

The Pennsylvania State University

The Graduate School

John and Willie Leone Family Department of Energy and Mineral Engineering

**THE DEVELOPMENT OF AN ARTIFICIAL NEURAL NETWORK AS A
PRESSURE TRANSIENT ANALYSIS TOOL WITH APPLICATIONS IN
MULTI-LATERAL WELLS IN TIGHT-GAS DUAL-POROSITY
RESERVOIRS**

A Thesis in

Energy and Mineral Engineering

by

Jacob Steven Cox

© 2014 Jacob Steven Cox

Submitted in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science

August 2014

The thesis of Jacob Steven Cox was reviewed and approved* by the following:

Turgay Ertekin

Head of the John and Willie Leone Family Department of Energy and Mineral Engineering
George E. Trimble Chair in Earth and Mineral Sciences
Professor of Petroleum and Natural Gas Engineering

Li Li

Assistant Professor of Petroleum and Natural Gas Engineering

John Yilin Wang

Assistant Professor of Petroleum and Natural Gas Engineering

Luis F. Ayala H

Associate Professor of Petroleum and Natural Gas Engineering
Associate Department Head for Graduate Education

*Signatures are on file in the Graduate School.

Abstract

The use of a commercial software package was used to produce pressure transient (PT) data via a drawdown test for a large variety of tight gas dual-porosity methane reservoirs with a multi-lateral well completion. The goal of this study was to create a tool with the use of an artificial neural network (ANN) that could quickly predict the inverse solution to the PT data. This inverse tool would be able to predict the user's reservoir parameters nearly instantaneously with known inputs of PT data and wellbore design. This tool will take ideas from current well test analysis to aid in the training of the neural network. However, once the network has been trained, the time consuming process of conventional well test analysis will no longer be an issue.

This tool will give results much like a history matching of production or pressure data would give, however this tool will require a large amount of training time. After the training has been completed the ANN will have the ability to predict reservoir characteristics nearly instantaneously. The predictions from this tool are valuable because it will save the user a large amount of time that would be spent analyzing the well test data. Also, there is an overwhelmingly large amount of data being taken and stored every day, and it is often never analyzed. This tool will make analyzing large volumes of well test data not only possible, but it is conceivable that this analysis could be completed in minutes rather than weeks or months. This tool will also be able to predict multiple reservoir properties whereas typical well test analysis can only predict one. Lastly, this tool's predictions will allow the user to have a much smaller data range when they make the next step into more advanced modeling techniques, saving them time and money.

Table of Contents

List of Figures	vii
List of Tables	x
Nomenclature add units	xi
Acknowledgements	xiii
Chapter 1: Introduction	1
Chapter 2: Well Test Analysis and Selection of Forward Model	9
2.1. <i>Typical Well Tests</i>	9
2.2. <i>Drawdown Test Analysis</i>	13
2.2.1. <i>Pressure Drawdown Analysis in Infinite Acting Reservoir</i>	14
2.2.1.1. Incorporating Skin	16
2.2.2. Dual Porosity Characteristics	19
2.3. <i>Reservoir Structure</i>	23
2.4. <i>Reservoir Parameters</i>	29
2.5. <i>CMG model specifications</i>	62
Chapter 3: Artificial Neural Networks	64
3.1. <i>Introduction to Neural Networks (NN)</i>	64
3.2. <i>History of Neural Networks (NN)</i>	66
3.2.1. Early Times (1890-1949)	66
3.2.2. 1950's and 1960's	67

3.2.2. 1980's.....	69
3.2.3. Applications	69
3.3. <i>How do NNs work?</i>	72
3.3.1. Biological Inspiration.....	72
3.3.2. Processing Elements (PE).....	75
3.3.3. Weighting Factors.....	75
3.3.4. Transfer Functions	76
3.3.5. Bringing it All Together.....	78
3.4. <i>How is Information Stored within the Network?</i>	82
3.4.1. Learning	83
3.4.2. Self-Organization.....	83
3.4.3. Training.....	83
3.4.4. Generalization.....	85
3.5. <i>Neural Network Learning Rate and Learning Laws</i>	85
3.5.1. Learning Rate.....	86
3.5.2. Learning Laws	86
3.6. <i>Back Propagation Network</i>	87
Chapter 4: Development Stages.....	90
4.1. <i>Forward Solution</i>	90
4.1.1. Forward Solution Introduction.....	90

4.1.2. Forward Solution Data Generation	91
4.1.3. Forward Solution ANN Training	95
4.2. <i>Inverse Solution</i>	106
4.2.1. Inverse Solution Introduction	106
4.3. <i>Creation of Graphical User Interface (GUI)</i>	113
Chapter 5: Discussion of Results	116
5.1. <i>Forward Solution</i>	116
5.2. <i>Inverse Solution</i>	121
5.2.1. High Matrix Permeability Reservoir.....	121
5.2.2. Low Permeability Reservoir	128
Chapter 6: Conclusions	135
Bibliography	138
Appendix A: Complete ANN Table	143
Appendix B: 2L Forward/Inverse Solution Results.....	144
Appendix C: 3L Forward/Inverse Solution Results.....	150
Appendix D: Example MATLAB Code Generating Data for CMG Analysis	156
Appendix E: Example Training Data.....	179

List of Figures

Figure 1: Commercial Software's representation of a multilateral well.....	2
Figure 2: Wellbore geometries studied.	3
Figure 3: Dual porosity reservoir representation.	5
Figure 4: Relationship between matrix pressure and fracture pressure	7
Figure 5: Basic design of Pressure Buildup test and the results of the design.....	10
Figure 6: Basic design of Pressure Drawdown test and the results of the design.....	11
Figure 7: Basic design of Injection test and the results of the design.....	12
Figure 8: Basic design of Fall Off test and the results of the design	13
Figure 9: Typical Horner plot for drawdown test	18
Figure 10: Diagnostic plot	19
Figure 11: Dual Porosity semi-log plot.....	23
Figure 12: Cumulative Gas as a function of Grid Block Size over 20 year span.	26
Figure 13: Cumulative Gas as a function of Grid Block Size over 20 day span.	26
Figure 14: Daily Production as a function of grid block size, over 20 year span.	27
Figure 15: Daily Production as a function of grid block size, over 20 day span.....	27
Figure 16: How fracture spacing affects the PT Drawdown Curve.....	32
Figure 17: How reservoir thickness affects the PT Drawdown Curve.	33
Figure 18: How matrix porosity affects the PT Drawdown Curve.	35
Figure 19: How fracture porosity affects the PT Drawdown Curve.	36
Figure 20: How Reservoir Temperature affects the PT Drawdown Curve.	37
Figure 21: How wellbore length affects the PT Drawdown Curve.	38
Figure 22: How Initial Pressure affects the PT Drawdown Curve.	40

Figure 23: How initial pressure affects the derivative of BHP curve.	41
Figure 24: How matrix permeability affects the PT Drawdown Curve (FS=1ft).	43
Figure 25: How matrix permeability affects the PT Drawdown Curve (FS=200ft).	44
Figure 26: How fracture permeability affects the PT Drawdown Curve (FS=1ft).	45
Figure 27: How matrix permeability affects the PT Drawdown Curve (FS=200ft).	46
Figure 28: How fracture spacing affects cumulative production.	49
Figure 29: How reservoir thickness affects cumulative production.	50
Figure 30: How matrix porosity affects cumulative production.	51
Figure 31: How matrix permeability affects cumulative production.	53
Figure 32: How reservoir temperature affects cumulative production.	54
Figure 33: How well length affects cumulative production.	55
Figure 34: How initial pressure affects cumulative production.	57
Figure 35: How matrix permeability affects cumulative production (FS=1 ft).	58
Figure 36: How matrix permeability affects cumulative production (FS=200 ft).	59
Figure 37: How fracture permeability affects cumulative production (FS=1 ft).	61
Figure 38: How fracture permeability affects cumulative production (FS=200 ft).	62
Figure 39: Swedish Proverb with vowels removed	65
Figure 40: Simplified Diagram of two biological neurons.	74
Figure 41: Schematic of a Feedback Network.	81
Figure 42: Forward Solution Diagram	91
Figure 43: Reservoir/Well Data Distribution.	94
Figure 44: Error vs. Iterations Plot.	102
Figure 45: Inverse Solution Diagram.	107

Figure 46: Inverse Solution ANN Final Structure	110
Figure 47: Inverse Solution GUI.....	114
Figure 48: Forward Solution GUI.....	115
Figure 49: Best Forward Solution PT data fits	117
Figure 50: Forward Solution Average PT Data Fits	118
Figure 51: Forward Solution Testing Error	119
Figure 52: Forward Solution Error Histogram.....	120
Figure 53: High Matrix Perm: Reservoir Training Results	122
Figure 54: High Matrix Perm: Reservoir Test Results	122
Figure 55: High Matrix Perm: Permeability Training Results	124
Figure 56: High Matrix Perm: Permeability Testing Results	124
Figure 57: Validation Test for high matrix permeability ANN	127
Figure 58: Low Matrix Perm: Reservoir Training Results	129
Figure 59: Low Matrix Perm: Reservoir Testing Results	129
Figure 60: Low Matrix Perm: Permeability Training Results	131
Figure 61: Low Matrix Perm: Permeability Test Results	131
Figure 62: Validation Test for low matrix permeability ANN	134

List of Tables

Table 1: Grid Block size and Corresponding Time of Simulation	25
Table 2: Reservoir Parameters for grid block size comparison	25
Table 3: Ranges of data found in the literature ² and used in the simulator.	30
Table 4: Fields and Applications for Neural Networks	69
Table 5: Various Transfer Functions used in MATLAB ANN program.....	78
Table 6: Forward Solution Input/Output Summary	96
Table 7: Inverse Solution Input/Output Summary	112
Table 8: Forward Solution Error Table.....	120
Table 9: High Matrix Perm: Reservoir Properties Error Table.....	123
Table 10: High Matrix Perm: Error Table	125
Table 11: Target and ANN data generated by high matrix permeability ANN.....	126
Table 12: Low Matrix Perm: Reservoir Properties Error Table	130
Table 13: Low Matrix Perm: Error Table	132
Table 14: Target and ANN data generated by low matrix permeability ANN.....	133

Nomenclature add units

ROMAN

ANN	Artificial Neural Network
<i>BHP</i>	Bottom Hole Pressure (psi)
BNN	Biological Neural Network
<i>c</i>	Compressibility (psi^{-1})
<i>FS</i>	Fracture Spacing (ft)
<i>h</i>	Reservoir Thickness (ft)
k_m	Permeability of Matrix (md)
k_f	Permeability of Fracture (md)
NN	Neural Network
\overline{P}	Average Reservoir Pressure (psi)
P_i	Initial Reservoir Pressure (psi)
P_{wf}	Well Flowing Pressure (psi)
P_{ws}	Well Shut-in Pressure (psi)
PE	Processing Element
S	Skin (dimensionless)
scfD	Standard Cubic Feet per Day
SCFD	Standard Cubic Feet per Day
SCF/d	Standard Cubic Feet per Day
T_{res}	Reservoir Temperature (°F)

GREEK

β	Formation volume factor
γ	Dimensionless constant
λ	Dimensionless constant

μ	Viscosity (cp)
\emptyset_m	Matrix Porosity (Percentage or Fraction)
\emptyset_f	Fracture Porosity (Percentage or Fraction)
θ	Angle (degrees)

SUBSCRIPTS

D	Dimensionless
f	Fracture
g	gas/vapor
i	Initial
m	Matrix
res	Reservoir
tm	Time dependent matrix
tf	Time dependent fracture
wd	Dimensionless wellbore
wf	Well Flowing

Acknowledgements

I would like to thank my thesis advisor Dr. Turgay Ertekin for his valuable assistance and discussions throughout this study. I appreciate being given the opportunity to work alongside one of Penn State's finest professors. I also wanted to extend a special thank-you to Dr. John Yilin Wang and Dr. Li Li for their interest in serving on my committee.

I am very grateful for my colleagues in the Petroleum and Natural Gas Engineering program. A personal thank-you is extended to Chukwuka Enyioha for his valuable assistance throughout this project.

I would like to thank my family for their constant support throughout this project. A special thank you to both of my parents who have always given me the tools and motivation that I needed to succeed.

And lastly, I want to personally thank my wife, Stephanie, for always being there when I needed her. There is and never will be a replacement for her love, support, and encouragement throughout this project and our life together.

Chapter 1: Introduction

As the interest and ability to harvest hydrocarbons from tight gas reservoirs expands, the longing to characterize these reservoirs expands with it. This thesis will focus on the characterization of a dual porosity tight gas reservoir. The completion technique being studied is a multilateral fishbone structure. The initial goal of the research was to create a tool that can characterize any tight gas reservoir that was completed via a multi-lateral well through data collected during a drawdown pressure test. Tight gas reservoirs that are of particular importance are both sandstone and shale reservoirs that have clean natural fractures. If the natural fractures have scaling or are completely closed off, they will not be able to transport hydrocarbons as expected, and a different completion technique other than multilateral wells will have to be implemented in order for the well to produce at an economic rate.

The term ‘multilateral well’ can refer to many different well structures. Some multilateral wells are used to intersect multiple tight formations that are vertically separated from one another via geologic layers. The multilateral well structure that we are studying is implemented to maximize wellbore connectivity within a single geologic layer. This multilateral well is merely a horizontal well that has additional wells which branch off from the main wellbore. A diagram of a multilateral well with four lateral wings that is typical of this study is shown below in Figure 1. As mentioned previously, other variations of multilateral wells exist, however this is one of the three structures that will be focused upon in this study. All three of the well geometries that were studied are shown in Figure 2. Note the symmetry that was used for this study.

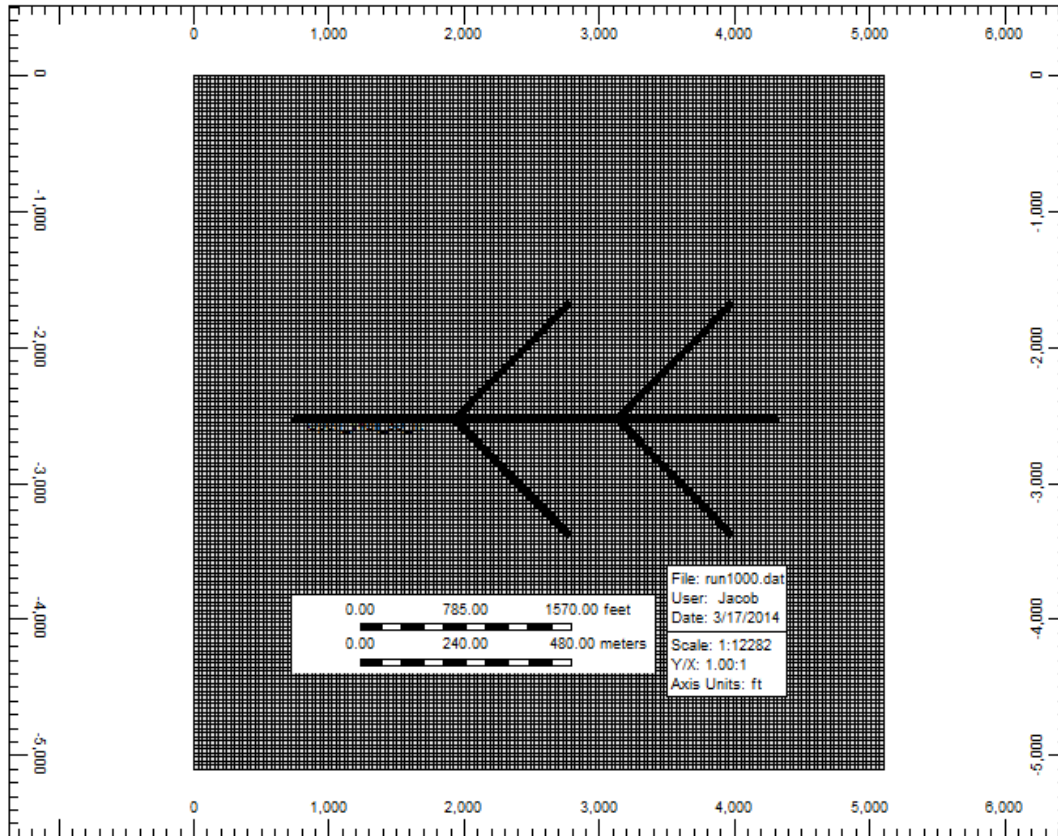


Figure 1: Commercial Software's representation of a multilateral well consisting of four lateral wells.

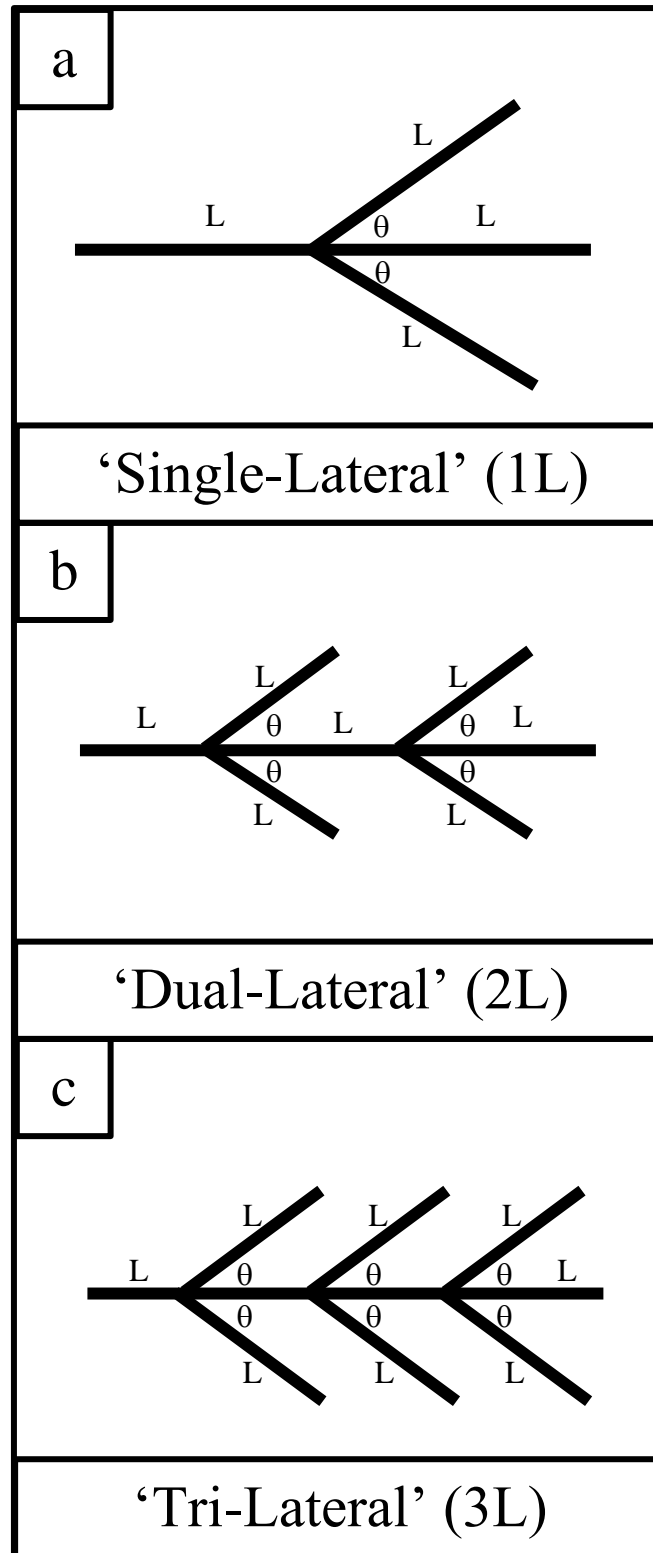


Figure 2: Schematic showing symmetry between the three cases that will be studied in this research.

Multilateral wells can be used in an attempt to connect the wellbore to a larger portion of the reservoir. Wellbore connectivity is particularly important in tight gas dual porosity systems because these systems tend to have areas of high permeability and areas of low permeability. By branching out the wellbore in the fishbone structure, there is a higher chance of intersecting a high permeability region. Because the multilateral well technique is currently being used today and has a moderate chance of being used extensively in the upcoming future, the development of an inverse tool through an artificial neural network which can successfully predict reservoir characteristics will be the goal of this research. A commercial modeling software¹ was used to create the forward solution (i.e. the Pressure Transient Data). This software was chosen to create the forward solution because extensive field data is difficult to access, the software package was readily available to use, and in the past individuals have used commercial modeling software to match production values by altering reservoir properties (Callard & Schenewerk, 1995), (Liu, Kelkar, Gang, & Dixon, 2007), (Watson, Lane, & Gatens III, 1990). Thus, because individuals use modeling software to history match their production data to characterize a reservoir, one can use the same modeling software to generate data that will be used to predict reservoir properties.

Dual porosity reservoirs are reservoirs which consist of natural fractures. This study used the Warren and Root method to represent the natural fractures. This method simplifies the complex structure of natural fractures via a ‘sugar-cube’ model. The matrix is then represented by cubic

¹ The commercial modeling software used was provided by Computer Modeling Group LTD (CMG). In particular, the IMEX software was used due to our single phase gas production.

units and the space between these cubic units represents the natural fractures. In this manner one can represent the two distinct porous systems. Figure 3 below shows an image from Warren and Root's journal article which shows the relationships between the reservoir and the model.

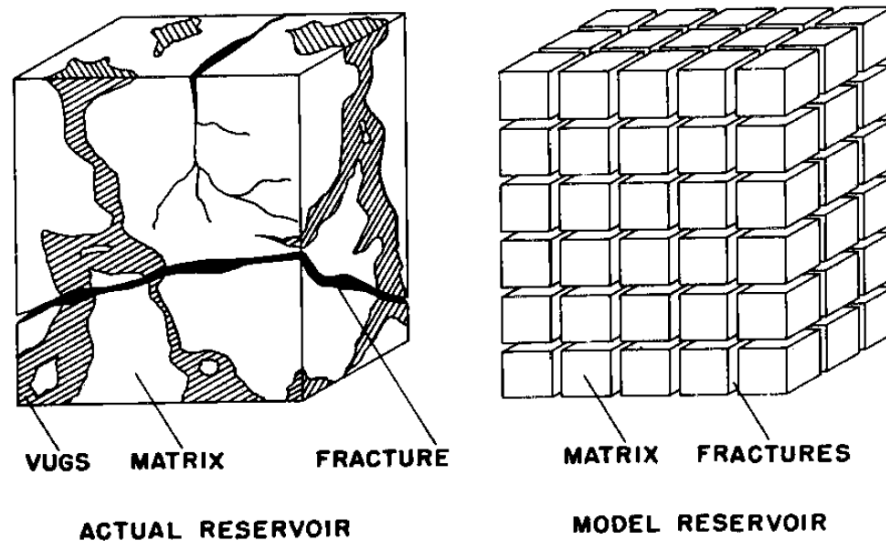


Figure 3: Warren and Root's sugar-cube model on the right is a representation of a dual porosity reservoir on the left (Warren & Root, 1962).

Warren and Root's assumptions for this reservoir model include, but are not limited to:

- a) The space comprising the matrix porosity is homogeneous and isotropic. The primary porosity is contained within an organized matrix of duplicate rectangular cubes.
- b) The fracture porosity can be represented by an orthogonal system of continuous uniform fractures. Orthogonality is such that the flow is in the same direction as the principal axes of permeability. The width of a fracture along a certain axis is constant, but different axes can have different widths as well as different fracture spacing to account for anisotropy.
- c) Flow can occur from the matrix to the fracture, but cannot occur from matrix to matrix element. (Warren & Root, 1962)

The third assumption is very important to the modeling of a dual porosity reservoir. When modeling the reservoir, one can visualize the matrix elements as source/sink terms and the fracture elements as the transport media for the reservoir. This being said, the pressure in the matrix porosity is assumed to be the average reservoir pressure. The pressure within the fracture must be lower to induce flow from the matrix to the fracture. This process is displayed below in Figure 4. The idea that the matrix cannot transport fluid except to the fracture network is the main reason why one needs to maximize connectivity of the wellbore to tight-gas dual-porosity reservoirs. Doing so enhances the chances that the highly permeable natural fractures will be intersected. This is why horizontal wells are being used more than vertical wells in recent completions of tight gas reservoirs. Moreover, in especially tight formations an even higher wellbore connectivity is required, and this is why the applications of multilateral wells must be implemented. It should be noted that there are several ‘dual-porosity’ simulators included in our software package including ‘Dual Permeability’, ‘Subdomain’, and ‘multiple interacting continua’ (MINC). These packages all can be used to represent dual porosity reservoirs; however, it was decided to use the ‘standard dual porosity’ due to its familiarity and simplicity. The other three packages are more complex and incorporate gravity effects along with other nuances. Because our system is a single phase gas reservoir, these additional nuances were not necessary to provide accurate representation of the reservoir.

