable and closest to the initial reservoir pressure, which was 7000 psi to 10,000 psi. **Figure 6.8** shows the simulation result with different gas injection pressure in terms of final and delta increment oil recovery. Delta oil recovery increment here means the difference between oil recoveries from the case with the higher pressure to the starting case which is 7000 psi. It is seen that the addition of oil recovery showed a stable trend until the case with a pressure of 10,000 psi.

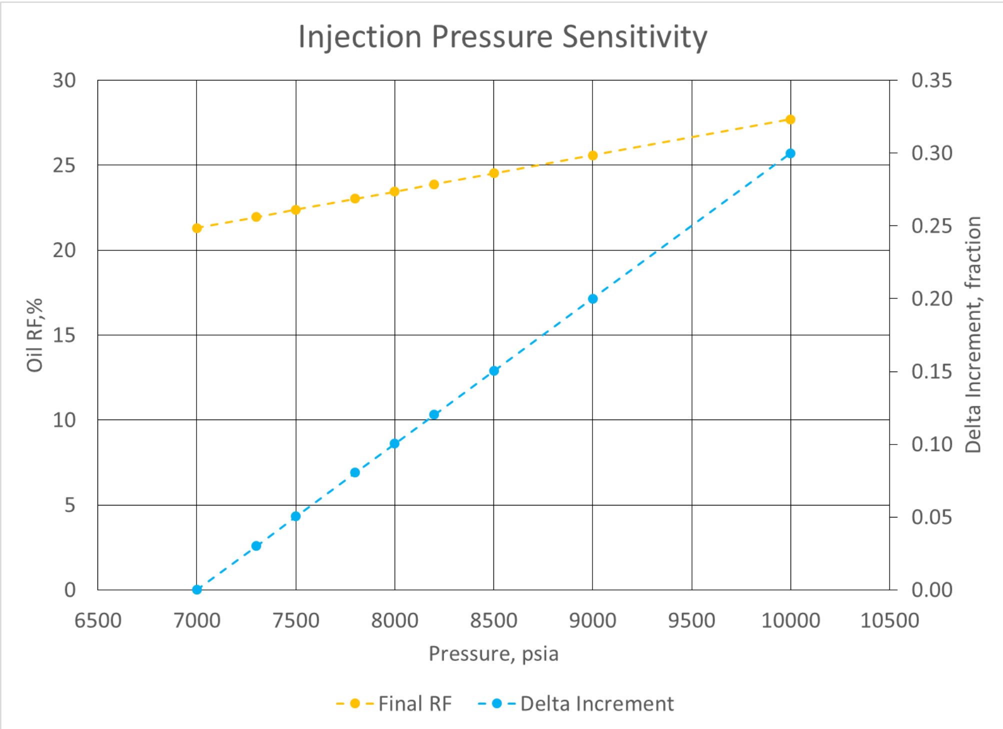

Figure 6.8 Sensitivity result of gas injection pressure (cyclic injection and production scenario with 250 ft hydraulic fracture half length)

Considering that the addition of injection pressure is proportional with the addition of oil recovery, and the real value of the gas injection pressure depends on each company's practical operational consideration, the author chose 7,000 psi gas injection as the default case in this study. The reason is because author wants to simulate this study with conservative thinking, in which the selected value is a number that has the smallest difference with the initial reservoir pressure, in case of gas injection pressure becomes a constraint that may be encountered in the field.

6.6 Fractures Configuration

The last adjustment in setting the default model was on the applied fractures configuration. Fracture configuration here means the placement configuration of injector and producer fracture in reservoir models. Author prepared four different fracture configurations to be simulated. **Figure 6.9** shows an illustration of how its configuration are categorized into scenario A, B, C and D, and how gas injection moves within the reservoir.

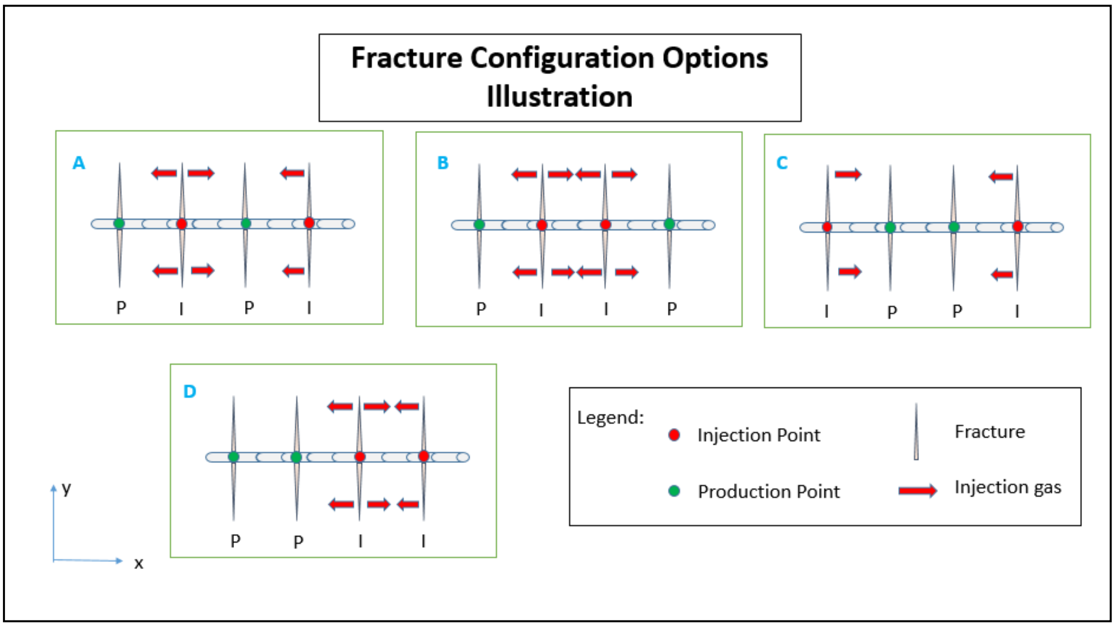

Figure 6.9 Top view of schematic fracture configuration options

Each prepared model consists of four fractures flanking equally large reservoir matrix and properties. For this sensitivity, reservoir models created from four adjoining fractures is referred as a displacement pattern. Author calls this as a displacement pattern because this configuration was deployed along horizontal well section, and the shape was repeated from the first to the last point of the well. Scenario A placed producer and injector fracture alternately on each fracture, scenario B put two injector fractures in adjoining position between two producer fractures on both outsides, scenario C was the opposite of the scenario B while scenario D put two fractures with the same function side by side in alternate position to fracture with different functions.

Figure 6.10 shows the simulation result of each scenario in terms of the final and depletion oil recovery. From the bar chart, it is seen that scenario A gives the greatest final oil recovery. In line with expectations for scenario A, the process of oil displacement by gas was more efficient where all oil matrixes were located between injector and producer fracture, so the entire oil matrixes were passed by the gas. In another scenario, seen in the gas injection movement direction, there was oil matrix areas between producer and injector fracture that were located side by side, thereby reducing the effectiveness of displacement.

In scenario B, matrix between the two injector fractures did not have suction point, but it received the same pressure from both injectors. In scenario C, matrix fracture between two producers did not receive support from injector fracture. In scenario D, the condition was the most severe because it had all the shortcomings found in scenario B and C where the oil matrix between the two producer fractures did not receive support and oil matrix between the two injector fractures did not have suction point.

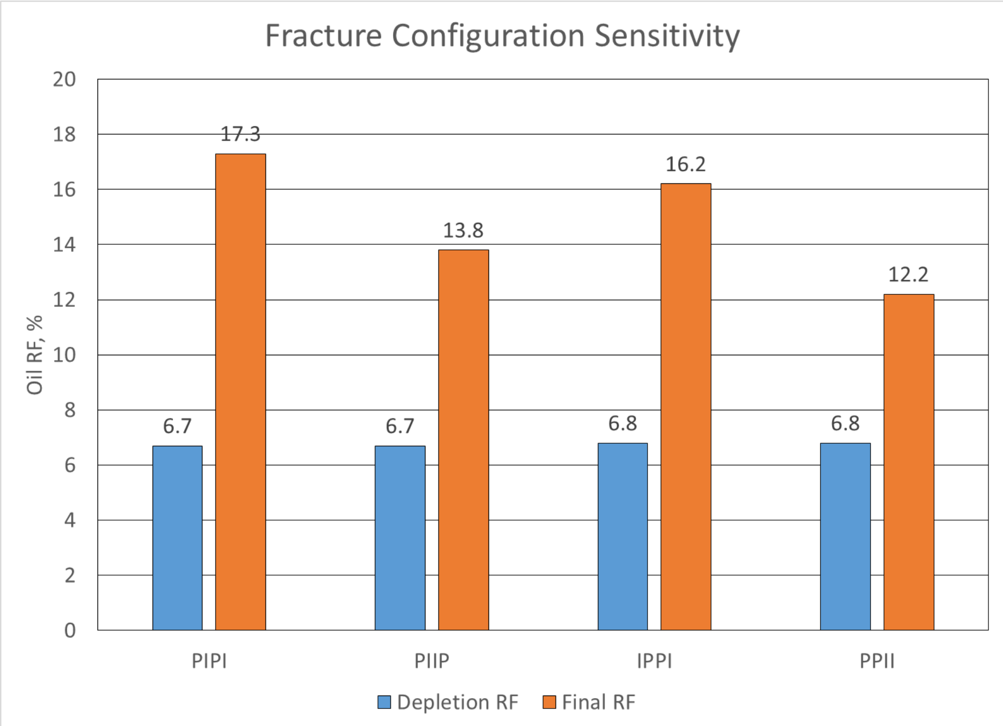

Figure 6.10 Sensitivity result of fracture configuration options (cyclic gas injection scenario with 250 ft hydraulic fracture half length and 365 days cycle time)

In line with the previous explanation, scenario A provided the greatest recovery because the displacement process is effective. Scenario C was the scenario with the second largest recovery because although there was a matrix that was skipped by the gas, but oil in the matrix was remain produced naturally. Scenario B was the scenario with the third largest oil recovery because oil matrix between the two injectors would not be produced since the location was quiet far from producer fracture and it received back pressure from the flanking injectors. Scenario B and C were still better than scenario D because in both scenarios, there were two matrix areas that received support from gas injection and left a less productive area. Scenario D was the worst because the displacement process occurs only in the oil matrix area and one injector fracture was not functioning optimally. As a result, scenario D has two unproductive oil matrix areas, where one area could not be producing and another areas that could only produce with a natural flow.

Considering the described simulation result, the author chose the configuration in which injector and producer fracture were placed alternately as the default reservoir models in this study. The reason is simply because this configuration is the most appropriate one to see the full potential of the same well EOR scenario that has been learned in this study.

7 SWEOR Potential

This study aims to assess the possibility of using one well for injection and production processes at once by applying cyclic gas injection scenario. Three simulation cases have been prepared, in which reservoir is developed through natural flow, continuous gas injection and cyclic injection. The results are presented as the well performance after 10 years simulation.

7.1 Depletion (Natural Flow)

First case as the reference to be compared with the gas injection EOR was the development through natural flow along the lifespan of the simulation. In this case, reservoir was produced naturally through all formed hydraulic fractures. Main driver of this case came from the reservoir itself, through the support from overburden, fluid and rock compressibility and solution gas drive that is represented as reservoir pressure. **Figure 7.1** displays the difference in reservoir simulation result between natural and continuous gas injection case. Simulation profile of natural depletion case is symbolized with dotted line.

7.2 Simultaneous Injection and Production Simulation

The case set as the initial reference for analyzing the potential of the gas injection EOR was the one with continuous injection and production. Continuous gas injection can be performed by two wells using conventional schemes well-to-well injection or by one well using same-well scheme with dual tubing strings, but the performed simulation applied the second scheme which used one well only to look at the potential of capital costs reduction. In this case, the well was put on production naturally for two years, then followed by development using gas injection. With the support from gas injection, a driving mechanism did not only come from the reservoir, but it also got additional support from injected gas to the reservoir pressure. The amount of additional pressure was proportional to the volume of injected gas. Performance profile of this case can be seen in **figure 7.1**, represented by solid line.

From the chart, it appears that the profile for the initial two years of both cases were the same because it used natural flow approach. Then in the third year, it started resulting gas injection in one case that made the difference in performance profile until the end of the simulation.

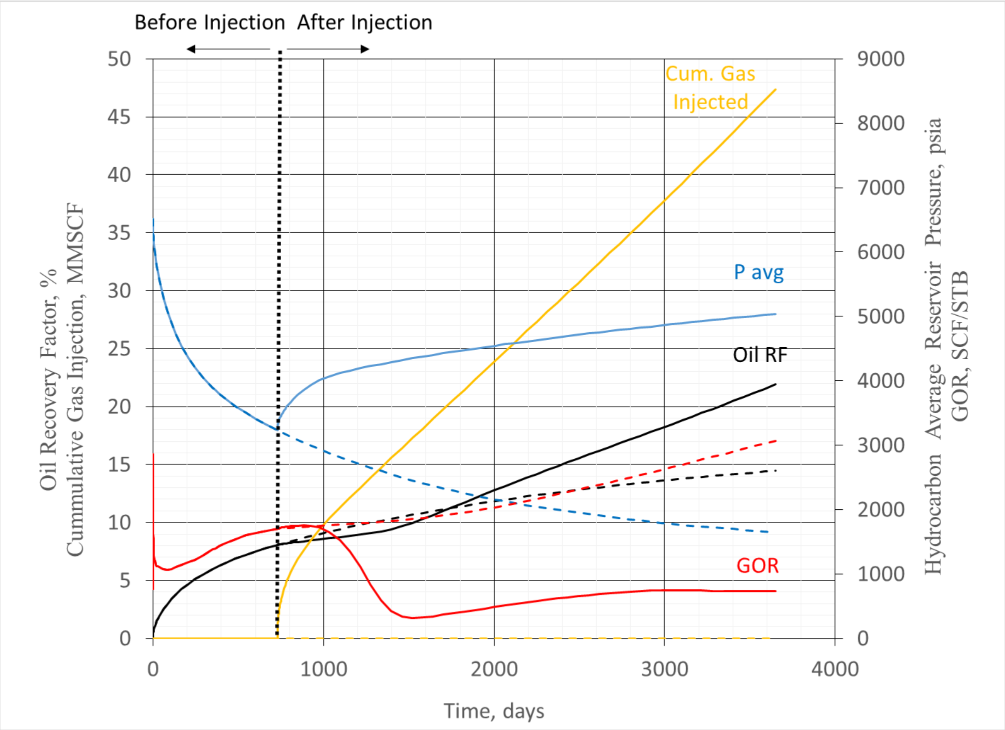

Figure 7.1 Continuous production-injection scenario performance curve (solid line) compared to natural production scenario (dotted line)

After the gas injection, reservoir pressure trend changed the direction from decline to incline. The given pressure support was very significant to the original reservoir pressure itself, as seen from the magnitude of separation between the solid blue line and dotted line. It also caused a significant increase in oil recovery as the results of gas displacement.

When viewed in more detail, there are at least two different trends in solid line profile at the time after injection. In the early years there was a sharp increase in pressure that is proportional to the increase in GOR. Then when GOR reached its peak, the pressure incline trend becomes lower but stable. This significant increase is the result of accumulated gas injection support and solution gas drive support which increased in this period because the reservoir pressure has reached its bubble point right before the start of the injection. After that, the reservoir pressure return back above the bubble point, thereby reducing the GOR as shown in solid red line.

The increase in reservoir pressure returning to above the bubble point also resulted in a slight slowdown in oil recovery due to reduced support from solution gas drive as seen between the two dotted blue lines in **figure 7.2**. Slowing oil recovery is also a result of changes in the function of some fractures that were previously used to produce oil but now used to inject gas. However, due to the reduced volume occupied by the gas solution, it resulted in increased volume of gas injection with greater pressure into the reservoir. Then oil production increased sharply, following the trend of its peak performance just before the start of the injection. The phenomenon is also seen in **figure 7.2** where oil recovery trend after the implementation of gas injection was now improved and stabilized following the peak of natural production recovery trend.

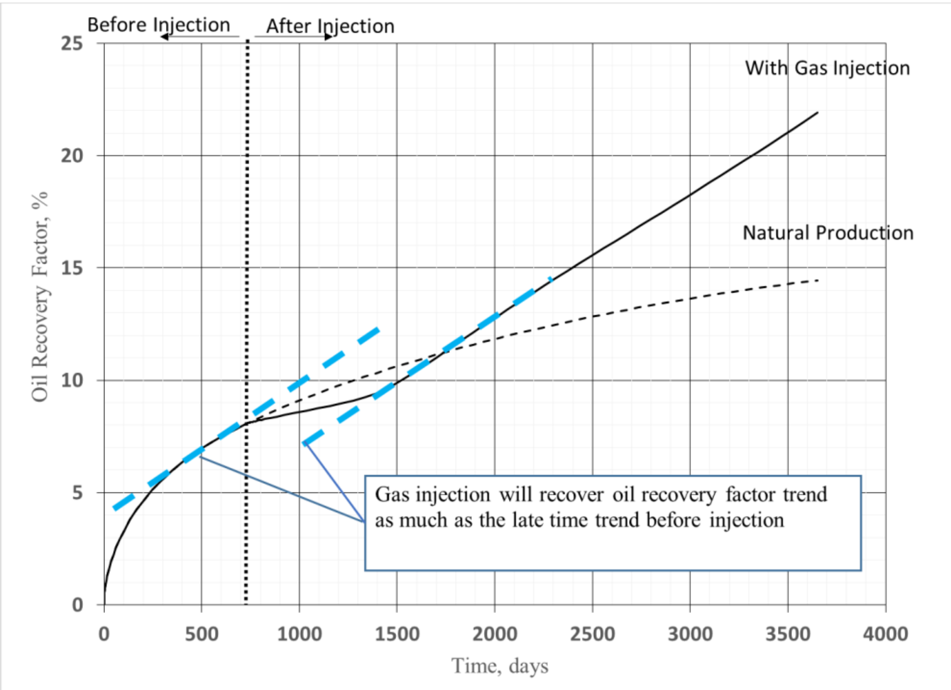

Figure 7.2 Oil recovery trend improvement from continuous gas injection scenario (solid line) compared to natural production (dotted line)

Figure 7.3 reinforces the conclusion that the gas injection can improve production as shown from generated oil production rate profile. Gas injection was able to restore the last oil production rate just like before the commencement of gas injection, but with lower decline rate. A little hollows that occurred in solid line was the result of the same phenomenon, which was due to the increase in reservoir pressure above its bubble point and function change of the several fractures. However, this production decrease interval is relatively very small compared to the production interval from the results obtained with gas injection implementation.

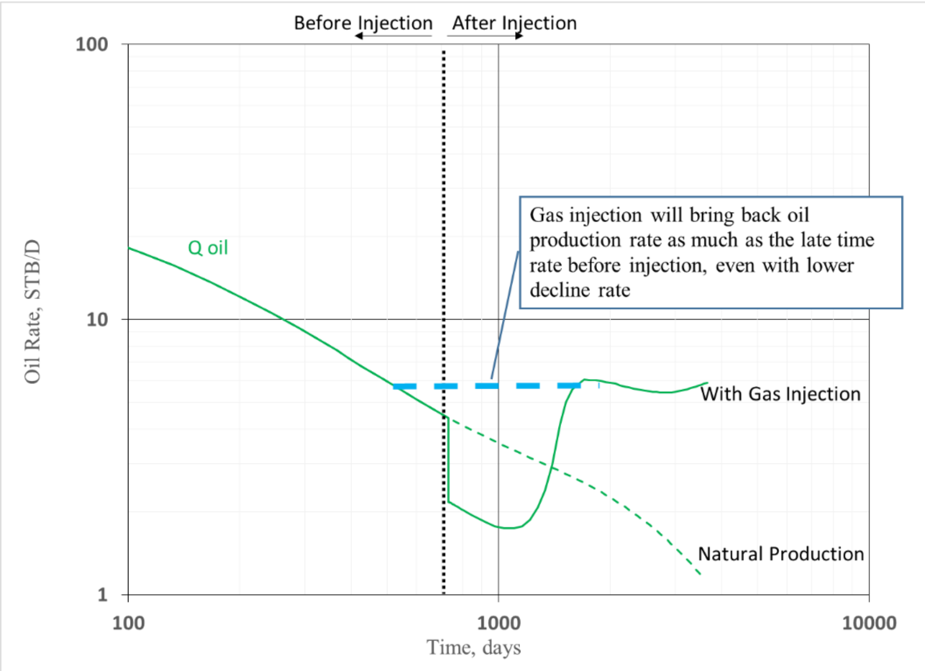

Figure 7.3 Oil rate improvement from continuous gas injection scenario (solid line) compared to natural production (dotted line)

7.3 Cyclic Injection and Production Simulation

Based on the good results of continuous gas injection implementation, then came the idea to try implementing cyclic gas injection for process effectiveness and cost-efficiency reason. Cyclic gas injection was done through the cheapest completion design and operating cost among other schemes, which was same-well with a single string of alternating injection and production fractures.

The simulated scenario was similar to the previous one which started the gas injection after two years of natural production. Simulation case with one-year cycle time interval was prepared to see the potential of this method. **Figure 7.4** shows typical performance plot of particular method. It is seen that there were two main processes that occurred in reservoir models, the injection and the production, which were clearly captured from the red GOR profile. At the time of injection, GOR value became zero, oil recovery plot was constant and reservoir pressure gradually increased proportional to the increase in its cumulative gas injection. At the time of production, there were produced gas and oil represented by the increase in GOR and oil recovery plot, while the cumulative gas injection remained constant and reservoir pressure gradually decreased proportional to the resulting oil recovery rate. Although the reservoir pressure experienced a period of decline, but its value was still above the bubble point and the overall trend was rising steadily, resulted in the increase of production compared to natural depletion scenario.

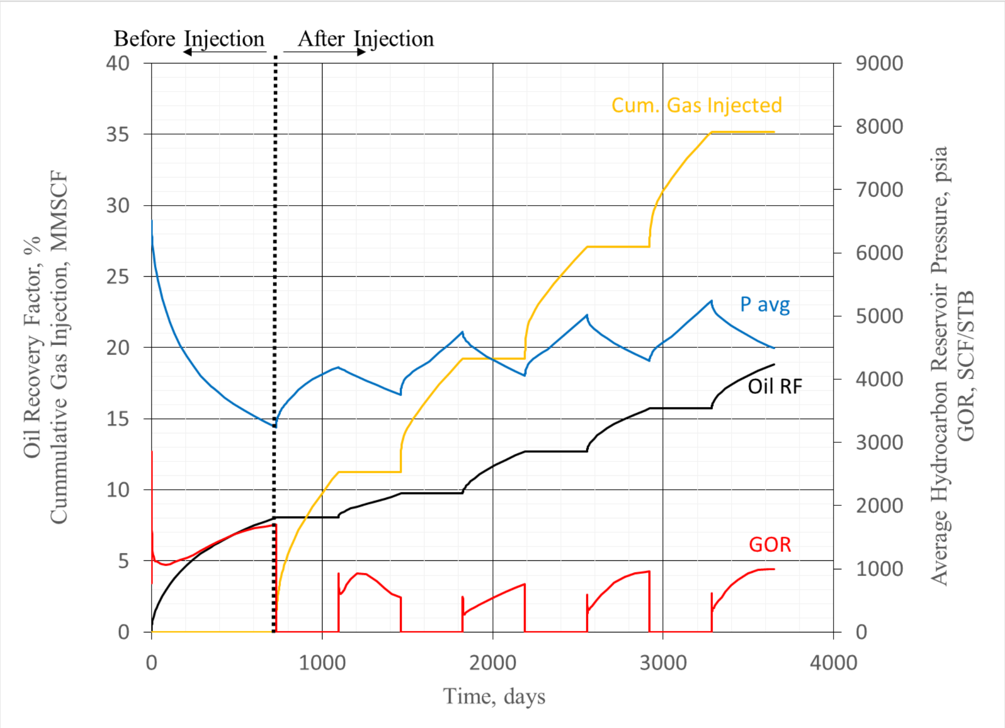

Figure 7.4 Cyclic production-injection scenario performance curve

Figure 7.5 shows the resulting plot performance before and after injection. The pressure gradually increased exceeding its bubble point pressure, as reflected from decreased GOR profile, resulting in improved oil recovery trend. Oil production rate is always larger than it was just before the injection, which indicates that there is improvement in reservoir production. When viewed in more detail, along with the addition of reservoir pressure, the required gas injection initial rate reduced but the initial oil production rate increased. This phenomenon gives the idea that in order to maintain optimum oil rate, it doesn't always require large gas injection rate because basically the increase in reservoir pressure from the gas injection is very quick to restore its optimum performance.

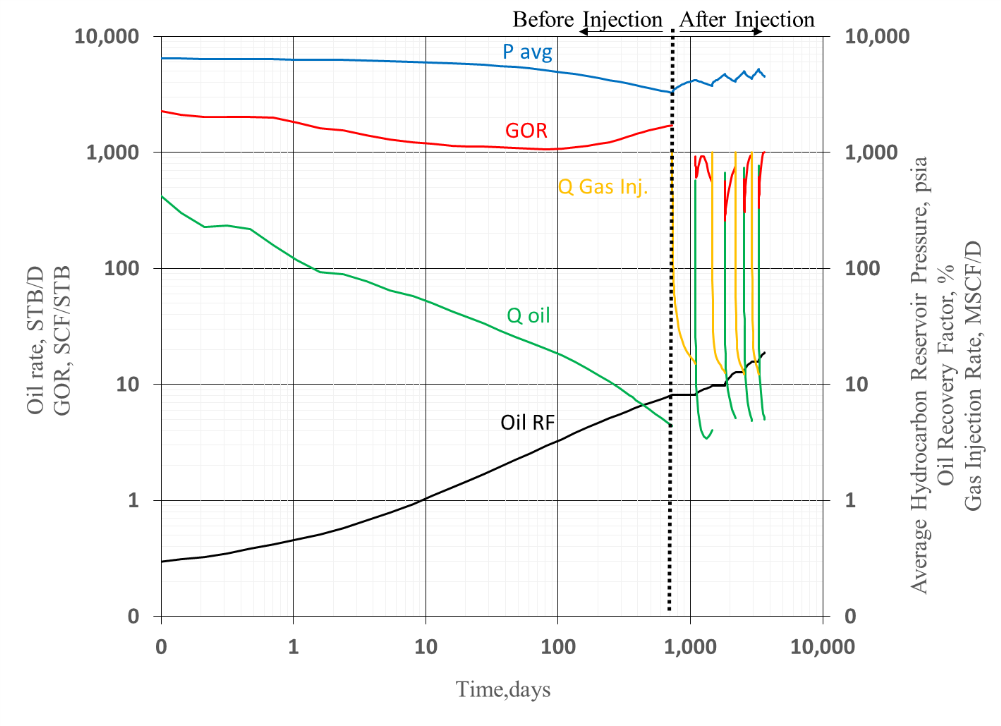

Figure 7.5 Cyclic production-injection scenario early time performance curve

Figure 7.6 shows interesting observation obtained by plotting normalized oil production rate after the gas injection to the late time production profile prior the injection, in which the following oil production after each injection cycle produced similar trend with the late time profile of natural production. This phenomenon is in accordance with the superposition concept in which every time the well is shut in, the resulted production profile will always follow the latest decline curve before the shut-in.

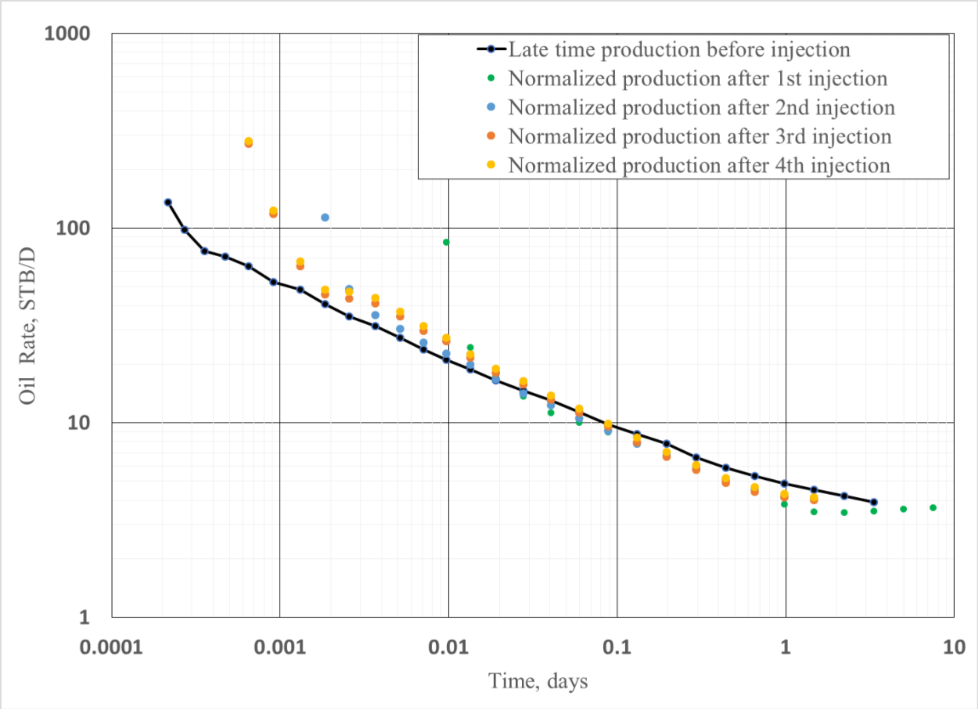

Figure 7.6 Normalized oil production rate after injection (cyclic injection scenario)

Figure 7.7 compares the performance plot of cyclic injection and production to the continuous injection and production development strategy. Case 1 is a development with cyclic gas injection which is denoted by a solid line, while the second case is the development with continuous gas injection which is denoted by dotted line. It is clearly seen that the cyclic injection has the potential to replace the continuous injection, since it will give considerably similar production results.