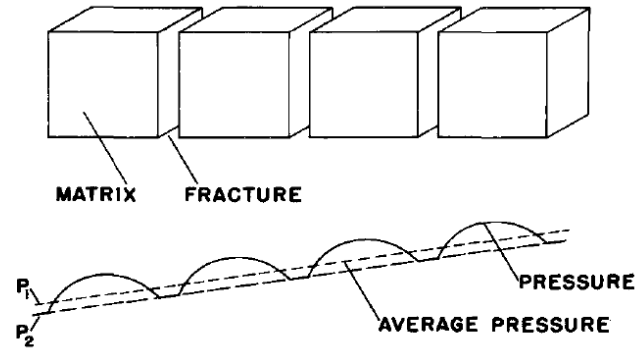


Figure 4: Showing relationship between matrix pressure and fracture pressure. (Warren & Root, 1962)

The remainder of this work will be split up into chapters. Chapter two will focus on well test analysis, which is a process in obtaining macroscopic or field wide data from well tests such as drawdown or buildup tests, to name a few. The application of well tests will be shown, but well test analysis of dual porosity reservoirs will be the main focus. This section will also go through the analysis used in determining the reservoir structure (such as grid size, drainage area, etc.) and selection of reservoir parameters' ranges ($k_m, k_f, \phi_m, \phi_f, h, P_i, T_{res}$, Fracture Spacing (FS) and wellbore length). This chapter will be concluded with a small discussion on the commercial software package used to generate the forward solution of the problem.

Chapter three will focus on the artificial neural network (ANN) which will be used as an inverse solution tool for this study. A brief history of ANN will be given followed by a generalized overview of neural networks.

Chapter four will give a quick overview of the techniques used to solve the problem at hand. The specific structures of ANN's that were used to solve a wide array of problems will be shown and the methodology used to create a better ANN structure, and consequently better prediction capability, will be evaluated.

Chapter five will be the results and discussion section. Here all of the relevant results for the Forward and Inverse Solution ANNs will be shown. The paper's main findings and conclusions will then be summarized in Chapter six.

Chapter 2: Well Test Analysis and Selection of Forward Model

Well test analysis is a historic process that uses either production or pressure data to aid in the characterization of a reservoir. For this study, it was chosen to use pressure data, and the well test that was chosen was a drawdown test. In this test, a well that was shut-in, such that the average reservoir pressure is very close to the actual reservoir pressure, is put on constant production. The corresponding flowing bottom hole pressure (BHP) of the well is recorded for a set period of time. This process will be explained in more detail in the following sections. The issues with this form of testing is that calculating the inverse solution, i.e. extracting reservoir properties from the pressure transient data can become laborious and intense, especially if there are multiple tests for several wells in a particular field. To decrease the amount of analysis time required to process a drawdown test, an ANN will be developed that will be able to read in pressure transient data and nearly instantaneously produce reservoir characteristics. However, in order for the ANN to perform well, the user designing it must have a thorough understanding of well test analysis. That is the purpose of this section.

2.1. Typical Well Tests

Typical well tests include but are not limited to pressure build up tests, drawdown tests, injection well tests, and fall off tests. Each of these tests requires specialized interpretation methods in order to obtain reservoir information. However, well test methods are all similar in the fact that the well is exposed to a given stimuli and the reaction to that stimuli is recorded and analyzed. This study focuses on pressure transient analysis, and thus the stimulus discussed in the

succeeding well tests will be flow rate and the results of that stimulus will be the pressure drop. However, in most cases the stimuli can be changed to pressure and the results of that stimuli will be the flowrate. This is referred to as rate analysis.

An example of a pressure buildup test is shown below in Figure 5a and 5b. For this test, a well is produced at a constant rate for an extended period of time. At the conclusion of the production time (t_p), the well is shut-in. The shut-in well pressure is recorded just prior to the well being shut-in, and then is continuously monitored for a specified amount of time. The data can then be manipulated and the results will give the user reservoir data such as permeability, initial pressure, well bore storage constant, and skin factor.

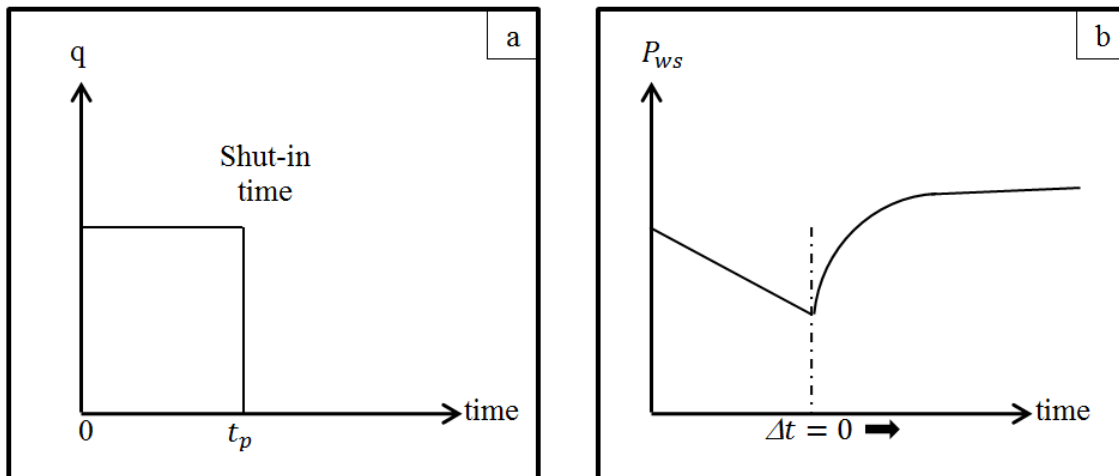


Figure 5: (a) Schematic showing basic design of Pressure Buildup test and the (b) results of the design

The drawdown test is similar to the buildup test, but the order of well operations is reversed. With a drawdown test a well that has been shut-in is put on constant production. The pressure is

recorded during shut-in as well as during production. Extended drawdown tests can be implemented to discover the drainage area of the well. This test can also give reservoir/well properties such as permeability, skin, wellbore storage effects, etc. The operations of this well test are shown in Figure 6a and b, and this method will be discussed in further detail in the next section.

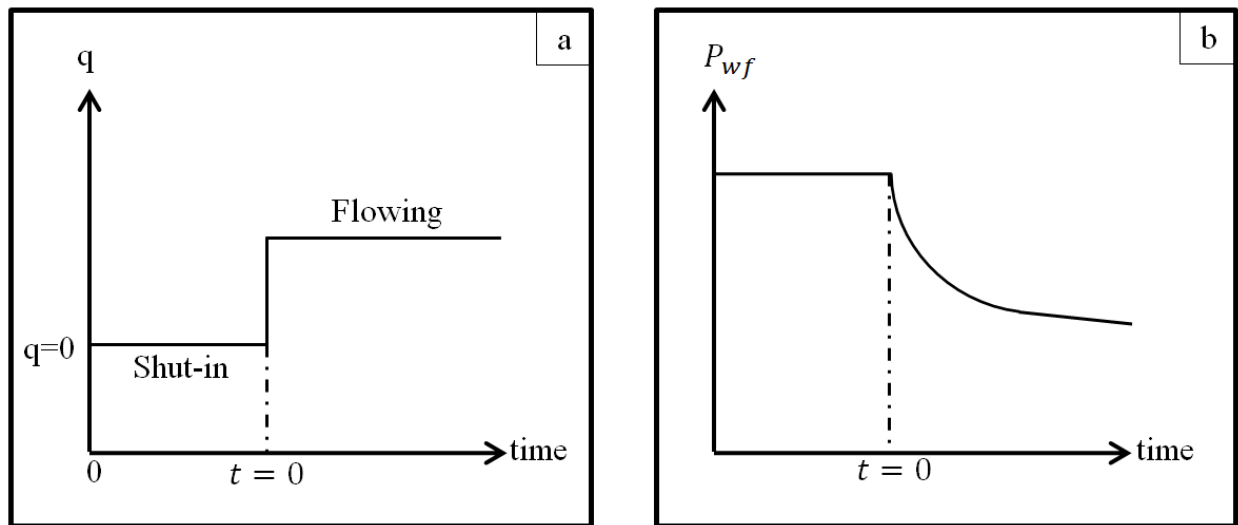


Figure 6: (a) Schematic showing basic design of Pressure Drawdown test and the (b) results of the design

Injection well tests are used to determine the unit mobility ratio of the reservoir. This test is similar to the drawdown test except the flowrate will be negative (injection vs. production). Thus, one can obtain the same reservoir/well parameters as in the drawdown test. In this test a fluid is injected into a shut-in well for a specific amount of time. The corresponding rise in wellbore flowing pressure is then recorded. This process is shown in Figure 7a and b below. Again, this test can be extended in order to find the reservoir limits and drainage area of the well.

The injection test is often followed by a fall off test. The fall off test is analogous to the buildup test, but in this case instead of the well producing for a set amount of time prior to being shut-in, it will be injected into. This procedure is shown in Figure 8a and b below. The well bore shut-in pressure will then decrease until it reaches the average reservoir pressure (\bar{P}). Typically this test will give the permeability of the reservoir as well as the skin factor.

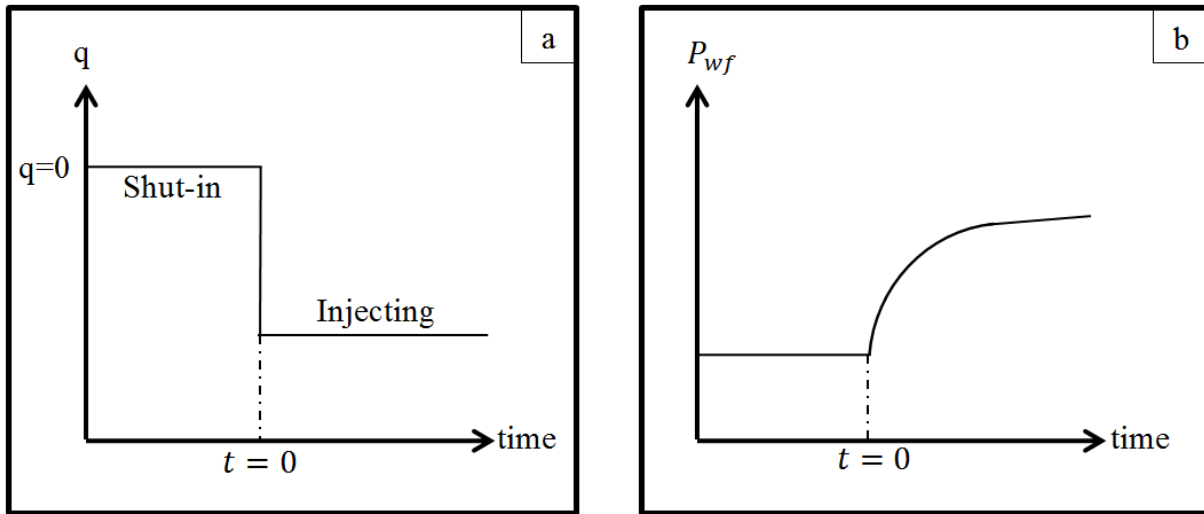


Figure 7: (a) Schematic showing basic design of Injection test and the (b) results of the design

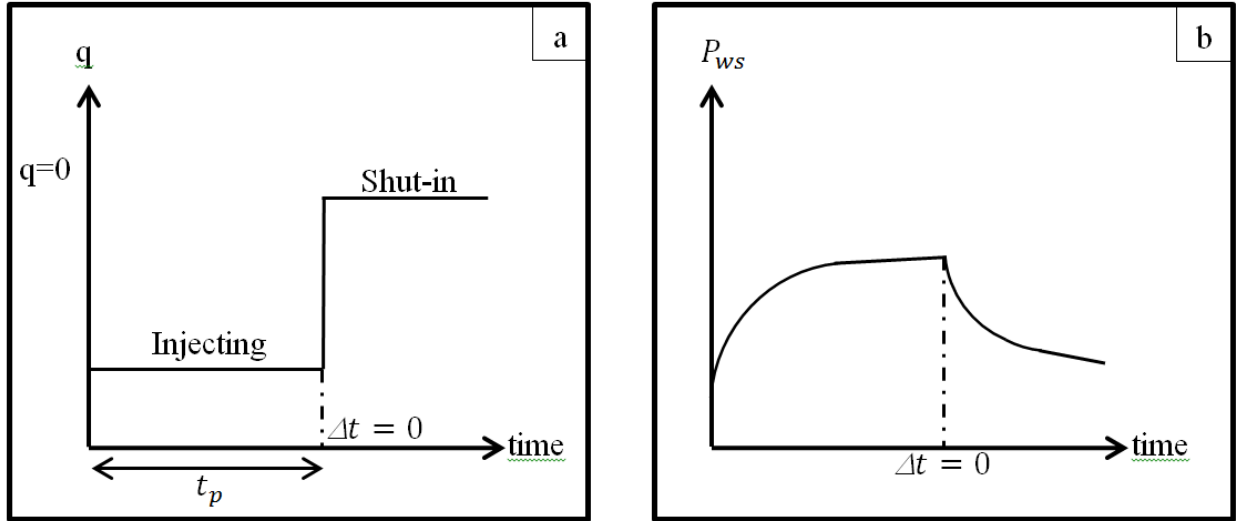


Figure 8: (a) Schematic showing basic design of Fall Off test and the (b) results of the design

2.2. Drawdown Test Analysis

In this section, the pressure drawdown test analysis will be elaborated upon. Recall, for the pressure drawdown test a well that was shut in is put on constant production. The resulting drop in well flowing pressure is recorded. It will be shown that plotting P_{wf} versus $\log(t)$ will generate a straight line. One can then calculate the slope of this line and by rearranging the equation, calculate an unknown reservoir value. The reservoir value typically calculated using this procedure is permeability, but it will be shown that one could calculate either β , μ , or height, given that all other variables are known.

2.2.1. Pressure Drawdown Analysis in Infinite Acting Reservoir

The reservoir in this study was constructed such that one can assume that the well behaves as if it is in the infinite acting regime at early times. Thus, none of the pressure transients have reached any of the boundaries. The basic equation to determine the dimensionless pressure drop at the wellbore is:

$$\Delta P_D = -\frac{1}{2} E_i \left(-\frac{r_D^2}{4t_D} \right) \quad \text{Eq. 1}$$

Where ΔP_D is the dimensionless pressure drop at the well bore, E_i is the exponential integral of the function $E_i(x) = \int_{-\infty}^x \frac{e^t}{t} dt$, r_D is the dimensionless wellbore radius, and t_D is the dimensionless time. However, this Equation reduces to:

$$P_t = \frac{1}{2} [\ln(t_D) + 0.809] \quad \text{Eq. 2}$$

due to the infinite acting assumption. Note the straight-line relationship similar to $y = mx + b$.

By substituting in the dimensionless terms:

$$P_t = \frac{P_i - P_{wf}}{\frac{P_i(\gamma\beta q_{sc}\mu)}{P_i kh}} \quad \text{Eq. 3}$$

$$t_D = \frac{\lambda kt}{\phi\mu C r_w^2} \quad \text{Eq. 4}$$

One is left with:

$$\frac{P_i - P_{wf}}{\frac{P_i(\gamma\beta q_{sc}\mu)}{P_i kh}} = \frac{1}{2} \left[\ln \left(\frac{\lambda kt}{\phi\mu C r_w^2} \right) + 0.809 \right] \quad \text{Eq. 5}$$

Canceling out like terms, and solving for P_{wf} yields:

$$P_{wf} = P_i - \frac{(\gamma\beta q_{sc}\mu)}{2kh} \left[\ln \left(\frac{\lambda kt}{\phi\mu C r_w^2} \right) + 0.809 \right] \quad \text{Eq. 6}$$

In field units, $\gamma = 141.2$ and $\lambda = 2.3026 * 10^{-4}$

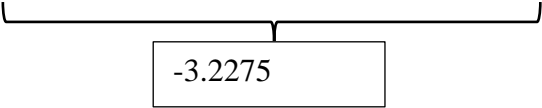
$$P_{wf} = P_i - \frac{(141.2\beta q_{sc}\mu)}{2kh} \left[\ln \left(\frac{(2.3026*10^{-4})kt}{\phi\mu C r_w^2} \right) + 0.809 \right] \quad \text{Eq. 7}$$

Using the rule that $\ln(x) = 2.3026 * \log(x)$

$$P_{wf} = P_i - \frac{(70.6\beta q_{sc}\mu)}{kh} \left[2.3026 \log \left(\frac{(2.3026*10^{-4})kt}{\phi\mu C r_w^2} \right) + 0.809 \right] \quad \text{Eq. 8}$$

Pulling out the 2.3016 from within the parenthesis, and splitting up the logarithm yields:

$$P_{wf} = P_i - \frac{(162.6\beta q_{sc}\mu)}{kh} \left[\log(t) + \log \left(\frac{k}{\phi\mu C r_w^2} \right) + \log(2.3026 * 10^{-4}) + 0.809/2.3026 \right] \quad \text{Eq. 9}$$



$$P_{wf} = P_i - \frac{(162.6\beta q_{sc}\mu)}{kh} \left[\log(t) + \log \left(\frac{k}{\phi\mu C r_w^2} \right) - 3.2275 \right] \quad \text{Eq. 10}$$

Now, it is clear if one plots P_{wf} versus $\log(t)$ a straight line will be produced, and the slope of that straight line is: $-\frac{(162.6\beta q_{sc}\mu)}{kh}$. This plot is referred to as a Horner Plot. Once the slope of this

line is known, one can obtain either β , μ , k , or h , so long as all of the other parameters are known. Typically, one will solve for the permeability of the reservoir.

2.2.1.1. Incorporating Skin

Next, it is often desirable to know the skin of the reservoir, or the dimensionless pressure drop due to either damage or stimulation of the near wellbore region. Equation 2 assumed that skin was equal to zero. By incorporating skin, Equation 2 becomes:

$$P_t + S = \frac{1}{2} [\ln(t_D) + 0.809] \quad \text{Eq. 11}$$

Incorporating skin into Equation 10 yields:

$$P_{wf} = P_i - \frac{(162.6\beta q_{sc}\mu)}{kh} \left[\log(t) + \log\left(\frac{k}{\phi\mu C r_w^2}\right) - 3.2275 \right] - 141.2 \left(\frac{\beta q_{sc}\mu}{kh} \right) S \quad \text{Eq. 12}$$

Bringing the Skin term into the main bracket yields:

$$P_{wf} = P_i - \frac{(162.6\beta q_{sc}\mu)}{kh} \left[\log(t) + \log\left(\frac{k}{\phi\mu C r_w^2}\right) - 3.2275 + 0.87S \right] \quad \text{Eq. 13}$$

And typically, for simplicity, one will solve this Equation at $t = 1\text{hour}$, because $\log(t)$ becomes $\log(1) = 0$.

$$P_{wf@t=1hr} = P_i - \frac{(162.6\beta q_{sc}\mu)}{kh} \left[\log\left(\frac{k}{\phi\mu C r_w^2}\right) - 3.2275 + 0.87S \right] \quad \text{Eq. 14}$$

Thus, even with the Skin factor, one would still plot P_{wf} versus $\log(t)$ and find the slope of the straight line. From the slope $\left(-\frac{(162.6\beta q_{sc}\mu)}{kh}\right)$ one can calculate the permeability of the system. Once the permeability is calculated Equation 13 can be rearranged to solve for skin. The permeability can then be plugged into this Equation yielding the skin.

A typical Horner plot is displayed in Figure 9 below. This plot is a representation of what field data looks like. The plot is broken up into the early time (ETR), middle time (MTR), and late time regions (LTR). The ETR is the time period in which wellbore storage effects can be seen (Ezekwe, 2011). This region will let the user know the magnitude and distance of the wellbore storage effects and can also be used to calculate the skin factor. The MTR is composed of data from areas of the reservoir that have not been affected by the wellbore. These areas are said to be composed of virgin permeability (Ezekwe, 2011). This gives the user the best idea of the reservoir properties, and is why the slope within the MTR is used to calculate the permeability of the system. Note the burgundy line drawn through the MTR. It is often very difficult to determine the region that is the MTR and using just the Horner Plot will often result in erroneous results. A diagnostic plot is typically used in order to find the region where the MTR begins. This makes determining the MTR a more scientific process. The diagnostic plot used to determine the MTR of the Horner plot in Figure 9 is shown in Figure 10. The diagnostic plot is very similar to the Horner plot; however $P_{ws} - P_{wf}$ is plotted versus $\log(t)$. This procedure will result in the wellbore storage effects having a signature of a unit slope. At the final point of the unit slope, the user will move 1.5 cycles to the right. This will give the beginning of the MTR. An interesting note is that when a half-unit slope is found, this indicates that the well is hydraulically fractured. A hydraulically fractured well

will have higher wellbore effects, thus our interpretation methods will change. In a hydraulically fractured well, one must go 1.5 cycles to the right starting at the end of the half-unit slope. It can be seen from both of the plots that the MTR must start at two hours. Finally, the LTR is the region where the reservoir boundaries or heterogeneities such as faults are shown by the Horner plot (Ezekwe, 2011). This region begins once the data deviates from the MTR slope. This region allows one to calculate the drainage area of the reservoir.

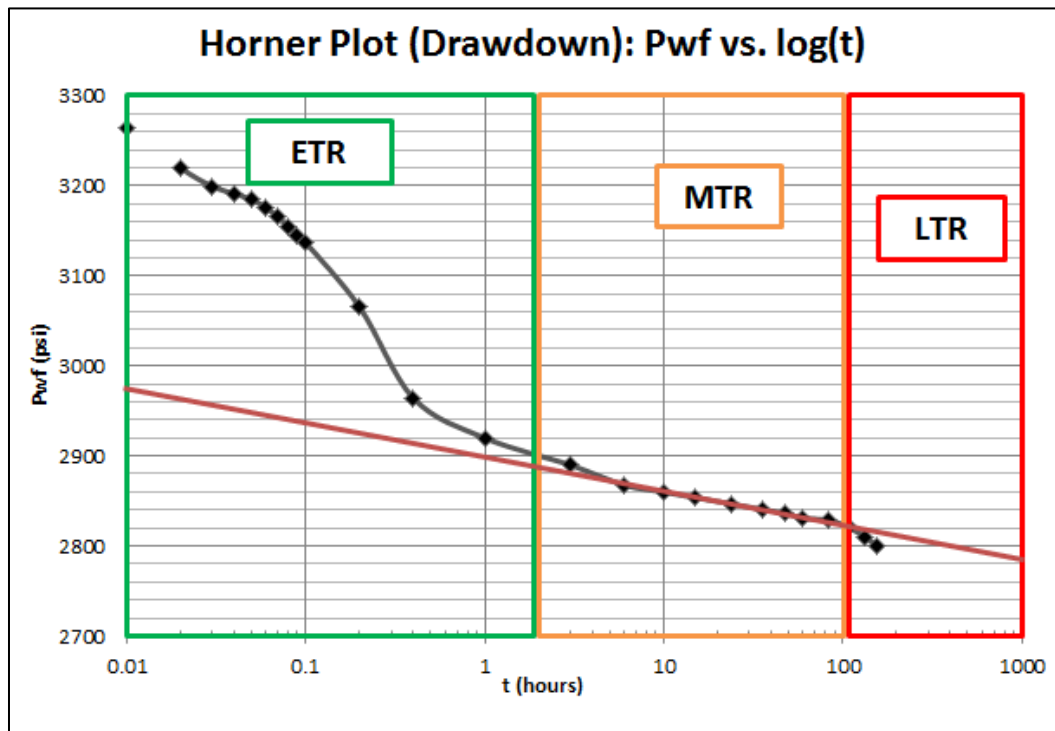


Figure 9: Typical Horner plot for drawdown test where ETR, MTR, and LTR are highlighted

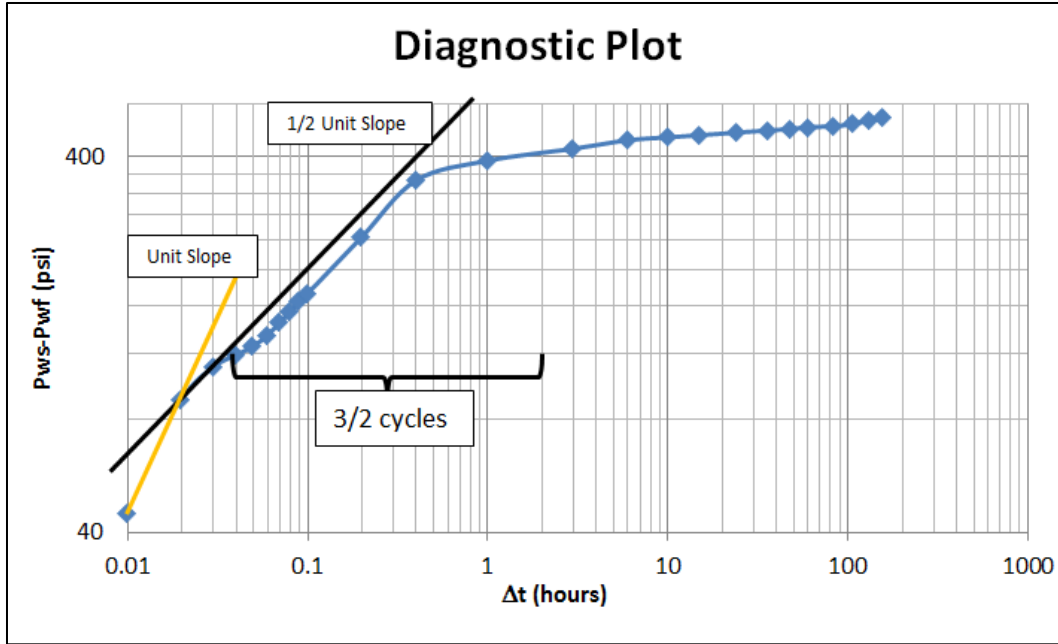


Figure 10: Diagnostic plot used to determine the beginning of the MTR on the Horner Plot.

2.2.2. Dual Porosity Characteristics

As mentioned previously, dual porosity reservoirs are used to describe reservoirs with natural fractures. The Warren and Root sugar cube model is used to model dual porosity reservoirs, and was described in detail in the introduction. Natural fractures have a high porosity in comparison to the matrix porosity, but a very low volume compared to the bulk volume of the system. Because of this, the natural fracture's porosity (ϕ_f) is actually very low. This is demonstrated by the equation:

$$\phi_f = \frac{\text{Pore Space in Fracture}}{\text{Bulk Volume of Reservoir}} \quad \text{Eq. 15}$$

For a dual porosity reservoir, the dimensionless equations and other important equations not mentioned previously are:

$$P_D = \frac{k_f h}{141.2 \beta \mu q} \quad \text{Eq. 16}$$

$$t_D = \frac{\lambda k_f t}{(\phi_f c_{tf} + \phi_m c_{tm}) \mu r_w^2} \quad \text{Eq. 17}$$

$$\omega = \frac{\phi_f c_{tf}}{\phi_f c_{tf} + \phi_m c_{tm}} \quad \text{Eq. 18}$$

$$\lambda' = \alpha \frac{k_m}{k_f} r_w^2 \quad \text{Eq. 19}$$

Where c_{tf} and c_{tm} are the total compressibility of the fracture and matrix along with the compressibility of formation fluid, respectively, ω is the storability ratio which quantifies how much fluid is stored in the natural fracture, and $\alpha = \frac{A}{XV}$ where A is the surface area of the matrix, V is the matrix volume, and X is the characteristic dimension of the reservoir.

The dimensionless pressure drop for a dual porosity well is given by the Equation:

$$P_t = \frac{1}{2} [\ln(t_D) + 0.809 + E_i\left(\frac{\lambda' t_D}{\omega(1-\omega)}\right) - E_i\left(\frac{-\lambda' t_D}{1-\omega}\right)] \quad \text{Eq. 20}$$

At short times, the E_i function can be approximated as: $E_i(x) = \ln(1.781x)$. Making this substitution and combining natural logs yields:

$$P_t = \frac{1}{2} \left\{ \ln(t_D) + 0.809 + \ln \left[\frac{\left(\frac{\lambda' t_D}{\omega(1-\omega)} \right)}{\left(\frac{-\lambda' t_D}{(1-\omega)} \right)} \right] \right\} \quad \text{Eq. 21}$$

Canceling out like terms yields:

$$P_t = \frac{1}{2} [\ln(t_D) + 0.809 - \ln(\omega)] \quad \text{Eq. 22}$$

Replacing natural log with log and distributing the $\frac{1}{2}$ yields:

$$P_t = 1.1513 \log(t_D) + 0.4045 + \log\left(\frac{1}{\omega}\right)^{1.1513} \quad \text{Eq. 23}$$

For long production times, t_D becomes very large. This results in the quantities $\frac{\lambda t_D}{\omega(1-\omega)}$ and $\frac{\lambda t_D}{(1-\omega)}$ to become roughly equivalent. Thus $E_i\left(\frac{\lambda t_D}{\omega(1-\omega)}\right) \approx E_i\left(\frac{\lambda t_D}{(1-\omega)}\right)$. The E_i terms will cancel out at large times resulting in the Equation:

$$P_t = \frac{1}{2} [\ln(t_D) + 0.809] \quad \text{Eq. 24}$$

Again, replacing natural log with log and distributing the $\frac{1}{2}$ yields:

$$P_t = 1.1513 \log(t_D) + 0.4045 \quad \text{Eq. 25}$$

One can see that the only difference between the short time equation (Eq. 23) and the long time equation (Eq. 25) is the term $\log\left(\frac{1}{\omega}\right)^{1.1513}$. This term dictates the difference between the matrix and the fracture properties. When plotting P_t versus $\log(t_D)$ for a dual porosity reservoir, two distinct straight lines with the same slope will appear. These two lines will be separated horizontally by a certain distance. This distance is determinant on how different the matrix and fracture properties are, i.e. the ω term. The larger the distance between the two lines the larger the difference in properties. When ω is equal to one, then $\log\left(\frac{1}{\omega}\right)^{1.1513}$ will go to zero. This means that the early time and late time equations will no longer have any differences, and the

natural fracture response of the well test will no longer be evident. This makes sense when we recall the equation for ω .

$$\omega = \frac{\phi_f c_{tf}}{\phi_f c_{tf} + \phi_m c_{tm}}$$

If the porosity of the matrix goes to zero, then $\omega = \frac{\phi_f c_{tf}}{\phi_f c_{tf}} = 1$. This indicates that all activity of the reservoir occurs within the fractures, and thus the ‘fractures’ are the matrix. Thus, there truly are not any fractures. ω being equal to one is an indication that one is in a conventional reservoir that has uniform properties of ϕ , μ , c_t , h , etc. A dual porosity reservoir semi-log plot given by Warren and Root is shown below in Figure 11.

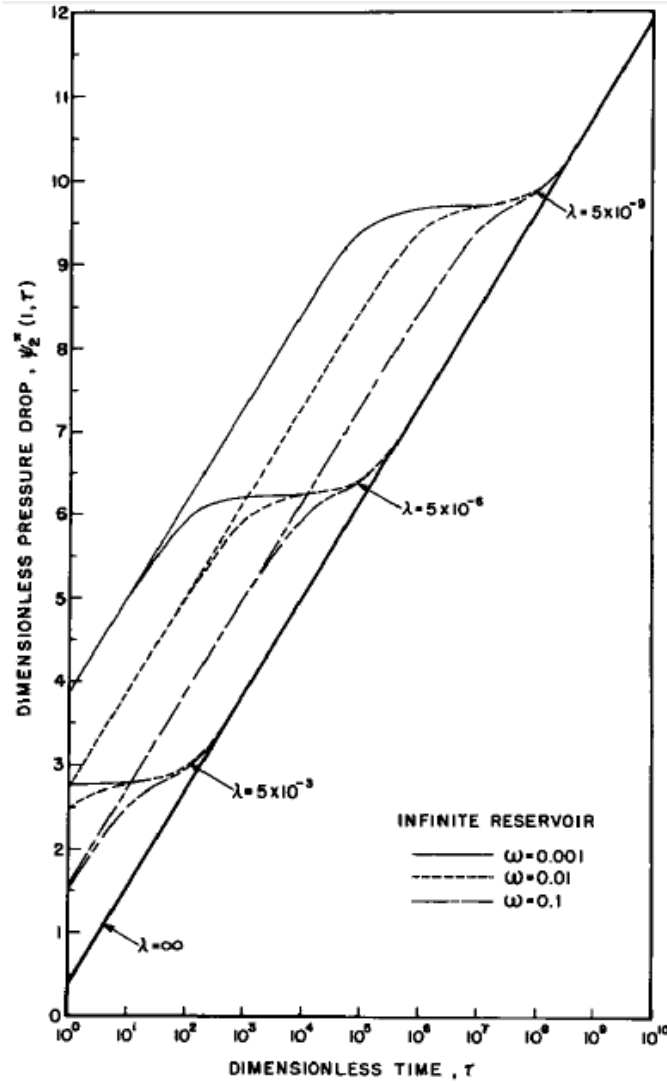


Figure 11: Dual Porosity semi-log plot (Warren & Root, 1962).

2.3. Reservoir Structure

Determining the reservoir structure was completed in a rudimentary fashion. Because the infinite acting solution was the target of the well test, the reservoir drainage area had to be made especially large. The reservoir was chosen to be a 5100' by 5100' reservoir with varying thickness. The area of this reservoir was $2.601 \times 10^7 \text{ ft}^2$, or 597.1 acres, much larger than a

typical dry gas well drainage area, which typically varies from 30 to 320 acres (Jenkins, DeGolyer and MacNaughton, & Boyer II, 2008). Additionally, the individual grid block size had to be selected. In order to have a more accurate representation of a reservoir, the grid block size should be made as small as possible. An infinitely small grid block will create an infinitely small error in the calculation, but will lead to an infinitely large calculation time. In order to determine the size of the grid block, a single-lateral well was placed in the center of the 597.1 acre reservoir. All reservoir and wellbore parameters were kept constant (including production time) and the only parameter that was changed was the grid block size. The minimum grid block attempted was 10 feet, which was not possible to run due to the large number of computations and lack of computing power. It should be noted that the goal of selecting the best grid block size is not simply to have the smallest grid block but to also have a relatively fast simulation. The smaller the grid block, the longer the simulation will run for. Thus, it was a goal to find a grid block that was small enough to give an accurate representation of the reservoir but was large enough that the simulation did not take a substantial amount of time to run. Table 1 below shows corresponding grid blocks and their respective time of simulation. Table 2 displays the reservoir and well inputs that were kept constant while the grid block size was varied. Figures 12-15 show how the predictive capabilities of the simulator changed with grid block size.

Table 1: Grid Block size and Corresponding Time of Simulation

Grid Block Size (ft)	Time of Simulation (s)
10	∞
20	445
30	295
40	175.72
50	168.8
80	61.02
100	7.35
200	1.95

Table 2: Reservoir Parameters for grid block size comparison

Reservoir Parameter	Value
ϕ_m	20%
ϕ_f	10%
k_m	.01 md
k_f	.1 md
h	150 ft
FS	7.5 ft
P_i	3000 psi
P_{wf}	300 psi
T_{res}	150°F
Wellbore Length	2000 ft
Lateral Length (single)	1000 ft

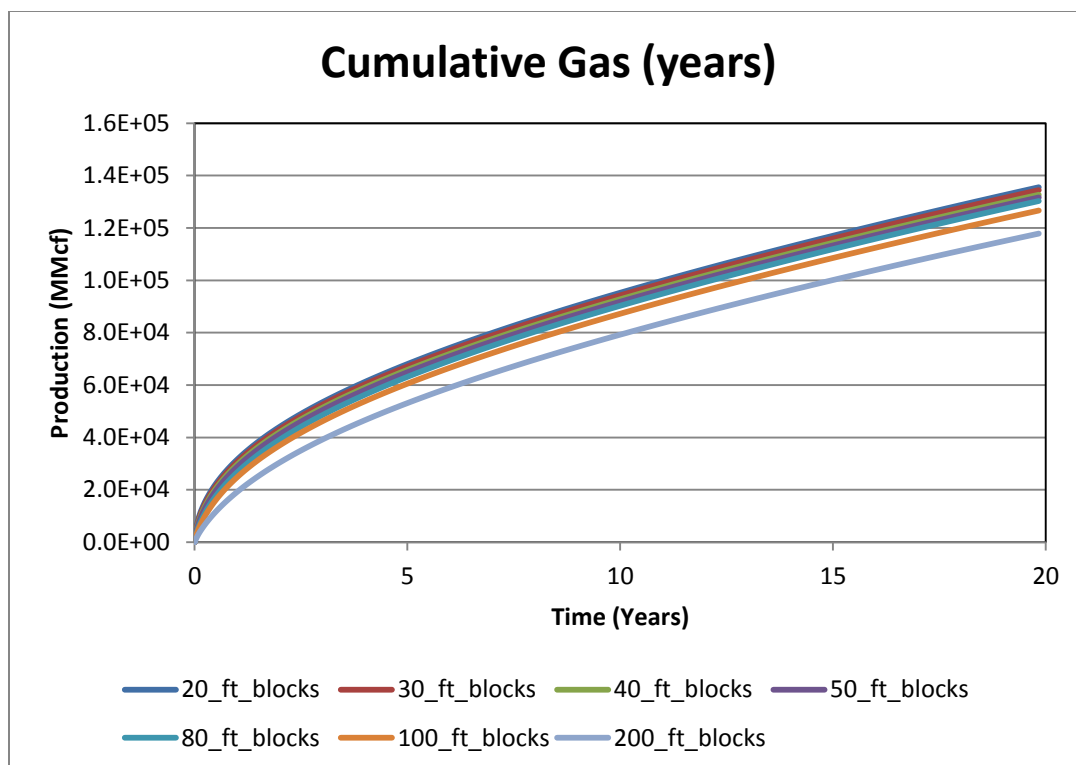


Figure 12: Cumulative Gas as a function of Grid Block Size over 20 year span. Drawdown test at 300 psia.

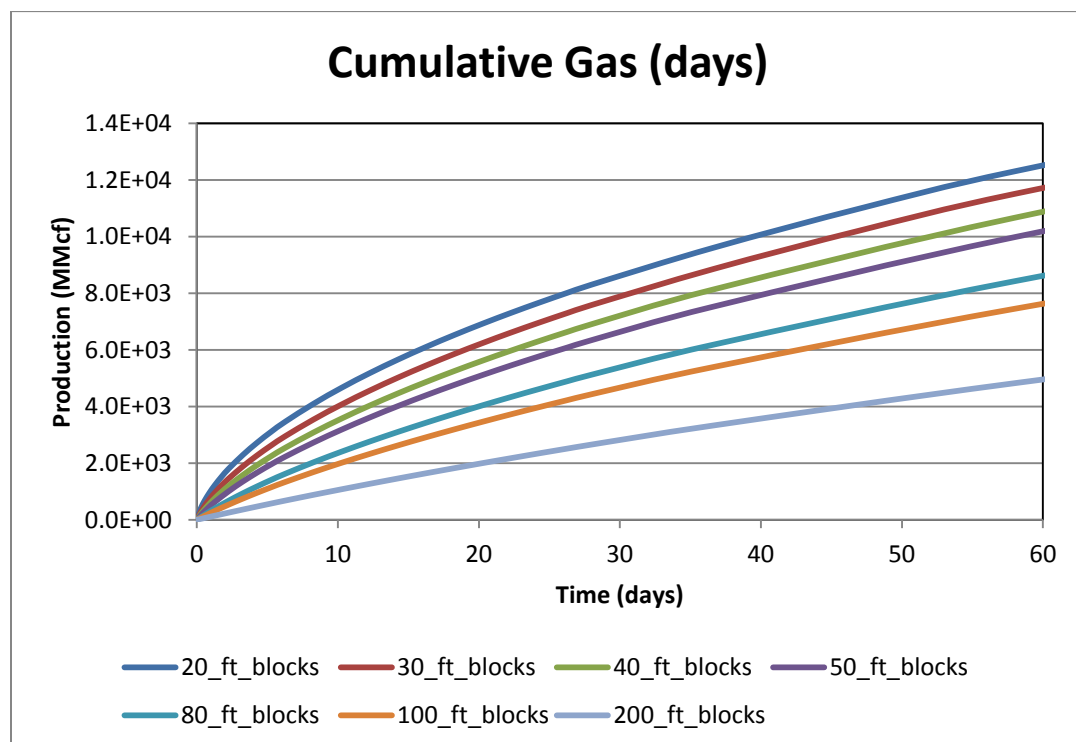


Figure 13: Cumulative Gas as a function of Grid Block Size over 20 day span. Drawdown test at 300 psia.

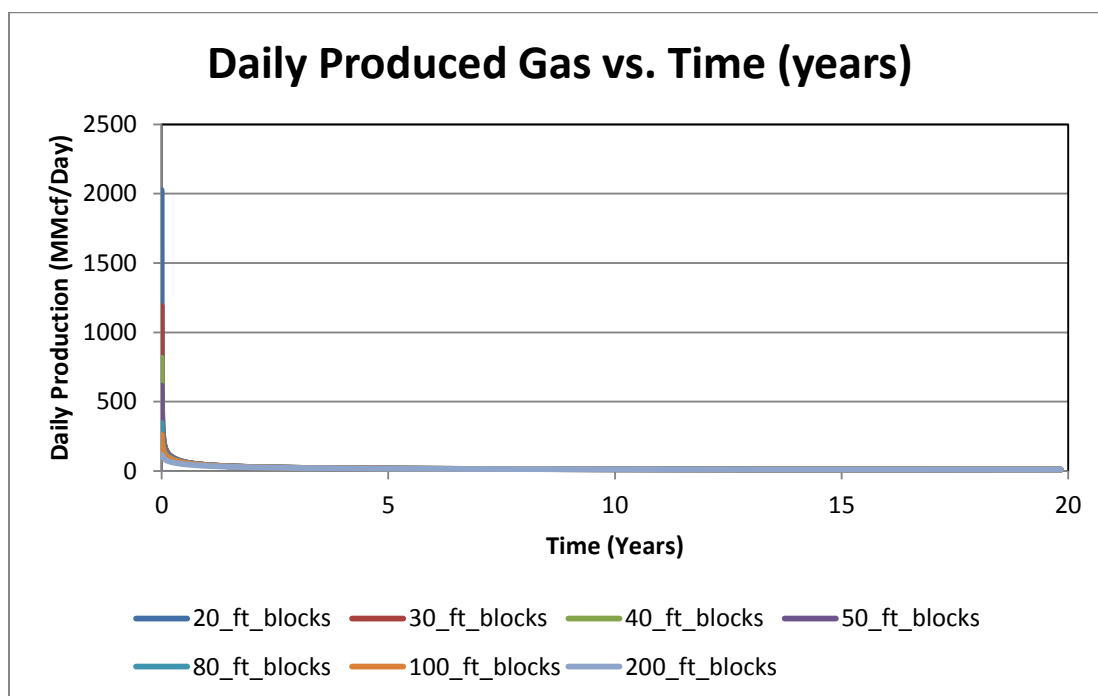


Figure 14: Daily Production as a function of grid block size, over 20 year span. Drawdown test at 300 psi.

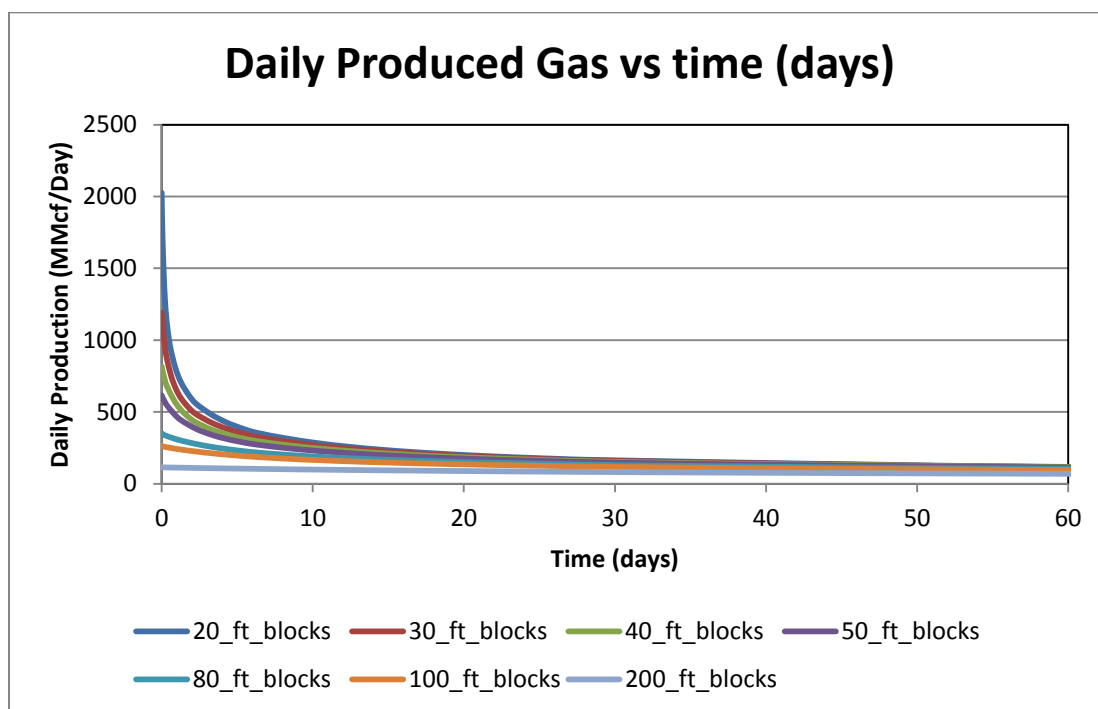


Figure 15: Daily Production as a function of grid block size, over 20 day span. Drawdown test at 300 psi.

When analyzing the simulation results over the entire 20 year range, it is difficult to see variance between the cases. Only the 200 and 100 foot grid blocks can be easily distinguished in Figure 12 as having a different cumulative production. Thus, the 200 and 100 foot grid blocks are eliminated as possible candidates in this study. For grid blocks smaller than 100 feet the percentage difference between cases is relatively small. The percentage difference in cumulative production from the 20 foot to the 80 foot blocks is 3.90% and the difference between the 20 foot and 30 foot blocks is only 0.824%. Both of these percentage differences may seem adequate, but these values were calculated at the end of 20 years whereas our data will be taken for only 60 days. It is important to also look at the how grid block size affects the early time region as well.

As mentioned above, because the drawdown test is being run for 60 days, variance between the grid blocks' performance within the first two months was significantly important. Analysis of Figures 13 and 15 showed that there was some distinction in all of the predicted values of the simulation when the block size was varied. It is assumed that the smallest grid block is giving the most accurate results. However, recall that it is ideal to have a simulator that is both accurate and fast. When comparing times in Table 1, one can see that the 20 foot case took nearly 450 seconds to complete whereas the 30 foot case only took about 300 seconds to complete. This shows that the 30 foot blocks are on average a third faster than the 20 foot case. Moreover, when comparing the cumulative production within the first 60 days, the percent error between the 20 foot and 30 foot cases was calculated to be 5.40%. Thus, the accuracy lost by switching to a coarser grid block size is around 5.5% but the speed gained is more than 30%. Because there will be thousands of trials ran, it was determined that the 30 foot blocks would be the best compromise.

This setup would still produce adequate results but would be more than 30% faster than the most accurate model.

2.4. Reservoir Parameters

Next, the reservoir parameters' ranges had to be selected. Past theses, journal articles, and simulation results were all used to aid in this process. The goal of this research was to create an artificial neural network that could accurately predict reservoir properties in an inverse manner for all tight gas reservoirs. This creates large ranges of reservoir parameters such as permeability, porosity, thickness, initial pressure, and fracture spacing but also leads to large ranges of well properties such as flow rate and wellbore length. The larger a range of input/output data, the more difficult it is for the ANN to be trained. This being said, it was vital that the ranges used were not only suitable for a tight-gas dual porosity reservoir, but also were shown to have an impact on the results; an example being the low permeability range of the matrix. It is known that tight gas reservoirs can have permeabilities in the nano-darcy range. However, it could be possible that once the matrix permeability reaches the nano-darcy range (i.e. 1-100 nano-darcy) even a large percentage change in value will not yield a large difference in how the reservoir behaves. Thus, it was not only important to find adequate ranges reported in the literature, but to also test these ranges in the simulator to verify what range of values truly affects the reservoir's performance. This will help to scale down the ranges tested and make constructing an accurate ANN a much simpler process.

Table 3: Ranges of data found in the literature² and used in the simulator.

Reservoir/Well Parameter	Trial Ranges	Default Value	Final Ranges
FS	.1-200 ft	1 ft	1-200 ft
k_m	$10^{-8} - 10^{-1} \text{ md}$	$1 * 10^{-6} \text{ md}$	$10^{-8} - 10^{-1} \text{ md}$
k_f	$1 * 10^{-5} - 5000 \text{ md}$	0.1 md	.1-250 md
h	5-300 ft	50 ft	50-300 ft
P_i	500-8500 psi	4000 psi	1500-8500
P_{wf}	300	300	-
T_{res}	100-500°F	200°F	100-500°F
ϕ_m	.5-15%	5%	6-15%
ϕ_f	.5-7%	3%	.5-3%
q	10Mcf-50MMcf	1 MMcf	1 MMcf-50 MMcf
Wellbore Length	1000-4000 ft	3000 ft	1000-4000 ft

2

The ranges selected for the dual porosity tight gas reservoir are listed in Table 3 above. These ranges were then further explored using the commercial software package. The simulator was operated with both constant production and constant flowrate, and the results have been plotted in Figures 16-27, and Figures 28-38 respectively below. From the simulator results as well as discussions with faculty, the final ranges were selected, which are displayed in the final column of Table 3.