In terms of reservoir pressure average, both cases exhibits the increasing trend in a relatively equal value ranges. In terms of final oil recovery, the different between these two cases was only 3%, where the continuous injection could recover 21.9% while cyclic injection could recover only 18.8%. The difference is very small if compared to the difference in cumulative gas injection used to achieve their respective production volume. The gas injected volume reduction after applying cyclic injection strategy is 26%, considered very significant relative to the oil recovery reduction. GOR plot also shows a difference, in which gas volume produced through cyclic injection is smaller than with continuous gas injection, which helps to reduce the volume of gas to be handled at the surface if it becomes a concern.

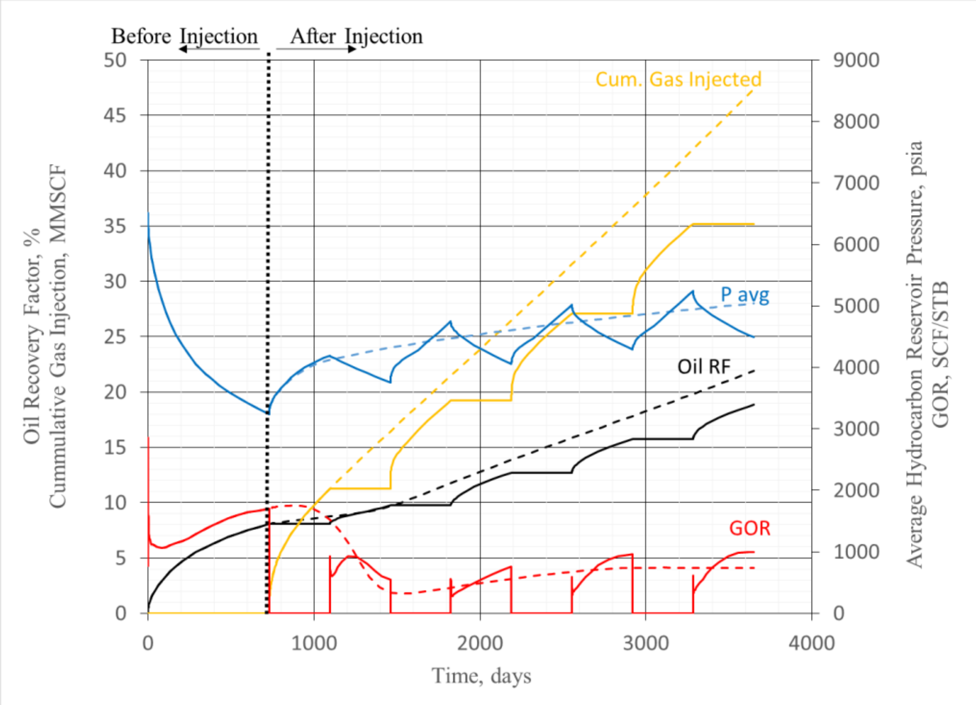

Figure 7.7 Continuous (dotted line) vs cyclic (solid line) production-injection scenario comparison

Table 7.1 Injection strategy comparison cases summary

Case	Scenario
1	Cyclic injection and production
2	Continuous injection and production

To see the gas movement within the reservoir, 3D visualization is also run using SensorMap. **Figure 7.8** illustrates the oil saturation map at the end of the 10th year of the continuous gas injection case simulation while **figure 7.9** illustrates the oil saturation map at the same simulation time for cyclic gas injection case. In simple terms, the figure illustrates how much is the oil saturation at each grid, denoted with a reddish orange color for high saturation and blue for low saturation. In the initial condition, the whole grid shows red color which symbolizes equilibrium condition.

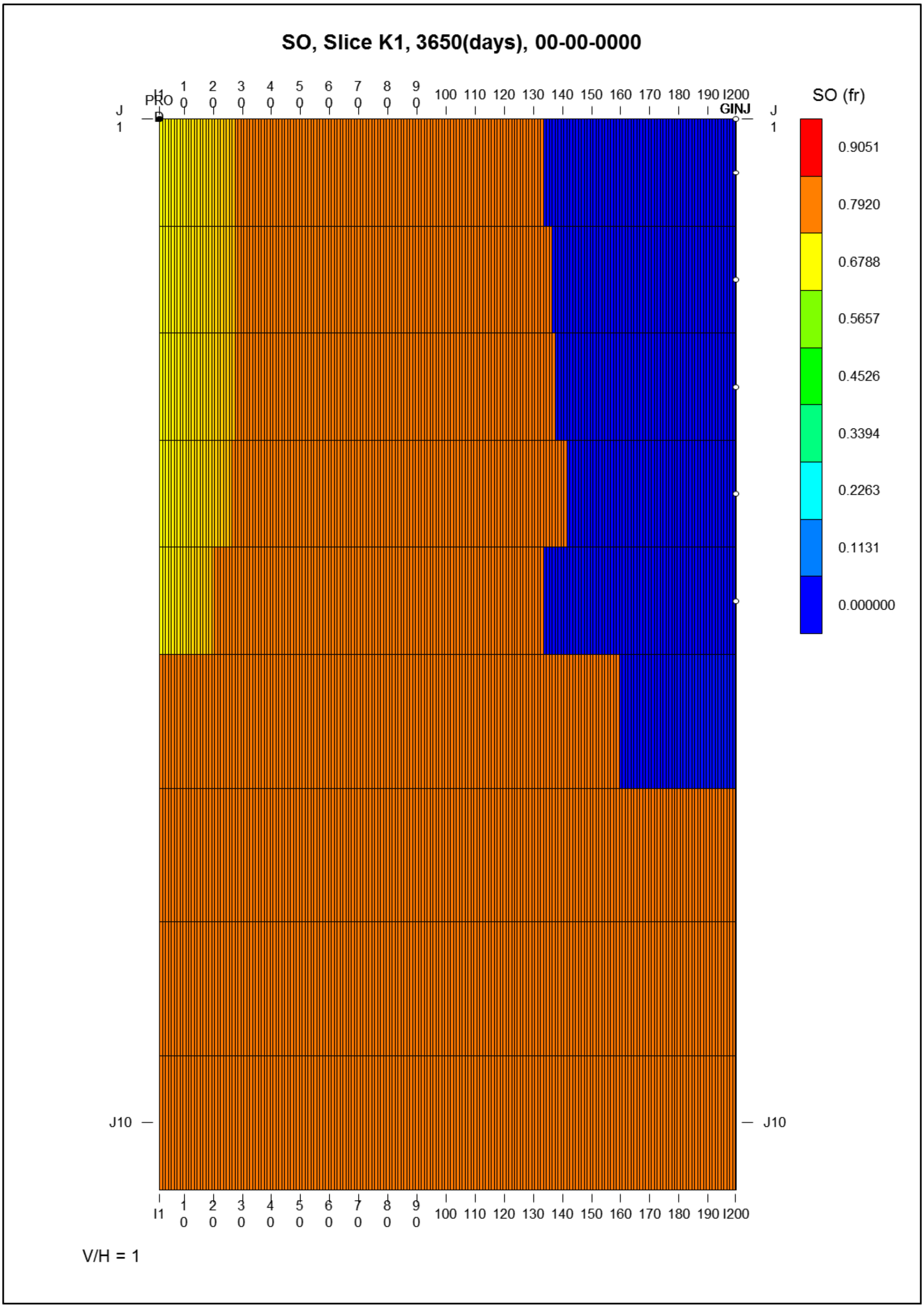

Figure 7.8 Oil saturation map at 10 years for continuous injection scenario

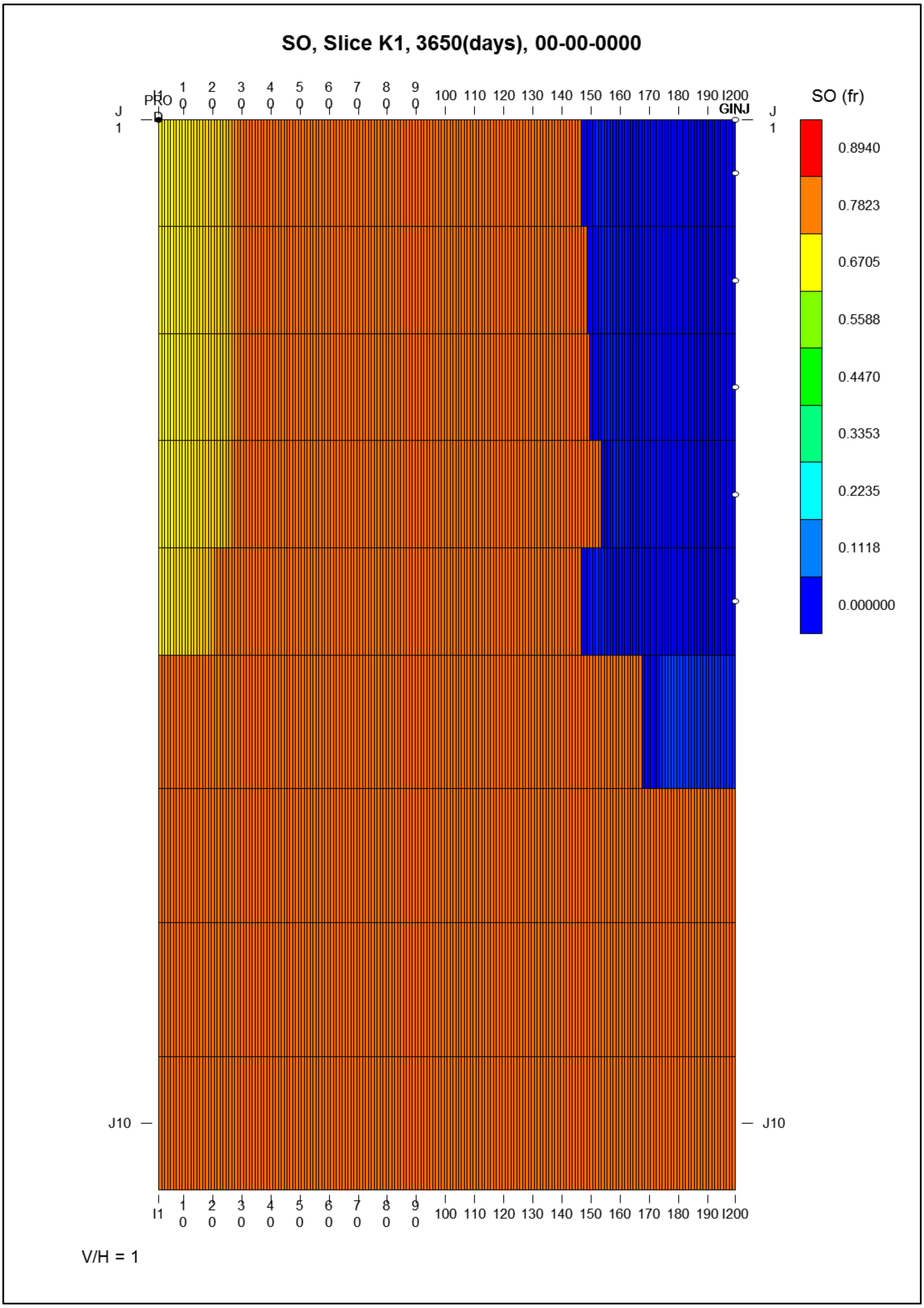

Figure 7.9 Oil saturation map at 10 years for cyclic injection scenario

Blue grid represents very low oil saturation which means that the area is already displaced by injected gas and yellow grid reflects that the area has reduced oil saturation that also means that it is the drainage radius from production fracture. It is seen that the gas movement in both strategy is not significantly different, since in average, the distance that has been displaced by the gas in continuous gas injection scenario is only about 10 ft longer than the cyclic injection. The distance difference is considered not significant compared to the total inter-fractures distance, which is 200 ft. From the figure, it is seen that the difference in injection strategy does not affect the production fracture drainage radius because the displayed results from both images are considered similar.

Considering the advantages and disadvantages of the two scenarios mentioned, it can be concluded that same-well cyclic injection is potential to be applied in shale oil reservoir as an alternative to continuous injection. The reason is because the obtained oil recovery is not significantly reduced but it helps to significantly reduce capital and operating cost. Reducing the volume of gas produced help to cope with limited gas handling capacity and reducing the gas injection volume means lowering the cost to provide gas and pump injection facilities. The produced gas which is not used in the process can also give commercial advantage by selling it.

For the purpose of further study, the next simulations will carry out the sensitivity analysis of cyclic injection scenario, which will be compared with continuous gas injection scenario as the base simulation case. This case is used as the base case because the purpose of the sensitivity is to find the optimum condition which does not much differ with the continuous injection case performance.

8 Sensitivity of Uncertain Variables

Number of simulations were conducted to see the effects of uncertain variables of the reservoir to the ability of its oil recovery. As discussed in the previous section, oil recovery is closely related to the gas volume injected and generated pressure increase in reservoir. The addition of accumulated gas injection volume increases reservoir pressure proportionally. The circumstances in which the reservoir pressure returns and surpasses its bubble point pressure is a condition that supports more effective oil displacement.

Considering that the relationship between the three parameters has been understood and the plot shape of these parameters are typical, then the simulation results shown here are focus only on oil recovery. Oil recovery have become the main key performance indicator because it is the ultimate goal of the application of the EOR strategy. Each sensitivity simulation is performed using cyclic injection strategy with a 30-day cycle time because this is the most optimum scenario, so then the result will be intuitive to illustrate the maximum capability that can be expected.

The simulation result will be shown in terms of final oil recovery, which means the oil recovery value that resulted after 10 years of simulation, consists of 2 years natural production and 8 years cyclic gas injection. The oil recovery of the natural production after 2 years is presented as depletion oil recovery. The result is also shown as delta oil recovery against the base case (the difference between oil recovery in simulated case and continuous gas injection scenario) to better describe the comparison.

8.1 Behind Pipe Communication

First sensitivity group is categorized as behind pipe communication. Variables used in this group are cement porosity and cement permeability. Base case reservoir model using 0% porosity and 0 md permeability assuming best cement job performed. But in actual, we often find bad cementing job which does not perfectly sealed separated segment, causing possible communication between the segments. Sensitivity aims to look at the not ideal situation and see the effects on production performance of cyclic gas injection.

8.1.1 Cement Porosity

Cement porosity variation simulated varied from 1% to 30%, simulated against 3 permeability values which are 0.1, 0.5 and 0.8 mD. The used permeability values are taken based on sensitivity result of cement permeability in which the values give significant effect to the performance, so it is considered to be able to represent the formed channel as the alternative paths for injected gas. **Figure 8.1** shows the final and depletion oil recovery obtained by each case, and **figure 8.2** shows its delta oil recovery against the base case (continuous gas injection).

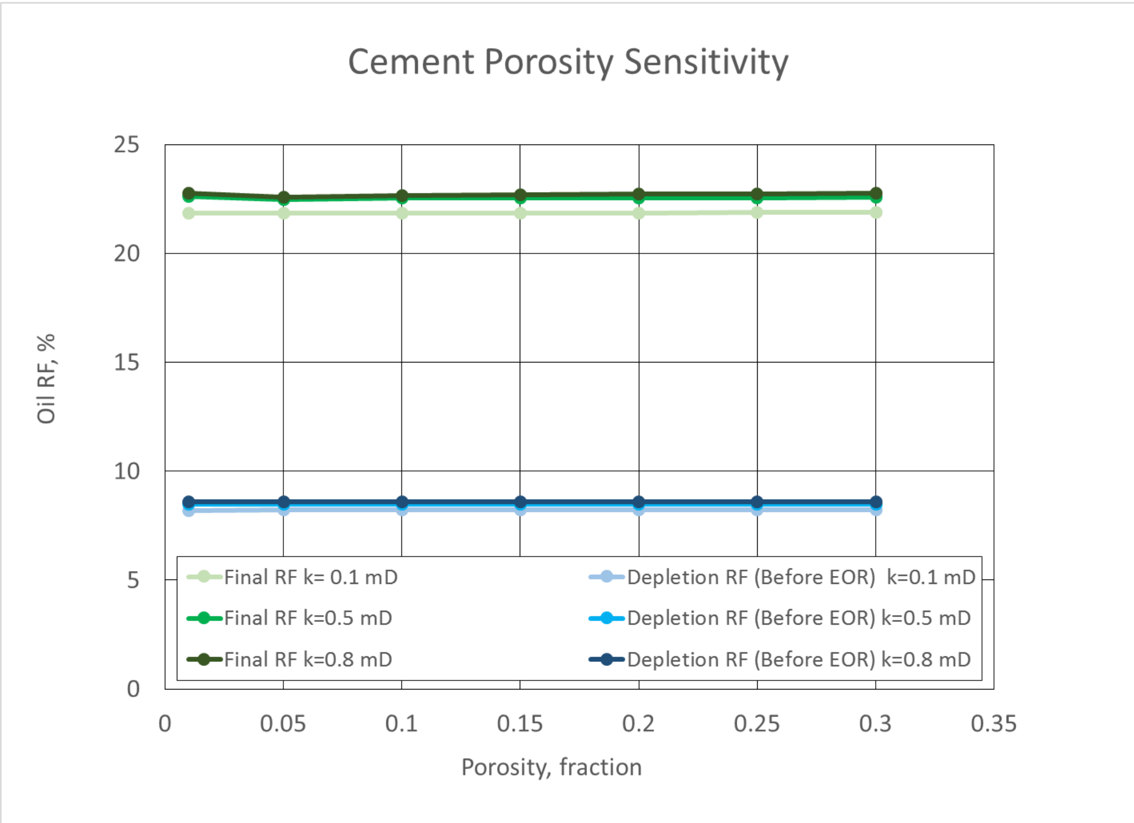

Figure 8.1 Figure 8 Oil recovery factor of cyclic injection scenario at 10 years for different cement porosity and permeability

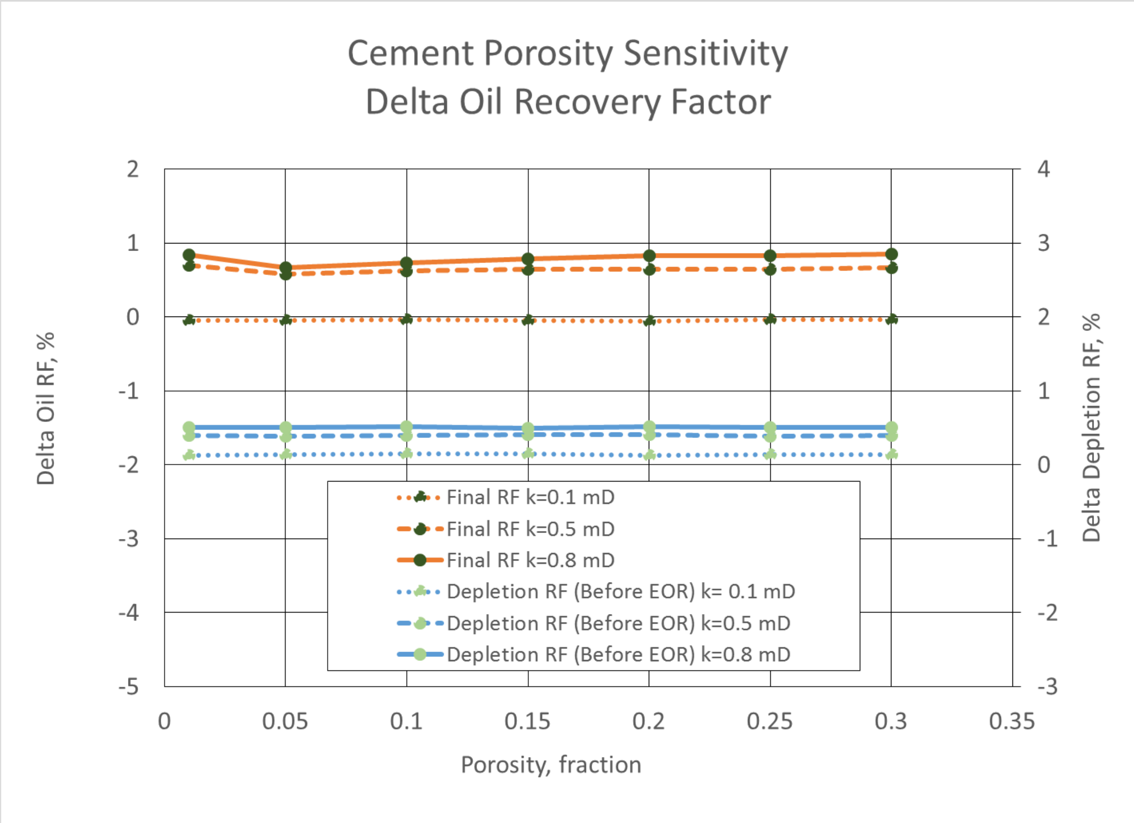

Figure 8.2 Delta oil recovery factor of cyclic injection scenario against base case (continuous gas injection) at 10 years for different cement porosity and permeability

From both charts displayed, it can be concluded that for all permeability values, the addition of cement pore volume is not sensitive to the production performance of the reservoir. It is shown from the range of porosity simulated, the results is considered constant, although the permeability value is relatively high. Small influence is seen in the final oil recovery of lower porosity interval but higher permeability, yet this change is relatively very small at around 0.1%. Increased porosity behind casing does not give any significant effect on the final oil recovery. Similarly to the depletion oil recovery, although the presence of communication will slightly increase the recovery, but its value is also very small in the range of 0.1 - 0.15% and the increase in the quality of communication does not give valuable effect.

8.1.2 Cement Permeability

Cement permeability variation in this simulation start from 1x10^{-7} md to 1 md, but with constant porosity 15%. The applied porosity is the median of porosity range in previous sensitivity. For displaying simulation result based on permeability sensitivity, author presents the results as two, which are smaller and higher permeability interval by using semi-log scale and cartesian scale respectively. Smaller values result is shown in **figure 8.3** and **figure 8.4**, while higher values result is exhibited in **figure 8.5** and **figure 8.6**. Final and depletion phase recovery factor as well as delta oil recovery against base case are also presented.

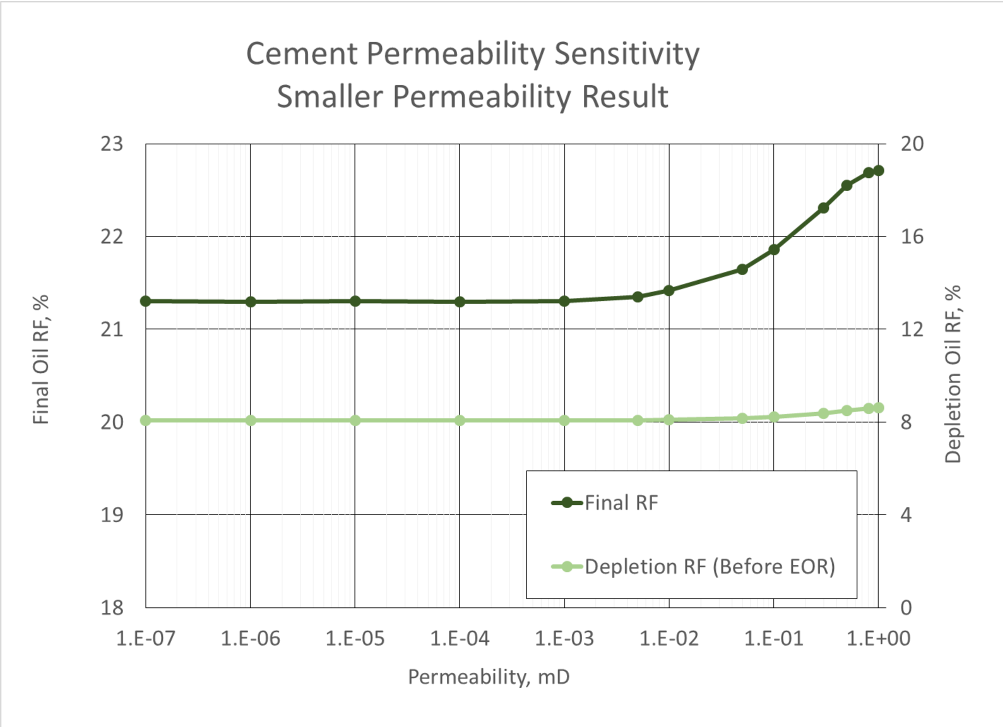

Figure 8.3 Oil recovery factor of cyclic injection scenario at 10 years for different cement permeability (smaller permeability interval)

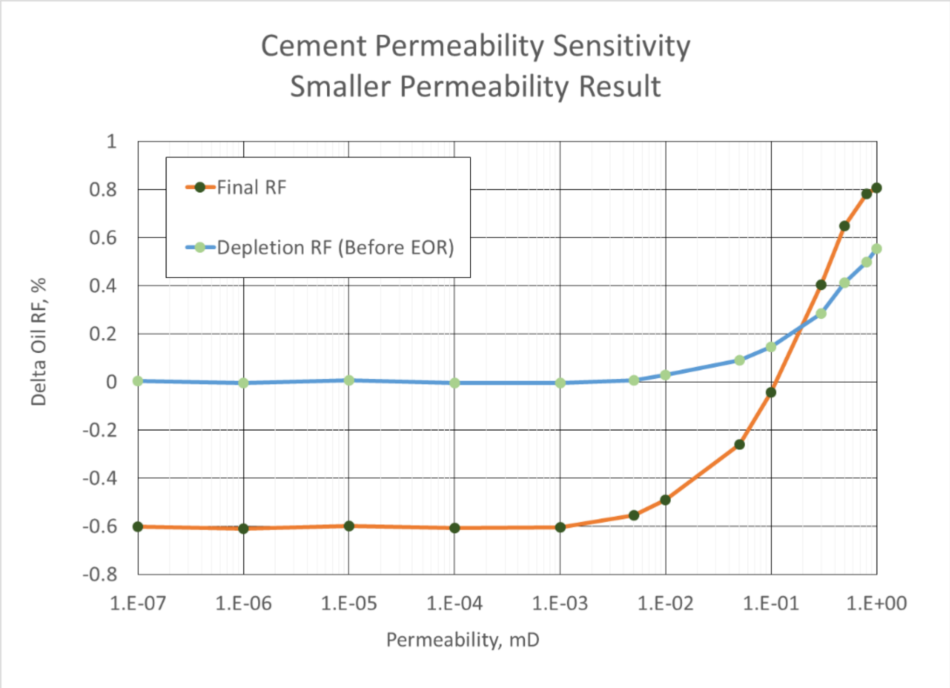

Figure 8.4 Delta oil recovery factor of cyclic injection scenario against base case (continuous gas injection) at 10 years for different cement permeability (smaller permeability interval)

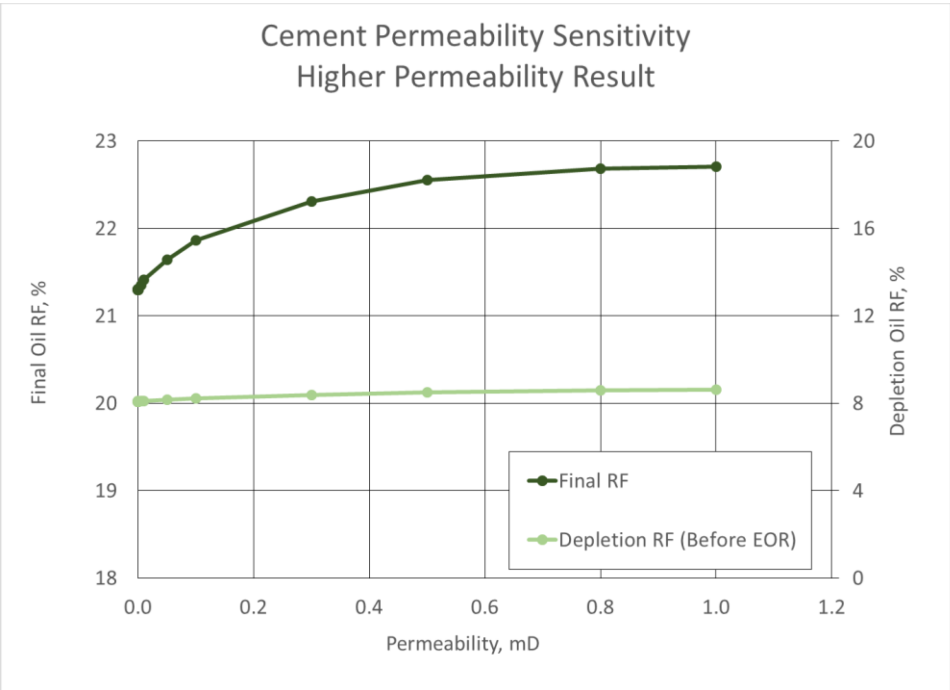

Figure 8.5 Oil recovery factor of cyclic injection scenario at 10 years for different cement permeability (higher permeability interval)

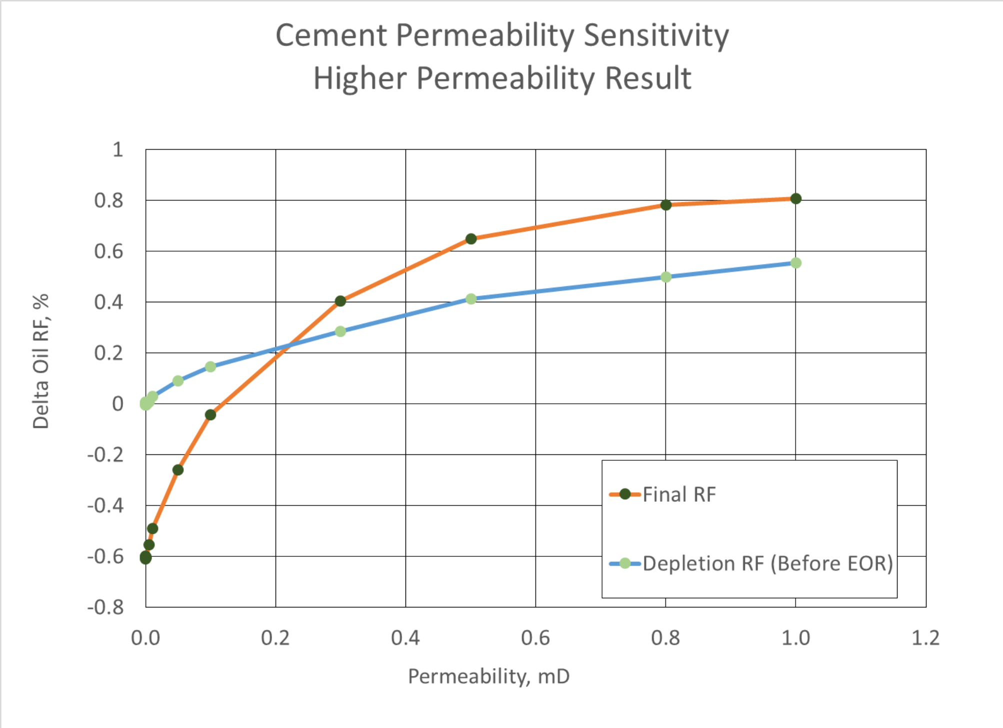

Figure 8.6 Delta oil recovery factor of cyclic injection scenario against base case (continuous gas injection) at 10 years for different cement permeability higher permeability interval)

From several graphs shown, it is visible that the communication behind cement doesn't affect the production in very small permeability range, until the threshold point reaches ($5x10^{-3}$ mD). Significant change to oil recovery was seen above this threshold value, where the channel behind casing slowly helps improving the recovery until it finally being stagnant at permeability above $8x10^{-1}$ mD.

From delta oil recovery against base case plot, there is interesting result showing that the presence of better communication behind casing increased the oil recovery both in depletion phase and after gas injection process. Even at permeability larger than $1x10^{-1}$ mD, the performance of cyclic gas injection started to improved and better than the base case (continuous gas injection without communication behind casing).

This improvement of cement permeability is an artificial effect in simulation but is not going to happen in real field. This result comes because the simulator will see the cement grid with pore volume, similar with other matrix grid. Bigger cement pore volume will give better transmissibility between cement grid and matrix grid. It is like the effect of addition in the matrix pore volume itself. Hence, it can be concluded that the change in cement permeability is not affecting the well performance.

8.2 Hydraulic Fracture Properties

The next sensitivity to be discussed is categorized as hydraulic fracture properties. Hydraulic fracture properties consist of hydraulic fracture length and number of hydraulic fracture half-length grid block. Base case reservoir model used 250 ft length, assigned in 6 grid blocks. The number is based on typical value in shale oil reservoir and 6 grid blocks are just the assumption that the author set for the initial models. This sensitivity aims to see whether the hydraulic fracture length and number of grid blocks assigned in the reservoir models will give impact to the simulation or not.

8.2.1 Hydraulic Fracture Half Length

Fracture half-length was simulated from 50 ft to 450 ft, reflecting the condition in which the hydraulic fracture was formed from 10% to 90% of total reservoir width. In this simulation, only the fracture length in the reservoir was changed, but the width of the reservoir remained the same. The goal is to see performance improvement in accordance with the increase in well conductivity to the reservoir, so it can be used as a reference when fracturing job is about to start. **Figure 8.7** shows the simulation result in terms of the final and depletion oil recovery and **figure 8.8** shows the generated change in oil recovery compared to continuous gas injection scenario.

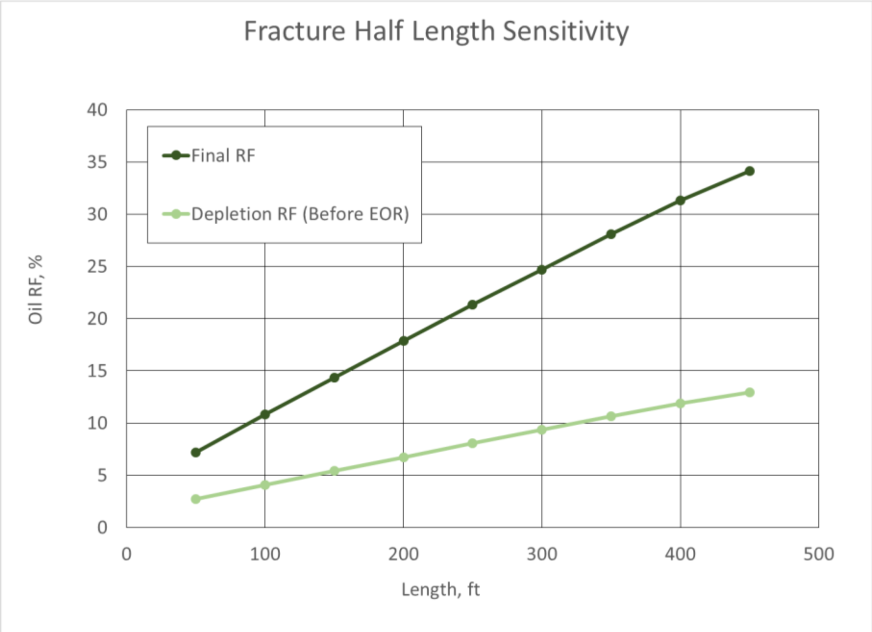

Figure 8.7 Oil recovery factor of cyclic injection scenario at 10 years for different fracture half length

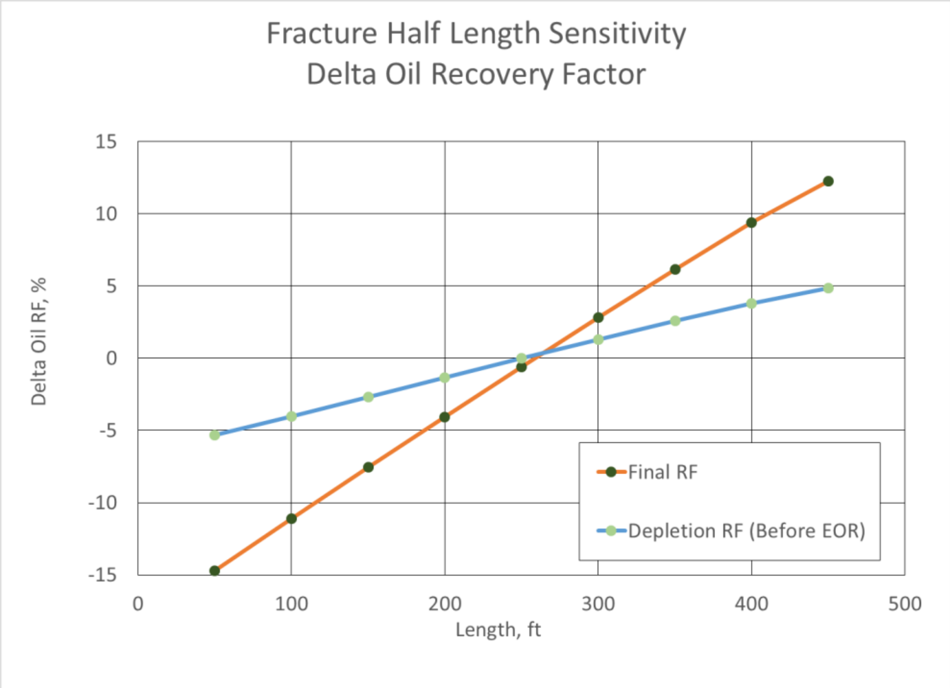

Figure 8.8 Delta oil recovery factor of cyclic injection scenario against base case (continuous gas injection) at 10 years for different fracture half length

The obtain result aligned with the expectation. It shows that increasing the length of hydraulic fracture generated significant increase in oil recovery. This is reasonable because the longer the fracture, the greater surface area of injected gas is exposed. As can be seen, each final and depletion oil recovery presented a constant inclining linear trend that is proportional to the increase in length. Seeing the final oil recovery factor trend is increasing greater than depletion oil recovery factor, it supports the idea that cyclic gas injection is very significant to increase productivity, which is much better improve the result compared to only adding the fracture length without any support from gas injection.

Based on the default model, 500 ft fracture length means that the fracture tip already reached no flow boundary. If this happens, then the process will become one dimensional flow from injector to producer fractures. This condition is more effective since the gas injection will eventually breakthrough the producer fracture if the gas is injected continuously.

The simulation result clearly delivers the message that oil recovery is very sensitive to how much improvement within well and reservoir conductivity is generated from adding fracture length. From the plot of delta oil recovery, it appears that even the small addition of length above 250 ft, brought the production to the level of base case scenario has (continuous gas injection). In summary, extending fracture depth when designing fracturing job is highly recommended.

8.2.2 Number of Grid along Hydraulic Fracture

Number of hydraulic fracture half-length grid block is the sum of the reservoir grids which were assigned as fractures in the reservoir models. The actual length of secondary fracture and reservoir half width was set at constant level, while it only modified the number of grid blocks assigned along the fractures. The simulation used initial reservoir model as the reference with length of 250 ft and reservoir half-length of 500 ft, which was represented by 10 rows grid in the y-direction. Default number is 6 grid block, so the sensitivity was initiated to see the effect if the number of grid is reduced to 2 or increased to 9. That number was chosen to ensure that the original shape of the reservoir model remains unchanged with cement in the first row and the remaining oil matrix interval beyond fracture. Two charts showing the final and depletion oil recovery as well as delta oil recovery factor to the base case is shown in **figure 8.9** and **figure 8.10**.

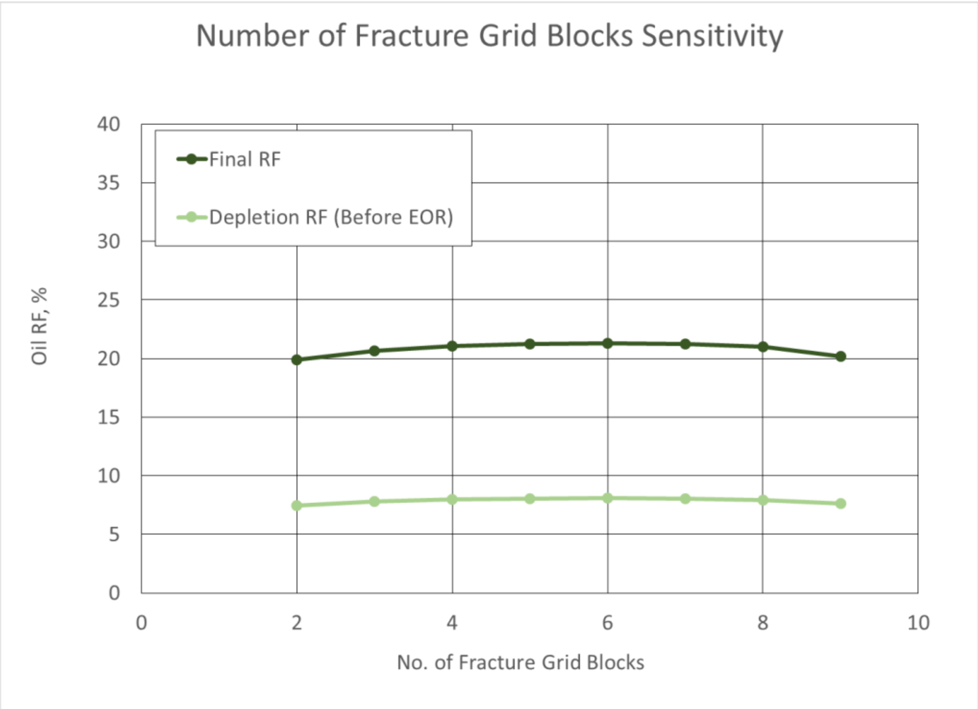

Figure 8.9 Oil recovery factor of cyclic injection scenario at 10 years for different number of grid block assigned as hydraulic fractures

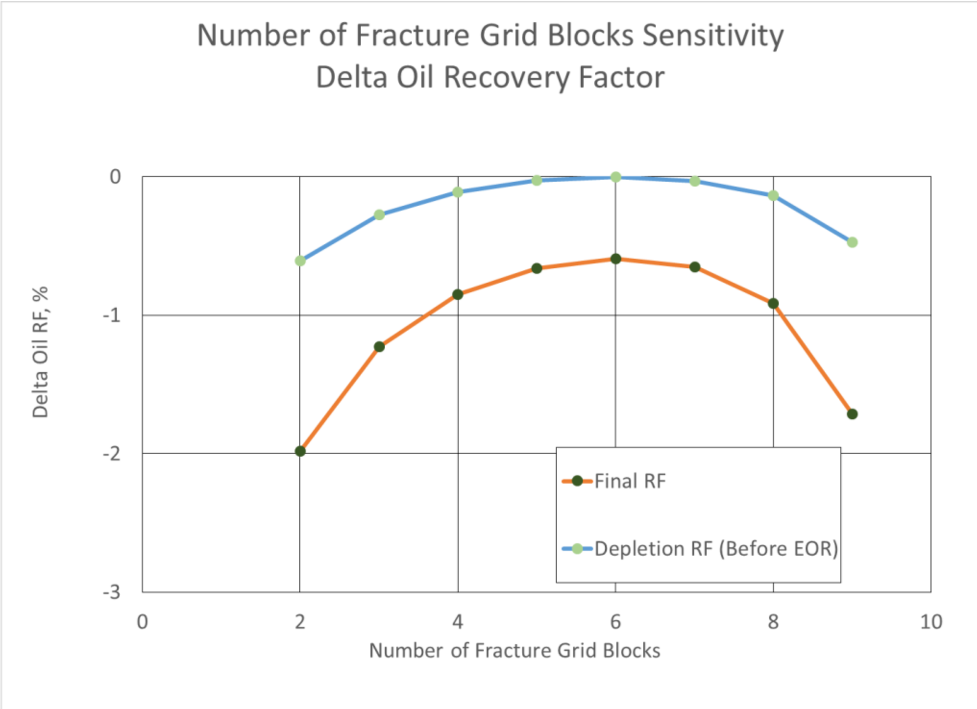

Figure 8.10 Delta oil recovery factor of cyclic injection scenario against base case (continuous gas injection) at 10 years for different number of grid block assigned as hydraulic fractures

Above plot exhibits that changes in the way the fracture grid is modeled is insensitive to the performance of the reservoir. The change was considered insignificant from 3 to 8 grid blocks. However, there was interesting behavior when assigning only 2 grid blocks or assigning almost all the y-direction blocks as the fracture. The oil recovery experienced a little decrease but this was considerably negligible since the difference is very small. Actual effect generated was more impactful to the numerical simulation instead of actual reservoir performance because in fact the reservoir condition is the same in all cases. Based on this result, it can be concluded that assigning extremely few or extremely large quantity of fracture grid block is not recommended. The fracture grid block assignment must be carefully planned by holding the principle of size uniformity throughout the entire model grid.

8.3 Reservoir Half Width (y-Direction)

The next sensitivity is about reservoir half width or y-direction length of reservoir models. In this case, only the reservoir width is changed from default value of 500 ft, while hydraulic fracture half-length and number of grid blocks assigned as the fractures were set constant. The purpose of this simulation is to see the effect on how big the influence is, by changing the remaining oil matrix interval beyond hydraulic fracture. Variation of simulation begins with 300 ft (bit higher than reservoir half width default of 250 ft) until 700 ft. In overall, it seems to have similar goal with sensitivity against fracture half-length, but since this variable is also one of uncertainties then it is also very important to see. The simulation result is summarized in **figure 8.11** for final and depletion oil recovery and **figure 8.12** for delta oil recovery against the base case.

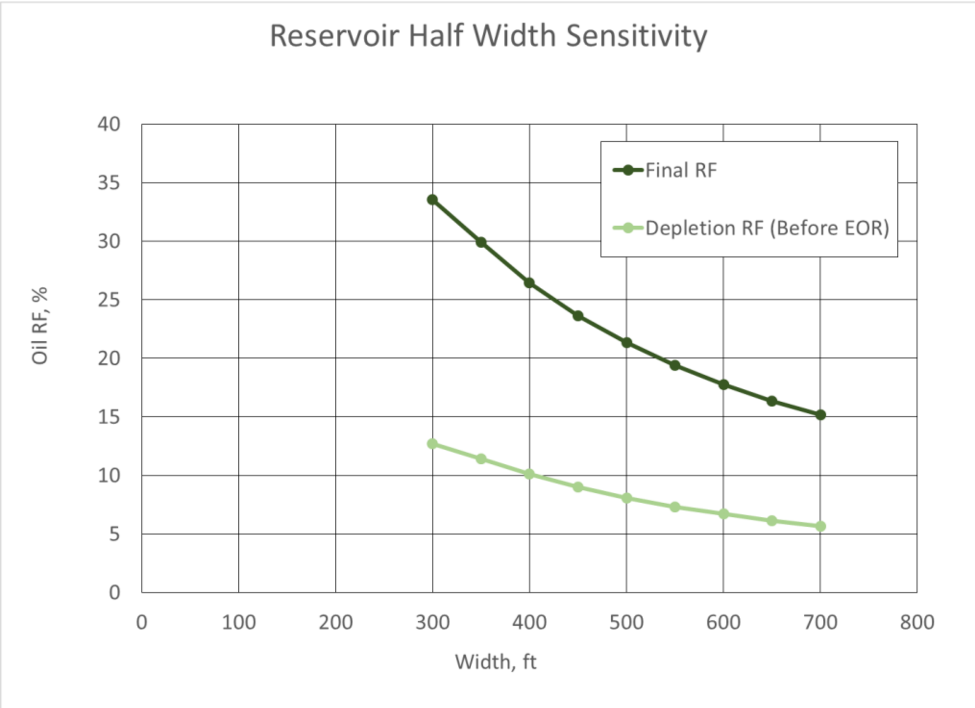

Figure 8.11 Oil recovery factor of cyclic injection scenario at 10 years for different reservoir half width

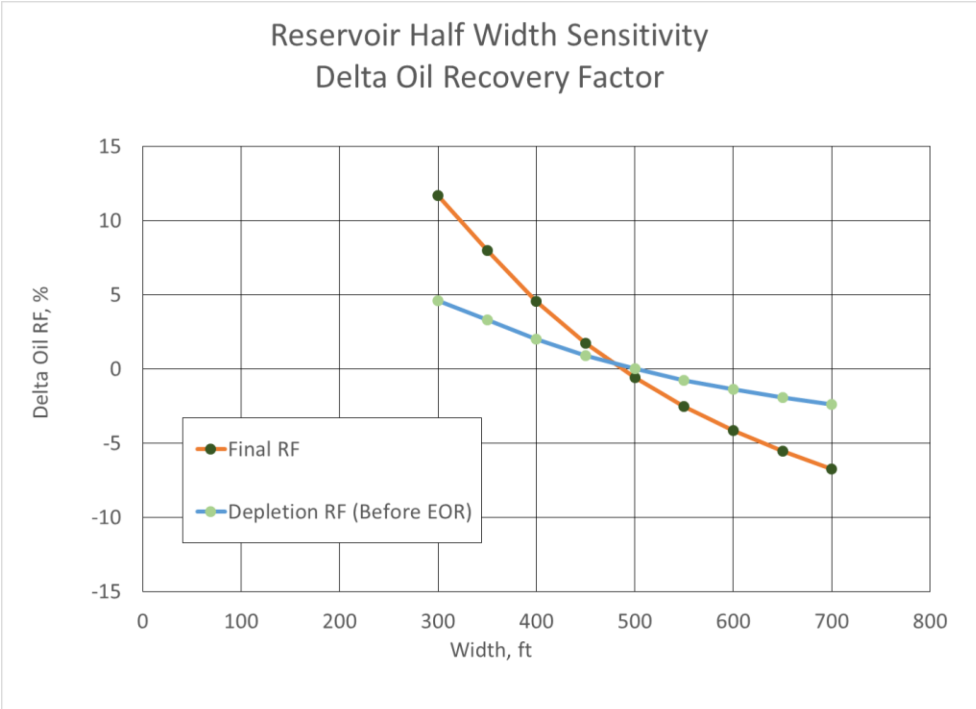

Figure 8.12 Delta oil recovery factor of cyclic injection scenario against base case (continuous gas injection) at 10 years for different reservoir half width

From the collection of the plots above, it is clear that increasing the length of the reservoir half width gradually decreases the oil recovery factor. This result is predictable especially because sensitivity against the length of hydraulic fracture has been performed previously in this study. The result is also similar where the final recovery trend is declining more sharply than the depletion phase recovery. However, interesting result is observed by the difference from the obtained trend lines. In this case, the curve obtained is not linear like that in fracture length sensitivity. The obtained curve looks exponential in which reservoir width below 500 ft will sharply bring the oil recovery lower. This result was not seen in the fracture half-length sensitivity. That is because in this case, when there is a change in reservoir half-width, there will be also a change in the initial oil in place. So the displacement efficiency generated will vary based on the volume itself. From the graph, it is seen that crucial interval is if the remaining oil length beyond fracture is less until equal to the hydraulic fracture length itself. Above that interval, the oil recovery decline is become linear. Although sensitivity result of this variable is not comparable because its initial oil in place is different in each case, but this plots gave an insight that cyclic gas injection can affect the reservoir beyond fracture until at least twice the length of its hydraulic fracture.

8.4 Secondary Fracture Properties

The next sensitivity is about the possibility of forming the secondary fractures as the side effect of its main hydraulic fracture. Included in this secondary fracture properties are the length, the permeability and the intensity. Secondary fracture is the byproduct of hydraulic fracture, whose location cannot be determined. This variable is very important because their existence in the reservoir create the possibility of increasing fracture conductivity. In the default case, secondary fracture is ignored since its existence, properties, and intensity are uncertain. The sensitivity then conducted to see its effect on production performance to allow for mitigation during the fracturing job design.

8.4.1 Secondary Fracture Length

The first variable in this simulation is secondary fracture length. As discussed earlier, properties of the secondary fracture cannot be deliberately formed, and once established, cannot be measured in the reservoir. Therefore, for simplification reason, the author used 1 ft as the smallest value and 98 ft as the most extreme value. This value is called extreme because based on the way the secondary fracture modeled (which has been first discussed in chapter Pipe-It project), that 98 ft represents a state in which both secondary fractures formed from both hydraulic fractures in the model unite and connect one another. In the simulation, secondary fracture permeability is assumed equal to main hydraulic fracture permeability, which is 10,000 mD. **Figure 8.13** and **figure 8.14** shows the result in terms of oil recovery factor and delta oil recovery compared to the base case.

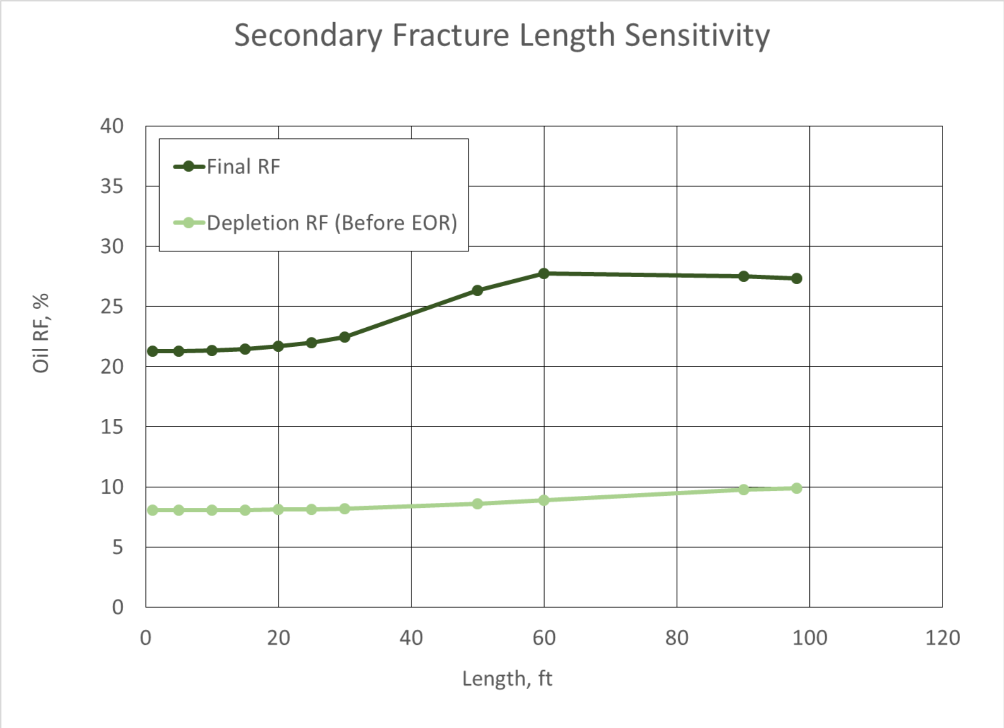

Figure 8.13 Oil recovery factor of cyclic injection scenario at 10 years for different secondary fracture length

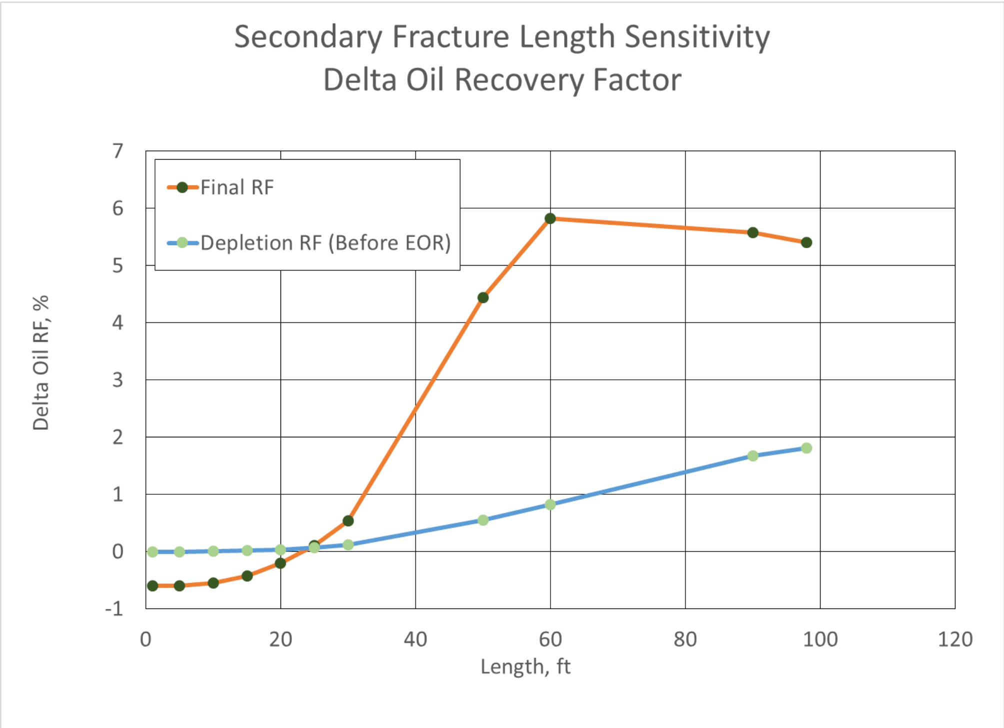

Figure 8.14 Delta oil recovery factor of cyclic injection scenario against base case (continuous gas injection) at 10 years for different secondary fracture length

Based on the above plots, it appears that the secondary fracture will give impact if it exceeds 25 ft, which is even exceeding the base case performance. Then the effects were relatively constant at the length above 60 ft. Depletion oil recovery looks gradually increasing starting from 30 ft which also reflected reservoir productivity improvement. In magnitude, the existence of secondary fracture at its optimum length produced significant increase in oil recovery. The result came as the effect of increasing conductivity and gas injection area exposure into the oil matrix. It was initially predicted that in extreme length, the gas will only flow through this communication without pushing oil and worsening performance, but the final result shows that it still produced excellent performance. This is because even in most extreme case, the volume of this high porosity and high permeability zone is very small relative to the reservoir matrix, thus the larger amount of injected gas will still do the good job in giving pressure to surrounding matrix area wherever they are exposed to oil matrix. Therefore it can be concluded that longer secondary fracture is more beneficial.