² (Artun, 2008), (Bodipat, 2011), (Bustin & Bustin, 2012), (Hyun Ahn, 2012), (Jenkins, DeGolyer and MacNaughton, & Boyer II, 2008), (Josh, et al., 2012), (Khattirat, 2004), (Nagel, Sanchez-Nagel, Zhang, Garcia, & Lee, 2013), (Siripatrachai, 2011), (Toktabolat, 2012), (Walton & McLennan, 2013)

It should be noted what is truly meant by the term ‘initial pressure’. The initial pressure is measured at the wellbore, and is thus a BHP. The simulator does not take into account a pressure gradient within the wellbore. Thus, each section of the wellbore has the same pressure. The simulator then sums up the flowrate for each section of the wellbore. This sum is then displayed to the user as the total flowrate of the well. Note that the well is considered to have an open wellbore completion.

2.4.1 Drawdown Test Analysis

The fracture spacing (FS) does affect the drawdown curve, as expected. When the fractures are close together the reservoir will be able to drain from a larger area more effectively. However, whenever the fractures are further apart, there is consequently less fractures within the reservoir, and this will limit the reservoir’s transportability of hydrocarbons. This decrease in transportability means that around the wellbore, the fluid being taken away will not be able to be replaced as quickly. This will consequently cause the pressure at the wellbore to decrease faster. One can see that over three orders of magnitude of FS, from 0.1 to 100 feet, results in a change in pressure of about 10 psi after one year. Thus, for the collective reservoir parameters used in this study, the fracture spacing will cause the PT data to change, but not drastically. Even when comparing the pressure difference between the 0.1 and 200 foot cases, there is a difference in pressure of less than 150 psi over the one year period. Moreover, it is difficult to distinguish difference between the cases when fracture spacing is between 0.1 and 50 feet at one year’s time. Differences at small time values can be seen, and it is proposed that the inverse tool created in this study will be able to use these differences to help distinguish between reservoirs with varying fracture spacing.

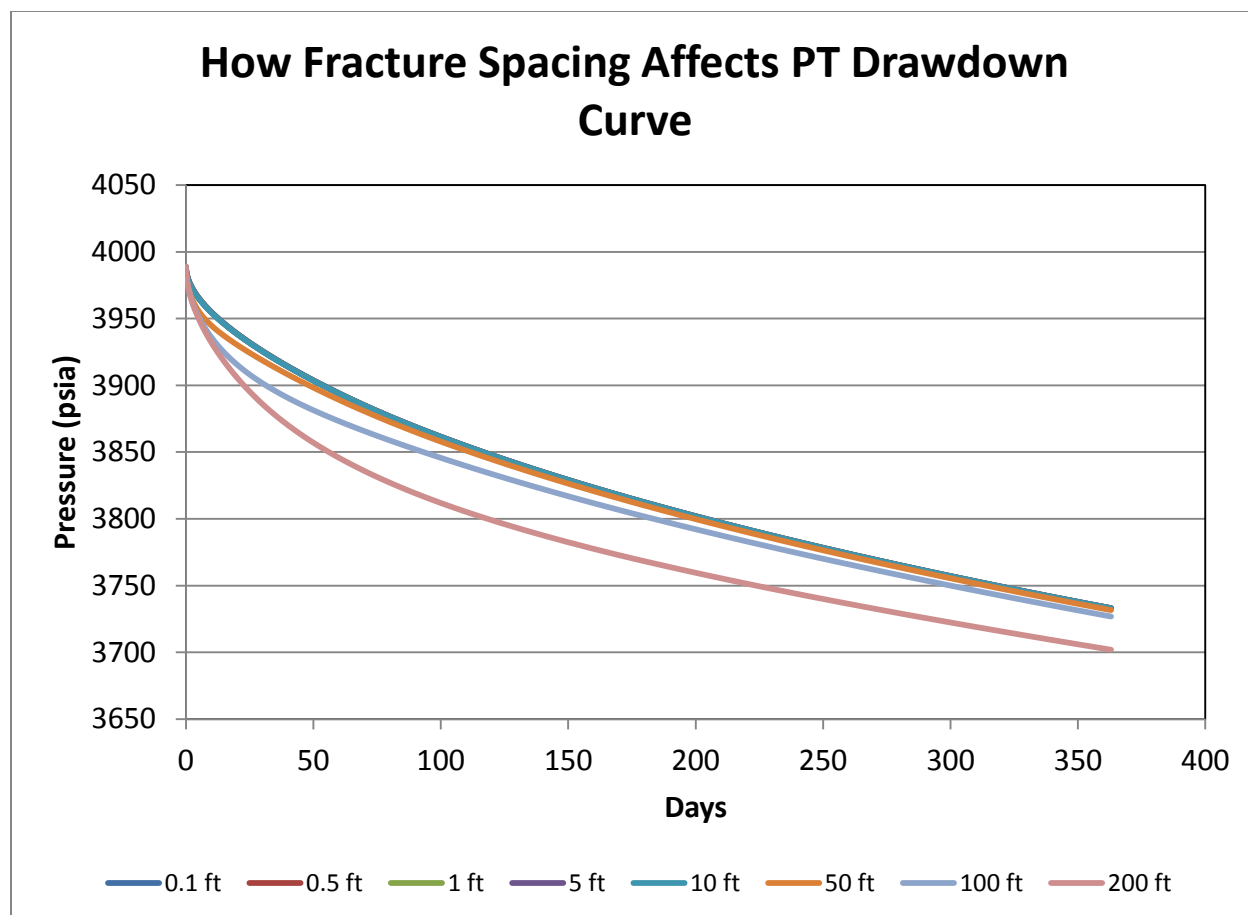


Figure 16: Plot demonstrating how fracture spacing affects the Pressure Transient Drawdown Curve.

Figure 17 displays how reservoir thickness affects the PT drawdown curve. One can note that the thinner the reservoir, the higher the pressure drop at the wellbore. This is due to the smaller volume of the thinner reservoirs coupled with the constant flowrate for the test for all reservoirs. Note that the difference between the five foot and 300 foot cases is more than 2500 psi after one year. This indicates that the reservoir's thickness will have a larger impact on the PT curve than the FS, for this particular reservoir. The thickness of the reservoir can be determined from more conventional methods other than well test analysis such as well logging. Thus, it may be beneficial to use the thickness of the reservoir as an input parameter into our neural network

rather than as an output parameter. It should also be noted that distinct differences are easily seen for reservoirs under 100 feet thick. Thicknesses at or above 100 feet are difficult to distinguish when testing for only a year. Our drawdown test will only be implemented for 60 days. Thus, it is hypothesized that our ANN may have difficulties distinguishing between reservoirs with thicknesses greater than 100 feet, but will be better at characterizing thinner reservoirs.

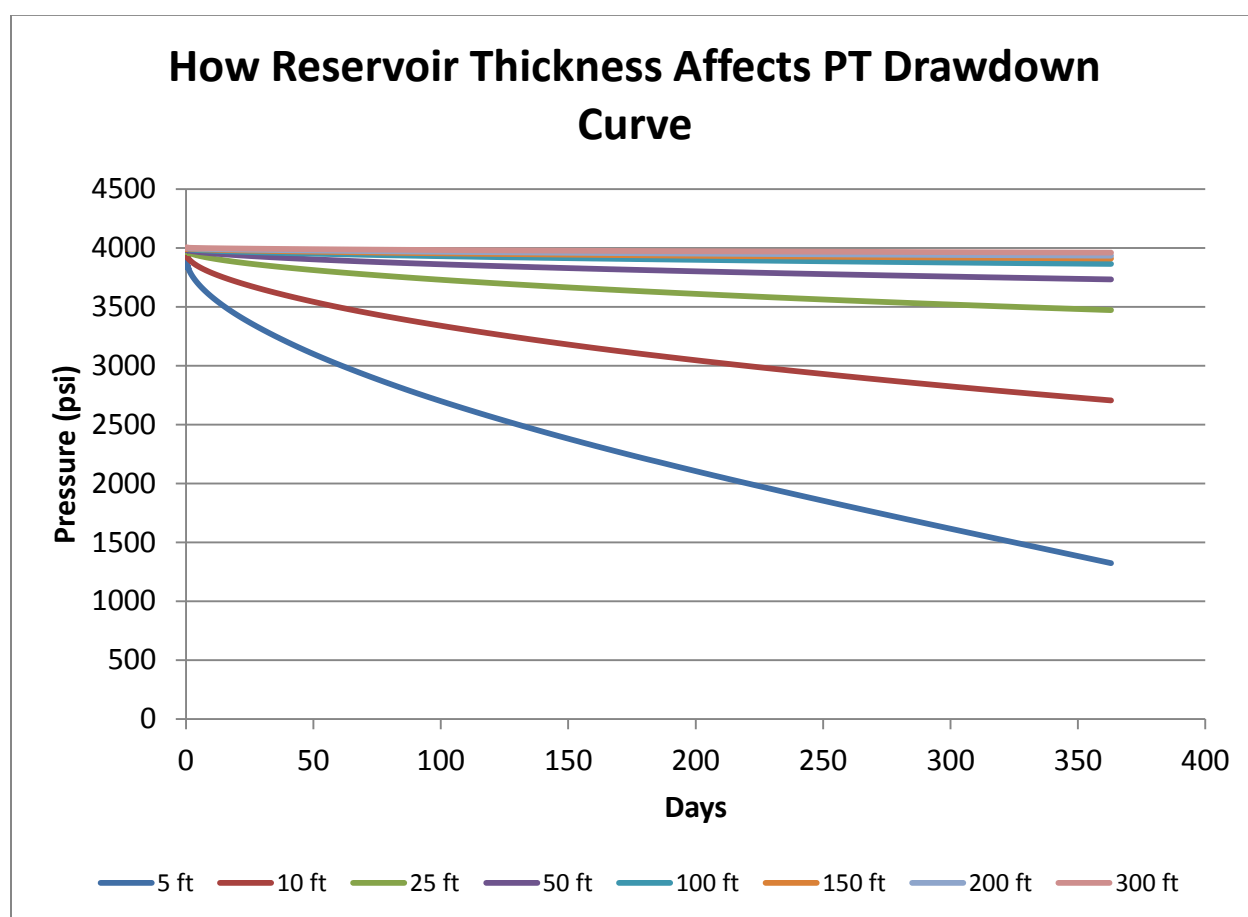


Figure 17: Plot demonstrating how reservoir thickness affects the Pressure Transient Drawdown Curve.

Figures 18 and 19 display how matrix and fracture porosity, respectively, affect the PT drawdown curve. The matrix porosity range of 0.5% to 15% causes the curve's shape and endpoints to change: after one year there is roughly a 225 psi change in pressure. The fracture porosity range also causes the curve's shape and endpoints to change, resulting in a difference of roughly 100 psi at the endpoints. The change in porosity of the matrix or the fracture cause similar results. One interesting note is for low porosities. Ignoring the 0.5% data, the porosity was increased by 2% from 1% to 15% for the matrix and 1% to 7% for the fracture. However, the difference in the curves is not uniform. The lower the porosity, the larger the difference in the curves as porosity is increased. This indicates that the ANN should be more efficient at predicting low matrix and fracture porosities because there is a larger distinction between these curves compared to that of the higher porosities.

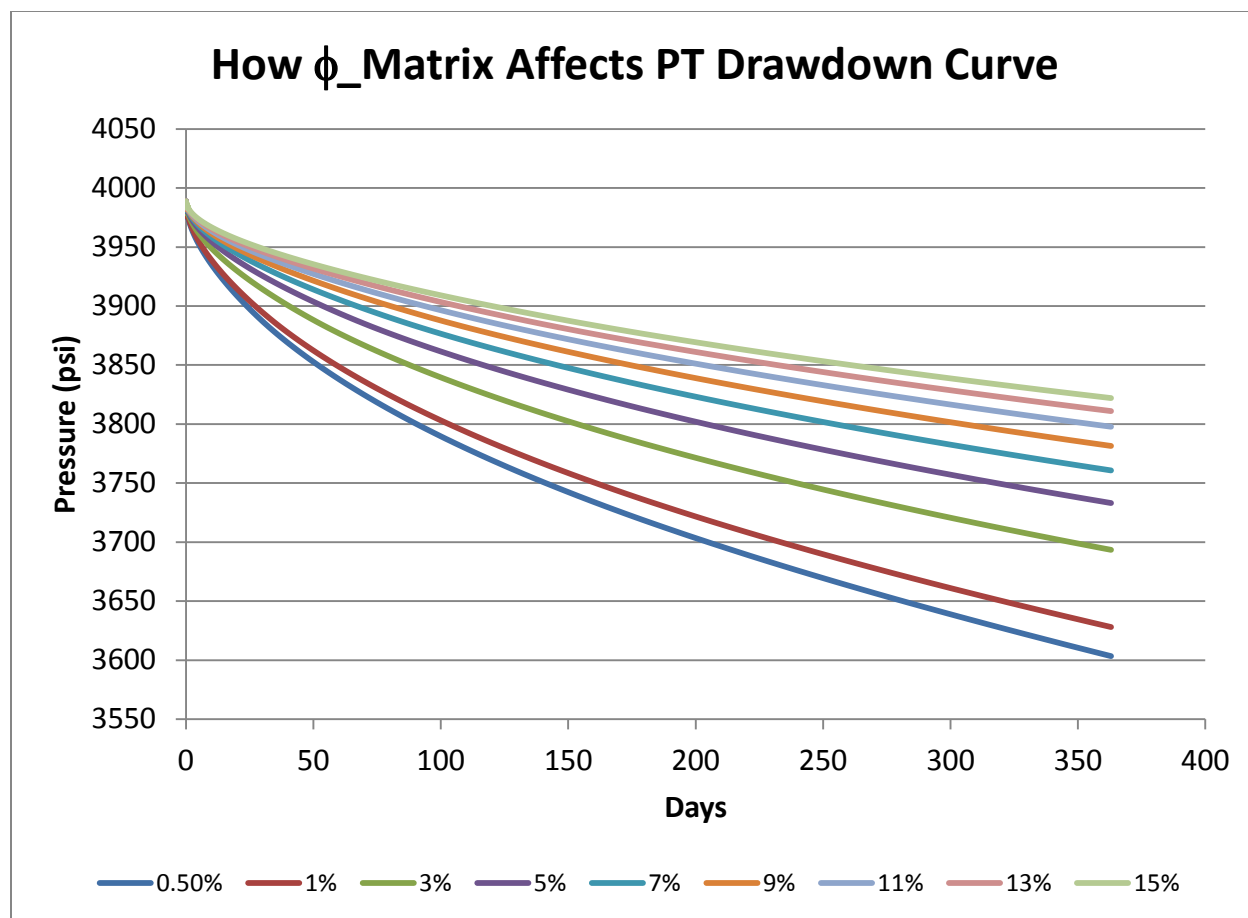


Figure 18: Plot demonstrating how matrix porosity affects the Pressure Transient Drawdown Curve.

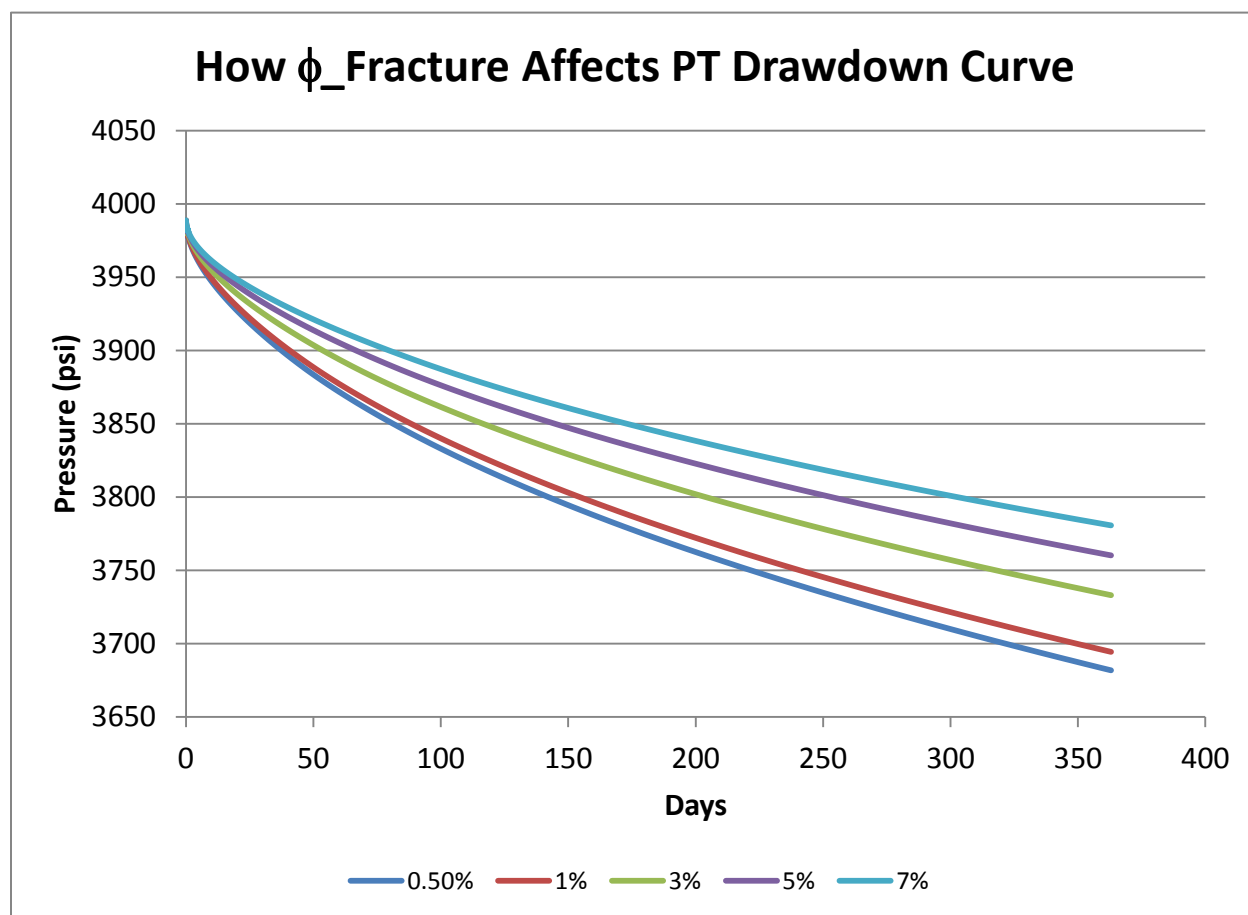


Figure 19: Plot demonstrating how fracture porosity affects the Pressure Transient Drawdown Curve.

Figure 20 represents how the reservoir temperature affects the PT drawdown curve. For our analysis we are assuming 100% natural gas, methane, production. Typically when the temperature is increased this will cause heavier elements in the hydrocarbon mixture to volatize, and this would cause the production to vary. However, in this case the hydrocarbon mixture is not a mixture, it is only one element. The average reservoir temperature range will cause the density of the gas to change, however these changes do not translate into distinct changes in the drawdown curve.

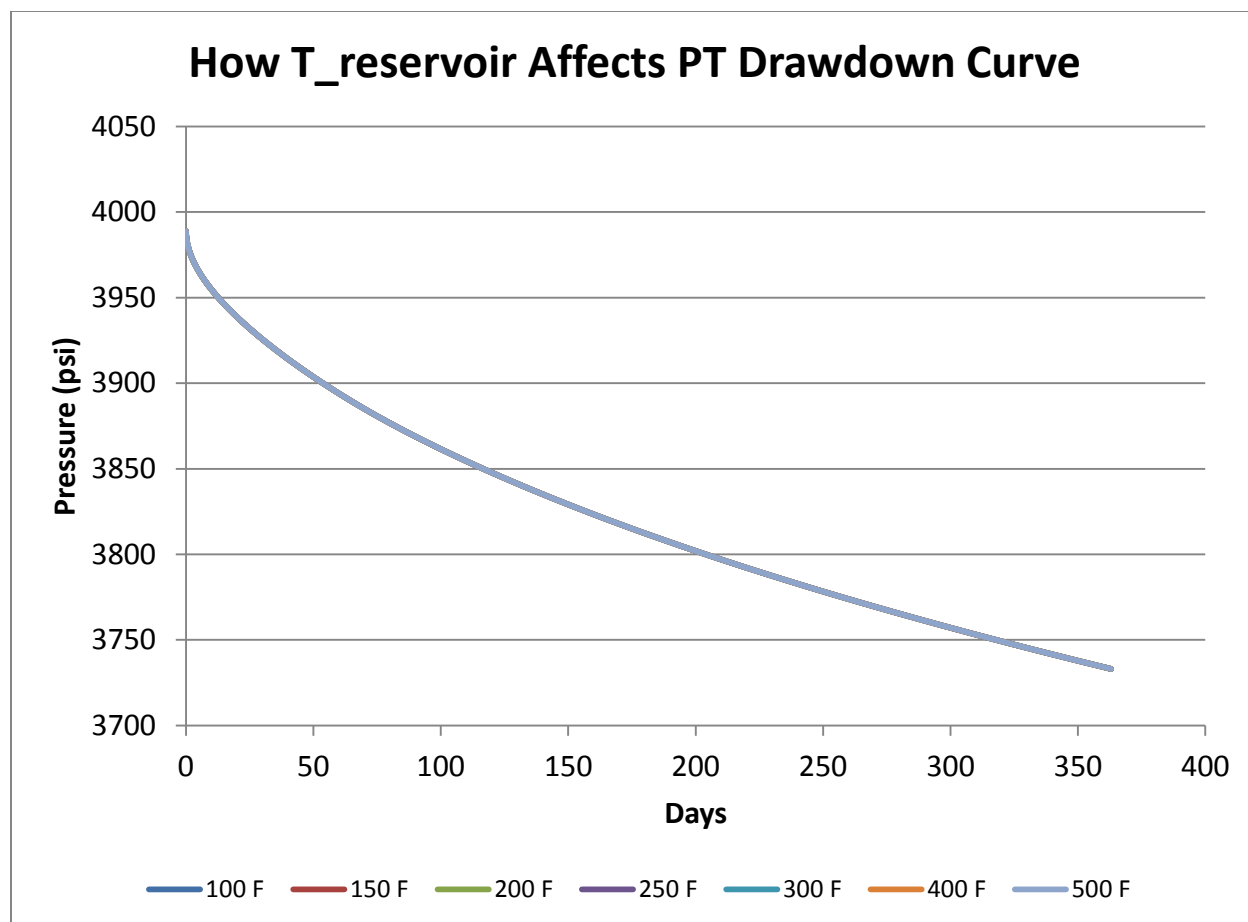


Figure 20: Plot demonstrating how Reservoir Temperature affects the Pressure Transient Drawdown Curve.

Figure 21 displays how the length of the main wellbore affects the PT drawdown curve. The wellbore length was varied by 500 foot intervals and one can see a difference of approximately 500 psi between the shortest and longest wellbore lengths after one year of constant production. The shape and endpoints of the curves change as the wellbore length increases. Again, starting with a smaller value of wellbore length and incrementally increasing it shows that there is a larger change with the shortest wellbore length and the smallest change with the longest wellbore lengths. As the wellbore length is increased, the wellbore connectivity is also increased. A larger wellbore connectivity will result in a smaller pressure drop at the wellbore. Moreover, increasing

the wellbore length from 3000 to 3500 feet will proportionally increase the connectivity by a smaller amount than increasing the wellbore length from 1000 to 1500 feet. The change in length for both cases is 500 feet, however the increase in length is $\frac{3500-3000}{3000} = \frac{1}{6}$ for the former case and $\frac{1500-1000}{1000} = \frac{1}{2}$ for the latter. Thus, this larger difference is what accounts for the larger variance in the PT drawdown curves at shorter wellbore lengths. Note that the wellbore length will always be assumed to be a known property for the inverse tool, and will never be a predicted property of the ANN. However, it is still important to see how the well structure will affect the PT curve.