8.4.2 Secondary Fracture Permeability

The next variable is the secondary fracture permeability. Similar to previous explanation that the value of the secondary fracture permeability cannot be set manually and cannot be measured. Therefore, in this sensitivity case, the author made an assumption regarding the range of permeability values that may be generated. For simplification, a variable called secondary fracture permeability multiplier is set to represent secondary fracture permeability. This multiplier was then multiplied by the hydraulic fracture permeability to represent permeability reduction that occurred due to the small volume of secondary fracture size relative to the main hydraulic fracture. Simulation case was run using 1 pair of secondary fractures with secondary fracture length 10, 30 and 60 ft and fracture permeability of 10,000 md. Length of secondary fracture was chosen based range of values that gave significant effect in previous sensitivity study about secondary fracture length. **Figure 8.15** until **figure 8.18** display the results of that sensitivity.

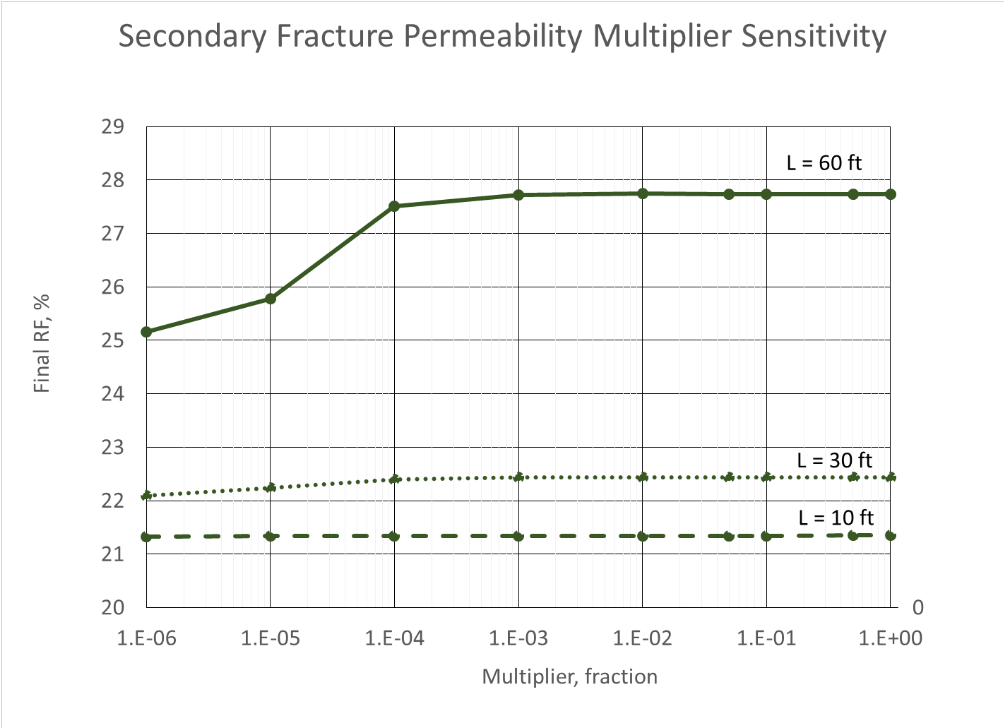

Figure 8.15 Final recovery factor of cyclic injection scenario at 10 years for different secondary fracture permeability multiplier and secondary fracture length

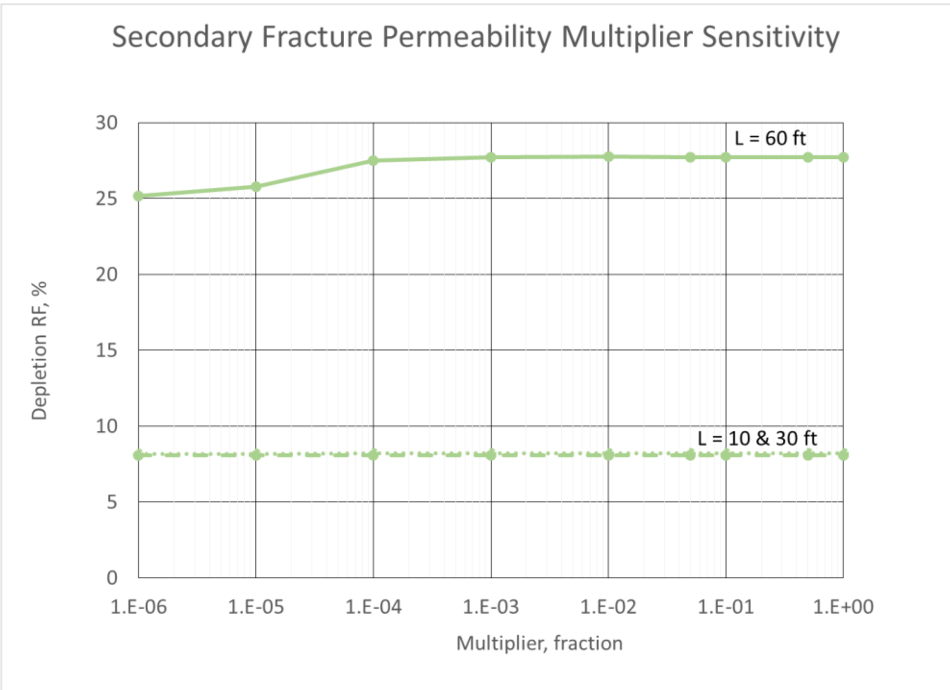

Figure 8.16 Depletion recovery factor of cyclic injection scenario at 10 years for different secondary fracture permeability multiplier and secondary fracture length

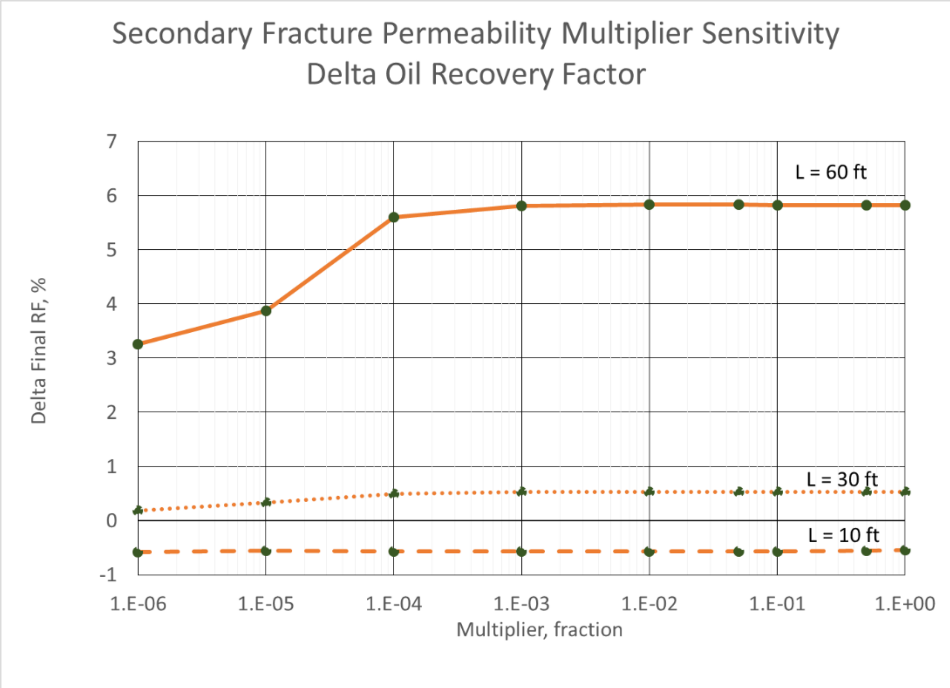

Figure 8.17 Delta final recovery factor of cyclic injection scenario against base case (continuous gas injection) at 10 years for different secondary fracture permeability multiplier and secondary fracture length

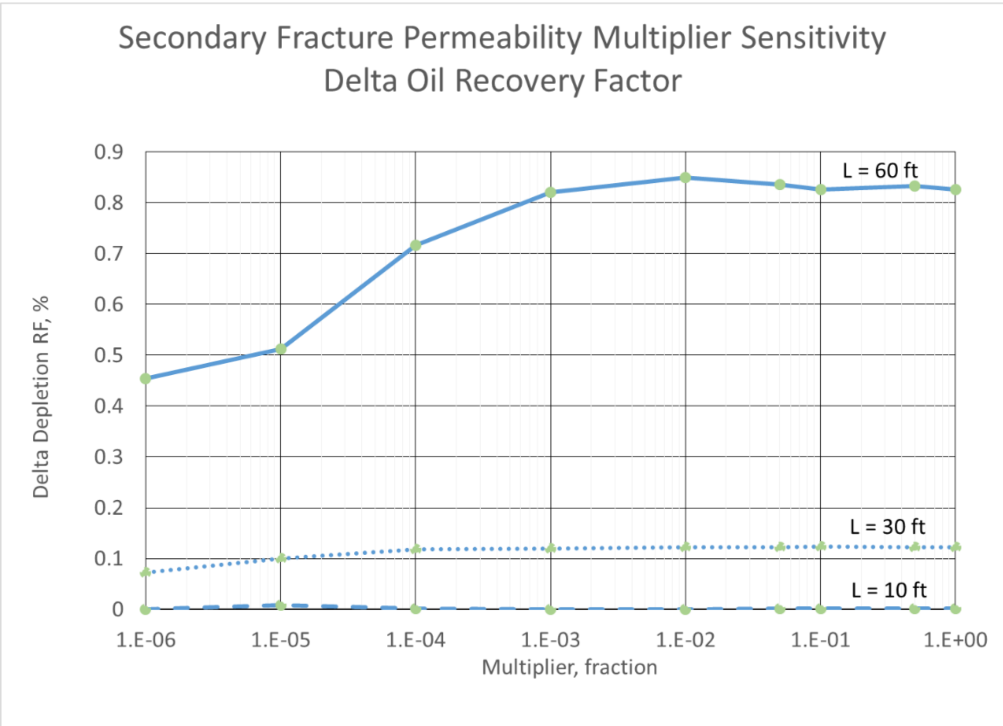

Figure 8.18 Delta depletion recovery factor of cyclic injection scenario against base case (continuous gas injection) at 10 years for different secondary fracture permeability multiplier and secondary fracture length

Based on the above plots, changes in secondary fracture permeability is insensitive against reservoir productivity for the case with length of 10 ft. However the change in final recovery result was detected in the case with length of 30 and 60 ft. In longer length, permeability multiplier $1x10^{-5}$ is become the threshold point at which the permeability decline beyond this point started to give effect to the performance of cyclic injection. The longer secondary fracture is generated, the greater reduction in oil recovery is resulted by the decline in secondary fracture permeability. The good news is, although the permeability value declines, cyclic gas injection will still provide better results compared to the base case of continuous gas injection as long as the secondary fracture's length exceeds 25 ft.

8.4.3 Secondary Fracture Intensity

The next variable to be simulated is secondary fracture intensity that is represented by the quantity of secondary fractures in reservoir model. In this case, simulation was performed with secondary fracture's number ranging from 1 pair up to 5 pairs. Properties of secondary fracture also used the same value with previous sensitivity; 10, 30 and 60 ft length and 10,000 mD permeability. **Figure 8.19** until **figure 8.21** present the result of sensitivity simulation mentioned.

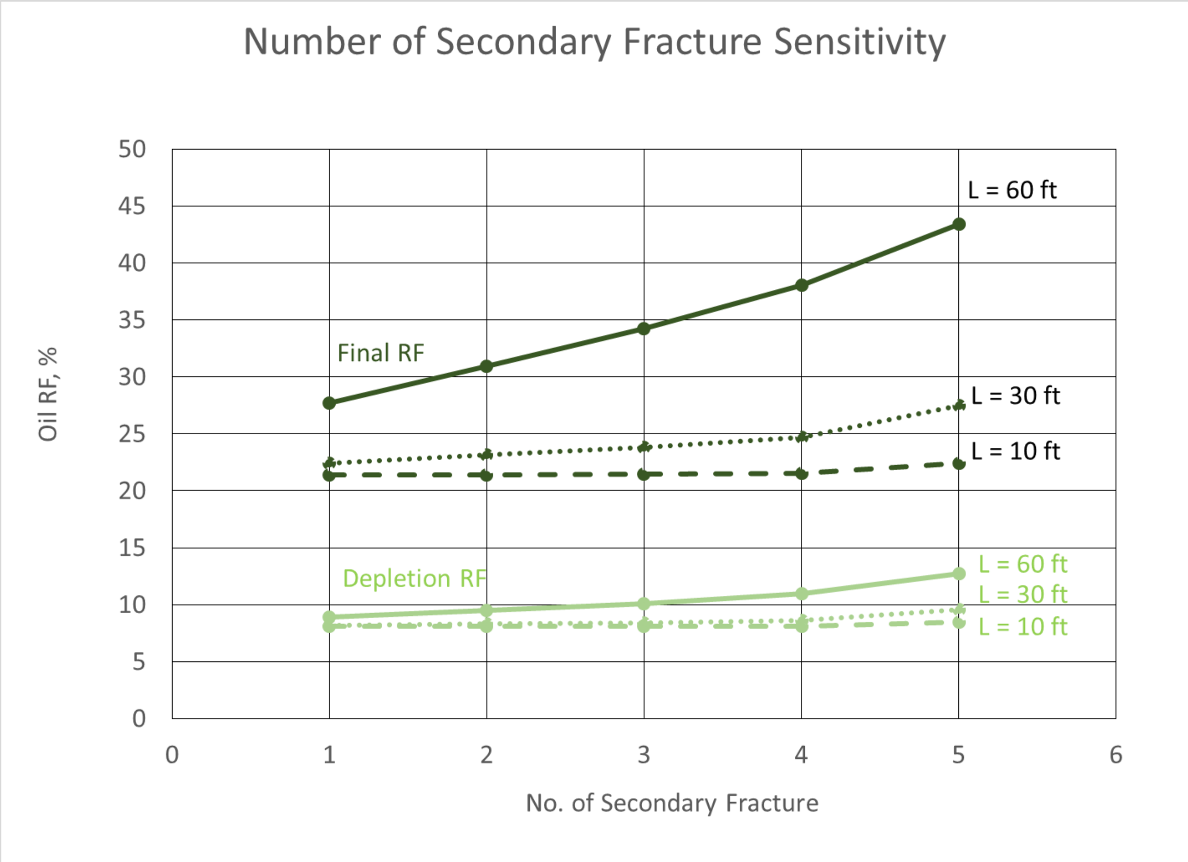

Figure 8.19 Oil recovery factor of cyclic injection scenario at 10 years for different number of secondary fracture and secondary fracture length

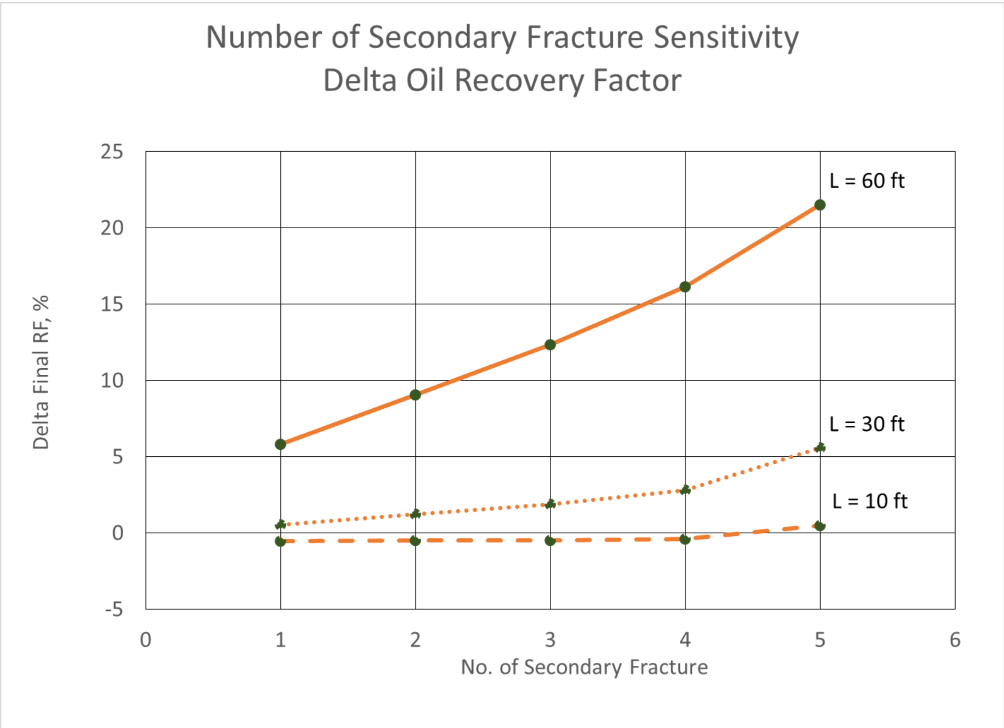

Figure 8.20 Delta final recovery factor of cyclic injection scenario against base case (continuous gas injection) at 10 years for different number of secondary fracture and secondary fracture length

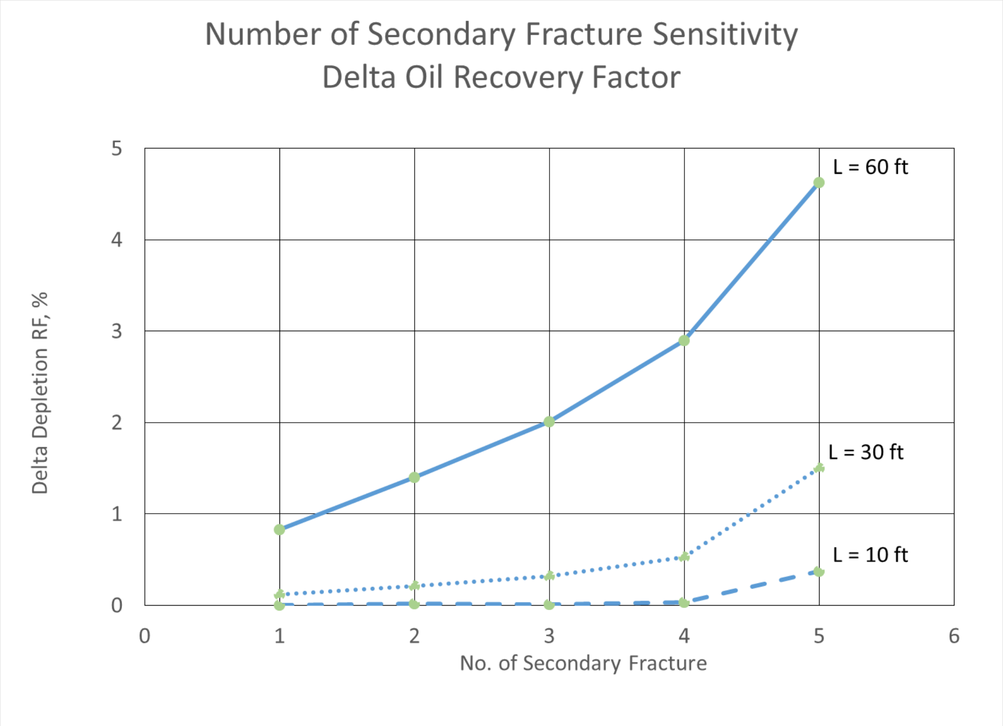

Figure 8.21 Delta depletion recovery factor of cyclic injection scenario against base case (continuous gas injection) at 10 years for different number of secondary fracture and secondary fracture length

Based on the simulation result with fracture length 10 ft, the intensity of secondary fracture is considered not influential to the reservoir performance. Effect started to be seen in extreme case which apply 5 pairs of secondary fractures. Extreme case assumed that secondary fractures are formed along its main hydraulic fracture. In this case, the performance of cyclic injection produced a slightly better result than the base case. However, because the magnitude of the generated recovery increase is considerably small even for the extreme case, it can be concluded that fracture intensity is insensitive to production performance for low secondary fracture length.

Then again the impact of intensity was seen in secondary fracture's length of 30 and 60 ft. Improvement in final oil recovery looks quite significant for any additions to secondary fracture intensity. Based on this results, it can be concluded that the greater number of secondary fractures formed, provided that they have reached the optimum length, the better effect it will give to the reservoir performance.

8.5 Matrix Permeability

The other sensitivity simulation was performed to the permeability matrix which is also one of reservoir uncertainties. The purpose of this sensitivity study is to see the potential of cyclic injection strategy to any possible reservoir matrix quality. In this study, matrix permeability is the only modified variable based on typical value of unconventional reservoirs, while the uncertainty of variable porosity was excluded from this study because the result is not comparable due to differences in initial oil in place at each case. The range of simulated numbers was from 1×10^{-5} mD to 1×10^{-1} mD. To see the result easier, the displayed plot is divided into two categories: smaller permeability and higher permeability. Each plot presents final and the depletion oil recovery as well as delta oil recovery against the base case. **Figure 8.22** and **figure 8.23** show the performance result for smaller permeability category while **figure 8.24** and **figure 8.25** show the result for higher permeability category.

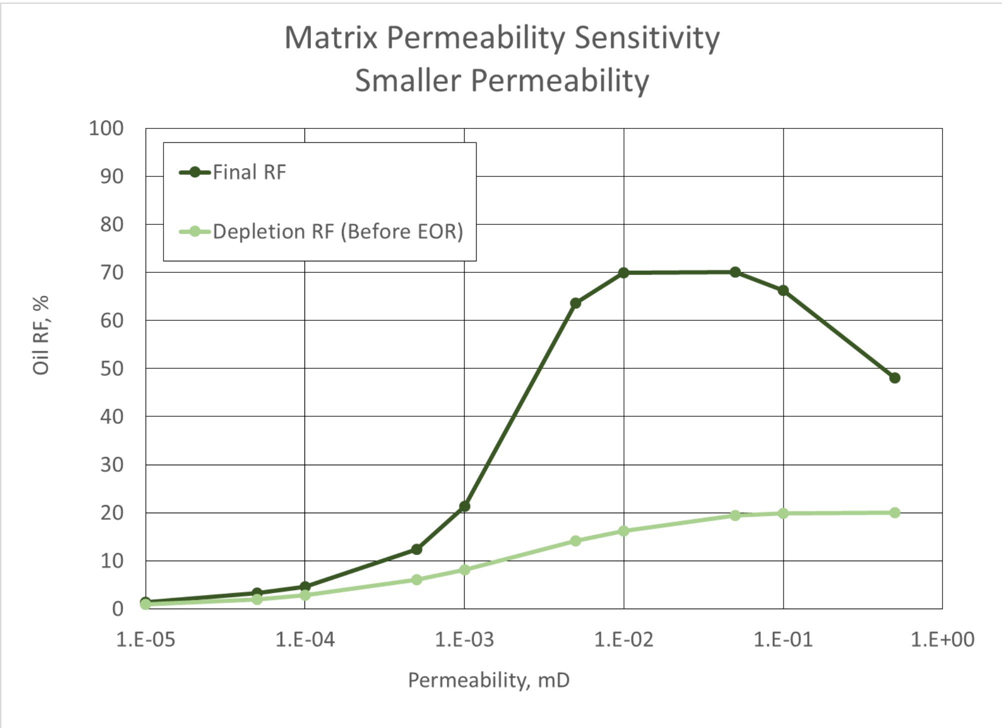

Figure 8.22 Oil recovery factor of cyclic injection scenario at 10 years for different reservoir matrix permeability (smaller permeability interval)

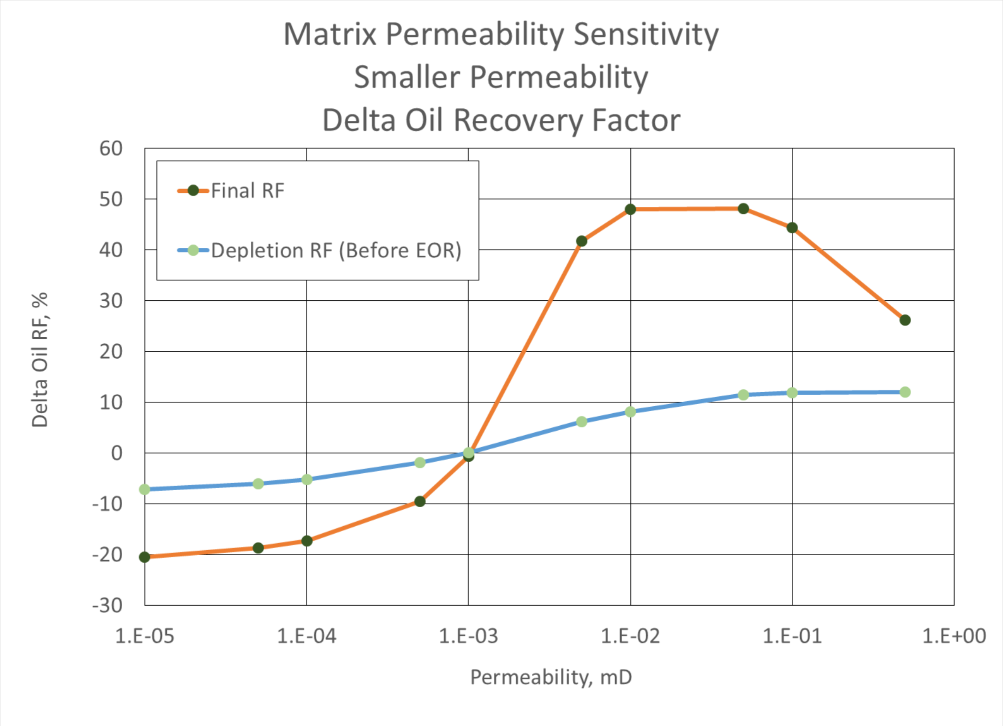

Figure 8.23 Delta oil recovery factor of cyclic injection scenario against base case (continuous gas injection) at 10 years for different reservoir matrix permeability (smaller permeability interval)

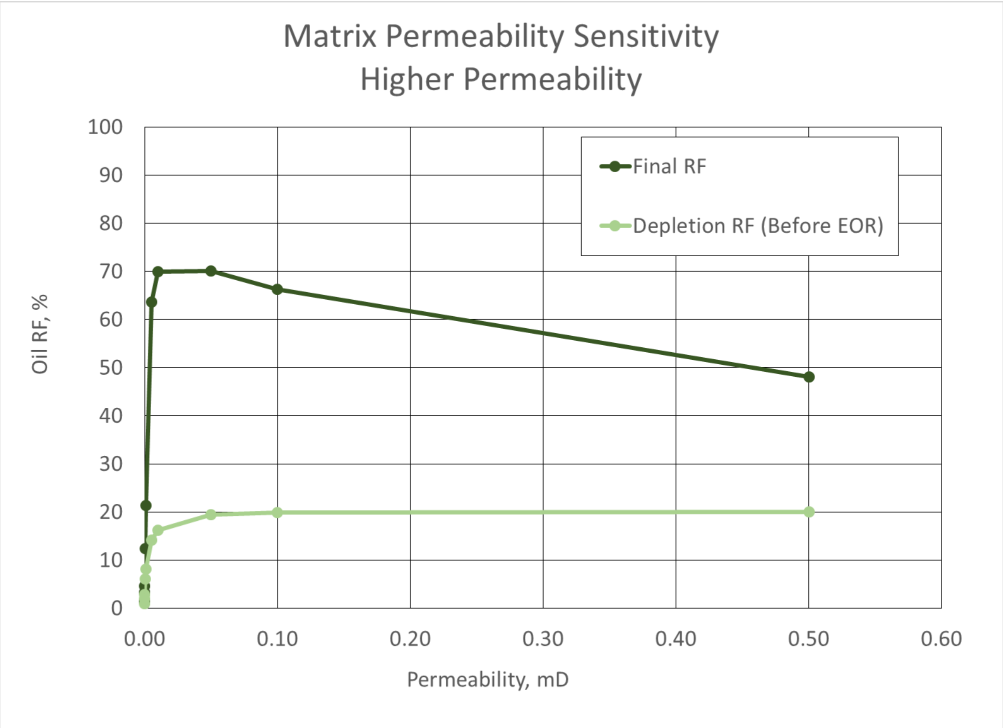

Figure 8.24 Oil recovery factor of cyclic injection scenario at 10 years for different reservoir matrix permeability (higher permeability interval)

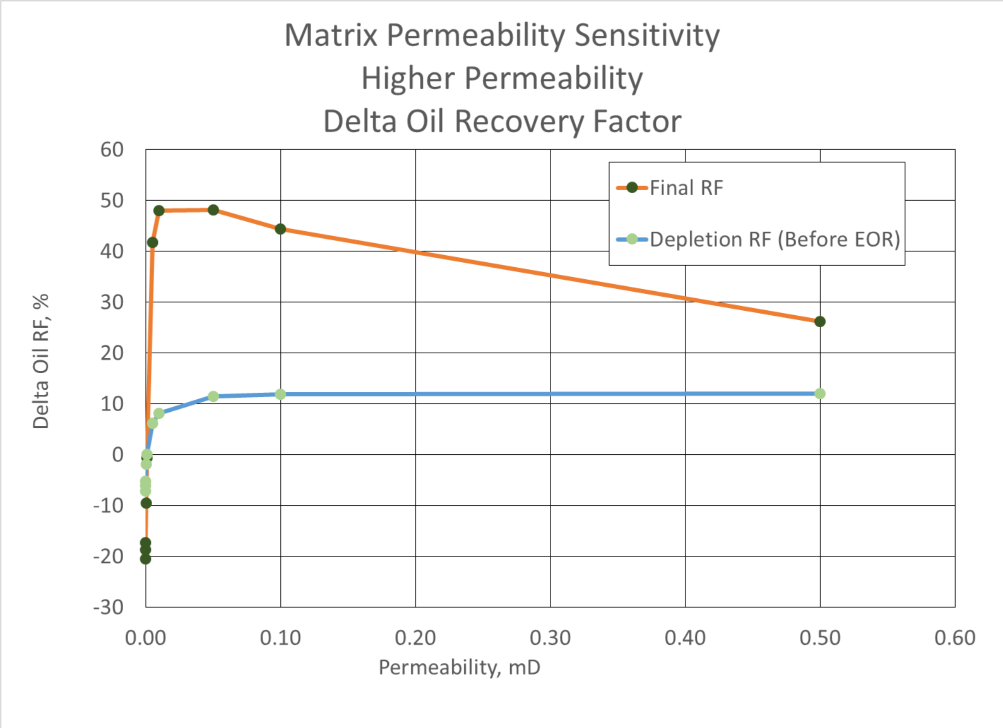

Figure 8.25 Delta oil recovery factor of cyclic injection scenario against base case (continuous gas injection) at 10 years for different reservoir matrix permeability (higher permeability interval)

Based on simulation results, we can conclude that better reservoir quality facilitates more effective use of cyclic gas injection. As seen from the increase in final oil recovery factor that far exceeding the increase in its depletion oil recovery. However, the efficiency of cyclic gas injection will become stagnant in very high permeability interval, which represented by value above $1x10^{-1}$ mD in the model.

At low permeability interval, the permeability default value ($1x10^{-3}$ mD) divided the simulation performance into two. In reservoir with poor quality below the default value, the increase in permeability only contributed a little effect to cyclic gas injection performance, as detected from the very small difference between the final oil recovery and depletion oil recovery. While at better reservoir quality above the default, the permeability increase gave significant effect then reached its stagnation point at permeability around $1x10^{-2}$ mD, then finally began to change the trend down as the permeability reach $1x10^{-1}$ mD.

Based on the simulation, we can conclude that there is an optimum matrix permeability interval for cyclic gas injection application. Based on the reservoir model used, the interval is ranging from about $5x10^{-4}$ mD to about $5x10^{-2}$ mD. At this range, the greater matrix permeability results in greater benefit generated by cyclic gas injection strategy. However, the benefit of cyclic injection will decrease at very large permeability matrix.

8.6 Lateral Heterogeneity

The previous matrix permeability sensitivity used an assumption that the reservoir is homogeneous and has uniform permeability in the entire reservoir. In fact, heterogeneity will be found within the reservoir and it becomes an important uncertainty to be assessed. So another simulation sensitivity which consider reservoir heterogeneity in lateral x and y direction were also performed. The model used for this simulation incorporate lower permeability region laterally in reservoir with purpose to represent permeability barrier as the worst case of its lateral heterogeneity.

To more easily understand the shape of the permeability barrier, **figure 8.26** illustrates the permeability barrier distribution on x-direction and **figure 8.27** illustrates the permeability barrier distribution on the y-direction. Permeability barriers modeled on both x and y direction are adjusted so the area affected by the permeability reduction is equal, despite its difference in distribution and orientation. The figure shows that the permeability barrier made on x-direction is in the form of alternating order consisting of 10 vertical columns with 11 ft in length (x-direction) and 500 ft in width (y-direction), while the permeability barrier on the y-direction is in the form of 5 horizontal lines with the same length, 200 ft (x-direction) but different width which adjusts the default geometry shape of reservoir grid models. With this adjustment, the total area of permeability barrier in both approaches is the same.

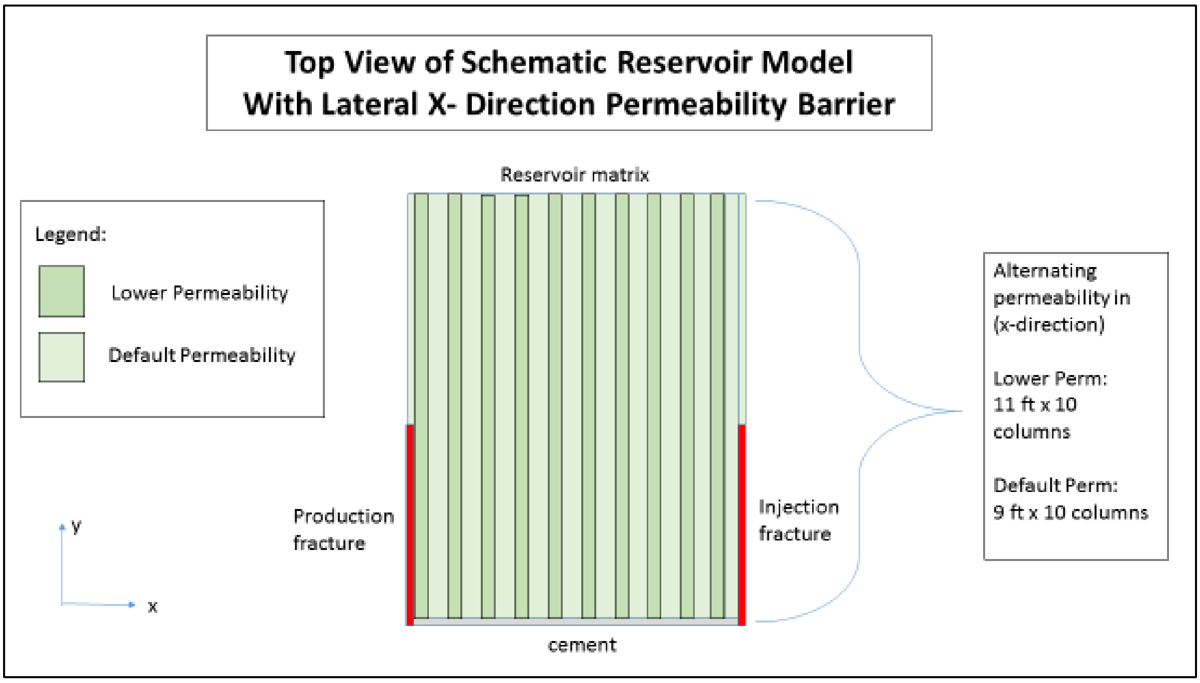

Figure 8.26 Illustration of the modeled lateral permeability barrier in x direction

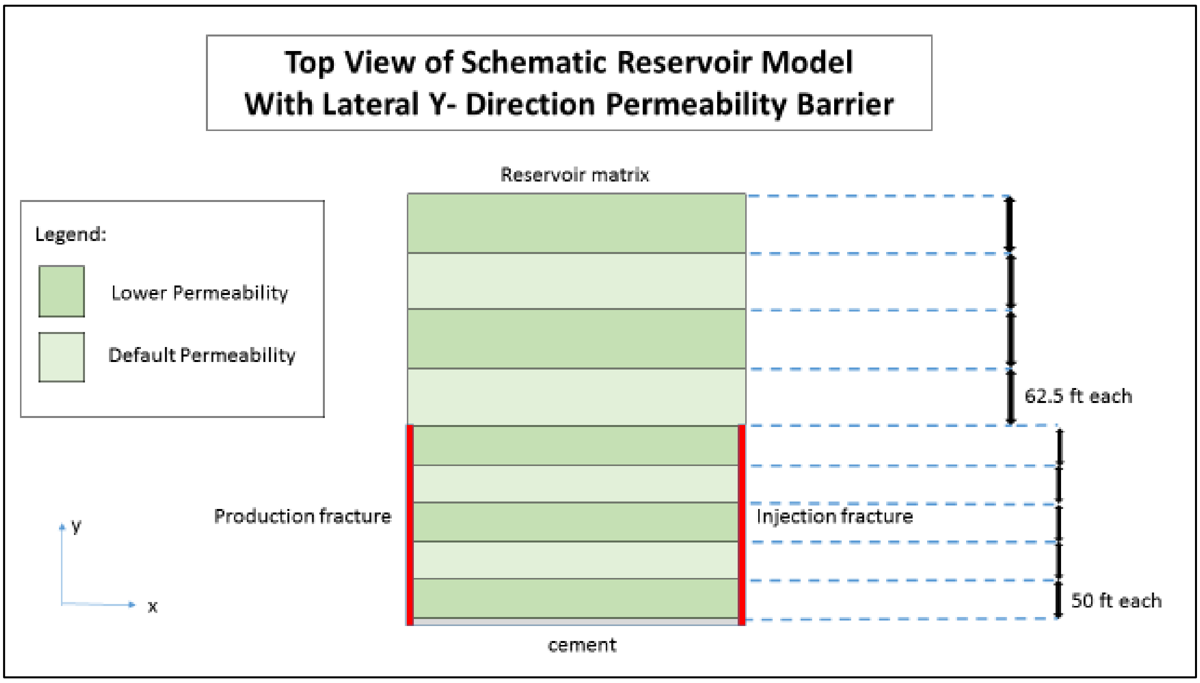

Figure 8.27 Illustration of the modeled lateral permeability barrier in y-direction

Two cases as the upper limit and lower limit cases were also simulated as references in comparing the performance of each simulation. The cases simulated consist of; upper limit case using main permeability value which is the default permeability 1×10^{-3} mD, the lower limit case using minus 30% default permeability, and the case with permeability barrier zone which has variation of default and lower permeability in the model.

Six main sensitivity cases were prepared to see the effect of barrier in x-direction, y-direction and both directions at the same time. **Figure 8.28** shows the result in terms of depletion recovery factor and recovery factor increment as the result of cyclic gas injection for particular reservoir condition. **Table 8.1** lists down the cases description with its respective permeability barrier value and the resulted final oil recovery factor.

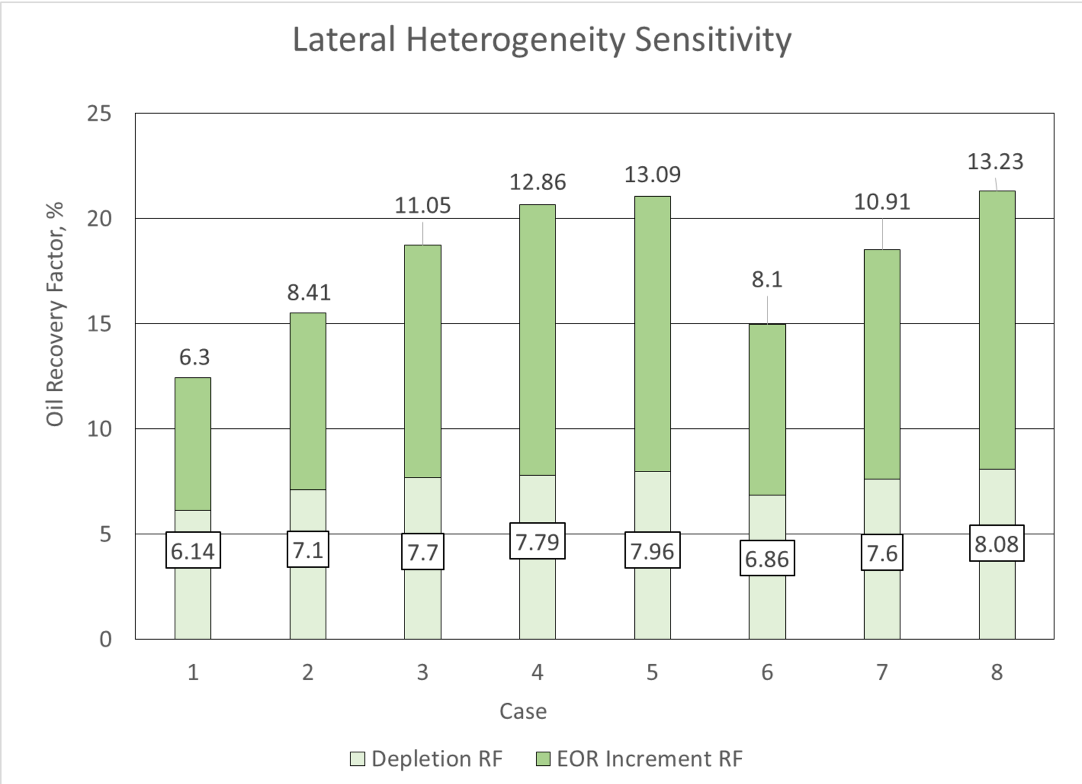

Figure 8.28 Oil recovery factor of cyclic injection scenario at 10 years for different lateral reservoir permeability barrier

Table 8.1 Lateral heterogeneity sensitivity cases summary

Case	Remarks	Final Recovery Factor (%)
1	Homogen, lower permeability case (constant 0.0005 mD)	12.44
2	X direction barrier case (0.0005 mD)	15.51
3	X direction barrier case (0.00075 mD)	18.75
4	Y direction barrier case (0.0005 mD)	20.65
5	Y direction barrier case (0.00075 mD)	21.05
6	X & Y direction barrier case (0.0005 mD)	14.96
7	X & Y direction barrier case (0.00075 mD)	18.51
8	Homogen, default permeability case (constant 0.001 mD)	21.31

Based on simulation result, it appears that the lateral heterogeneity in x-direction (represented by the second and third cases) has larger effect compared to the lateral heterogeneity in y-direction (represented by fourth and fifth cases). Case 4 and case 5 did not significantly harm the oil recovery compared to the ideal case (case 8), where the recovery reduction is less than 0.66%. Even the combination of permeability barrier in both direction is just resulted in small decrease compared to only x-direction. Then it concluded that the permeability barrier that is parallel to direction of gas injection movement does not give much productivity reduction compared to the perpendicular permeability barrier. From **table 8.1**, based on the final oil recovery factor, it is seen that the value of all cases with lateral heterogeneity is still relatively much larger than the lower limit case. These results also support the recommendation to apply cyclic injection strategy even in very heterogenic reservoir, although it allows the permeability barrier in any direction.

9 Optimization of Controllable Variables in Simulation

Study sensitivity was also conducted to see the effect of the controllable variables to the reservoir performance. Just like the previous series of sensitivity study, the results displayed in this sensitivity only covered the oil recovery factor. The reason is the same with previous section, because of oil recovery is the main goal of the cyclic gas EOR implementation.

The result will be shown in terms of final oil recovery, which means the oil recovery value that resulted after 10 years of simulation, consists of 2 years natural production and 8 years cyclic gas injection. The result is also shown as delta oil recovery against the base case (the difference between oil recovery in simulated case and continuous gas injection scenario) to better describe the comparison.

This section discussed about the simulation result by modifying the value of variable that is considered as controllable from the surface. This study is intended to determine what effect that would be generated by the change in technical scenario or simulation control to EOR application, which aiming to know the optimum operating condition for the field development.

9.1 Cyclic Interval

The most important variable in this section is the cycle time. Cycle time is the duration of injection or production that is set while EOR process is taking place. Knowing the optimum cycle time is essential because its application becomes the main driver for EOR performance. Also to determine the benefits of applying cyclic gas injection since it is closely related to how much gas injection volume can be reduced to keep producing its optimum production rate. The simulated value ranges started from as low as 10 days to as high as 365 days. **Figure 9.1** exhibits the simulation results in terms of oil recovery factor and its delta to the base case. Depletion oil recovery is not shown here because variable cyclic interval will only affect the result after the gas injection process begins, not before the process.

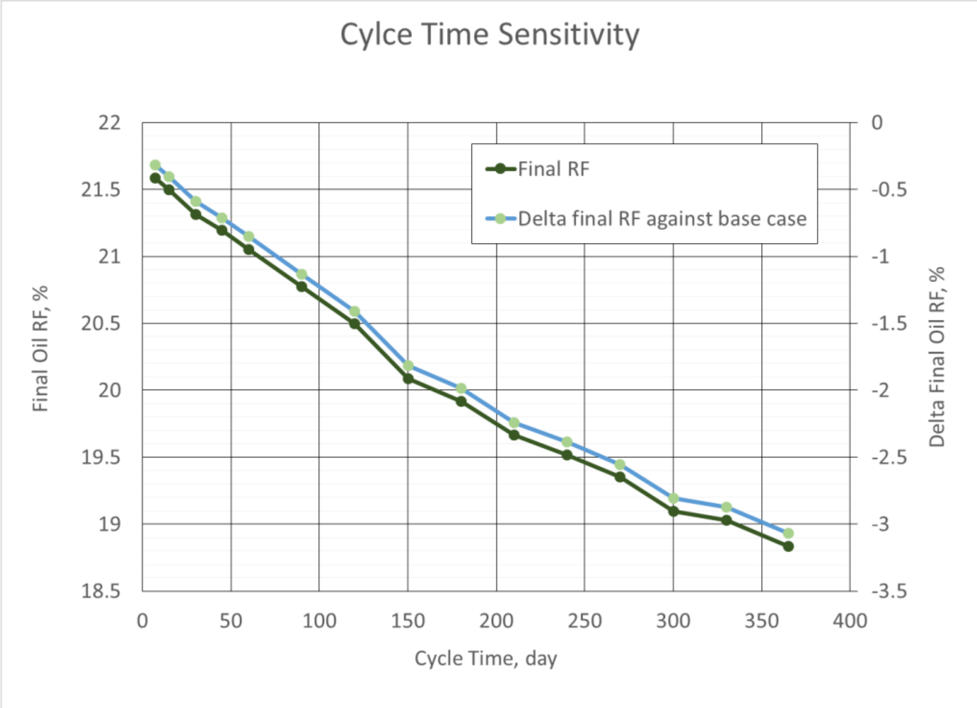

Figure 9.1 Oil recovery factor of cyclic injection scenario at 10 years for different cycle time and the delta oil recovery against base case (continuous gas injection)

Based on the simulation, cycle time of 30 days is the optimum value since the oil recovery reduction from continuous gas injection scenario in the range of 10 to 30 days is relatively minor and acceptable. It is considered acceptable because gas injection reduction obtained will greatly reduce the operating cost by sacrificing insignificant amount of production, in the range of 0.5% only. By applying longer cycle time, oil recovery reduction began to increase and efficiency began to decrease.

9.2 Injection Rate Target

The next variable is operating gas injection volume target during injection cycle. Gas injection rate target is one of variable which becomes the simulation control. The goal is to find out the minimum target required to get optimum production performance, as well as to know the production lost due to injection rate reduction. Initial model were using 1000 MSCFD as injection target, the number is made high enough to allow the simulation adjusting the target as maximum as it can, as determined by injection bottom hole pressure constraint. **Figure 9.2** shows the simulation result in terms of oil recovery factor and its delta to the base case.

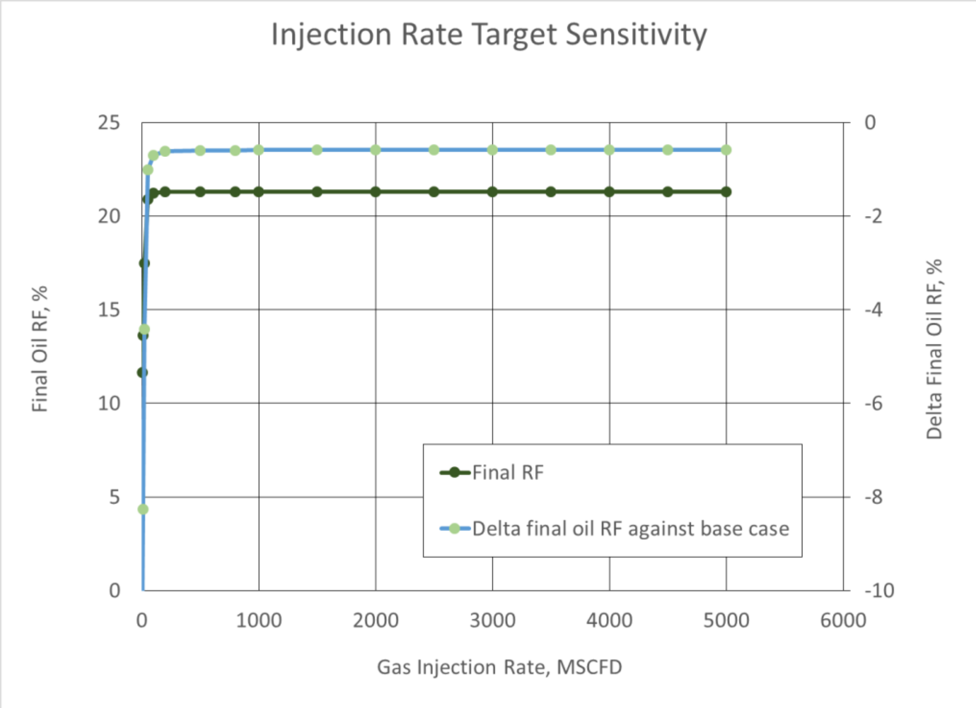

Figure 9.2 Oil recovery factor of cyclic injection scenario at 10 years for different injection rate target and the delta oil recovery against base case (continuous gas injection)

Based on the simulation result, the minimum target injection required to achieve optimum production is 100 MSCFD. The change in injection target above it, is insensitive to the production performance. It is seen that above 100 MSCFD injection rate, the production performance will give constant result where there is no point to increase the target beyond, since the reservoir will only need relatively small volume to achieve the pressure needed. As discussed in chapter 6, because of low porosity and low permeability, a little volume of injected gas would directly increase the reservoir pressure significantly. In short injection period, reservoir pressure returned to its bubble point and improved the efficiency of displacement process of cyclic gas injection.

However, if the scale is reduced as seen in the delta oil recovery against a base case, the point at which it began to produce more stable performance is actually starting from 1000 MSCFD injection target rate, which is the default case of injection target control. Therefore, for the simulation study purpose, injection target rate used is the default value.

9.3 Production Rate Target While Cyclic

The next variable is the production volume target during the production cycle. Production rate target is one variable of simulation control. The goal is similar to sensitivity study in previous simulation control, which is to determine the minimum target required to obtain optimum production performance. By knowing the minimum value, it will also show how much potential that may be lost if choke is used. Initial model is also set as quite high value of 1000 STBD for the same reason, to let the simulation adjusts itself according to bottom hole pressure constraint. **Figure 9.3** shows the simulation results in terms of oil recovery factor and its delta to the base case.

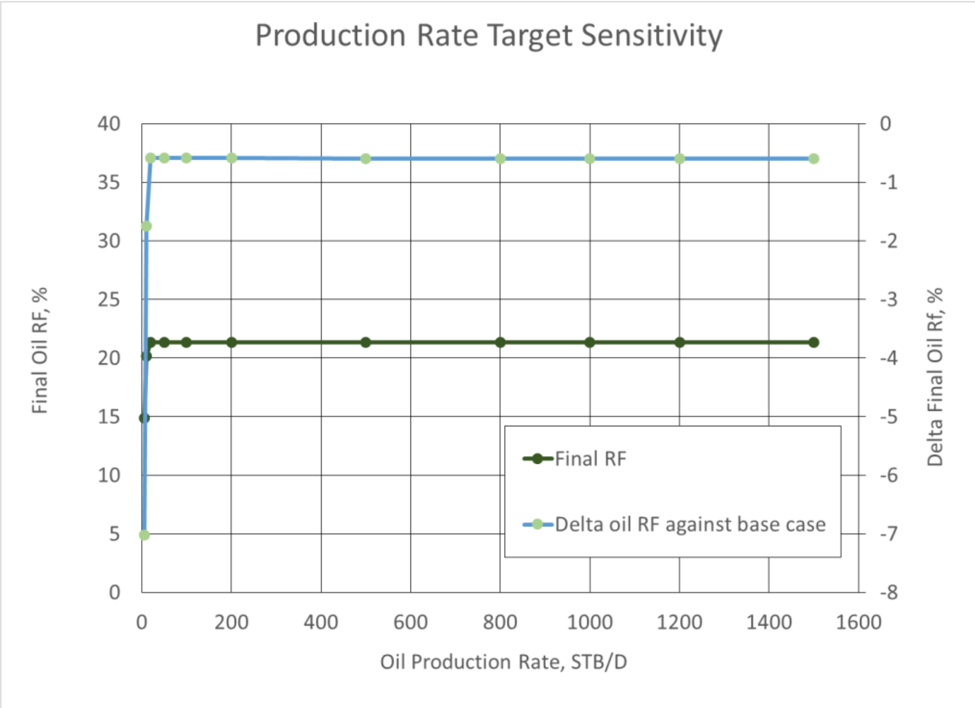

Figure 9.3 Oil recovery factor of cyclic injection scenario at 10 years for different production rate target and the delta oil recovery against base case (continuous gas injection)

Based on simulation result, it appears that the minimum production targets required to produce optimum production performance is 20 STBD. Similar to the injection rate target, changing the target rate above its minimum required value was also proven to be insensitive to the production performance. Above that value, reservoir performance will give relatively constant result. It is also illustrated from the result that the maximum reservoir ability in each cycle is to produce accumulated volume that is equal to the volume generated by the average rate of 20 BOPD.

For simulation purpose, the author chose to keep the default 1000 STBD production target control in case a better scenario will be found in further study which could increase the production even higher.

9.4 Cyclic Injection Start Day

Basically, this sensitivity is similar to the performed sensitivity in chapter 3 concerning the gas injection start day. The difference here is the sensitivity purpose, which is to know whether there is any difference regarding the optimum time to begin the gas injection with cyclic injection strategy. Cyclic injection start day is another variables which is categorized as technical scenario. **Figure 9.4** shows the simulation results in terms of final and depletion oil recovery.

Based on the simulation, the results obtained are the same with continuous gas injection start day sensitivity, which came up with 730 days as the optimum time to begin the cyclic gas injection strategy. The reason is also the same, because the decrease in oil recovery with two years delay is not significant compared to the longer delay. The delay of one year to two years would only reduce the final oil recovery of about 0.5%, while another delay to the third year already reduce the final oil recovery by more than 1.5%. The generated depletion oil recovery profile looks quite stable with a constant increase in every delayed year. It can be concluded that final oil recovery becomes a major factor in determining the cyclic injection start day.

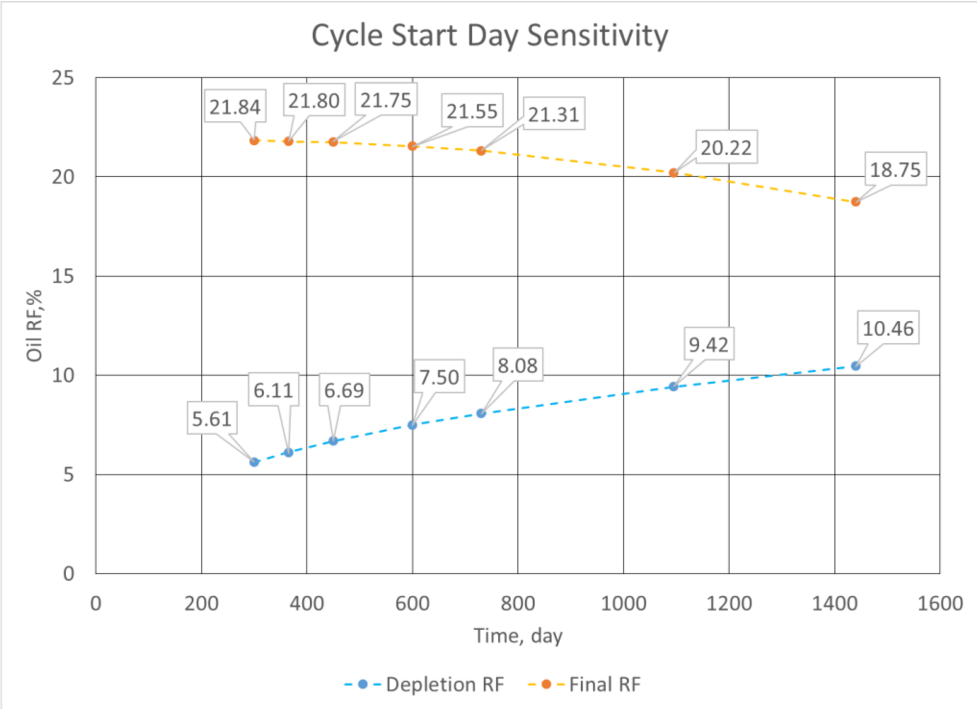

Figure 9.4 Oil recovery factor of cyclic injection scenario at 10 years for different cyclic injection start day

9.5 Inter-Fractures Distance

Sensitivity was then performed to the variable distance between two neighboring fractures. Inter-fractures distance is also categorized as variable of technical scenarios. This simulation involved some value that falls around its default value (200 ft) which is the typical distance for hydraulic fracturing in shale oil reservoir. The purpose of this sensitivity is to give an idea about the effect of hydraulic fractures placing to the cyclic gas injection performance. **Figure 9.5** and **figure 9.6** displays the simulation result in terms of oil recovery and its delta to the base case respectively.