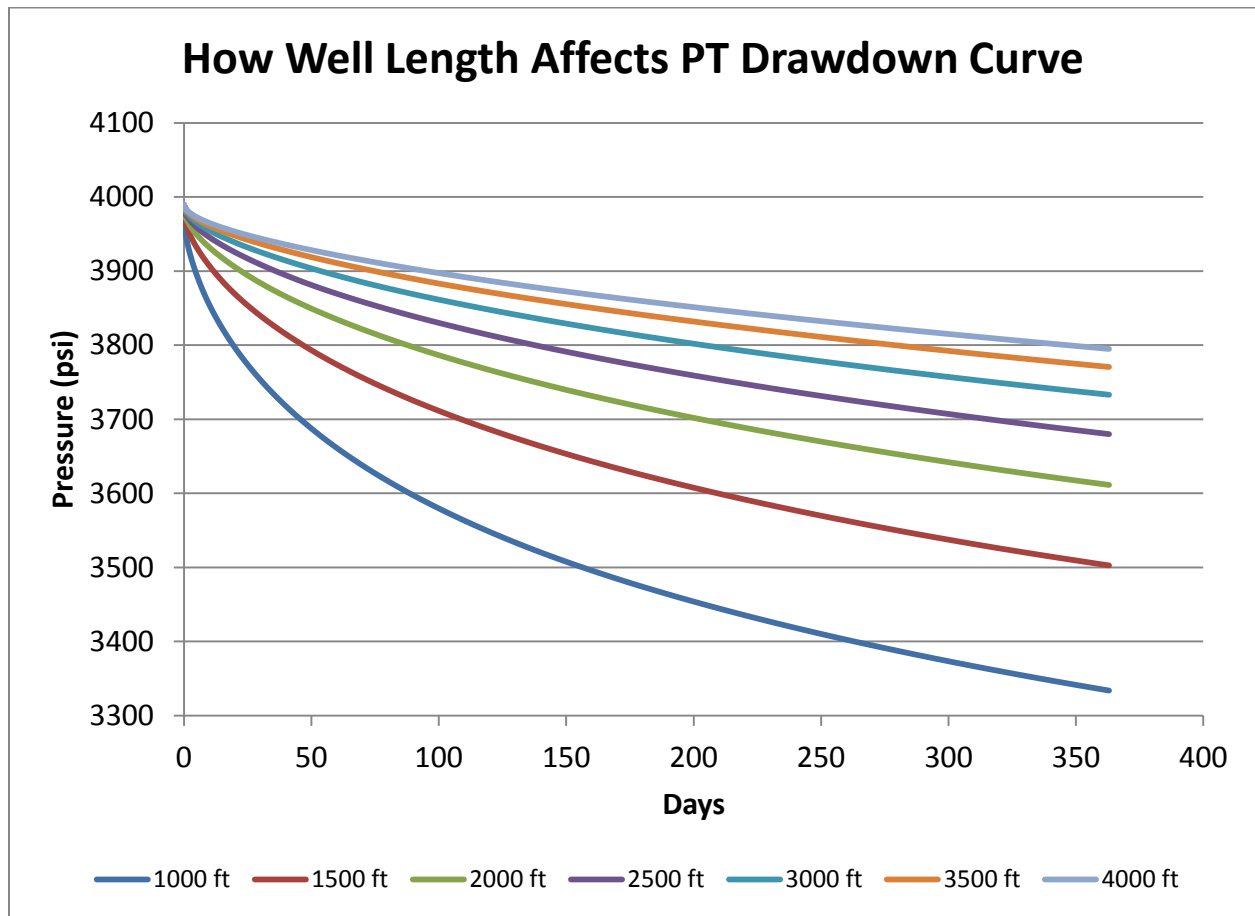


Figure 21: Plot demonstrating how wellbore length affects the Pressure Transient Drawdown Curve.

Figures 22 and 23 represent the PT drawdown test in two separate ways. Figure 22 plots the BHP versus time whereas Figure 23 plots the derivative of BHP versus time. One can see that as the pressure is increased from 500 to 8500 psi, the PT curve is shifted up. However, the general shape of the PT curve appears to be consistent between all cases. In order to prove this, the derivative of the BHP was plotted versus time in order to detect any small changes in the pressure. Figure 23 shows that for all pressure values over 1000, the PT curve decreases at the same rate. The 500 psi trial does not follow this trend because the BHP went to zero prior to the end of the drawdown test. This caused the spike in the derivative plot. From analysis of the derivative plot, one can see that changing the initial pressure causes only the vertical placement of the curve to change. Knowing this will affect how the ANN is designed. Later functional links will be discussed. Functional links will help the ANN map the given input data to the given output data, essentially increasing the training abilities of the ANN. The results of Figures 22 and 23 seem to indicate that the initial pressure will not be a valid functional link for mapping the entire PT curve. However, it may be advantageous to use the initial pressure to aid in mapping the initial point on the PT curve. Note that because we are performing a pressure drawdown test, the initial pressure will always be an input for the inverse solution and will never have to solely be predicted by the ANN.

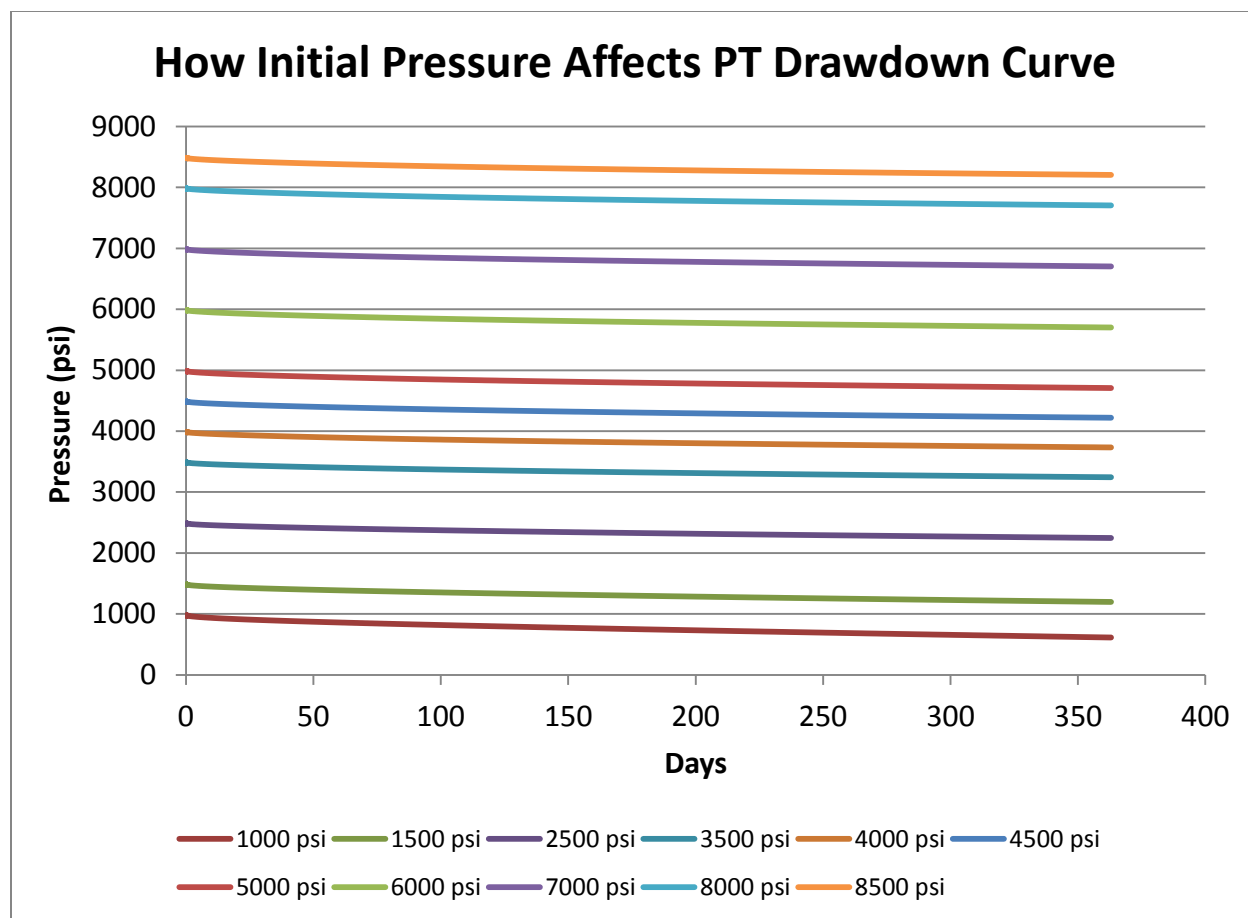


Figure 22: Plot demonstrating how Initial Pressure affects the Pressure Transient Drawdown Curve.

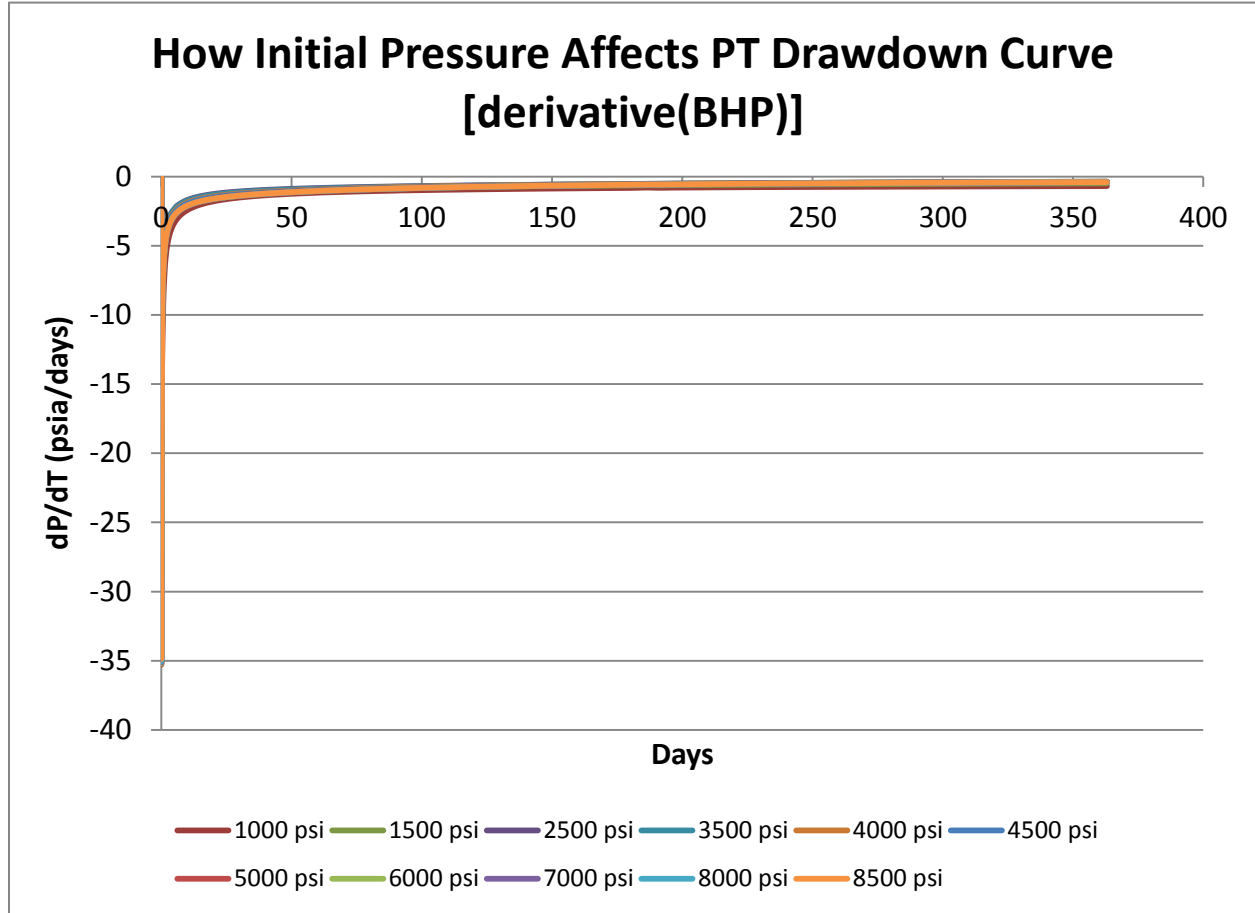


Figure 23: Plot demonstrating how initial pressure affects the derivative of BHP curve.

Figures 24 and 25 below show how matrix permeability (k_m) affects the PT drawdown curve. Note that Figure 24 shows the PT curve for reservoirs with fracture spacing equal to one foot whereas Figure 25 shows the PT curve for reservoirs with fracture spacing equal to 200 feet. Note that with smaller fracture spacing k_m does not affect the PT drawdown curve over the entire studied range of $1 * 10^{-8}$ to $1 * 10^{-1}$ md. Because the fracture spacing is so close the reservoir will be primarily dominated by the fracture's properties. Thus, the matrix permeability will have a limited impact on the PT curve. However, once the fracture spacing is increased, the

fracture properties no longer will dominate the reservoir's characteristics. When varying k_m over the same range we see that high permeability cases ($1 * 10^{-1}$ md to $1 * 10^{-2}$ md) will produce the same PT curve as we saw in Figure 24. However, at $1 * 10^{-3}$ md a difference in the PT curve is detected. Finally, the range from $1 * 10^{-4}$ to $1 * 10^{-8}$ md all lie on the same curve, but produce a higher drop in BHP than the subsequent cases. Recall that in the dual porosity system the matrix acts as a source and the fractures will transport the fluids from the source to the wellbore. The larger drop in BHP for a lower permeability system arises because the matrix cannot supply the hydrocarbons to the wellbore as effectively as a high permeability system. Thus, the pressure at the wellbore will decrease faster than cases that have high matrix permeability. These two cases show the difficulties that can arise during history matching. There are many solutions to the same problem. This issue can still arise with an ANN inverse solution tool, but because the network will be trained from each data point, it is expected that the ANN tool will be able to distinguish between these 'duplicate' cases and give a more accurate representation of the reservoir being studied.

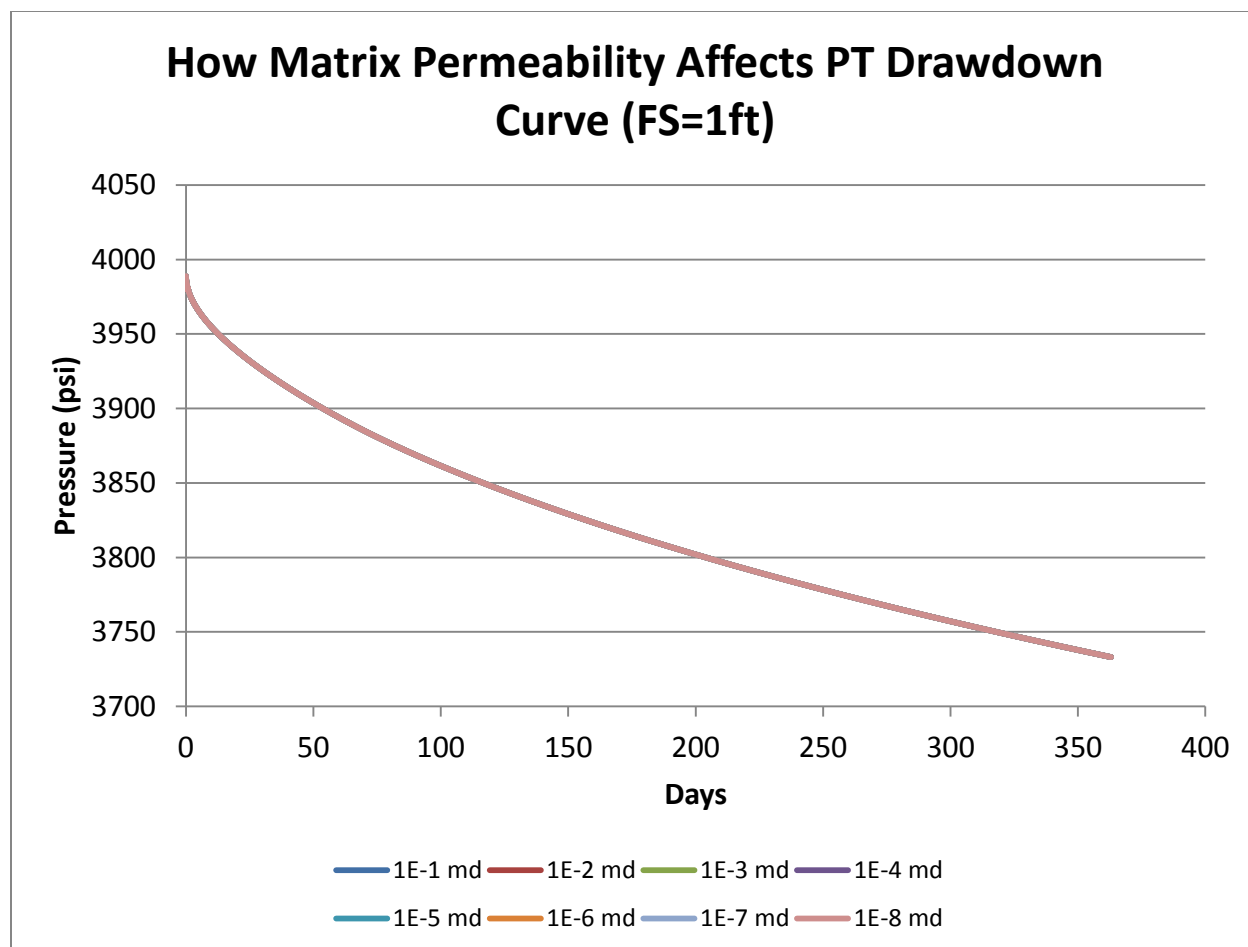


Figure 24: Plot demonstrating how matrix permeability affects the Pressure Transient Drawdown Curve (FS=1ft).

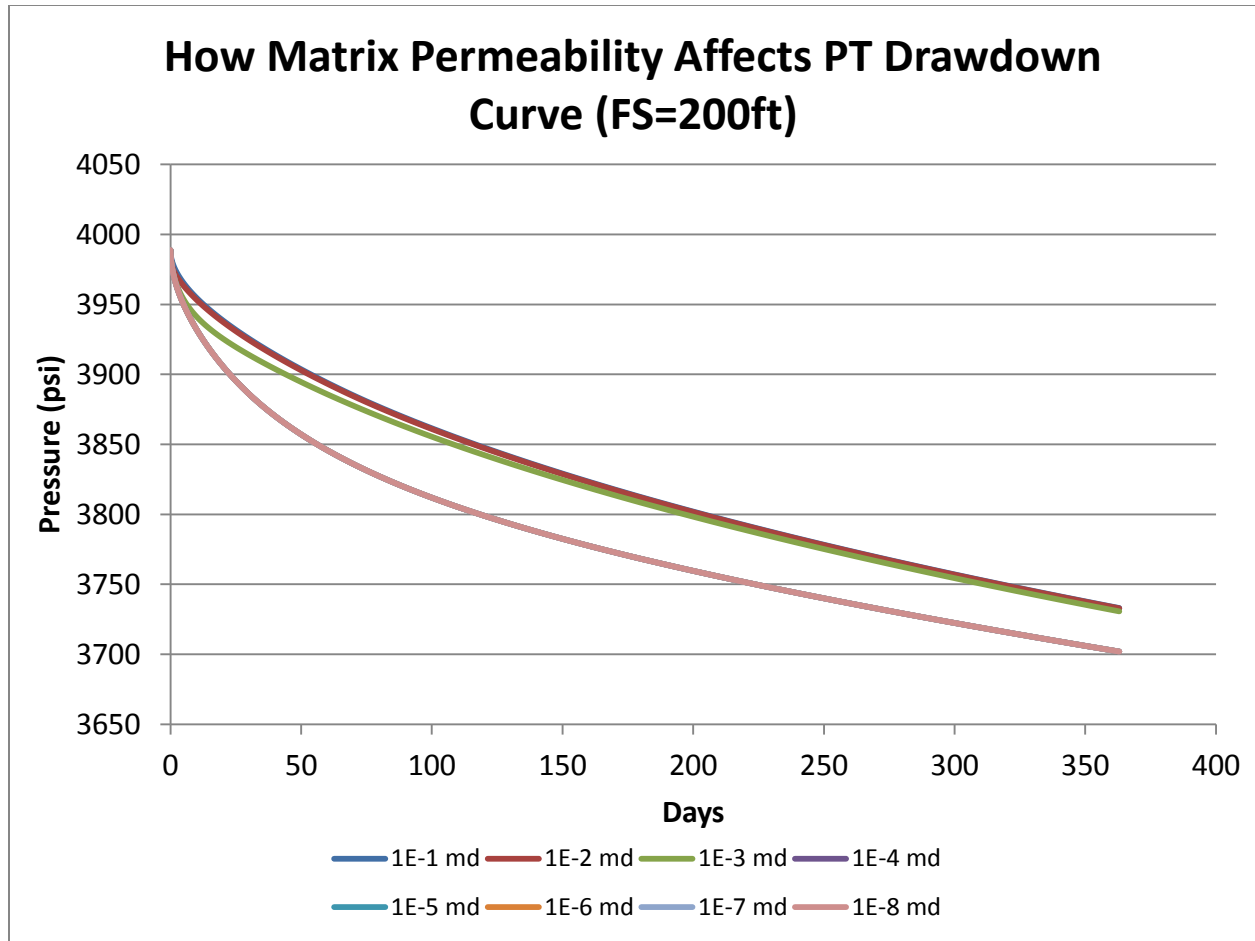


Figure 25: Plot demonstrating how matrix permeability affects the Pressure Transient Drawdown Curve (FS=200ft).

Lastly, Figures 26 and 27 show how the PT drawdown curve is affected by the fracture permeability (k_f). Note that Figure 26 shows the PT curve for reservoirs with fracture spacing equal to one foot whereas Figure 27 shows the PT curve for reservoirs with fracture spacing equal to 200 feet. As the fracture spacing is increased, the fracture's effect on the reservoir's behavior will decrease. Thus, in order to understand this phenomenon, both reservoirs' performance is shown graphically below. When k_f is less than 0.1 md, differences between the PT curves in both figures can be seen. Also note that the pressure drop for the reservoir with

fracture spacing equal to 200 feet is larger than the pressure drop for the reservoir with fracture spacing equal to 1 foot. This is the same phenomenon we saw when varying the fracture spacing for the k_m trials above.

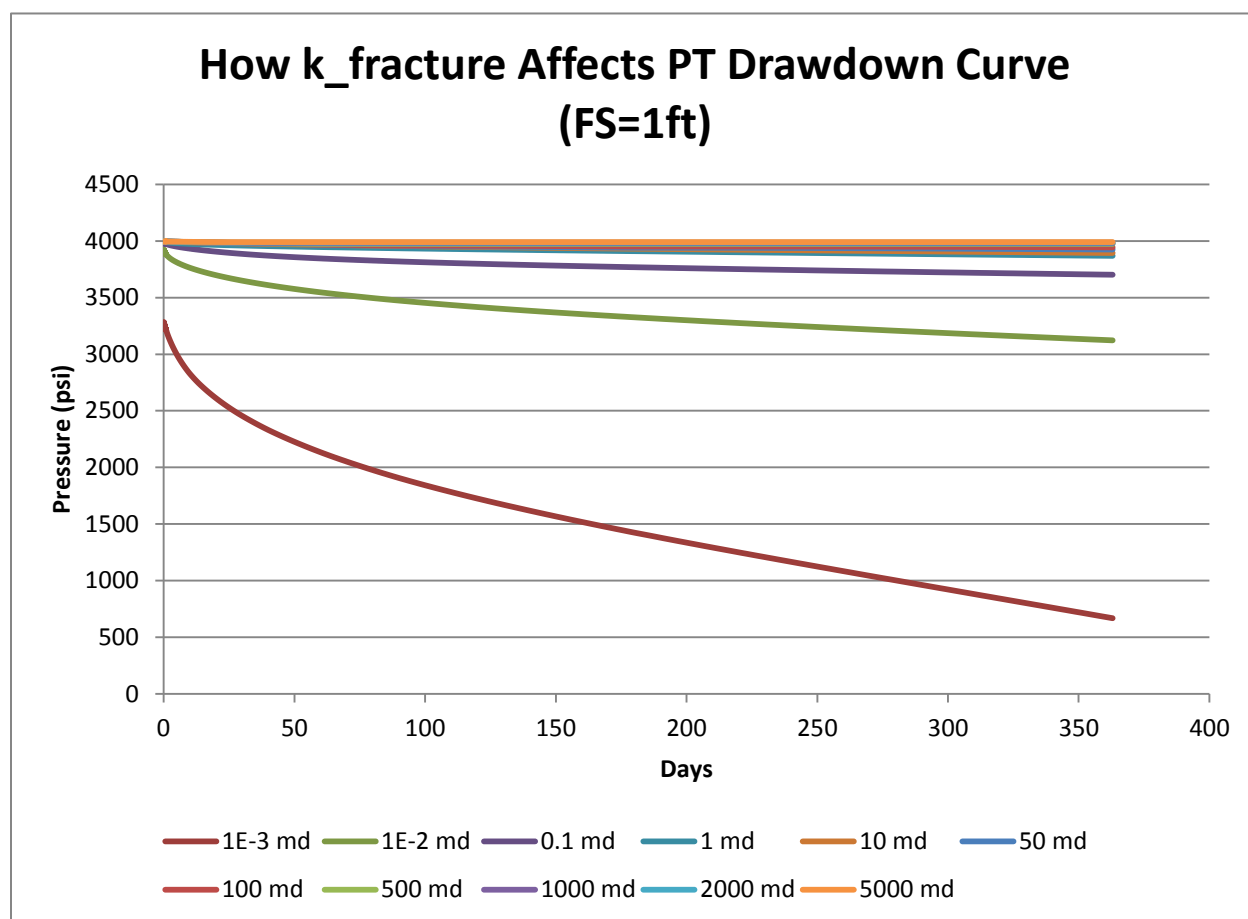


Figure 26: Plot demonstrating how fracture permeability affects the Pressure Transient Drawdown Curve (FS=1ft).