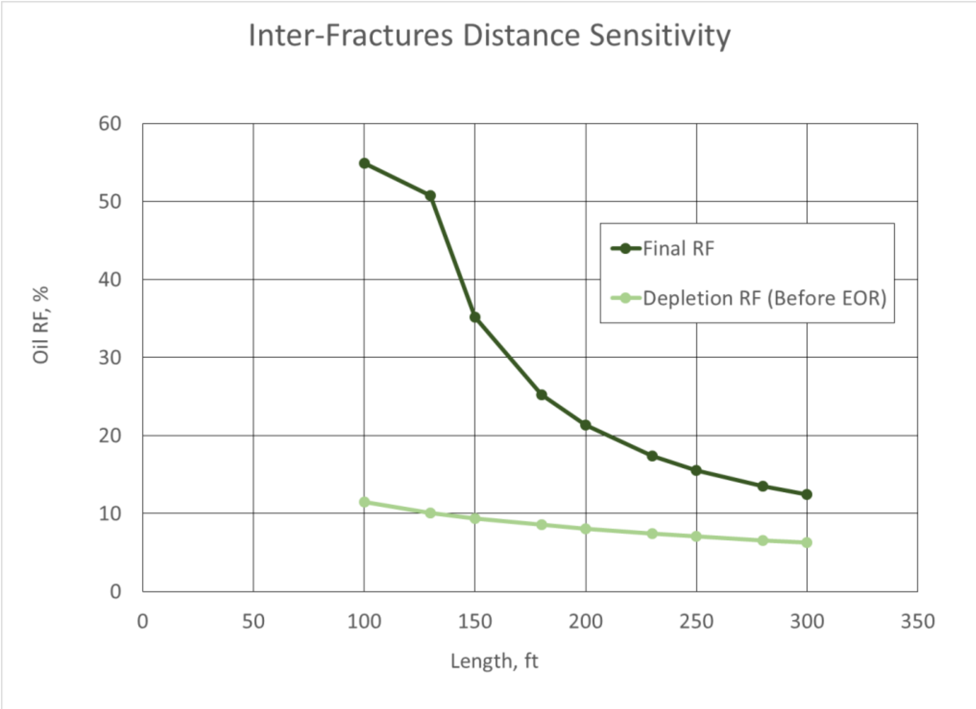

Figure 9.5 Oil recovery factor of cyclic injection scenario at 10 years for different inter-fractures distance

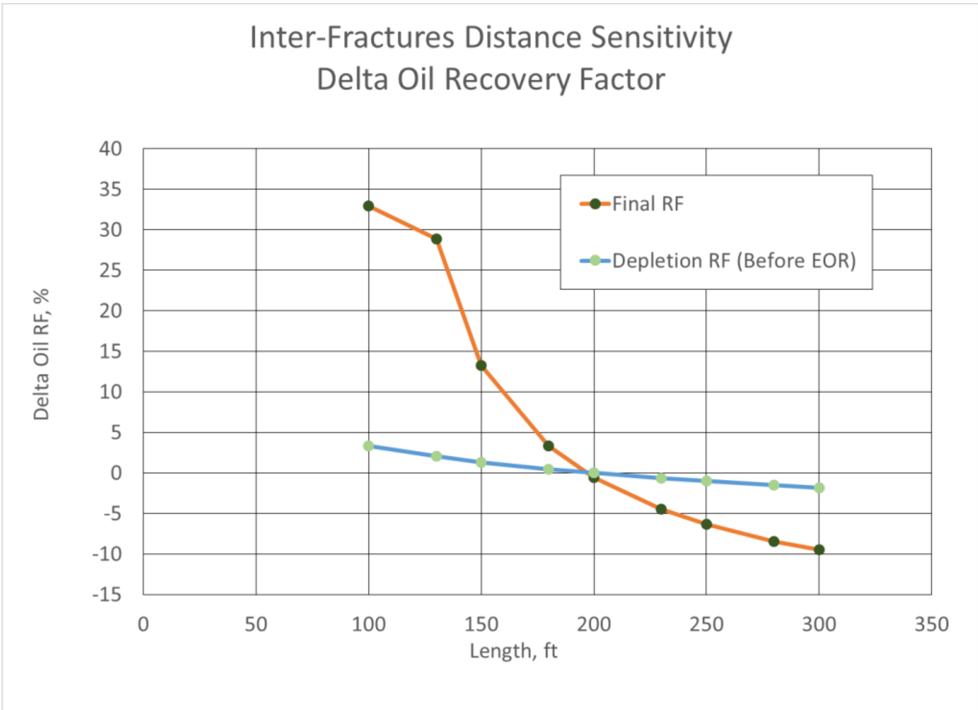

Figure 9.6 Delta oil recovery factor of cyclic injection scenario against base case (continuous gas injection) at 10 years for different inter-fractures distance

Based on the simulation result, change in inter-fracture distance is highly influential to the production performance. Shortening the distance between injection and production fracture could add significant support to the efficiency of displacement by the injected gas. The closer the distance between fractures, the greater the resulting oil recovery will be. Even very small change to reduce the distance from its default value 200 ft has been able catch up the performance of the base case scenario (continuous gas injection). Although the compared base case was still using the default distance of 200 ft, assuming that oil recovery of that case is the benchmark of comparison, a little effort to reduce fractures spacing could generate huge improvement.

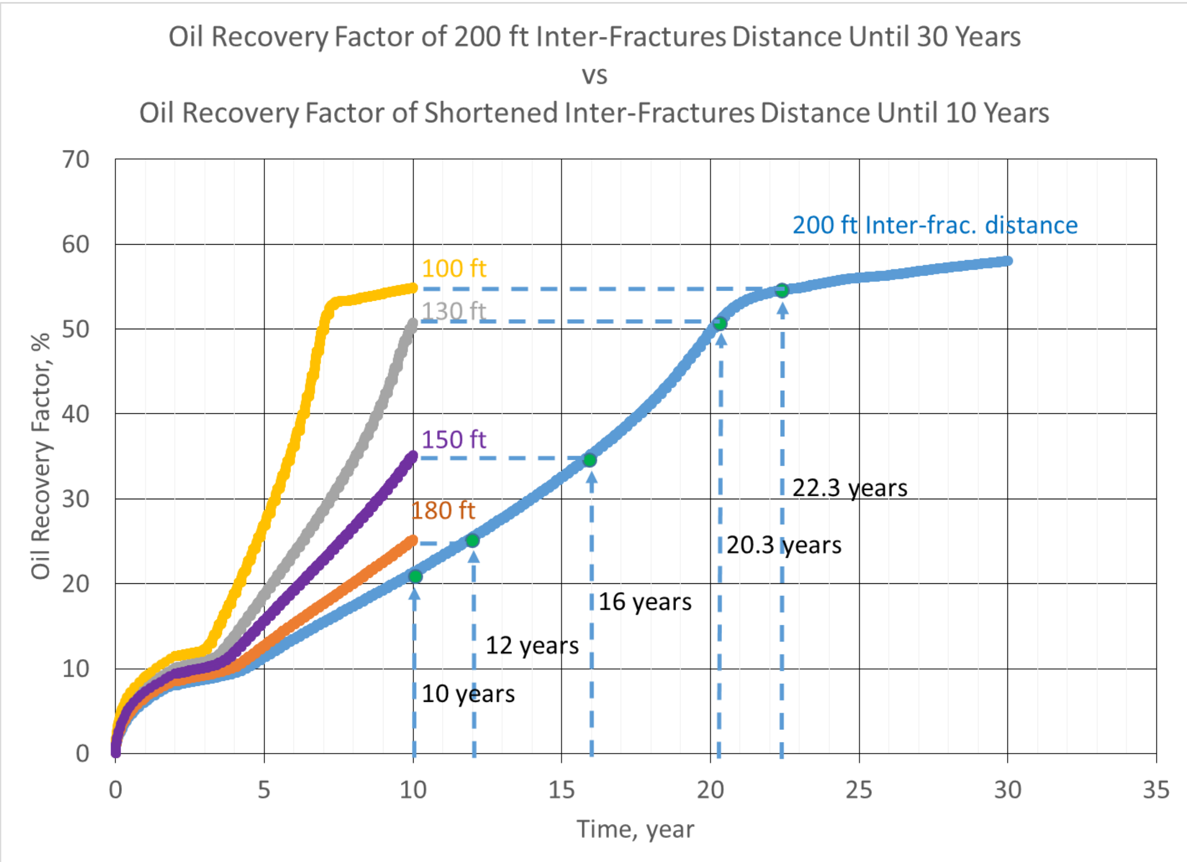

Figure 9.7 Oil recovery factor of different inter-fractures distance at 10 years against case with 200 ft inter-fractures distance until 30 years

Figure 9.7 shows the comparison of simulation result that were obtained by shortening inter-fracture distance to the simulation using default inter-fractures distance of 200 ft that were simulated by the same cyclic injection strategy but with longer simulation time. It showed that shortening the inter-fracture distance resulted in oil recovery acceleration which will also be acquired by default distance with a longer time. For example, if the base case with 200 ft inter-fracture distance resulted in about 21% oil recovery at the end of the year 10, shortening the inter-fracture distance to 180 ft distance could increase the oil recovery to about 27% in the year 10. Similar result was obtained to the base case with 200 ft distance in year 12. Even shortening the distance further to 150 ft, in only 10 years it will produced the same oil recovery with the base case of 200 ft in the 15 years, and it continues that way.

10 SWEOR Alternative Strategies

Reservoir models created could be occupied to perform simulation on the alternative application of same-well EOR. In this section, author discusses the simulation results of applying the shut in between injection cycles and huff and puff. Comparison will be shown in terms of final oil recovery factor against proposed EOR strategy that is further referred as normal cyclic injection strategy to facilitate comparison. The simulation was performed with the same scenario, in which each case consists of two years depletion phase, followed by EOR in the third up to tenth year. Cycle time was set as 30 days, in accordance with the optimum cycle time obtained for normal cyclic gas injection, to make it comparable to each strategy's optimum condition.

10.1 Shut In Period between Cycles

The first strategy simulated was shutting-in wells between the injection and production. In this strategy, the shut-in was done to see whether allowing injection of gas and oil in the reservoir matrix to interact much longer before the start of the injection process or before the production process will give good results or not. Fracture configuration applied was the optimum configuration with alternating injector and producer fractures.

10.1.1 Shut In Before Injection

The first scenario for this strategy was inserting a shut-in period prior the injection. In this case, the injection cycle was inserted with the shut-in period before the gas injection began. Two cases with different shut-in duration was prepared. In first case, shut-in period was as long as the injection period, in which the well was shut for 15 days, followed by gas injection for 15 days. In second case, shut in period was shorter than the injection period, where the well was shut for 8 days then followed by injection of gas for 22 days. The total duration of the shut-in and the gas injection was set according to the interval cycle, so that the production period in the simulation remains the same with the optimum cycle time of a normal cyclic injection, which is 30 days. **Figure 10.1** shows a comparison of the normal cyclic injection strategy of the two cases mentioned.

10.1.2 Shut In After Injection

Not much different from the shut-in before injection, this strategy also basically inserted a shut-in period on the gas injection cycle. The difference was only on the placement of shut-in period, since it was applied in reverse of the previous scenario. Two cases using the same shut-in duration with previous scenario was prepared, which are the case with the 15-days injection followed by 15-days shut-in, and a case with 22-days of injection, followed by 8-days shut-in. **Figure 10.1** shows its comparison to the normal cyclic gas injection strategy.

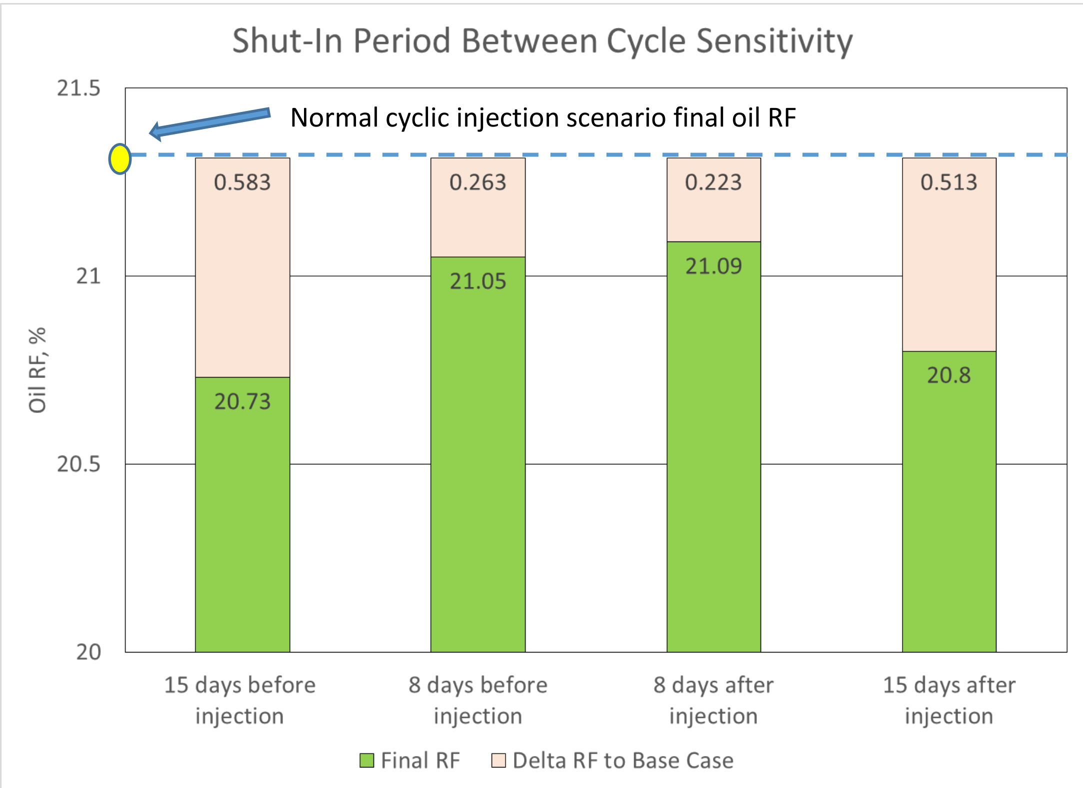

Figure 10.1 Oil recovery factor of cyclic injection scenario at 10 years for cases with shut-in period between cycles and the oil recovery loss compare to normal cyclic gas injection scenario (amount of brown box)

The bar chart in **figure 10.1** consists of final recovery factor result that is represented by green bars and its delta oil recovery against normal cyclic gas injection strategy that is represented with light brown bars. It is seen that all alternative cases produced smaller final oil recovery. There were no significant differences between doing the shut-in before or after the injection. Greater reduction in oil recovery was detected if shut-in period was extended. Based on that simulation results, we can conclude that adding shut-in period between cycles does not provide any benefits for reservoir production but it can reduce oil recovery, especially if the shut-in period lasts longer.

10.2 Huff-n-Puff

Another tested alternative strategy to be simulated is huff and puff. Huff and puff duration were set similar to the cycle time of normal cyclic injection strategy, which is 30 days each. In this strategy, all fractures were similarly functioned in each cycle, either as the injector or the producer. **Figure 10.2** shows the simulation results of this strategy.

The bar chart presents that, based on this model, huff and puff strategy did not produce a significant improvement in the reservoir production. Even the resulted final recovery factor in this strategy was much smaller than the normal cyclic injection strategy. Depletion phase for 2 years produced 8.09% oil recovery, and huff and puff could only add less than 1% oil recovery at the end of year 10 in the simulation.

Figure 10.3 shows the result of performed sensitivity by modifying the cycle time of huff and puff. It can be interpreted that shortening the cycle time does not provide meaningful improvement, where 10-days intervals only produced a final oil recovery slightly larger than 10%. From the orange graph, it appears that shortening cycle time of normal cyclic injection strategy's optimum cycle time (30 days) only contributed to addition of less than 2% in final recovery factor. While rising the duration from 30 days to 90 days did not result in significant reduction of final oil recovery. Then it can be assumed that cycle time is insensitive and unhelpful for huff and puff to boost the production.

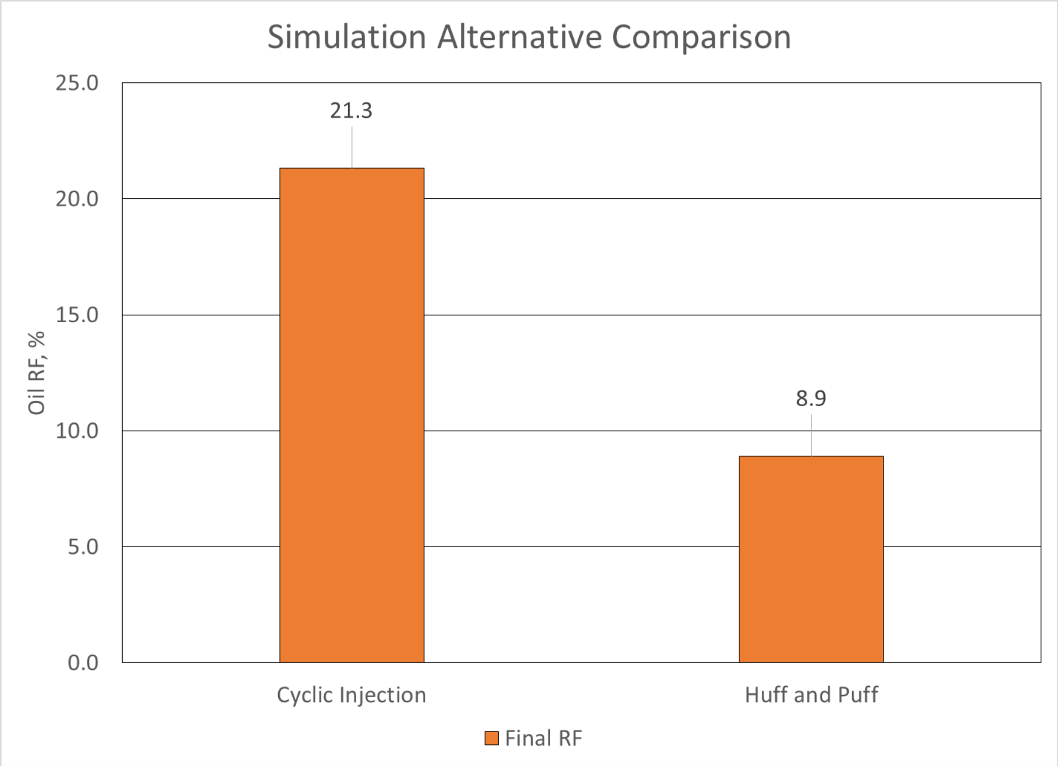

Figure 10.2 Oil recovery factor at 10 years comparison for cyclic injection against huff and puff

Figure 10.4 presents the sensitivity result that is performed by replacing the inter-fractures distance. The best final recovery from case with 150 ft inter-fractures distance gave relatively poor performance, which is not far exceeding 10%. Changing the distance length was also proven as not helping to increase the production of the reservoir in which the results obtained by shortening the distance to 150 ft only contributed to addition of less than 2% in year 10. From the orange line, it is shown that the change in distance from the base case of 200 ft just produces little change in final oil recovery factor. The addition of distance by up to 300 ft only reduced the final recovery by less than 2%. Thereby, the distance change in huff and puff strategy can also be concluded as insensitive and it cannot help to catch up with the performance in normal cyclic gas injection strategy.

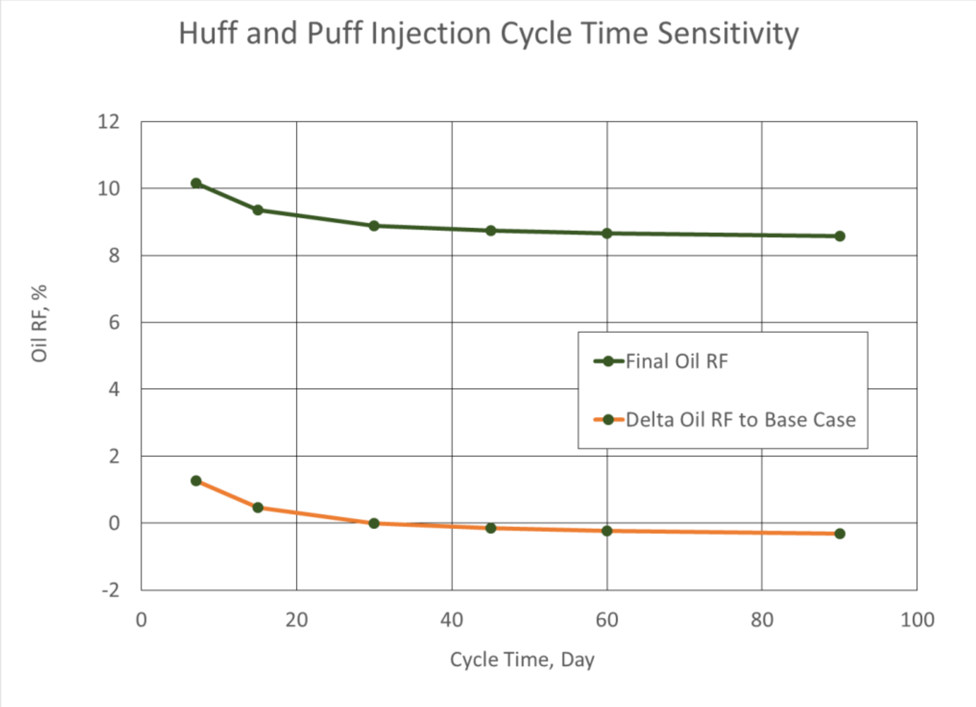

Figure 10.3 Oil recovery factor of huff and puff scenario at 10 years for different cycle time

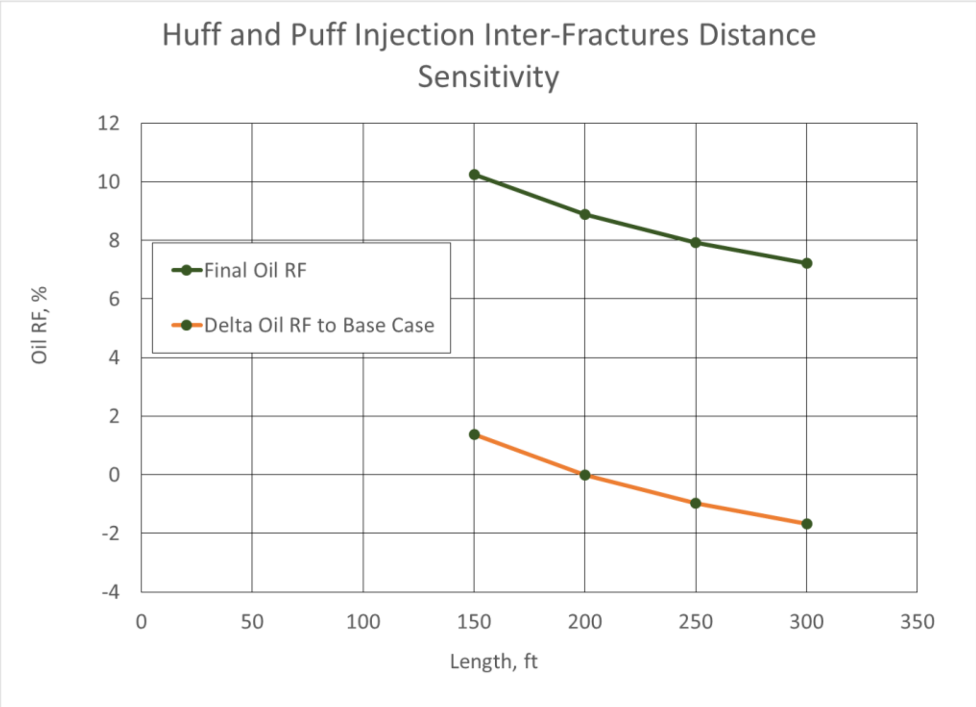

Figure 10.4 Oil recovery factor of huff and puff scenario at 10 years for different inter-fractures distance

11 Results

This part discusses summary of all sensitivity simulation result that has been done along with its uncertainty analysis to determine how the reservoir behaves, and how the optimum strategy could be applied in the development of shale oil reservoir using gas injection EOR. In the elaboration, the result of the simulation is described in general and it focuses on oil recovery factor performance comparison with the base case scenario. This aligns with the main objective of the cyclic injection EOR strategy implementation and can be used as the reference for decision making in real field implementation.

11.1 Sensitivity Analysis

In this study, author carried out number of sensitivity simulations against all variables that can be set during reservoir simulation scenario preparation. All displayed result shows that not all parameters involved in the sensitivity study affect the reservoir performance. In general, the simulation result shows that there are some variables that are very influential, some have little impact, some have effects within certain value range and some have no effect at all.

Since the performed sensitivity is divided into two categories, then the obtained result are also grouped according to the respective category. For the uncertain variable category, hydraulic fracture length and width are impactful to the reservoir performance. While the permeability variable of cement behind casing, secondary fracture properties including length, permeability and intensity, and variable matrix permeability have a pretty significant effect to certain interval value but have no effect at all outside the range. The other parameters which are cement porosity and number of fracture grid block on the other hand, are considerably insensitive. According to sensitivity against lateral heterogeneity, permeability barrier that is parallel to the direction of gas displacement only gives very small reduction in recovery compared to the perpendicular permeability barrier.

For controllable variables category, every change in cyclic interval and inter-fractures distance significantly affected obtained oil recovery, while the variable production rate and injection rate target did not really affect the performance of the well. The sensitivity result in those two variables only showed that there is a minimum rate required to achieve optimum production, but because this minimum value is also very small compared to the typical practice value that is commonly applied in the field, then these two variables are considered insensitive on reservoir performance.

11.2 Optimal Environment Condition

Optimal operating scenario based on performed simulation study is by applying same well EOR strategy supported by cyclic gas injection scenario. The recommended fracture configuration is placing injector and producer fracture alternately. Gas injection was performed in beginning of year 3 and cyclic injection scheme was performed continuously without shut in between cycle. Huff and Puff scheme is not recommended based on obtained simulation result in this model.

Based on sensitivity study conducted, optimal cyclic gas injection condition can be obtained by reducing cycle time, shortening the fractures spacing, and extending as well as increase the intensity of secondary fractures. The best combination based on the optimum value is by performing cyclic gas injection cycle time in 30 days, setting the fracture spacing less than 200 ft, creating significant number of secondary fracture with length more than 10 ft.

The obtained result by considering the optimum value, is expected to be able to incorporate the goal of “make the most out of what we have” in the field to obtain the desirable oil recovery factor, and also to reduce the gas injection volume and cost. Fracture spacing and secondary fracture typically depend on the field properties and tools availability that make it uncertain and challenging to be set as we want. However the success of bringing it to the optimum condition ultimately adds value to the implementation of the cyclic gas injection EOR.

11.3 Uncertainty Analysis (Monte Carlo Simulation)

Based on the result of the sensitivity study analysis, applying optimal scenario in cyclic gas injection resulted in average final oil recovery 21% with average depletion oil recovery by 8%. To get an idea about the result that can be expected because of the influence of uncertain variables involved, the author also run Monte Carlo simulation for all uncertain variables by assuming that every variable has triangular probability distribution and use the ± 30% of typical value as min and max value. Uncertainty simulation was done in two approaches. First one is the simulation to assess the diversity of each uncertain variables but using homogeneous reservoir matrix properties which produce result as presented in **figure 11.1** and **table 11.2** for its final oil recovery, also **figure 11.2** and **table 11.3** for its depletion oil recovery. There were also simulations on permeability diversity which represent lateral reservoir matrix heterogeneity which presented in **figure 11.3** and **table 11.4** for its final oil recovery and **figure 11.4** and **table 11.5** for its depletion oil recovery. Case 1 use 7 different uncertain variables assuming that each variable has triangular probability distribution that can be seen in **table 11.1**. Case 2 use the reservoir model which has 34 different zones consists of group of matrix, which will be set to random permeability values by assuming that the rock permeability probability has triangular distribution with 0.001 mD mode, 0.0005 mD as min and 0.0015 mD as max value.

Table 11.1 Monte Carlo case 1 uncertain variables summary

Variable	Min	Mode	Max
Cement Porosity, fraction	0	0.1	1
Cement Permeability, mD	0	0.01	0.1
Frac Half Length, ft	200	250	300
Matrix Permeability, mD	0.0007	0.001	0.0013
Secondary Fracture Length, ft	0	10	99
Secondary Fracture Permeability Multiplier, fraction	0	0.9	1
Number of Secondary Fracture	0	3	5

As additional note, Monte Carlo simulation involved 100 and 700 iterations for each variable due to simulation time constraints at each iteration. These 2 different iteration number are used to see the effect of adding it to the simulation result, since it will require quite a long time to run the proper number of iterations with so many variables being used. In this chapter, author will focus to discuss the 700 iterations simulation result since it is obviously more reliable than the 100 iterations result. For simplification reason, the resulting probability result is considered valid, provided that the generated error needs to be further consideration.

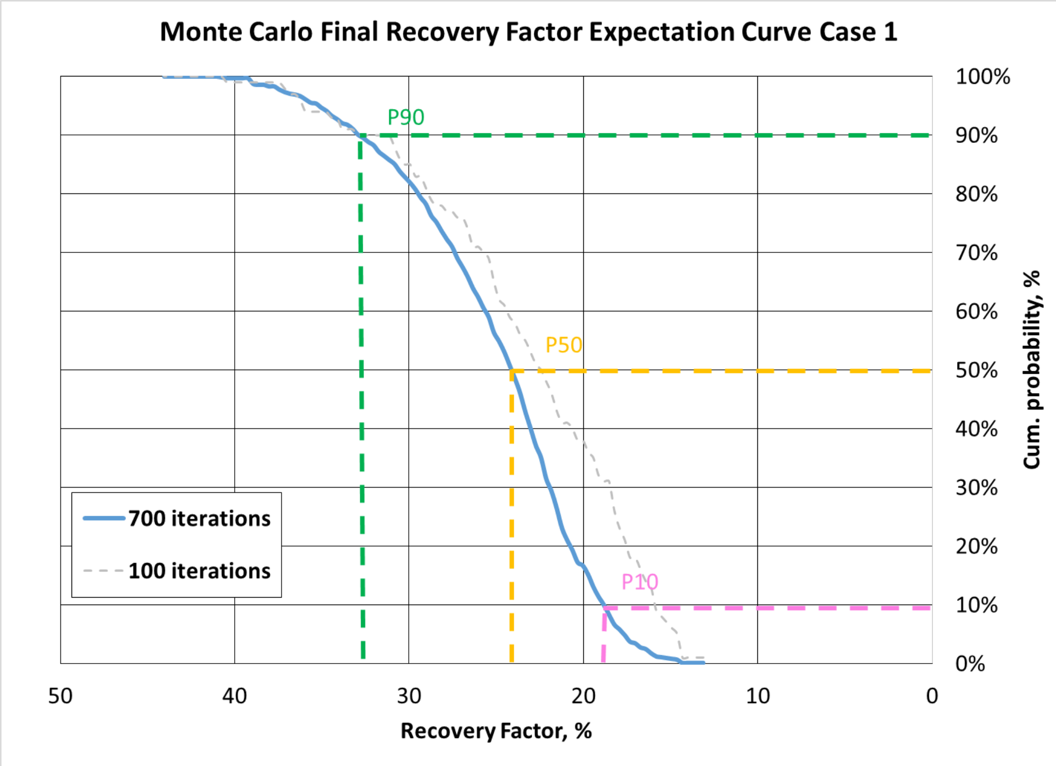

Figure 11.1 Final oil recovery expectation curves generated by 100 & 700 iterations Monte Carlo simulation with 7 uncertain variables (case 1)

Table 11.2 Final oil recovery probability summary resulted from 700 iterations (case 1)

Monte Carlo Simulation Result	
(700 iterations)	RF, %
Expected Value (mean)	25.24
P10	18.83
P50 (median)	24.10
P90	32.92

Based on the simulation result, the first approach shows that expected final oil recovery values that can be expected due to uncertainty in its every variable is 25.24%. These value is considered satisfactory because it is still better than the base case. These result also provides information that the minimum result that can be expected is still good in showing quite good increase in oil recovery since its application after the depletion phase, as seen from difference of 10.8% against 8%, which is the average of its depletion phase's oil recovery.

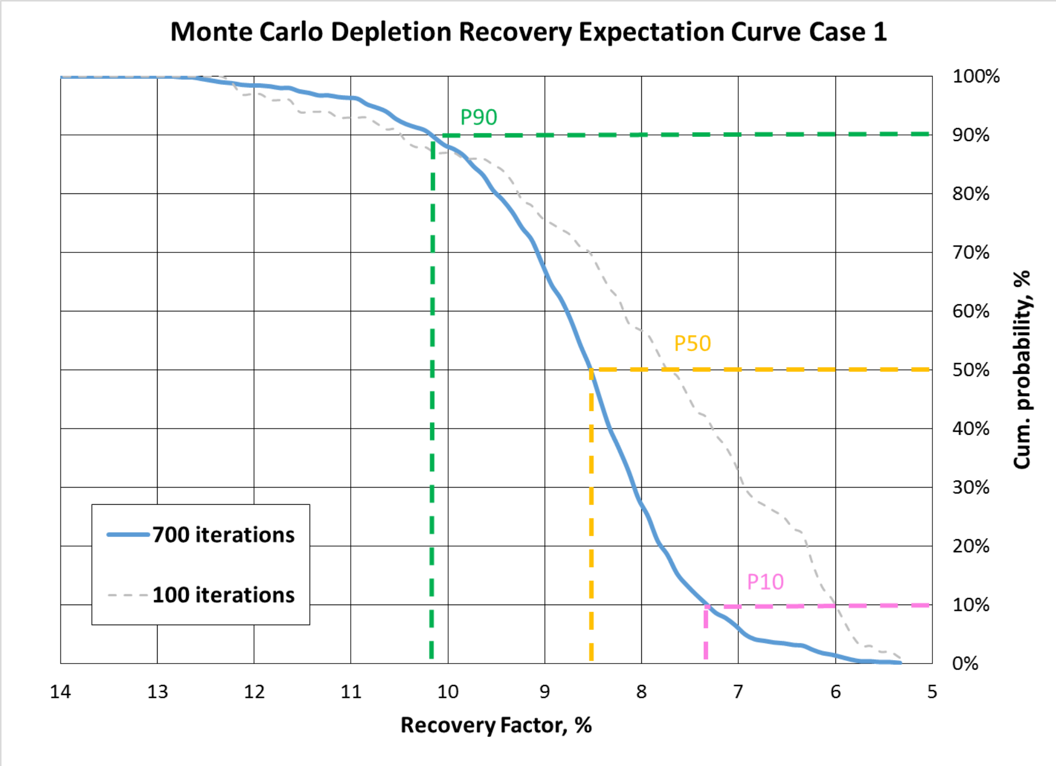

Figure 11.2 Depletion oil recovery expectation curves generated by 100 & 700 iterations Monte Carlo simulation with 7 uncertain variables (case 1)

Table 11.3 Depletion oil recovery probability summary resulted from 700 iterations (case 1)

Monte Carlo Simulation Result	
(700 iterations)	**RF, %**
Expected Value (mean)	8.70
P10	7.32
P50 (median)	8.53
P90	10.17

Based on the first approach's simulation result, it is also shown that the expected depletion oil recovery by considering the uncertain variables matches the average depletion oil recovery obtained in the earlier simulations, at around 8%. This result also confirms that the simulations performed on sensitivity analysis is valid.

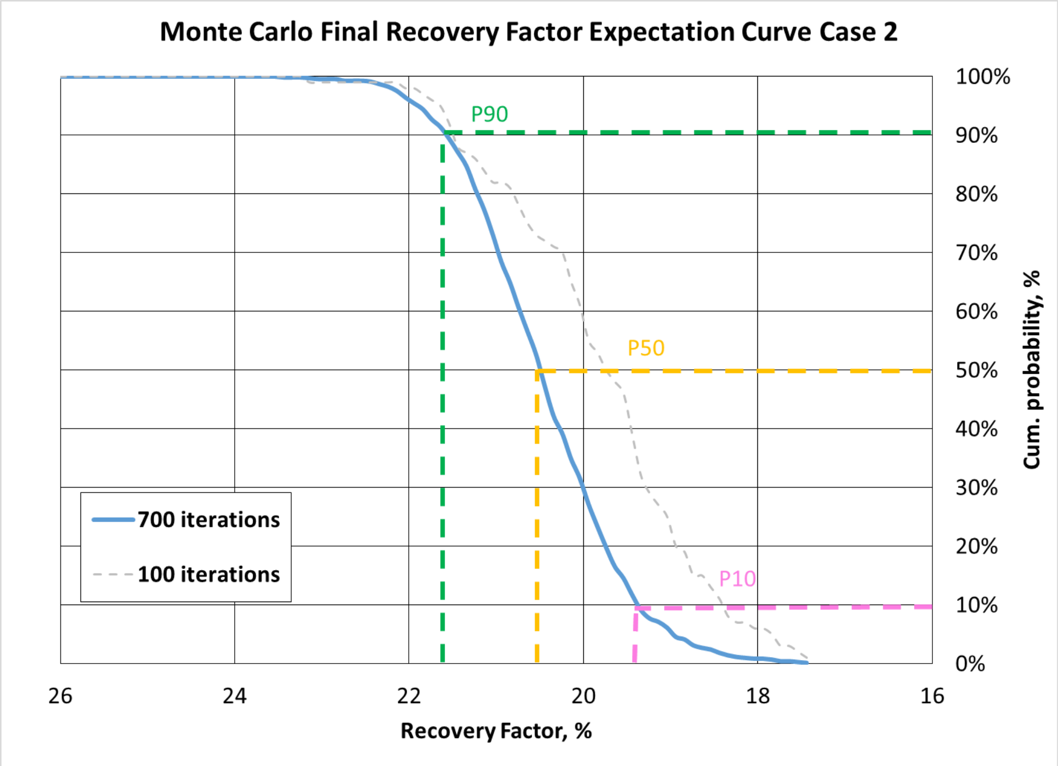

Figure 11.3 Final oil recovery expectation curves generated by 100 & 700 iterations Monte Carlo simulation for lateral matrix permeability uncertainty (case 2)

Table 11.4 Final oil recovery probability summary resulted from 700 iterations (case 2)

Monte Carlo Simulation Result	
(700 iterations)	RF, %
Expected Value (mean)	20.52
P10	19.38
P50 (median)	20.50
P90	21.58

Based on the simulation results on the second approach, it is seen that the expected final oil recovery value due to lateral reservoir permeability heterogeneity is 20.52%. This value decreased from its optimum value because of the permeability barrier that appears on the reservoir matrix also generates considerable impacts. However, from the result, the rate of 20.52% is considered quite satisfactory because based on performed sensitivity analysis, possibly obtained worst case by the effect of the permeability barrier is about 15%. The minimum expected result is also shows considerably very good result compare to average depletion phase oil recovery.

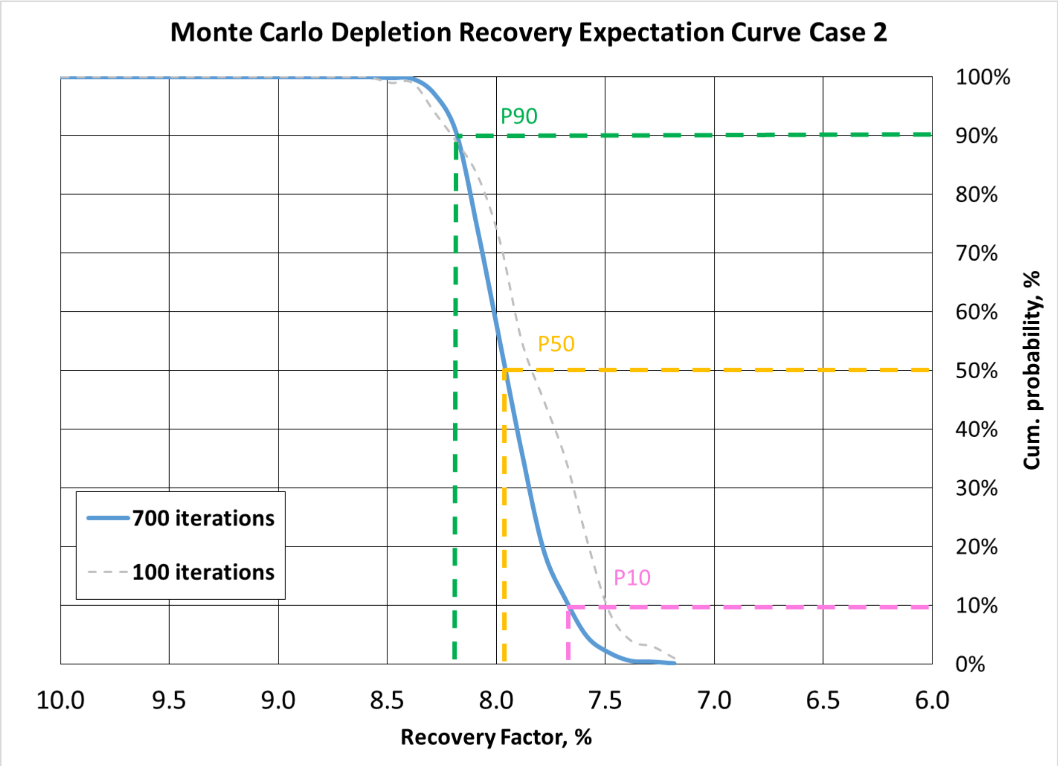

Figure 11.4 Depletion oil recovery expectation curves generated by 100 & 700 iterations Monte Carlo simulation for lateral matrix permeability uncertainty (case 2)

Table 11.5 Depletion oil recovery probability summary resulted from 700 iterations (case 2)

Monte Carlo Simulation Result	
(700 iterations)	**RF, %**
Expected Value (mean)	7.99
P10	7.67
P50 (median)	7.95
P90	8.18

Based on the simulation result of the second approach, it is also obtained that expected value of depletion oil recovery is considerably similar to average value (in the range of 8%), after considering lateral heterogeneity in the model. This result also serves as the confirmation that performed simulation in this study is valid.

From all the charts can be seen that the lower iteration number will result in slightly pessimist oil recovery, reflected from the generated expectation curve represented as dotted light gray line in all the charts. The 100 iterations expectation curve looked shifting to the lower values compare to more iterations and the generated line looks erratic. The 700 iterations expectation curve, on the other hand, generated smoother expectation curve line. This curve's expected P5-value also looks move to the higher oil recovery as the result. Based on this phenomenon, it can be inferred that the probability resulted with 700 iterations is acceptable.

11.4 Discussion

Various sensitivity studies as well as uncertainty analyses had been performed in this study. The general result shows that the cyclic injection approach is very potential to be applied based on the model. However, the variables variation here are still limited to typical value and several simplifications used to assess uncertainty. It will be more interesting to bring the advance study to see the potential impact by utilizing real field data. Important things from a real field like heterogeneity in vertical and horizontal direction, also surface facility constraints are expected to provide significant impact to the cyclic gas injection performance. The result may possibly change the optimum case strategy that is considered the most appropriate accordance with needs and characteristics of a field.

As the additional consideration to reassure the result, another study approach should also be considered to be performed as comparison. Possible approach is by using another simulator software that has additional features that SENSOR simulator doesn't have. It is also possible to do laboratory experiment against each simulation result by utilizing similar process and variables. Thus, this process is expected to capture aspects that were previously missing to be further considered, and to see its realization in laboratory before being implemented in real field scale.

11.5 Summary and Conclusions

The main idea of this study is to simulate the performance of shale oil reservoir using the same-well cyclic gas injection strategy at hydraulically fractured horizontal well. This approach was chosen because it is an approach that provides optimum result with the lowest cost. In this study, the simulation was performed using SENSOR reservoir simulator software.

Reservoir model validation study has indicated that any value of matrix Δ_x, solver that was utilized and the way producer perforation in fracture grid assigned do not affect the simulation result. Thus the simplified model (matrix Δ_x = 1 ft, solver = RBILU (0), and producer perforation grid only at first grid (i=1, j=1, k=1)) were used in whole simulation study.

The first phase of this study is designing and developing Pipe-It project template that would be used during the study process. Project template is intended to provide various simplifications in doing a lot of simulations and integrating SENSOR simulator with GAWK programming tool. The produced Pipe-It project templates could also be used by another user for further study since its interface has been created in such a way that is able to incorporate almost all most important functions and very user friendly.

Various simulations have been carried out to see which optimum scenario is suitably applied to the reservoir that is classified as an ultra-tight and in poor quality. The most recommended strategy is to apply the cyclic gas injection scenario that began after two years of natural flow, assisted by utilizing the same-well EOR configuration with alternating injector and producer fracture. Same-well cyclic gas injection strategy is the process of injecting gas and oil as well as producing oil through the same well in turns for certain cycle time. Based on the simulation results, the implementation of this strategy in shale oil reservoir greatly helped to increase the oil recovery. The result obtained by applying the most optimum operating condition could produce almost equal performance with continuous gas injection scenario which has much expensive cost. The oil recovery loss for 365 days injection cycle time is only 3% compare to continuous gas injection case, while the gas injection reduction is 26%. It will be even better if the cycle time is reduced to 30 days. This strategy is proven effective in pursuing the objective of maximizing existing resources while still obtaining the maximum profit, since the result shows that the oil recovery loss was very insignificant while gas injection efficiency was much better. Some alternative strategies were also tested in this study, but did not give better results.

Study was also conducted to assess the generated effects after modifying uncertain and controllable variables in several aspects related to well and reservoir that were made either intentionally or unintentionally to reservoir performance. Results of the study showed that not all variables provide significant effect. There are several variables that affect the performance, but there are also variables with little or even no effect to the performance. The variables that gave such effects are mainly them who are related to influence well conductivity and injection volume. They are inter-fracture distance, hydraulic fracture length, secondary fractures and injection cycle time. Study also shows that there is unique value interval for some variables such as cement, matrix and secondary fracture permeability, at which the effect is detected in the simulation.

12 References

1. Canadian Society for Unconventional Resources. Understanding Tight Oil. Available at: http://www.csur.com/sites/default/files/Understanding_TightOil_FINAL.pdf
2. Tran, T., Sinurat, P. D., & Wattenbarger, B. A. (2011, January 1). Production Characteristics of the Bakken Shale Oil. Society of Petroleum Engineers. doi:10.2118/145684-MS
3. Updates to the EIA Eagle Ford Play Maps. Available at: https://www.eia.gov/maps/pdf/eagleford122914.pdf
4. Lake, L.W., Johns, R.T., Rossen, B., Pope, G., 2014. Fundamentals of Enhanced Oil Recovery. Society of Petroleum Engineers, Richardson, Texas, USA.
5. Sheng, J. J., Cook, T., Barnes, W., Mody, F., Watson, M., Porter, M., & Viswanathan, H. (2015, November 13). Screening of the EOR Potential of a Wolfcamp Shale Oil Reservoir. American Rock Mechanics Association.
6. Nguyen, N. T. B., Dang, C. T. Q., Chen, Z., & Nghiem, L. X. (2015, June 1). Optimization of Hydraulic Fracturing Design with Future EOR Considerations in Shale Oil Reservoirs. Society of Petroleum Engineers. doi:10.2118/174307-MS
7. Wan, T., Sheng, J. J., & Soliman, M. Y. (2013, August 12). Evaluate EOR Potential in Fractured Shale Oil Reservoirs by Cyclic Gas Injection. Society of Petroleum Engineers. doi:10.1190/URTEC2013-187
8. Gamadi, T. D., Elldakli, F., & Sheng, J. J. (2014, August 28). Compositional Simulation Evaluation of EOR Potential in Shale Oil Reservoirs by Cyclic Natural Gas Injection. Society of Petroleum Engineers. doi:10.15530/urtec-2014-1922690
9. Coats Engineering, Inc. *Sensor Manual*, 2011
10. Whitson, C. H., & Sunjerga, S. (2012, January 1). PVT in Liquid-Rich Shale Reservoirs. Society of Petroleum Engineers. doi:10.2118/155499-MS

13 Appendices

Appendix A

Base Case Sensor (Master) Data File

```
TITLE
    Same-Well Cyclic EOR
    1/4 of injector & producer fractures
    Frac-to-frac distance: 200 ft
    Frac half-length: 250 ft
    Well to boundary: 500 ft
    Fracture-like (optional) representing cement
    Pseudoized compositional simulation with EOS9 (Bakken)
    Started: Sept. 17, 2015
    With Reizky
ENDTITLE

KEYWORD

GRID  200 10 1    ! NX NY NZ
CPU
IMPLICIT

MAPSFILE  P  SAT  SG SO KX
MAPSPRINT 1  P PSAT SG SO GG GO KX

C      Bwi    cw   denw   visw  cr   pref
MISC    1    3E-6  63.1    .6   4E-6 3500

DELX  ! VALUE
0.0833 198*1 0.0833
DELY  ! VALUE
0.0833 5*50 4*62.5

POROS CON
0.05
MOD
```

```
  1   1   1   5   1   1 = 1.00 ! injection fracture
200 200   1   5   1   1 = 1.00 ! production fracture
  2 199   1   1   1   1 = 0.00 ! cement

C Rocktypes: 1 = matrix | 2 = fractures | 3 = cement
ROCKTYPE CON
1
MOD
  1   1   1   5   1   1 = 2 ! injection fracture
200 200   1   5   1   1 = 2 ! production fracture
  2 199   1   1   1   1 = 3 ! cement

KX CON
1000E-6
MOD
  1   1   1   5   1   1 = 10000  ! injection fracture
200 200   1   5   1   1 = 10000  ! production fracture
  2 199   1   1   1   1 = 0      ! cement

KY EQUALS KX
KZ EQUALS KX

DEPTH CON
10000
THICKNESS CON
200
C ---------------------------------------------------------------------------
C Relative permeability curves
C ---------------------------------------------------------------------------
KRANALYTICAL 1
 0.2 0.2 0.38 0.05 ! Swc Sorw Sorg Sgc
 1 1 1             ! krw(Sorw) krg(Swc) kro(Swc)
 3 3 2 3           ! nw now ng nog
  KRANALYTICAL 2   ! For fractures
 0.0 0.0 0.0 0.0  ! Swc Sorw Sorg Sgc
 1 1 1            ! krw(Sorw) krg(Swc) kro(Swc)
```

```
 1 1 1 1          ! nw now ng nog
  KRANALYTICAL 3   ! For cement
 0.0 0.0 0.0 0.0  ! Swc Sorw Sorg Sgc
 1 1 1            ! krw(Sorw) krg(Swc) kro(Swc)
 2 2 2 2          ! nw now ng nog
C ---------------------------------------------------------------------------
C Initialization regions
C ---------------------------------------------------------------------------
PRINTKR 3

C =================================================================
C Fluid Properties
C ===================================================================
PVTEOS SRK
240    ! Reservoir temperature (deg F)
```

CPT	MW	TC	PC	ZCRIT	SHIFT	AC	PCHOR
H2S	34.082	672.12	1300	0.28292	0.10153	0.09	80.1
CO2	44.01	547.42	1069.5	0.27433	0.21749	0.225	80
N2C1	17.135	331.17	650.14	0.28666	-0.00234	0.01266	69.91
C2C3	36.69	606.72	659.03	0.27761	0.07656	0.12462	129.88
C4C5	63.881	793.27	520.66	0.27282	0.11088	0.21674	206.63
C6C9	104.93	1012.5	438.12	0.30723	0.105	0.31764	286.83
C10C17	177.2	1216.1	303.63	0.29442	0.14617	0.62464	460.28
C18C29	308.78	1428.7	199.32	0.29041	0.1885	1.1802	776.07
C30+	548.58	1634.8	136.8	0.32159	0.16213	2.2561	1351.6

BIN

0.12	0.08125	0.07	0.06	0.03326	0.03	0.03	0.03
0.11246	0.15	0.15	0.10814	0.1	0.1	0.1	
0.00288	0.00337	0.00337	0.00337	0.00337	0.00337		
0	0	0	0	0			
0	0	0	0				
0	0	0					
0	0						
0							

```
C   Lohrenz, Bray, Clark Viscosity  Coefficients
CVIS0  0.1023   0.023364   0.058533   -0.0333815   0.00599457
SEP  1
100   150
14.7  60

INITIAL 1
DEPTH
10000 2364.02   ! depth  psat
0.00000000
0.00305556
0.31403421
0.17744379
0.10705390
0.17600376
0.14159629
0.05844783
0.02236467

PINIT   6500
ZINIT   10000
ENDINIT
C     END OF INITIAL DATA

C     START RECURRENT DATA
WELL
   I    J1 J2  K    PI
  PROD
   1   1    1    1   10
  GINJ
 200  1    5   1   10

WELLTYPE
  PROD   STBOIL
  GINJ   STBOIL        ! Initially produce from what becomes an injector
BHP
```

```
  PROD  500
  GINJ  500
RATE
  PROD  1.E10
  GINJ  1.E10
MAPSFREQ 1

DT 0.01
TIME 730 365
WELLTYPE
  GINJ   MCFINJ
BHP
  GINJ  7000
RATE
  GINJ  1000
TIME 3650 365 1

END
```

Appendix B

Pipe-It Project Template Sensor Main Data File

```
TITLE
    Same-Well Cyclic EOR
    1/4 of injector & producer fractures
    Frac-to-frac distance: 200 ft
    Frac half-length: 250 ft
    Well to boundary: 500 ft
    Fracture-like (optional) representing cement
    Pseudoized compositional simulation with EOS9 (Bakken)
    Started: Sept. 17, 2015
    With Reizky
ENDTITLE

KEYWORD

INCLUDE
grid-relperm.inc

C ---------------------------------------------------------------------------
---------------
C Relative permeability curves
C ---------------------------------------------------------------------------
---------------

KRANALYTICAL 1
 0.2 0.2 0.38 0.05 ! Swc Sorw Sorg Sgc
 1 1 1             ! krw(Sorw) krg(Swc) kro(Swc)
 3 3 2 3           ! nw now ng nog

 KRANALYTICAL 2   ! For fractures
 0.0 0.0 0.0 0.0  ! Swc Sorw Sorg Sgc
 1 1 1            ! krw(Sorw) krg(Swc) kro(Swc)
 1 1 1 1          ! nw now ng nog
```

```
 KRANALYTICAL 3    ! For cement
 0.0 0.0 0.0 0.0  ! Swc Sorw Sorg Sgc
 1 1 1            ! krw(Sorw) krg(Swc) kro(Swc)
 2 2 2 2          ! nw now ng nog

C ---------------------------------------------------------------------------
---------------
C Initialization regions
C ---------------------------------------------------------------------------
---------------

PRINTKR 3

C ========================================================================
C Fluid Properties
C ========================================================================
PVTEOS SRK
240    ! Reservoir temperature (deg F)
```

CPT	MW	TC	PC	ZCRIT	SHIFT	AC	PCHOR
H2S	34.082	672.12	1300	0.28292	0.10153	0.09	80.1
CO2	44.01	547.42	1069.5	0.27433	0.21749	0.225	80
N2C1	17.135	331.17	650.14	0.28666	-0.00234	0.01266	69.91
C2C3	36.69	606.72	659.03	0.27761	0.07656	0.12462	129.88
C4C5	63.881	793.27	520.66	0.27282	0.11088	0.21674	206.63
C6C9	104.93	1012.5	438.12	0.30723	0.105	0.31764	286.83
C10C17	177.2	1216.1	303.63	0.29442	0.14617	0.62464	460.28
C18C29	308.78	1428.7	199.32	0.29041	0.1885	1.1802	776.07
C30+	548.58	1634.8	136.8	0.32159	0.16213	2.2561	1351.6

BIN

0.12	0.08125	0.07	0.06	0.03326	0.03	0.03	0.03
0.11246	0.15	0.15	0.10814	0.1	0.1	0.1	
0.00288	0.00337	0.00337	0.00337	0.00337	0.00337		
0	0	0	0	0			
0	0	0	0				

```
0	0	0
0	0
0

C   Lohrenz, Bray, Clark Viscosity  Coefficients
CVIS0  0.1023   0.023364   0.058533   -0.0333815   0.00599457

SEP  1
100   150
14.7  60

INITIAL 1

DEPTH

10000 2364.02   ! depth  psat
0.00000000
0.00305556
0.31403421
0.17744379
0.10705390
0.17600376
0.14159629
0.05844783
0.02236467

PINIT   6500
ZINIT   10000

ENDINIT

C     END OF INITIAL DATA

INCLUDE
recurrent.inc
END
```