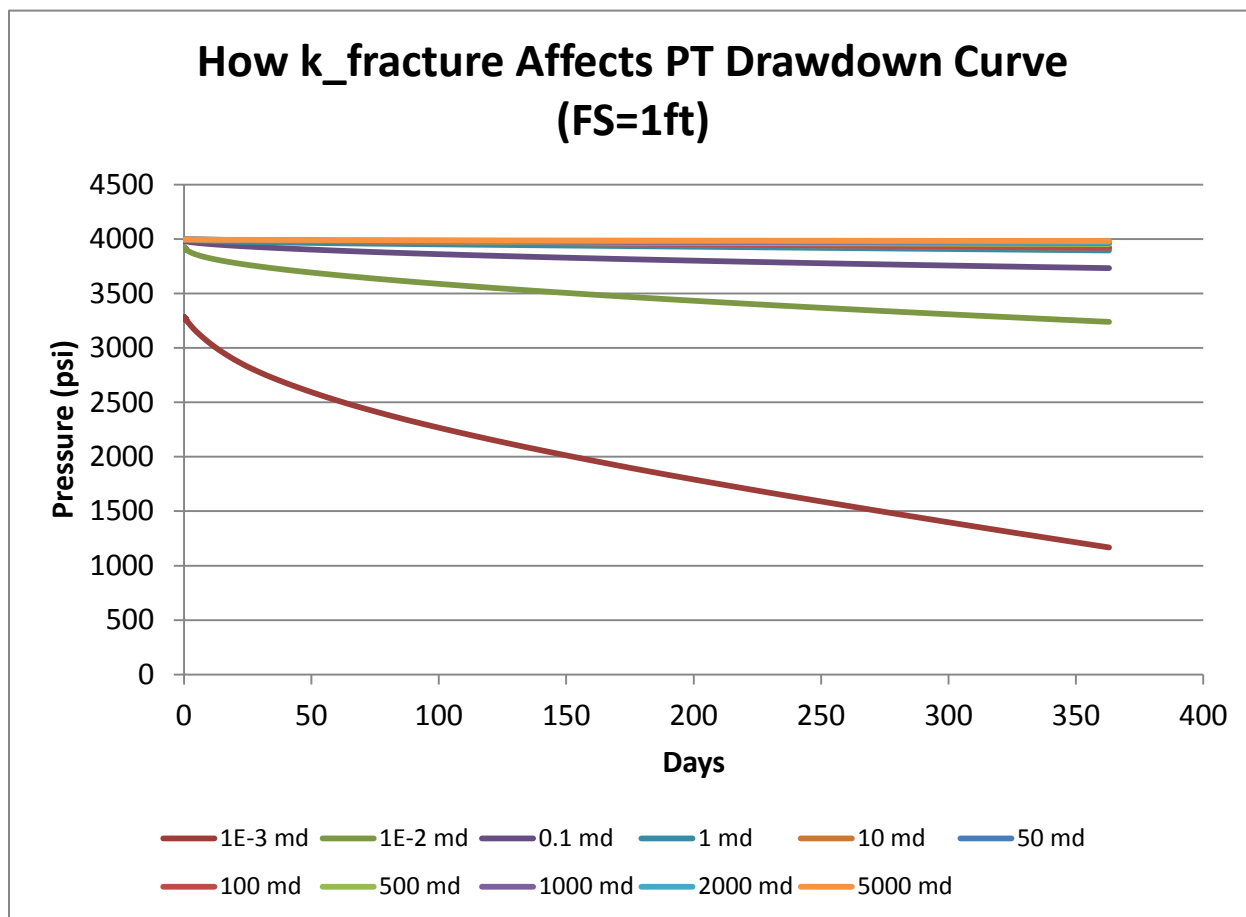


Figure 27: Plot demonstrating how matrix permeability affects the Pressure Transient Drawdown Curve (FS=200ft).

Lastly, the flowrate of the wells will surely change the shape of the PT curve. However, distinctions in the PT curve are not the only parameter that needs to be considered when it comes to deciding the range of production values to be studied. The costs of the multilateral wells suggested in this study would range from five to ten million dollars. Assuming the price of one SCF of natural gas to be four dollars, we can perform a rough economics assessment to determine the minimum flowrate of the wells. For example, when examining the lower limit of the flowrate range (10 MSCFD), we see that this well would only make \$40 per day. Assuming

the drilling and completing the well costs five million dollars, operational costs are zero, there is no interest, and the value of money does not depreciate over time, this well would take 125,000 days or 342.5 years to pay off. Assuming the well produced at a constant rate of 50 MMSCFD and that the above assumptions are valid, then the well would make \$200,000 per day, and it would only take 25 days to pay off the five million dollar well and 50 days to pay off the ten million dollar well. Thus, the upper limit is reasonable, but the lower limit still has to be established. Varying the production and quantifying how long it would take to pay off the well was completed for all flowrates within the range listed in Table 3. The flowrate chosen for the lower end of the production range was one MMSCFD. Using the same assumptions stated above, it was determined that this well would make \$4,000 per day. The well would be paid off between 1250 and 2500 days, or 3.42 and 6.85 years. For an expensive well, or a well with more laterals, this would be an extensive payback time. We did not want to have a longer payback time than this, thus this flowrate was chosen as the lower limit. The final flowrate range for this study will be one MMSCFD to fifty MMSCFD.

2.4.2 Cumulative Production Analysis

In order to get a better understanding of how reservoir and well properties affected production, a constant pressure drawdown test was also performed. For this test, it was assumed that the well was producing at its minimum BHP, 14.7 psia. The test was run for ten years, but over this long range it was difficult to notice changes in the plots. The plots only show the first 60 days of production, which is the same amount of time the constant production drawdown test will be run. Figures 28-38 display the results of this study. However, the reason for the change in production

is very similar to the reasons described above in Figures 16-27, and thus if more information is desired, please refer to the above figures.

Figure 28 displays how fracture spacing (FS) affects cumulative production. Note that there is not a large difference in the cumulative production after 60 days between the 0.1 and 10 foot cases. However, once the step between FS sizes becomes larger, the differences can be seen. Note that the general slope at later times appears to be constant for all values less than 100 feet. However, the 100 and 200 foot cases appear to have a lower slope. This is expected as the fracture network is the sole transportation route for the hydrocarbons in a dual porosity reservoir. Once the FS reaches a certain spacing the production will begin to diminish at a more noticeable rate.

Based on the literature and Figures 16 and 28, a large change in the reservoir's performance is not noticed for small changes of FS. Low values of FS will consequently be omitted. The range selected for our study will be 1 to 200 feet.

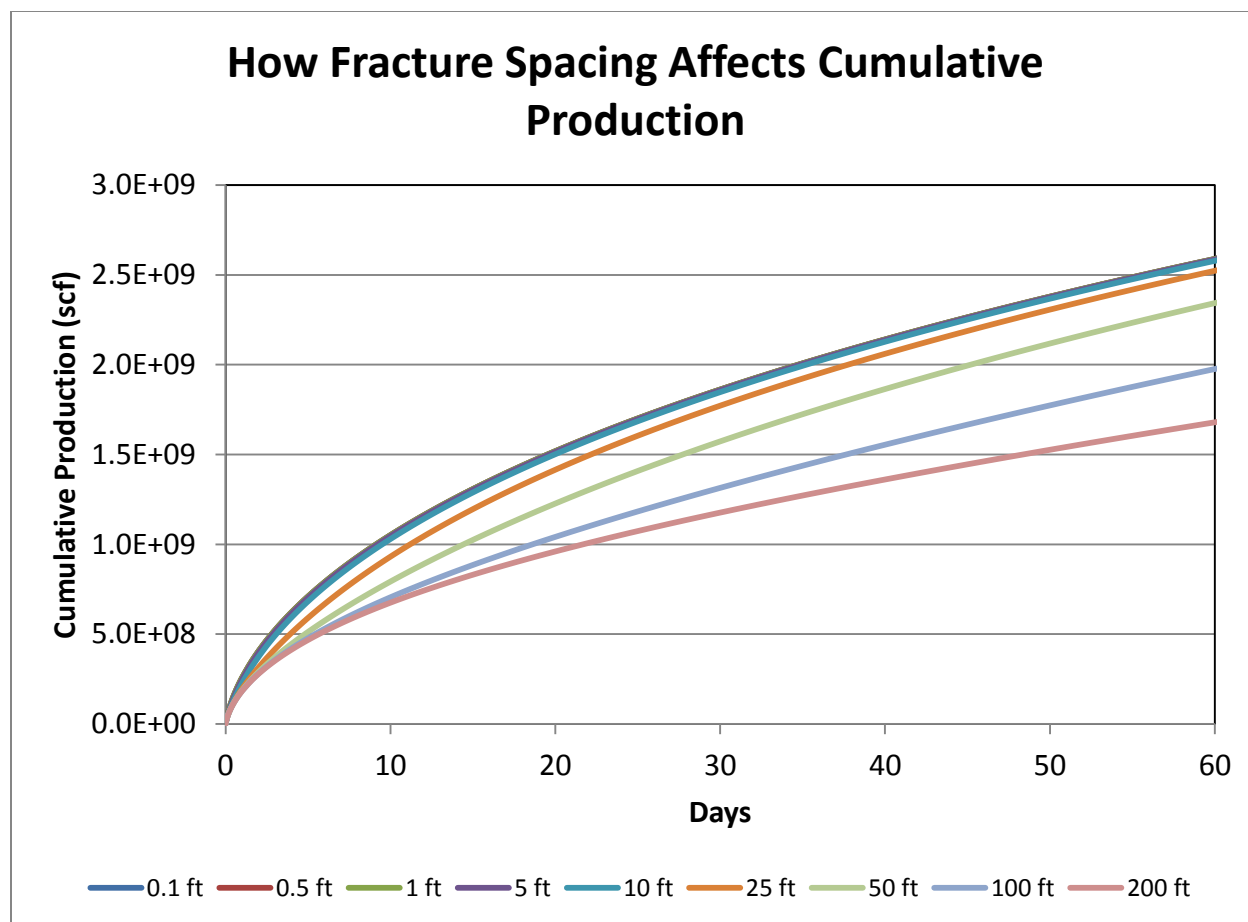


Figure 28: Plot demonstrating how fracture spacing affects cumulative production.

Figure 29 displays how reservoir thickness will affect cumulative production. We would expect for a larger volume of reservoir to be able to produce at a higher rate for a longer time, and the opposite to be true for a smaller volume. This is what we see when analyzing Figure 29.

Based on the literature and Figures 17 and 29, it is determined that reservoirs under 50 feet thick will not be able to sustain the large production values that are associated with expensive

multilateral wells. Because of this, the range of thicknesses examined in this study are 50 to 300 feet.

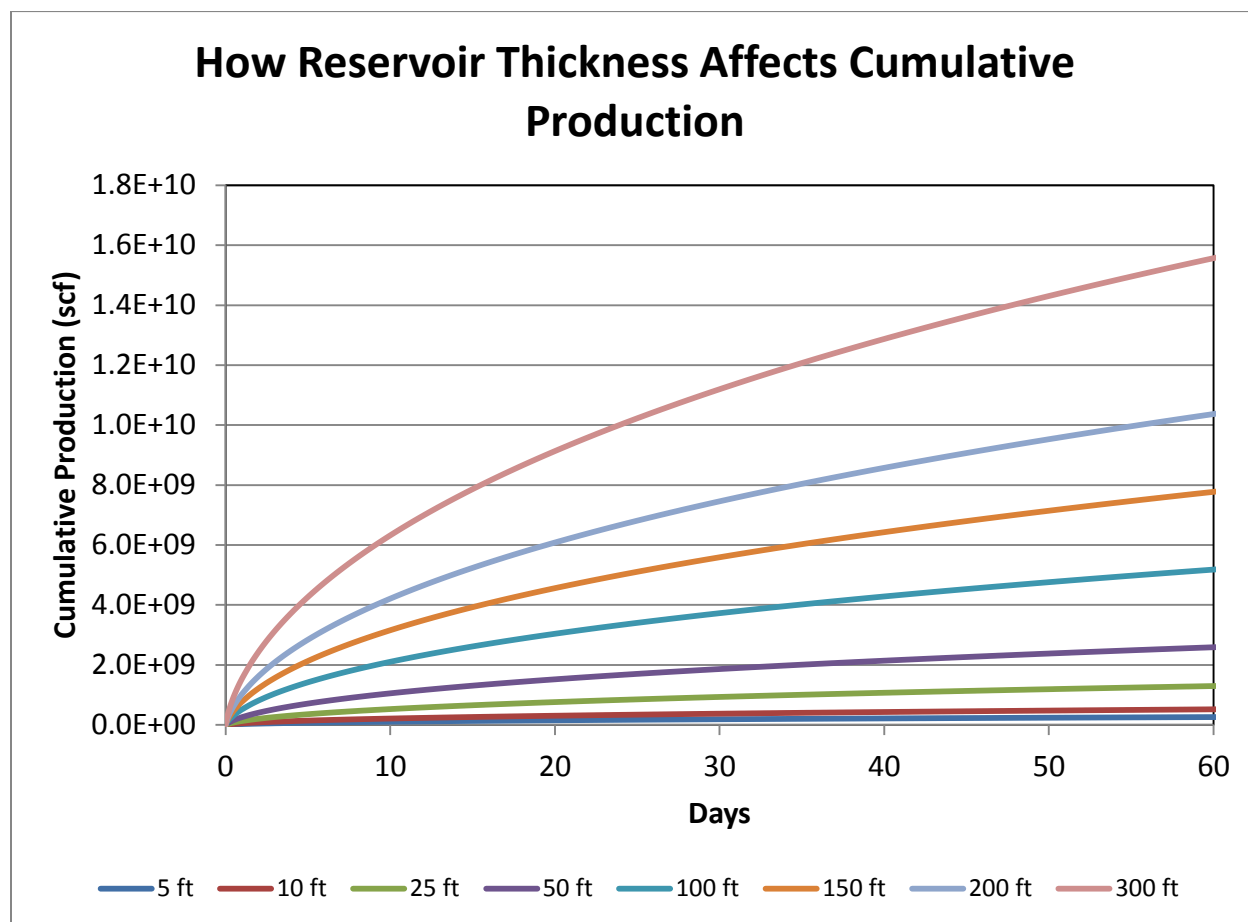


Figure 29: Plot demonstrating how reservoir thickness affects cumulative production.

Figure 30 demonstrates how matrix porosity affects cumulative production for reservoirs of the same volume. The larger the porosity, the larger the volume of available producible hydrocarbons, and consequently the larger the cumulative production.

Based on the literature and Figures 18 and 30 it was determined that the matrix porosity will range from 6% to 15% for the study at hand. Again, in order to produce at an economic rate for an extended period of time, the reservoir needs to have a large volume of hydrocarbons stored within it. Recall that all fluid is stored within the matrix elements. Thus, lower values of matrix porosity will result in a lower volume of hydrocarbons, and this will yield wells that cannot maintain production for extended periods of time. These wells will not be drilled because they will not be economic, thus they were omitted from this study.

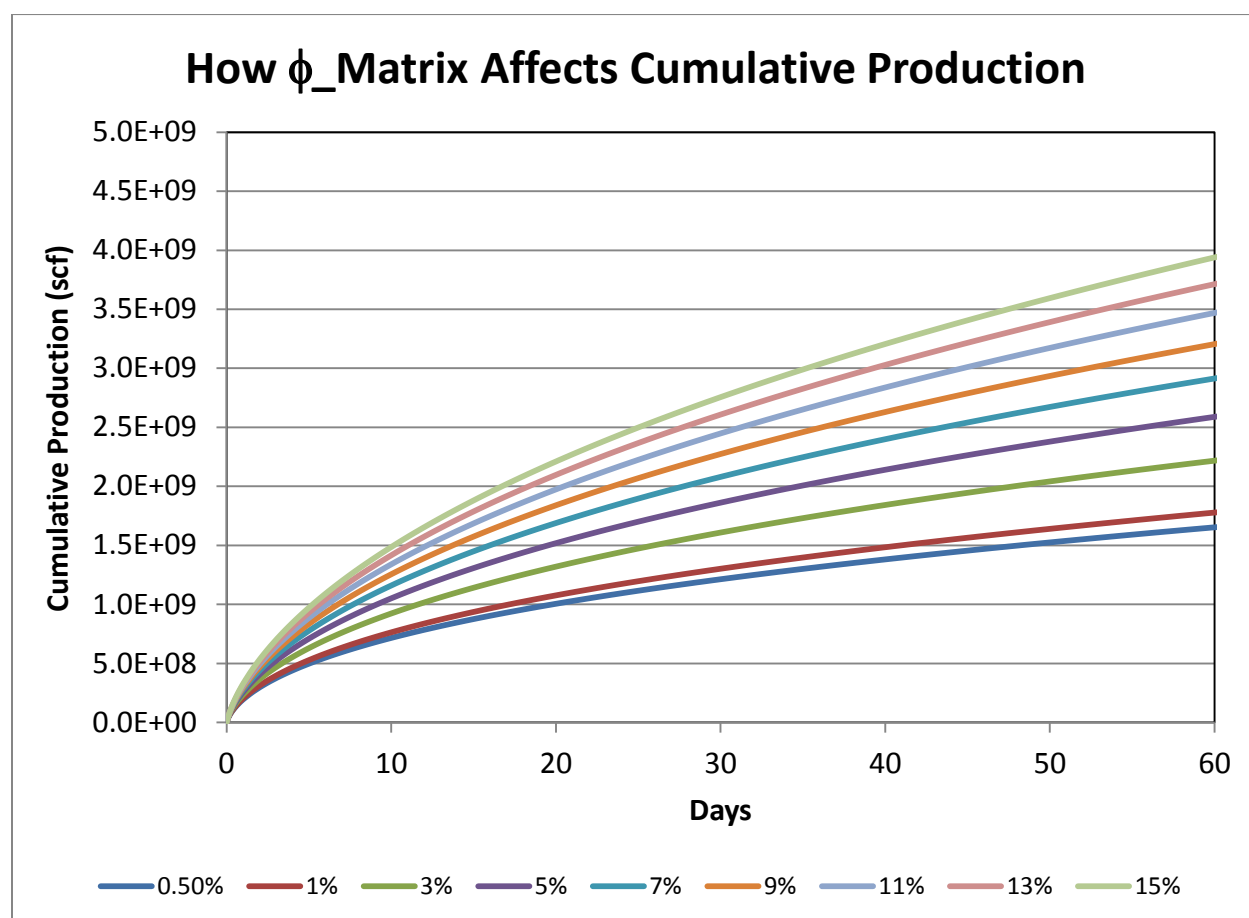


Figure 30: Plot demonstrating how matrix porosity affects cumulative production.

Figure 31 shows how fracture porosity affects the cumulative production. The larger the porosity of the fracture, the larger the diffusibility of the fracture network. This will lead to higher production at early times, but because the matrix acts as the source/sink terms of the reservoir, all values of fracture porosity will eventually converge to the same cumulative production if given enough time to do so.

Based on the literature and Figures 19 and 31, the fracture porosity range for this study was determined to be 0.5% to 3%. Values outside of this range are uncharacteristic of a typical natural fracture, and these values will be omitted from the study. A natural fracture will always have a lower porosity compared to the matrix because the matrix represents a larger fraction of the bulk volume of the reservoir, as described before. Thus, even though the natural fracture itself will have a higher porosity compared to the matrix, when the entire volume of the reservoir is taken into consideration, the fracture porosity will always be less than the matrix porosity.

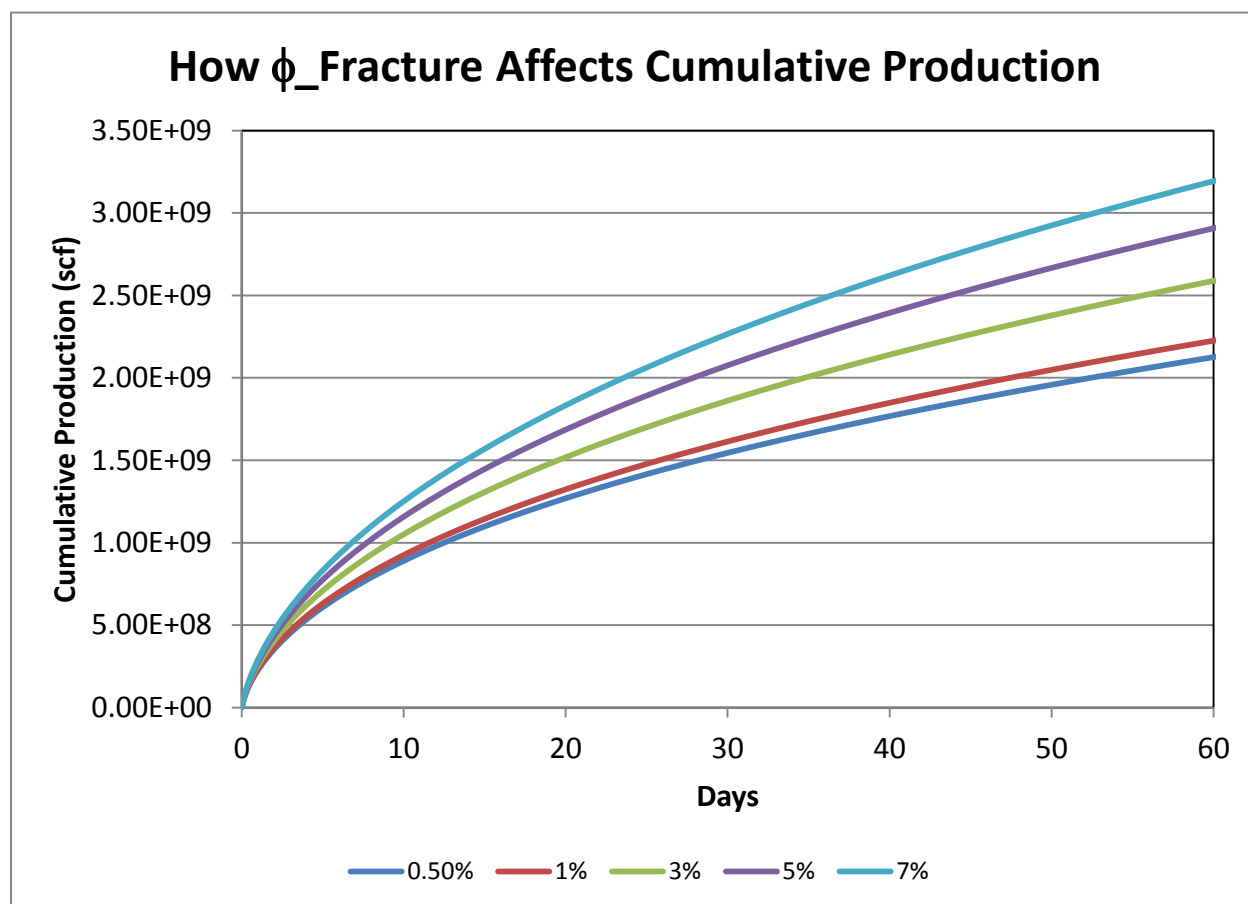


Figure 31: Plot demonstrating how matrix permeability affects cumulative production.

Figure 32 shows how the reservoir temperature affects the cumulative production. Again, because the reservoir is assumed to have one component, methane, the production does not vary with reservoir temperature.

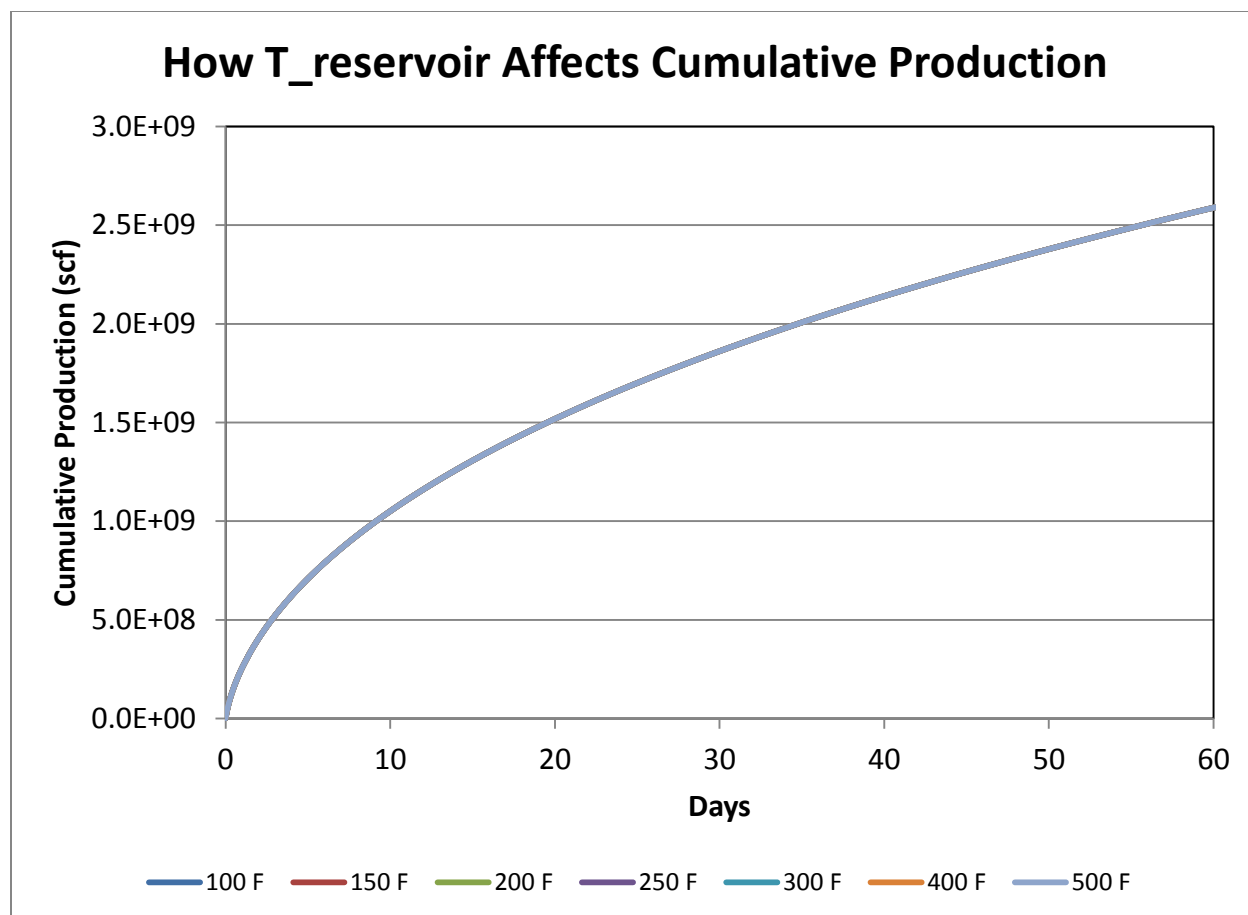


Figure 32: Plot demonstrating how reservoir temperature affects cumulative production.

Figures 20 and 32 demonstrate that the range of reservoir temperature used in this study does not affect the production significantly. However, it was determined that in order to maintain an ANN that could be used universally for a dual-porosity tight gas reservoir, the original temperature range was not altered. Maintaining this large range in reservoir temperatures will make it more difficult for the ANN to be able to distinguish between different reservoir temperatures. However, the end product will be a much more versatile analysis tool if the temperature range is unaltered.

Figure 33 shows how the well length will affect the cumulative production. As the well length increases, the connectivity to the reservoir increases. This means that a larger volume will be contacted and produced from a longer wellbore, and this is what Figure 33 displays below.

Because the wellbore length is within the limit of the reservoir, and there are distinct differences between the individual cases shown in Figures 21 and 32, the range for this study will not be changed.

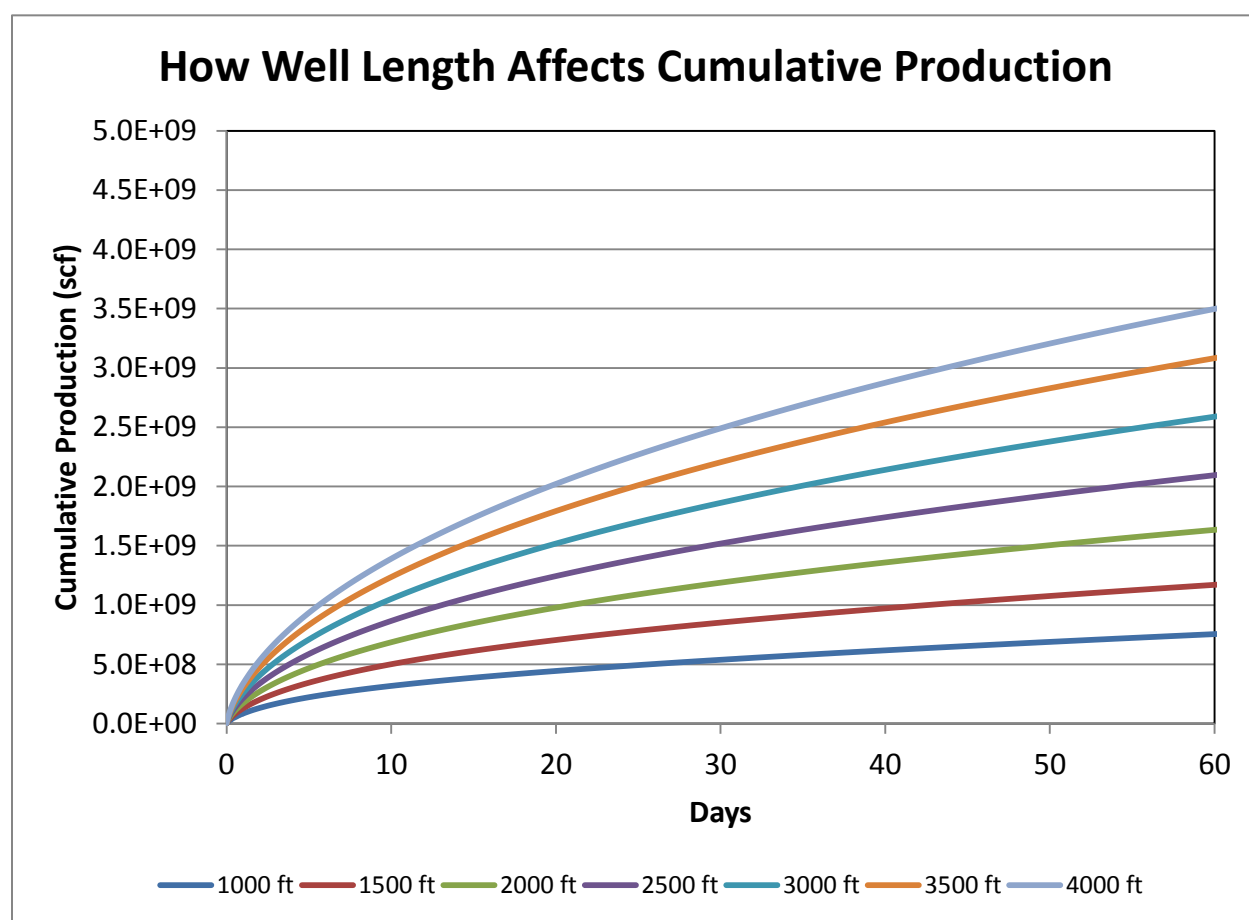


Figure 33: Plot demonstrating how well length affects cumulative production.

Figure 34 shows how the initial pressure affects cumulative production. Within the first 60 days of constant BHP production it is evident that the initial reservoir pressure has an effect on the production rate. After all, the pressure is just a form of measuring the volume of hydrocarbons within the reservoir. Just as the case with gas in a cylinder, the pressure is recorded by how many collisions occur against the inner walls of the container. As the number of molecules increases, the collisions increase, and the pressure increases. Thus, the high pressured reservoirs will have a larger cumulative production because they have a larger volume of producible hydrocarbons within them.

At the start of this research, the effects of adsorbed gas production were not taken into consideration. Instead, the production of free gas alone was considered. Natural gas wells have a large volume of free gas compared to their counterpart coalbed methane, and that is why this assumption can be made for the reservoirs being studied, but not for coalbed methane reservoirs. The free gas will be produced first, and this will cause the pressure of the reservoir to decrease. Once the pressure of the reservoir reaches a very low range, around 200-300 psi, the adsorbed gas will then be produced. For a typical tight-gas well, the pressure will not reach this range for years. Thus, the production of adsorbed gas would not occur during the short time that we are studying the well. Because adsorbed gas production was not taken into consideration for this study, the range of reservoir pressures must be above the range where adsorbed gas can be produced. For this reason, the range of initial reservoir pressure chosen for this study is 1500 psi to 8500 psi.

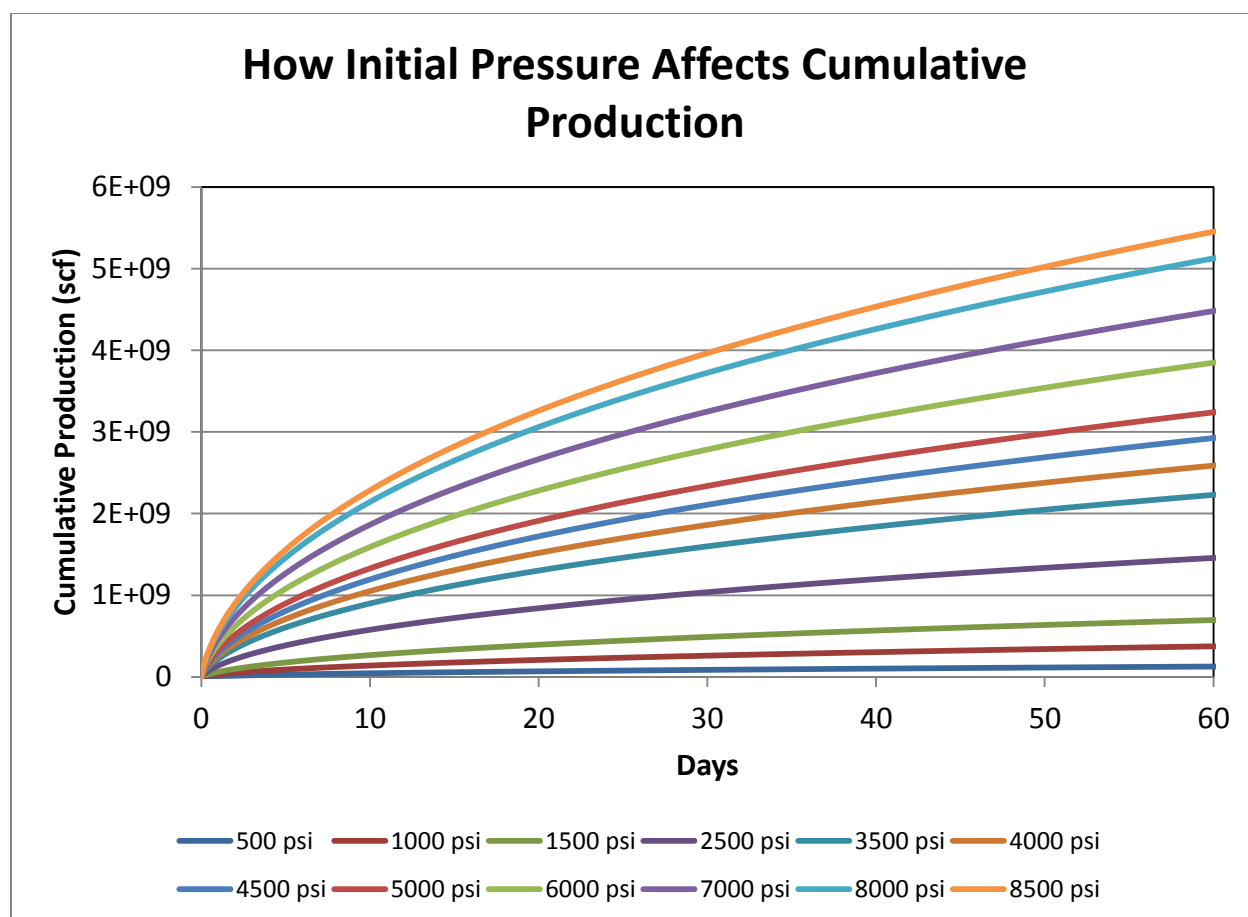


Figure 34: Plot demonstrating how initial pressure affects cumulative production.

Figures 35 and 36 below show how the permeability of the matrix affects the cumulative production of a reservoir with fracture spacing of 1 and 200 feet, respectively. One can see from Figure 35 that when the FS is equal to one, the fracture properties will dominate the reservoir's behavior, and the permeability of the matrix will not alter cumulative production plot. However, when the fracture spacing is increased, the matrix properties begin to dominate the reservoir's behavior. As was seen before in Figure 25, a distinction can be made when the matrix permeability is increased from 1×10^{-3} to 1×10^{-1} md. However, for all other matrix permeabilities equal to or lower than 1×10^{-4} md, no distinction can be seen. Thus, regardless of

fracture spacing, the changes in the cumulative production curves are negligible for permeabilities smaller than $1 * 10^{-4}$ md.

From the literature and discussions with faculty, it was determined that the permeability of the matrix over the entire range is vital to ultra-tight natural gas reservoirs. Thus, the matrix permeability range will not be altered in further steps, and the range focused upon will remain $1 * 10^{-8}$ md to $1 * 10^{-1}$ md.

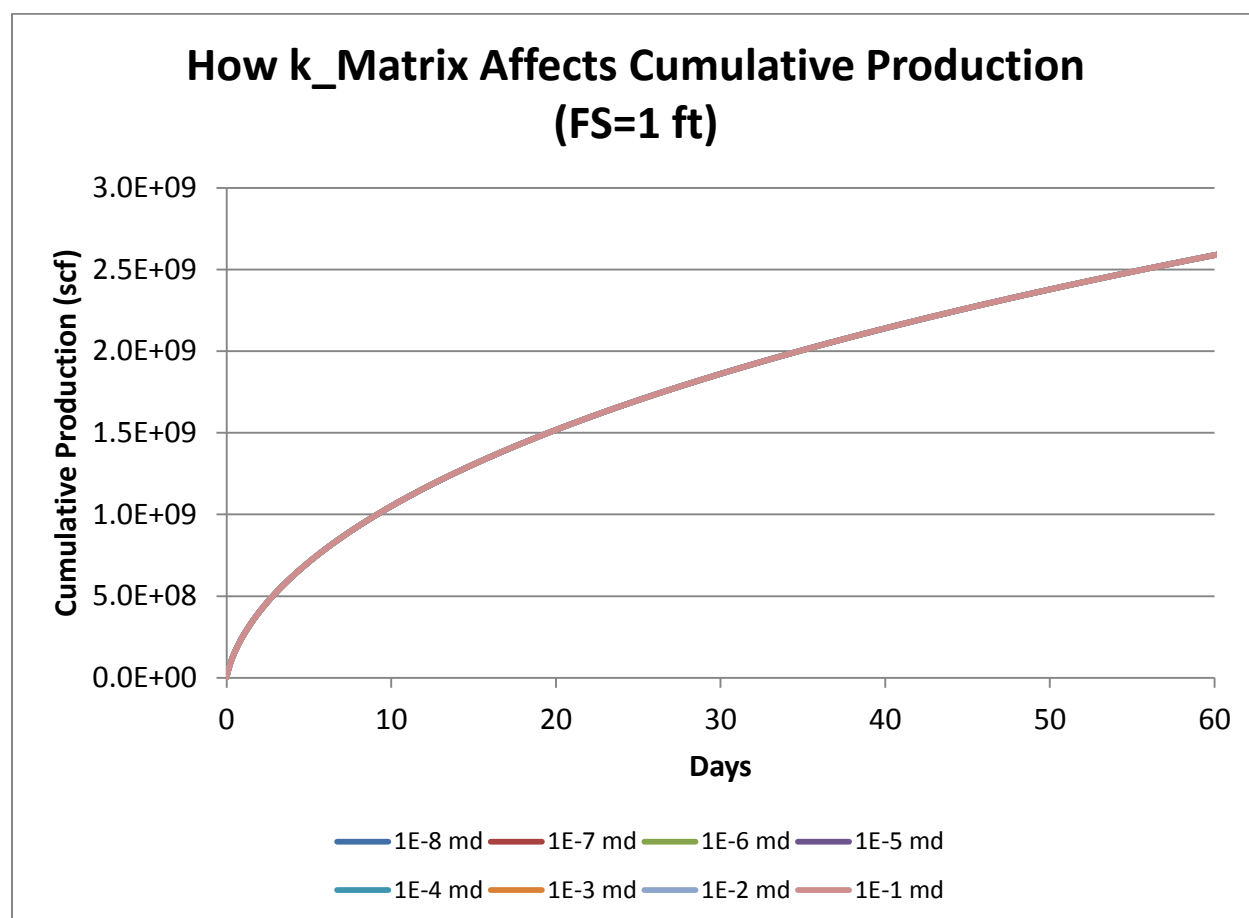


Figure 35: Plot demonstrating how matrix permeability affects cumulative production (FS=1 ft).

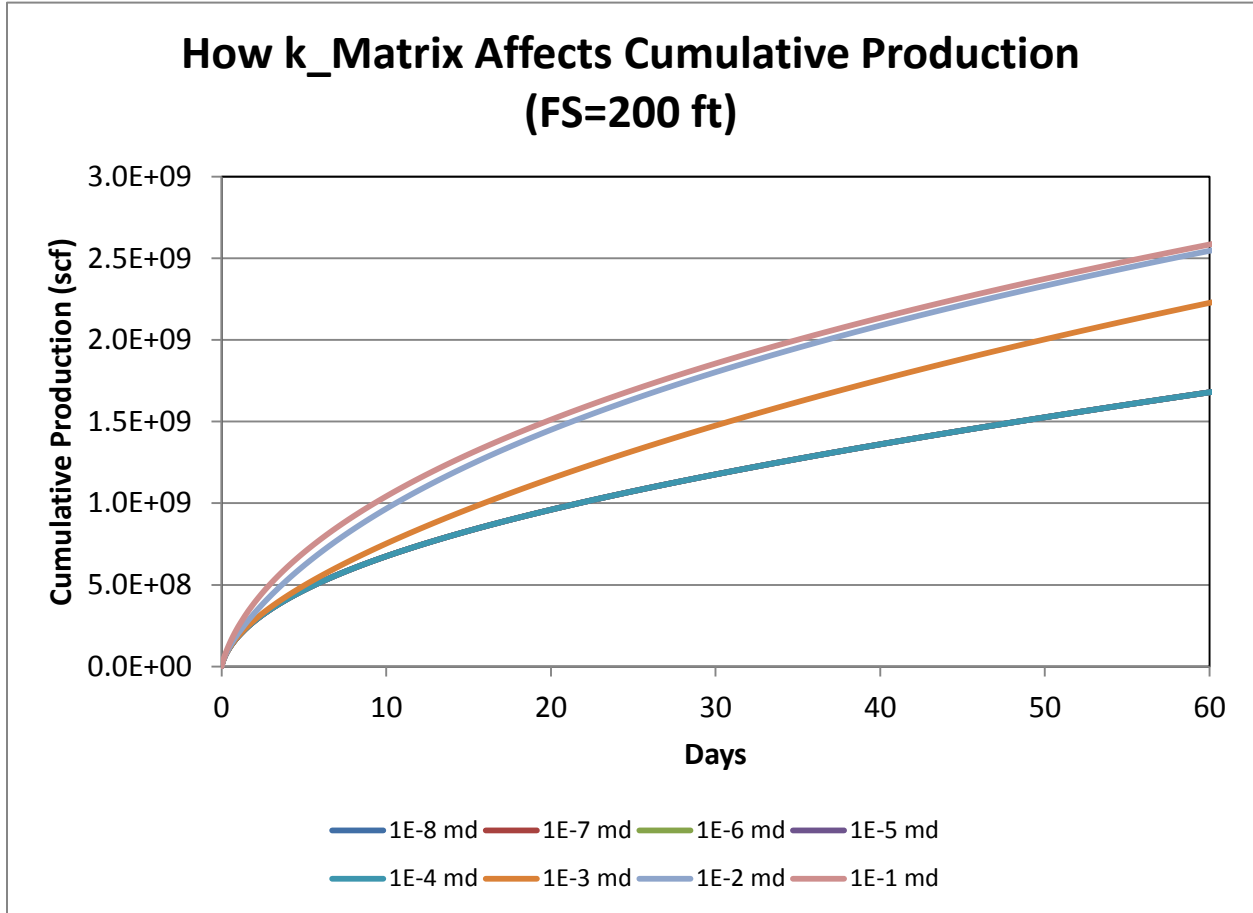


Figure 36: Plot demonstrating how matrix permeability affects cumulative production (FS=200 ft).

Figures 37 and 38 below show how the permeability of the fracture (k_f) affects the cumulative production of a reservoir with fracture spacing of 1 and 200 feet, respectively. As can be seen in Table 3, a larger range of k_f was used initially in this study. However, k_f values equal to and larger than 500 md produced inadequate results. When the permeability is that large, the matrix cannot produce at a high enough rate to maintain production. These values were then omitted from the study. When examining Figures 37 and 38, one notices at early times the larger the

value of k_f , the larger slope of the cumulative production. However, after a certain time the low permeable and high permeable cases production approaches zero, and the slopes of their cumulative production are nearly equivalent. However, the k_f values in the middle, 0.1-10 md, have production greater than zero for the entire 60 day test. Recall, that the matrix permeability affects the transportation of the hydrocarbons, not the overall volume of hydrocarbons within the reservoir. Thus, if these reservoirs were ran at a constant BHP for an infinite amount of time, all of the cumulative production values would eventually converge.

From the literature as well as the evidence presented in Figures 26, 27, 35, and 36, it was determined that the matrix permeability will range from 0.1 md to 250 md.

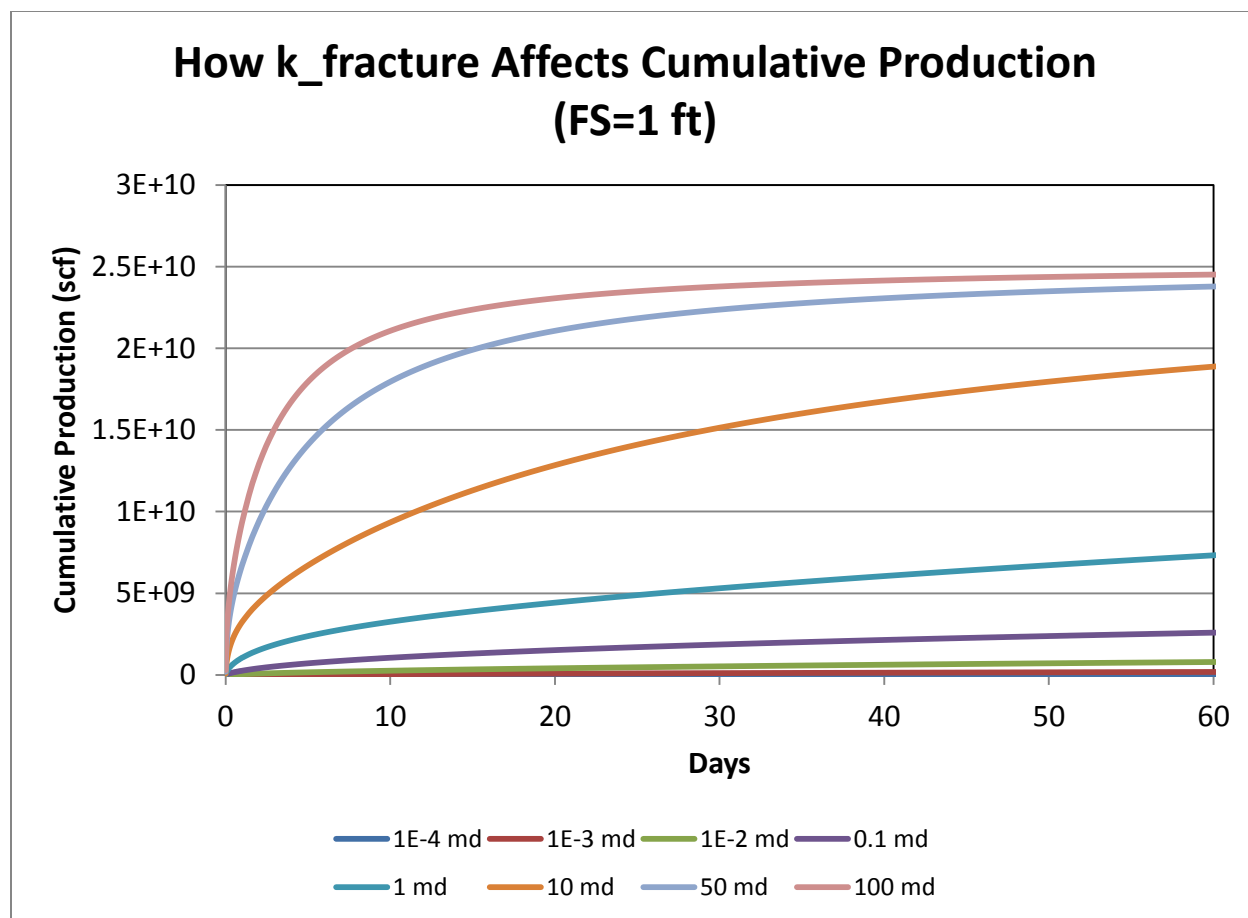


Figure 37: Plot demonstrating how fracture permeability affects cumulative production (FS=1 ft).