Appendix C

GAWK1 Grid & Rel.Perm Script

```
BEGIN {
#      awk: each program line consists of pattern {action}
#      each input file line is processed through the whole program
       MP=0.05             # Matrix porosity input
       CP=0                # Cement porosity input
       MK=0.001            # Matrix permeability input
       CK=0                # Cement permeability input
       XL=200              # Frac to frac distance input
       YL=500              # Reservoir width (y-dir) input
       FL=250              # Fracture length input
       FLG=6               # No. of fracture grid block input
       NX=200              # No. of X-dir. grid block input
       NY=10               # No. of Y-dir. grid block input
       NS=1                # Number of secondary fracture input
       SFL=10              # Secondary fracture length input
       FM=0.1              # Secondary fracture permeability multiplier input
       counter = 0;
}
       x=1
       SF=0.1
       AA=0.0833
       s0=1

{          NLC = (NX - 1)
           NXM = (NX - 2)
           DELX = (XL - 2) / NXM
           DELYF = FL / (FLG - 1)
           FLGM = (FLG - 1)
           MLG = NY - FLG
           DELYM = (YL - FL)/(NY - FLG + 0.00000000000000001)
           FR = FLG - 1 - (NS *2)
           NSFM = FLG - 1
```

```
        NYN = NY + NS
        FLGN = FLG + NS
        KS = FM * 10000

        if (FL > YL)
        {
        print "ERROR : Fracture length is too high."
        exit}

        if (FLG > NY)
        {
        print "ERROR : Number of fracture is too high."
        exit}

        if (NS > NSFM)
        {
        print "ERROR : Number of secondary fracture is too high."
        exit}

        ## print into outfile
        printf ("GRID  %s %s 1    ! NX NY NZ", NX, NYN)
        print ""
        print "CPU"
        print ""
        print "IMPLICIT"
        print ""
        print "MAPSFILE  P  SAT  SG SO KX"
        print "MAPSPRINT 1  P PSAT SG SO GG GO KX"
        print ""
        print "C      Bwi    cw   denw   visw  cr   pref"
        print "MISC    1    3E-6  63.1    .6   4E-6 3500"
        print ""
        print "DELX  ! DELX, ft"
        printf ("0.0833 %s*%s 0.0833\n", NXM, DELX)
        printf("\n");
}
```

```
{
if (NS == 0)
{
          {
          if (MLG > 0)
          {
          print "DELY  ! DELY, ft"
          printf ("0.0833 %s*%s %s*%s\n", FLGM, DELYF, MLG, DELYM )
          }
          else
          {
          print "DELY  ! DELY, ft"
          printf ("0.0833 %s*%s\n", FLGM, DELYF )
          }}
}
else
{
          {
          if (MLG > 0)
          {
          print "DELY  ! DELY, ft"
          printf ("%s ", AA)
          while(1)
          {
          DYN = DELYF - 0.1
          printf ("%s %s ", DYN, SF);
          if ( x==NS )
          break;
          x++;
          }     {
          DE = NSFM - NS
          if (DE == 0)
          {
          printf ("%s*%s\n", MLG, DELYM);
          }
          else
```

```
          {
          printf ("%s*%s %s*%s\n", DE, DELYF, MLG, DELYM);
          }}}
            else
            {
          print "DELY  ! DELY, ft"
          printf ("%s ", AA)
          while(1)
          {
          DYN = DELYF - 0.1
          printf ("%s %s ", DYN, SF);
          if ( x==NS )
          break;
          x++;
          }{
          DE = NSFM - NS
          if (DE == 0)
          {
          printf ("\n");
          }
          else
          {
          printf ("%s*%s\n", DE, DELYF);
          }}}}}}
{         printf("\n");
          print "POROS CON"
          printf ("%s \n", MP)
          print "MOD"
          printf ("  1    1  1   %s   1   1 = 1.00 ! production
fracture\n", FLGN)
          printf ("%2.0f  %2.0f  1   %s   1   1 = 1.00 ! injection
fracture\n", NX, NX, FLGN)
          printf ("  2  %s  1   1   1   1 = %s    ! cement porosity\n",
NLC, CP)
}{
if (NS == 0)
{         print ""
```

```
}
else
{         DF = SFL / DELX
            x2 = 1 + DF
            x3 = NX - DF
            x4 = NX - 1
            s1 = s0 + 2
            x = 1
            while(1)
            {
            printf ("  2   %2.0f  %s   %s   1   1 = 1.00 ! production
fracture\n", x2, s1, s1)
            printf ("%2.0f  %s  %s   %s   1   1 = 1.00 ! injection
fracture\n", x3, x4, s1, s1)
            s1 = s1 + 2;
            if ( x==NS )
            break;
            x++;
            }
              print ""
}}
{         print "C Rocktypes: 1 = matrix | 2 = fractures | 3 = cement"
          print "ROCKTYPE CON"
          print "1"
          print "MOD"
          printf ("  1    1  1   %s   1   1 = 2 ! production fracture\n",
FLGN)
          printf ("%2.0f  %2.0f  1   %s   1   1 = 2 ! injection
fracture\n", NX, NX, FLGN)
          printf ("  2  %s  1   1   1   1 = 3 ! cement\n", NLC)
}{
if (NS == 0)
{         print ""
}
else
{         DF = SFL / DELX
            x2 = 1 + DF
            x3 = NX - DF
```

```
            x4 = NX - 1
            s1 = s0 + 2
            x = 1
            while(1)
            {
            printf ("  2   %2.0f  %s   %s   1   1 = 2 ! production
fracture\n", x2, s1, s1)
            printf ("%2.0f  %s  %s   %s   1   1 = 2 ! injection
fracture\n", x3, x4, s1, s1)
            s1 = s1 + 2;
            if ( x==NS )
            break;
            x++;
            }
              print ""
}}
{          print "KX CON"
          printf ("%s \n", MK)
          print "MOD"
          printf ("  1    1  1   %s   1   1 = 10000 ! production
fracture\n", FLGN)
          printf ("%2.0f  %2.0f  1   %s   1   1 = 10000 ! injection
fracture\n", NX, NX, FLGN)
          printf ("  2  %s  1   1   1   1 = %s     ! cement porosity\n",
NLC, CK)
}{
if (NS == 0)
{          print ""
}
else
{          DF = SFL / DELX
            x2 = 1 + DF
            x3 = NX - DF
            x4 = NX - 1
            s1 = s0 + 2
            x = 1
            while(1)
            {
```

```
            printf ("  2   %2.0f  %s   %s   1   1 = %s ! production
fracture\n", x2, s1, s1, KS)
            printf ("%2.0f  %s  %s   %s   1   1 = %s ! injection
fracture\n", x3, x4, s1, s1, KS)
            s1 = s1 + 2;
            if ( x==NS )
            break;
            x++;
            }
              print ""
}}
{          print "KY EQUALS KX"
           print ""
           print "KZ EQUALS KX"
           print ""
           print "DEPTH CON"
           print "10000"
           print ""
           print "THICKNESS CON"
           print "200"
           print ""
           print "C -------------------------------------------------------"
           print "C Relative permeability curves"
           print "C -------------------------------------------------------"
           print ""
           print "KRANALYTICAL 1"
           print " 0.2 0.2 0.38 0.05 ! Swc Sorw Sorg Sgc"
           print " 1 1 1             ! krw(Sorw) krg(Swc) kro(Swc)"
           print " 3 3 2 3           ! nw now ng nog"
           print ""
           print " KRANALYTICAL 2   ! For fractures"
           print " 0.0 0.0 0.0 0.0  ! Swc Sorw Sorg Sgc"
           print " 1 1 1            ! krw(Sorw) krg(Swc) kro(Swc)"
           print " 1 1 1 1          ! nw now ng nog"
           print ""
           print " KRANALYTICAL 3   ! For cement"
           print " 0.0 0.0 0.0 0.0  ! Swc Sorw Sorg Sgc"
```

```
        print " 1 1 1            ! krw(Sorw) krg(Swc) kro(Swc)"
        print " 2 2 2 2          ! nw now ng nog"
        print ""
                }{
              print "C -----------------------------------------------"
              print "C Initialization regions"
              print "C -----------------------------------------------"
              printf("\n");
              print "PRINTKR 3"
              counter = 1;
              }
END {}
```

GAWK2 Schedule Script

```
BEGIN {
#     awk: each program line consists of pattern {action}
#     each input file line is processed through the whole program
      FLG=6               # No. of fracture grid block input
      PP0=500             # Producer BHP constraint before cyclic input

      GP0=500             # Injector BHP constraint before cyclic input
      PO0=10000000000     # Production target rate before cyclic input
      x1=730              # Cylic operation start input
      xf=3650             # End of simulation input
      dx=365              # Cylic duration input
      P=7000              # Injector BHP constraint while cyclic input
      RI1=1000            # Injector inj. rate target while cyclic input
      PO2=1000            # Producer prod. rate target while cyclic input
      DT=0.1              # Simulation delta time input
      NX=200              # No. of X-dir. grid block input
      NS=1                # Number of secondary fracture input
      SFL=10              # Secondary fracture length input
      XL=200              # Frac to frac distance input
      EOR=1               # EOR type input
      counter = 0;
}
      RI2=0
      PO1=0
      s0=1
{
      NXM = (NX - 2)
      DELX = (XL - 2) / NXM
      FLGN = FLG + NS
      NSFM = (FLG - 1) / 2
      if (NS > NSFM)
      {
      print "ERROR : Number of secondary fracture is too high."
      exit}}
```

```
{ if (EOR == 1)
{ if (NS == 0)
{ if (x1 == 0)
{
          ## print into outfile
          print "C      START RECURRENT DATA"
          print ""
          print "WELL"
          print "I J1 J2 K  PI"
          print "PROD"
          print " 1 1  1  1 10"
          print "GINJ"
          printf ("%s 1  %s  1 10\n", NX, FLGN)
          print ""
          print "WELLTYPE"
          print "  PROD   STBOIL"
          print ""
          print "BHP"
          printf ("  PROD   %s \n", PP0)
}
else
{
          ## print into outfile
          print "C      START RECURRENT DATA"
          print ""
          print "WELL"
          print "I J1 J2 K  PI"
          print "PROD"
          print " 1 1  1  1 10"
          print "GINJ"
          printf ("%s 1  %s  1 10\n", NX, FLGN)
          print ""
          print "WELLTYPE"
          print "  PROD   STBOIL"
          print "  GINJ   STBOIL"
          print ""
```

```
        print "BHP"
        printf ("  PROD   %s \n", PP0)
        printf ("  GINJ   %s \n", GP0)
        print ""
        print "RATE"
        printf ("  PROD   %s \n", PO0)
        printf ("  GINJ   %s \n", PO0)
        print ""
        print "MAPSFREQ 1"
        print ""
        printf ("DT  %s \n", DT)
        printf ("TIME  %s 365", x1)
        printf("\n");
}}
else
{ if (x1 == 0)
{
        ## print into outfile
        print "C     START RECURRENT DATA"
        print ""
        print "WELL"
        print " I1   I2  J1  J2  K    PI"
        print "PROD"
        print " 1     1   1   1   1   10"
        print "GINJ"
        printf ("%s  %s   1   %s   1   10\n", NX, NX, FLGN)

        DF = SFL / DELX
        x3 = NX - DF
        x4 = NX - 1
        s1 = s0 + 2
        x = 1
        while(1)
        {
        printf ("%2.0f  %2.0f  %2.0f  %2.0f   1   10\n", x3, x4, s1, s1)
        s1 = s1 + 2;
```

```
            if ( x==NS )
            break;
            x++;
            }
            print ""
            print "WELLTYPE"
            print "  PROD   STBOIL"
            print ""
            print "BHP"
            printf ("  PROD   %s \n", PP0)
}
else
{
            ## print into outfile
            print "C     START RECURRENT DATA"
            print ""
            print "WELL"
            print " I1   I2  J1  J2  K    PI"
            print "PROD"
            print " 1     1   1   1   1   10"
            print "GINJ"
            printf ("%s  %s   1   %s   1   10\n", NX, NX, FLGN)

            DF = SFL / DELX
            x3 = NX - DF
            x4 = NX - 1
            s1 = s0 + 2
            x = 1
            while(1)
            {
            printf ("%2.0f  %2.0f  %2.0f  %2.0f   1   10\n", x3, x4, s1, s1)
            s1 = s1 + 2;
            if ( x==NS )
            break;
            x++;
            }
```

```
          print ""
          print "WELLTYPE"
          print "  PROD   STBOIL"
          print "  GINJ   STBOIL"
          print ""
          print "BHP"
          printf ("  PROD   %s \n", PP0)
          printf ("  GINJ   %s \n", GP0)
          print ""
          print "RATE"
          printf ("  PROD   %s \n", PO0)
          printf ("  GINJ   %s \n", PO0)
          print ""
          print "MAPSFREQ 1"
          print ""
          printf ("DT  %s \n", DT)
          printf ("TIME  %s 365", x1)
          printf("\n");
}}}
else { if (NS == 0)
{ if (x1 == 0)
{
          ## print into outfile
          print "C     START RECURRENT DATA"
          print ""
          print "WELL"
          print "I J1 J2 K  PI"
          print "PROD"
          print " 1 1  1  1 10"
          print "PINJ"
          printf (" 1 2  %s  1 10\n", FLGN)
          print "GINJ"
          printf ("%s 1  %s  1 10\n", NX, FLGN)
          print ""
          print "WELLTYPE"
          print "  PROD   STBOIL"
```

```
          print "  PINJ   STBOIL"
          print ""
          print "BHP"
          printf ("  PROD   %s \n", PP0)
          printf ("  PINJ   %s \n", GP0)
}
else
{
          ## print into outfile
          print "C     START RECURRENT DATA"
          print ""
          print "WELL"
          print "I J1 J2 K  PI"
          print "PROD"
          print " 1 1  1  1 10"
          print "PINJ"
          printf (" 1 2  %s  1 10\n", FLGN)
          print "GINJ"
          printf ("%s 1  %s  1 10\n", NX, FLGN)
          print ""
          print "WELLTYPE"
          print "  PROD   STBOIL"
          print "  PINJ   STBOIL"
          print "  GINJ   STBOIL"
          print ""
          print "BHP"
          printf ("  PROD   %s \n", PP0)
          printf ("  PINJ   %s \n", GP0)
          printf ("  GINJ   %s \n", GP0)
          print ""
          print "RATE"
          printf ("  PROD   %s \n", PO0)
          print "  PINJ   0 "
          printf ("  GINJ   %s \n", PO0)
          print ""
          print "MAPSFREQ 1"
```

```
          print ""
          printf ("DT  %s \n", DT)
          printf ("TIME  %s 365", x1)
          printf("\n");
}}
else
{ if (x1 == 0)
{
          ## print into outfile
          print "C     START RECURRENT DATA"
          print ""
          print "WELL"
          print " I1   I2  J1  J2  K    PI"
          print "PROD"
          print " 1     1   1   1   1   10"
          print "PINJ"
          printf (" 1     1   2   %s   1   10\n",FLGN)

          DF = SFL / DELX
                xtwo = 1 + DF
                xone = 2
                s1 = s0 + 2
                x = 1
                while(1)
                {
                printf ("%2.0f    %2.0f  %2.0f  %2.0f   1   10\n", xone,
xtwo, s1, s1)
                s1 = s1 + 2;
                if ( x==NS )
                break;
                x++;
          }
          print "GINJ"
          printf ("%s  %s   1   %s   1   10\n", NX, NX, FLGN)

          DF = SFL / DELX
          x3 = NX - DF
```

```
          x4 = NX - 1
          s1 = s0 + 2
          x = 1
          while(1)
          {
          printf ("%2.0f  %2.0f  %2.0f  %2.0f   1   10\n", x3, x4, s1, s1)
          s1 = s1 + 2;
          if ( x==NS )
          break;
          x++;
          }
          print ""
          print "WELLTYPE"
          print "  PROD   STBOIL"
          print "  PINJ   STBOIL"
          print ""
          print "BHP"
          printf ("  PROD   %s \n", PP0)
          printf ("  PINJ   %s \n", GP0)
}
else
{
          ## print into outfile
          print "C     START RECURRENT DATA"
          print ""
          print "WELL"
          print " I1   I2  J1  J2  K    PI"
          print "PROD"
          print " 1     1   1   1   1   10"

          print "PINJ"
                printf (" 1     1   2   %s   1   10\n",FLGN)

                DF = SFL / DELX
                      xtwo = 1 + DF
                      xone = 2
```

```
            s1 = s0 + 2
            x = 1
            while(1)
            {
              printf ("%2.0f    %2.0f  %2.0f  %2.0f   1   10\n",
              xone, xtwo, s1, s1)
            s1 = s1 + 2;
            if ( x==NS )
            break;
            x++;
}
print "GINJ"
printf ("%s  %s   1   %s   1   10\n", NX, NX, FLGN)

DF = SFL / DELX
x3 = NX - DF
x4 = NX - 1
s1 = s0 + 2
x = 1
while(1)
{
printf ("%2.0f  %2.0f  %2.0f  %2.0f   1   10\n", x3, x4, s1, s1)
s1 = s1 + 2;
if ( x==NS )
break;
x++;        }
print ""
print "WELLTYPE"
print "  PROD   STBOIL"
print "  PINJ   STBOIL"
print "  GINJ   STBOIL"
print ""
print "BHP"
printf ("  PROD   %s \n", PP0)
printf ("  PINJ   %s \n", GP0)
printf ("  GINJ   %s \n", GP0)
print ""
```

```
        print "RATE"
        printf ("  PROD   %s \n", PO0)
        print "  PINJ   0"
        printf ("  GINJ   %s \n", PO0)
        print ""
        print "MAPSFREQ 1"
        print ""
        printf ("DT  %s \n", DT)
        printf ("TIME  %s 365", x1)
        printf("\n");
}}}}
{ if (EOR == 1)
{
      x2_half = x1 + dx
      for (i=x2_half; x2_half <= xf;  i++)
            {
            x2 = x1 + 2*dx
            x2_half = x1 + dx
            if (x2 < xf && x2_half < xf)
                  {
                  printf("\n");
                  print "WELLTYPE"
                  print "GINJ   MCFINJ"
                  print "BHP"
                  printf ("GINJ  %s \n", P)
                  print "RATE"
                  printf ("GINJ  %s \n", RI1)
                  print "RATE"
                  print "  PROD  0"
                  printf ("DT  %s \n", DT)
                  printf ("TIME  %s \n", x2_half)
                  print "RATE"
                  print "  GINJ  0"
                  print "RATE"
                  printf ("PROD  %s \n", PO2)
                  printf ("DT  %s \n", DT)
```

```
   printf ("TIME  %s", x2)
   printf("\n");
   x1 = x2
   }
else if (x2 == xf && x2_half < xf)
   {
   printf("\n");
   print "WELLTYPE"
   print "GINJ   MCFINJ"
   print "BHP"
   printf ("GINJ  %s \n", P)
   print "RATE"
   printf ("GINJ  %s \n", RI1)
   print "RATE"
   print "  PROD  0"
   printf ("DT  %s \n", DT)
   printf ("TIME  %s", x2_half)
   printf("\n");
   x1 = x2
   }
else if (x2 > xf && x2_half < xf)
   {
   printf("\n");
   print "WELLTYPE"
   print "GINJ   MCFINJ"
   print "BHP"
   printf ("GINJ  %s \n", P)
   print "RATE"
   printf ("GINJ  %s \n", RI1)
   print "RATE"
   print "  PROD  0"
   printf ("DT  %s \n", DT)
   printf ("TIME  %s", x2_half)
   printf("\n");
   x1 = x2
   }
```

```
            else if (x2 > xf && x2_half > xf)
               {
               printf("\n");
               print "RATE"
               print "  GINJ  0"
               print "RATE"
               printf ("PROD  %s \n", PO2)
               printf ("DT  %s \n", DT)
               printf ("TIME  %s 365 1", xf)
               x1 = x2
               }}}
else
{
      x2_half = x1 + dx
      for (i=x2_half; x2_half <= xf;  i++)
            {
            x2 = x1 + 2*dx
            x2_half = x1 + dx
            if (x2 < xf && x2_half < xf)
                  {
                  printf("\n");
                  print "WELLTYPE"
                  print "GINJ   MCFINJ"
                  print "PINJ   MCFINJ"
                  print "PROD   MCFINJ"
                  print "BHP"
                  printf ("GINJ  %s \n", P)
                  printf ("PINJ  %s \n", P)
                  printf ("PROD  %s \n", P)
                  print "RATE"
                  printf ("GINJ  %s \n", RI1)
                  printf ("PINJ  %s \n", RI1)
                  printf ("PROD  %s \n", RI1)
                  printf ("DT  %s \n", DT)
                  printf ("TIME  %s \n", x2_half)
                  print "WELLTYPE"
```

```
    print "GINJ   STBOIL"
    print "PINJ   STBOIL"
    print "PROD   STBOIL"
    print "BHP"
    printf ("GINJ  %s \n", GP0)
    printf ("PINJ  %s \n", GP0)
    printf ("PROD  %s \n", PP0)
    print "RATE"
    printf ("PROD  %s \n", PO2)
    print "PINJ  0"
    printf ("GINJ  %s \n", PO2)
    printf ("DT  %s \n", DT)
    printf ("TIME  %s", x2)
    printf("\n");
    x1 = x2
    }
else if (x2 == xf && x2_half < xf)
    {
    printf("\n");
    print "WELLTYPE"
    print "GINJ   MCFINJ"
    print "PINJ   MCFINJ"
    print "PROD   MCFINJ"
    print "BHP"
    printf ("GINJ  %s \n", P)
    printf ("PINJ  %s \n", P)
    printf ("PROD  %s \n", P)
    print "RATE"
    printf ("GINJ  %s \n", RI1)
    printf ("PINJ  %s \n", RI1)
    printf ("PROD  %s \n", RI1)
    printf ("DT  %s \n", DT)
    printf ("TIME  %s", x2_half)
    printf("\n");
    x1 = x2
    }
```

```
else if (x2 > xf && x2_half < xf)
   {
   printf("\n");
   print "WELLTYPE"
   print "GINJ   MCFINJ"
   print "PINJ   MCFINJ"
   print "PROD   MCFINJ"
   print "BHP"
   printf ("GINJ  %s \n", P)
   printf ("PINJ  %s \n", P)
   printf ("PROD  %s \n", P)
   print "RATE"
   printf ("GINJ  %s \n", RI1)
   printf ("PINJ  %s \n", RI1)
   printf ("PROD  %s \n", RI1)
   printf ("DT  %s \n", DT)
   printf ("TIME  %s", x2_half)
   printf("\n");
   x1 = x2
   }
else if (x2 > xf && x2_half > xf)
   {
   printf("\n");
   print "WELLTYPE"
   print "GINJ   STBOIL"
   print "PINJ   STBOIL"
   print "PROD   STBOIL"
   print "BHP"
   printf ("GINJ  %s \n", GP0)
   printf ("PINJ  %s \n", GP0)
   printf ("PROD  %s \n", PP0)
   print "RATE"
   printf ("GINJ  %s \n", PO2)
   print "PINJ  0"
   printf ("PROD  %s \n", PO2)
   printf ("DT  %s \n", DT)
```

```
                    printf ("TIME  %s 365 1", xf)
                    x1 = x2
                    }}}}
{                   printf("\n");
}END {}
```

GAWK3 Depletion RF Script

```
BEGIN {
#      awk: each program line consists of pattern {action}
#      each input file line is processed through the whole program
       DRF=8.08195        # Simulation result of depletion recovery factor
       TRF=18.8291        # Simulation result of final recovery factor
       DCO=8.19341        # Simulation result of depletion cumulative oil
produced
       TCO=19.0888        # Simulation result of final cumulative oil
produced
       counter=0;
}
counter == 0
{      DeltaRF = TRF - DRF
       DeltaCOP = TCO - DCO
       if (DRF < TRF)
{      print ""
       printf ("Depletion Oil Recovery Factor = %.4s %\n", DRF)
       printf ("Depletion Cumulative Oil Produced (1 Pattern)= %.4s MSTB\n",
DCO)
       printf ("Final Oil Recovery Factor (FRF) = %.4s %\n", TRF)
       printf ("Final Cumulative Oil Produced (FCO) (1 Pattern)= %.4s
MSTB\n", TCO)
       print ""
       printf ("Delta Cumulative Oil Produced (1 Pattern)= %.4s MSTB\n",
DeltaCOP)
       printf ("Delta RF = %.4s %\n", DeltaRF)
}
else
{      print ""
       print "Depletion Oil Recovery Factor = FRF"
       print "Depletion Cumulative Oil Produced (1 Pattern)= FCO"
       printf ("Final Oil Recovery Factor (FRF) = %.4s %\n", TRF)
       printf ("Final Cumulative Oil Produced (FCO) (1 Pattern)= %.4s
MSTB\n", TCO)
       print ""
       print "Depletion RF = Final RF"
```

```
}
        print ""
        print "Run completed"
}
END {
}
```

GAWK4 Scale Up Factor Script

```
BEGIN {
#      awk: each program line consists of pattern {action}
#      each input file line is processed through the whole program
       RLM=1              # Reservoir length input in miles
       DFF=198            # Inter-fractures distance
       counter=0;
counter == 0
{      WLM = RLM
       WL = WLM * 5280
       NF = (WL / DFF)
       print ""
       printf ("Reservoir Length = %3.0f mile\n", WLM)
       printf ("Reservoir Length = %s ft\n", WL)
       printf ("Number of Fracture = %s \n", NF)
       printf ("Number of Fracture (rounded) = %3.0f \n", NF)
       printf ("Frac to Frac Distance = %s \n", DFF)
}
       print ""
       print "Run completed"
}
END {
}
```

GAWK5 Well Data Smoothing Script

```
BEGIN {
#     awk: each program line consists of pattern {action}
#     each input file line is processed through the whole program
      NFL=26.4            # Calculated number of segment
      NFR=26              # Rounded number of segment
      DFF=200             # Inter-fractures distance
      counter=0;
}
counter == 0
{     WL = NFR * DFF
      IZ = NFR * DFF
      SUF = NFR * 2
      NUMFRAC = NFR + 1
      NFRN = NFR - 1
      WLN = NFRN * DFF
      SUFN = NFRN * 2
      NUMFRACN = NFR
      if (NFR > NFL)
{     print ""
      printf ("Well Horizontal Length = %s ft\n", WLN)
      printf ("Well Length Multiplier = %s \n", NFRN)
      printf ("Interest Zone = %s ft\n", WLN)
      printf ("Scale Up Factor = %s \n", SUFN)
      printf ("Number of Fracture = %s \n", NUMFRACN)
}
      else
{     print ""
      printf ("Well Horizontal Length = %s ft\n", WL)
      printf ("Well length Multiplier = %s \n", NFR)
      printf ("Interest Zone = %s ft\n", IZ)
      printf ("Scale Up Factor = %s \n", SUF)
      printf ("Number of Fracture = %s \n", NUMFRAC)
}
      print ""
```

```
    print "Run completed"
}
END {
}
```

GAWK6 Well Scale Calculation Script

```
BEGIN {
#     awk: each program line consists of pattern {action}
#     each input file line is processed through the whole program
      SUF=52            # Calculated scale up factor
}
{     total = SUF * $3
      print $1, total
}
END {
}
```

GAWK7 Script QC Case 1

```
BEGIN {
      SS=281      # total simulation step result
      CW=0        # cumulative water produced from simulation result
      CO=19       # cumulative oil produced from simulation result
      CG=18       # cumulative gas produced from simulation result
      WI=0        # cumulative injected water from simulation result
      GI=35       # cumulative injected gas from simulation result
      WP=0        # water production rate from simulation result
      OP=5        # oil production rate from simulation result
      GP=5        # gas production rate from simulation result
      WR=0        # water injection rate from simulation result
      IR=0        # gas injection rate from simulation result
      GO=998      # gas oil ratio from simulation result
      WC=0.0      # water cut from simulation result
      OR=18.8     # oil recovery from simulation result
      GR=-22.5    # gas recovery from result
      PA=4489     # pressure average from simulation result
      counter=0;
}
counter == 0
{     SSE = 255 - SS
      CWE = 0 - CW
      COE = 15 - CO
      CGE = 14 - CG
      WIE = 0 - WI
      GIE = 28 - GI
      WPE = 0 - WP
      OPE = 4 - OP
      GPE = 4 - GP
      WRE = 0 - WR
      IRE = 0 - IR
      GOE = 978 - GO
      WCE = 0 - WC
      ORE = 15.8 - OR
```

```
      GRE = -17.6 - GR

      PAE = 4572 - PA

      print "Case 1 : 1 year cycle time, 200 ft frac to frac distance, 0% cement porosity, 0 md cement permeability"

      print ""

      print "Field Summary"

      print "    Time     Cum Production       Cum Injection
      Production Rates  Injection Rates             %       %          %
HC          Remark"

      print "Days Step  Water Oil   Gas   Water   Gas    Water     Oil  Gas
        Water        Gas          GOR        WCUT  Oil Rec  Gas Rec   PAVG"

      print "3650 255   0     15    14    0       28      0       4
      4           0      0        978          0        15.8         -17.6
4572      Original"

      printf ("3650     %s    %.2s  %.2s  %.2s  %.2s    %.2s          %.2s
            %.2s       %.2s        %.2s      %.2s      %s      %.2s
%.4s     %.5s    %s      Pipe-It \n", SS, CW, CO, CG, WI, GI, WP, OP, GP,
WR, IR, GO, WC, OR, GR, PA)

      printf ("Error    %.4s  %.2s   %.2s  %.2s %.2s     %.2s          %.2s
            %.2s       %.2s        %.2s      %.2s     %.5s     %.2s
%.5s       %.5s     %.5s \n", SSE, CWE, COE, CGE, WIE, GIE, WPE, OPE, GPE,
WRE, IRE, GOE, WCE, ORE, GRE, PAE)

      print ""

      print "Note: Error calculation is original minus simulation value"

}
```