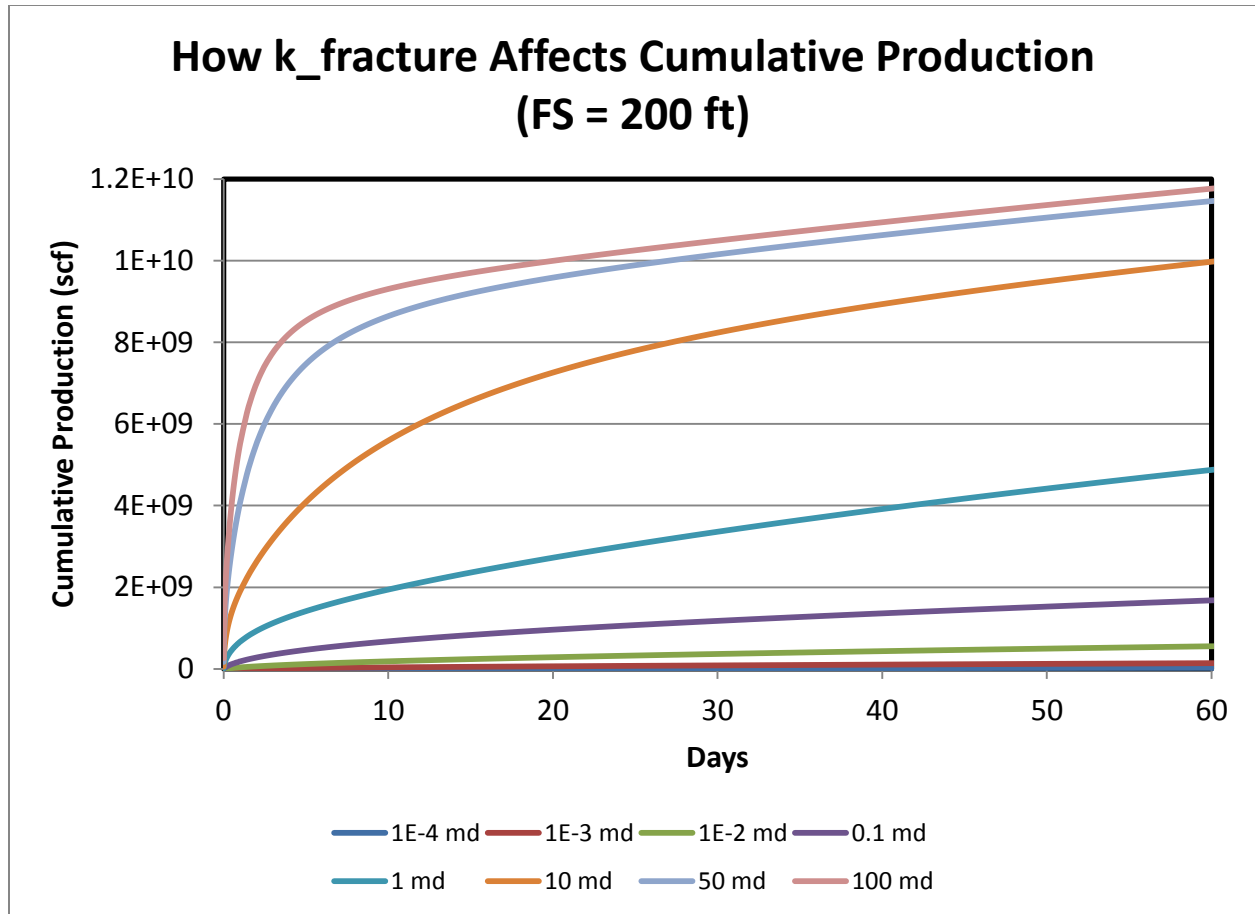


Figure 38: Plot demonstrating how fracture permeability affects cumulative production (FS=200 ft).

2.5. CMG model specifications

For this study, CMG's IMEX conventional black-oil/gas simulator was used. This simulator is used when "changing fluid composition and reservoir temperature are not important factors in the accurate modeling of hydrocarbon recovery processes" (Computer Modelling Group Ltd., 2012). This system was chosen because it was readily available, it is the self-acclaimed fastest black-oil simulator, and it has the ability to accurately represent a naturally fractured reservoir. Also, because this project was focused on a single component gas reservoir, the simulator did not

have to be as complex, and that made IMEX a perfect candidate. This commercial software package has the ability to represent dual porosity systems in many different ways, but because we were using Warren and Root's sugar cube model, the standard dual porosity model was used with Warren and Root's shape factor.

Chapter 3: Artificial Neural Networks

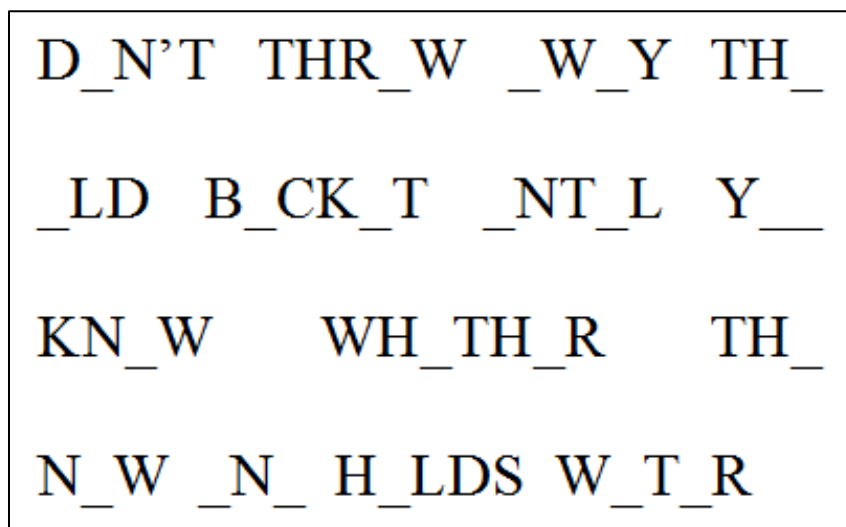
This chapter will attempt to give a brief history and summary of Artificial Neural Networks (ANN). It will also explain in more detail the specific pieces of the ANN that were used in this particular study.

3.1. Introduction to Neural Networks (NN)

Every day our brains do things that are incredible, and we often take it for granted. Just reading, conceptualizing, and storing the information from this one sentence requires an immense amount of neurological processing. Couple the reading with breathing, thinking, and movements, and we are using somewhere around $1 * 10^{11}$, or 100 billion interconnected neurons (Hagan, Demuth, & Beale, 1996). Each neuron has the processing ability of a computer processor, but is six orders of magnitude slower. However, the human mind can still do many things with minimal effort that would be difficult and time consuming for a computer. How can this be one may ask. The reason the brain is so powerful is the sheer magnitude of the processors (neurons), the immense amount of interconnections between the processors, and brain's ability to use parallel processing (Nelson & Illingworth, 1994). An example of where parallel processing is used is shown in the example below.

Figure 39 shows a Swedish proverb that has had the vowels removed from it. Because the brain has the ability of parallel processing we do not look at the characters one at a time. Instead, we can see the big picture and see the letters that are before and after the missing vowels. As we

begin to read we figure out some of the combinations, and knowing these words aids us in determining the subsequent words. This process continues until we have completed the task at hand, and now can recite the Swedish proverb. A computer does not behave in the same manner. A computer uses sequential processing. This would be the equivalent of us having a piece of paper that would only allow us to see one character at a time. We could not see the proverb in its entirety. We could not see the letters before and after the character we were at. This would make solving this problem much more difficult and time consuming. The adaptation of neural networks will help create computers or software packages that use parallel processing instead of sequential processing to solve problems. (Nelson & Illingworth, 1994)



D _N'T THR _W _W _Y TH _
 _LD B _CK _T _NT _L Y __
 KN _W WH _TH _R TH _
 N _W _N _ H _LDS W _T _R

Figure 39: Swedish Proverb with vowels removed

There are two main neural network factions in today's society. One faction is attempting to use a neural network to model how the brain functions. From these studies one can better determine significant features about the brain. The members of this faction develop biological neural

networks (BNN). The second faction uses simplified ideas of biology to create neural networks that are designed to solve a certain problem. This faction develops artificial neural networks (ANN) and will be the focus of this study.

3.2. History of Neural Networks (NN)

The history of how neural networks (NN) came to be is very interesting. NNs did not appear overnight, and their progress was not slow and steady. NNs saw periods of great advancement mixed with periods of little to no progress. This occurred for multiple reasons, but mainly due to the limitations of computing power and lack of funding for research. Hagan, Demuth, and Beale put it best when they stated that ‘two ingredients are necessary for the advancement of a technology; concept and implementation’. A concept is great, and necessary for innovation, however if the concept cannot be implemented due to technological barriers, then one will not have advancement. This often plagued the early pioneers of NNs, however through diligent work and perseverance, the concept of NNs never died, and once the technology was available to test these concepts, NNs began to expand very rapidly into multiple fields of research.

3.2.1. Early Times (1890-1949)

During this time the advancements in NNs were mainly advancements in biological and psychological concepts. In 1890, *Psychology* was written by William James. In this work James discussed the ‘excitement’ of neurons. James linked this excitement to how the brain transmits information and how certain brain processes are linked. Later, in 1943 a neurophysiologist by the name of Warren McCulloch and a young mathematician by the name of Walter Pitts produced a

paper discussing the concepts of how neurons may work. In their paper they showed that networks of artificial neurons, in theory, could compute any logical or arithmetic function. They then modeled a simplified neural network using electric circuits. Their work is often linked to the origin of the neural network field (Hagan, Demuth, & Beale, 1996). Finally, in 1949 Donald Hebb released his book *The Organization of Behavior*. In it he stated that neural pathways are reinforced when that same pathway is used over and over again. ‘Hebb’s Learning Rule’ was derived from this notion, and is still referenced today. (Nelson & Illingworth, 1994)

3.2.2. 1950’s and 1960’s

The 1950’s saw improvements in computer systems, making it possible to test theories that had originally been limited to paper. Hebb and Nathaniel Rochester of IBM teamed up to create successful adaptations of neural networks. Then, in 1956 the ‘birth’ of NNs was said to occur at the Dartmouth Summer Research Project on Artificial Intelligence. Previously, NNs were lumped into the same category as artificial intelligence. At this meeting there was a distinction made between the two. High level, or artificial intelligence (AI), was seen as “intelligent” machine behavior whereas neural networks were seen as low level because they used neural networks processes of the brain to achieve “intelligence”. However, the distinction between the two parties had been made, and this would help separate the two strategies in the future. If a process could not be completed using AI it could still be attempted with NNs and vice versa. (Nelson & Illingworth, 1994)

In 1957 the Perceptron was developed by Frank Rosenblatt. The Perceptron is seen as the oldest neural network which still has application today. The Perceptron demonstrated its ability to perform pattern recognition, and this early success brought a large amount of attention to the NN world. However, the Perceptron had limitations and could only solve linear problems. Two years later in 1959 Bernard Widrow and Marcian Hoff created the Widrow-Hoff learning rule, which was similar to the Perceptron's learning rule, and is also still used today (Hagan, Demuth, & Beale, 1996). Widrow and Hoff developed ADALINE and MADALINE with stands for Multiple ADaptive LINear Elements. This network was used for the first real-world application; canceling out echoes in telephone wires. Later in 1967 Stephen Grossberg developed Avalanche. This NN was used extensively on activities such as speech recognition, but also found use in teaching motor commands to robotic arms. (Nelson & Illingworth, 1994)

The hype brought forward by the Perceptron and these other NNs was felt worldwide. However, with this popularity came much scrutiny. Marvin Minsky and Seymour Papert wrote a book titled *Perceptrons, An Introduction to Computational Geometry*. This book openly criticized the Perceptron's and other networks' lack of ability to solve complex problems. Widrow and Hoff proposed new learning algorithms that could overcome this dilemma, but they could not successfully implement them. Minsky and Papert's criticisms coupled with a lack of invention and computing power caused NN's progress to nearly come to a halt in the 1970's and early 80's. (Hagan, Demuth, & Beale, 1996)

3.2.2. 1980's

In 1982 John Hopfield presented a paper on NNs that would have an extreme impact on the future of the technology. Hopfield noted that many researchers had tried to use NNs to emulate the brain as their first task. Later they would attempt to manipulate this same network to solve some sort of problem. Hopfield suggested that NNs should first be designed to solve specific problems. Using this mindset, NNs began to fill in the gaps that AI could not. Military interests in image recognition/retrieval led to a large amount of funding being put into the NN field. This led to more research projects and fast progress during these years. The expansion of interest led to an expansion of conferences. This led to more collaboration and success. Also, interest from Nobel laureates Leon Cooper (Superconductivity Theory) and Francis Crick (DNA structure) added to the hype of this 'new' and emerging field. (Nelson & Illingworth, 1994)

3.2.3. Applications

Currently, there are applications in many different fields. These fields and applications include but are not limited to:

Table 4: Fields and Applications for Neural Networks (Hagan, Demuth, & Beale, 1996)

Field	Applications
Aerospace	<ul style="list-style-type: none"> • Flight Path Simulators • High Performance Aircraft Autopilots • Aircraft Component Default Detectors
Automotive	<ul style="list-style-type: none"> • Automobile Automatic Guidance Systems • Warranty Activity Analyzers

Banking	<ul style="list-style-type: none"> • Credit Application Evaluators • Check and Document Readers
Defense	<ul style="list-style-type: none"> • Weapon Steering • Target Tracking • Object Discrimination • Facial Recognition • Signal/Image Identification
Electronics	<ul style="list-style-type: none"> • Chip Failure Analysis • Voice Synthesis
Entertainment	<ul style="list-style-type: none"> • Animation • Market Forecasting • Special Effects
Financial	<ul style="list-style-type: none"> • Loan Advisor • Real Estate Appraisal • Mortgage Screening • Currency Price Prediction
Insurance	<ul style="list-style-type: none"> • Product Optimization • Policy Application Evaluation
Manufacturing	<ul style="list-style-type: none"> • Process Control • Real-Time Particle Identification <ul style="list-style-type: none"> • Beer Testing • Welding Quality Analysis

	<ul style="list-style-type: none"> • Project Bidding • Dynamic Modeling of Chemic Process Systems
Medical	<ul style="list-style-type: none"> • Breast Cancer Cell Analysis • Prosthesis Design • EEG and ECG Analysis • Optimization of Transplant Times • Emergency Room Test Advisement
Oil and gas	<ul style="list-style-type: none"> • Exploration
Robotics	<ul style="list-style-type: none"> • Forklift Robot • Trajectory Control • Vision Systems
Speech	<ul style="list-style-type: none"> • Speech Compression • Text to Speech Synthesis • Speech Recognition
Securities	<ul style="list-style-type: none"> • Automatic Bond Rating • Market Analysis • Stock Trading Advisory Systems
Telecommunication	<ul style="list-style-type: none"> • Automated Information Services • Image and Data Compression • Real-Time Translation of Spoken Language
Transportation	<ul style="list-style-type: none"> • Vehicle Scheduling • Routing Systems

3.3. How do NNs work?

3.3.1. Biological Inspiration

When discussing how Neural Networks (NN) work, it is often best to think in metaphors. All NNs operate based on the principle concepts of the neurons within the body. The body is quite complex, and there are over 700 different types of neurons within it. Some neurons are microscopic and others can be nearly a meter long. The neuron used in our metaphor is a neuron found in the retina. Neurons consist of four main parts that one is concerned with when creating metaphors for a NN. These parts are dendrites, nucleus, axon, and the synapse. The dendrites are what carry the input signal into the nucleus. This signal is generated from chemical and electrical changes within the brain, and can be either positive or negative. Each neuron has many dendrites, so multiple neurons can be connected to the input of the particular neuron being studied. The input signals from neighboring neurons are transmitted through the dendrites to the nucleus, where all of these inputs are summed. Each neuron has a certain threshold value. If the sum of inputs is greater than this threshold then the input signal will continue on through the neuron. If the sum of inputs is less than the threshold, the input signal will not continue through the neuron. If the threshold is reached the signal then travels through the axon. The axon will split into many root-like structures, similar to the dendrites, and in this way the neuron can be connected to multiple neighboring neurons. The ‘connection’ to the neighboring neurons isn’t a physical connection, but a small gap between the axon and the dendrites, known as the synapse (Hagan, Demuth, & Beale, 1996). Once the neuron fires, the signal will branch the gap, and continue to the next neuron. The synapse activity is what an artificial neural network uses as the motivation to model. (Nelson & Illingworth, 1994)

All signals will travel at roughly the same speed through a given neuron. Once the signal has transmitted through the neuron, the neuron will return to its unexcited state, ready to receive the next impulse. The strength of a signal cannot be determined from the speed of the signal but rather from the rate of the neuron firing. A simplified diagram representing two biological neurons is shown in Figure 40 below. Recall a biological neural network's (BNN) main goal is to simulate how neurons interact so that the brain's functions can be studied and understood. An artificial neural network's (ANN) main goal is to use the ideas of neurons and how they interact to develop a system that can learn to solve a specific problem (Nelson & Illingworth, 1994). However, BNNs and ANNs do have some similarities. First, the building blocks of both networks are simple computational devices; however BNN's neurons are much more complex than an ANN's neurons. These building blocks are highly connected throughout the network to other neurons. And lastly, the connections between the neurons determine the functionality of the network. (Hagan, Demuth, & Beale, 1996)

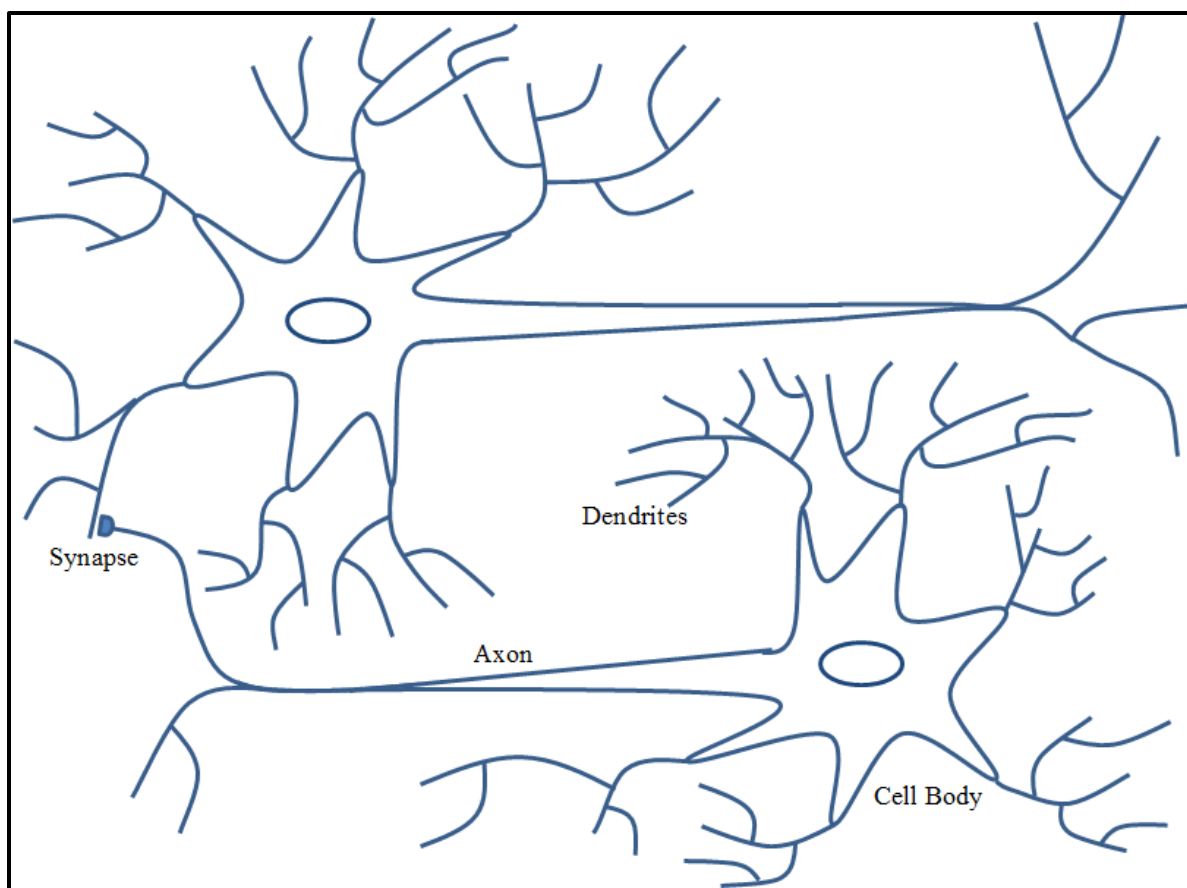


Figure 40: Simplified Diagram of two biological neurons.

Neurons that are interconnected form nerve structures. These nerve structures will connect input data from the eyes or the ears to effector organs such as muscles or glands. Neurons are highly interconnected and can have as many as 200,000 connections, however 1,000 to 10,000 is common. Signals that come into the dendrites are given a certain weight. An intense signal will be given a higher weight than a mild signal. This idea will be used directly within the ANN, as signals that cause the input data to be better mapped to the output data will gain in weight, and signals that do not have an effect on mapping will lose weight until their weight is nearly zero. Moreover, processes can occur within the brain that will affect the rate of firing. For example, oxygen deficiency, anesthetics, and fatigue will lead to a higher resistance within the neuron.

This higher resistance would increase the threshold of the neuron, causing the neurons' firings to become less frequent. Other parameters could cause the firing process to speed up. The idea that the neurons' firing rate can be changed will be used by the ANN. The ANN will be able to adjust signals so that it will be able to learn more efficiently. (Nelson & Illingworth, 1994)

3.3.2. Processing Elements (PE)

Processing Elements (PE) are the basic components of neural networks. PEs process the input signals, assigning a specific weight to each. The PE then sums up the inputs and determines what the output signal will be. Biological neurons will either fire or not fire. Because our brain uses this functionality we know that it can be used to solve complex problems. However, the brain has 10^{11} neurons, and our neural network will not be this sophisticated. Thus, in order to solve non-linear problems our PEs cannot have outputs of either one or zero, but must be equipped with non-linear transfer functions, which will be discussed in a future section. The PEs behave like biological neurons in that they take a set of input signals, sum it, and then determine the output.

3.3.3. Weighting Factors

Weighting factors are applied in order to see which information is most important. The input data are a column matrix and the weighting factors are also a column matrix. The dot product of the two matrices will determine the strength of the signal. Recall, if two vectors are in the same direction, the magnitude of the dot product is at a maximum, and if two vectors are in the opposite direction, i.e. 180° , then the dot product is a minimum. The weighting factors will be changed frequently at the end of each run by the neural network. As the weighting factors are

changed, the inputs' 'mapping' to the outputs will change. If the mapping is changed for the better, than these weighting factors are kept, if the mapping is changed for the worse, then new weighting factors will be generated. As the neural network operates it will be able to determine which input data is most significant to the output data, and this data will be given the highest weighting factor. Input data that is unimportant to the output will be given a smaller weighting factor until the weighting factor is equal to zero.

3.3.4. Transfer Functions

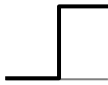
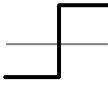







Transfer functions are used to manipulate the input data. Transfer functions can range in complexity from step-wise functions, to linear functions, to non-linear functions. With today's problems being primarily focused on non-linear problems, non-linear transfer functions are necessary in order to create non-linear solutions. To refresh, five transfer functions will be explained, and diagrams of these and other transfer functions are displayed in Table 5 below. The first function is the step-wise or 'hard-limit' function. For this function if x is less than some threshold value, then the output is negative one (or zero). If x is greater than the threshold value then the output is positive one. The second function is a linearly increasing function with constant slope. The third transfer function is known as the ramping function. This function combines the hard-limit and linear transfer functions. If x is less than the lower threshold value, then the output is zero. If x is greater than the lower threshold value but less than the larger threshold value, then the output becomes a linearly increasing function with constant slope. Finally, if x is greater than the larger threshold value, the output is equal to one. Lastly, there are two sigmoid or 'S' functions which are commonly used in ANNs. These functions do not have any linear segments, but rather increase at varying rates from either negative one to positive one,

or from zero to positive one. Both of these Equations and their derivatives are continuous. This will allow using either this function or the derivative of this function to calculate error, which will add an extra level of adaptability for the network. The Equations of these two curves are:

$$\textit{Log - Sigmoid:} \quad y = \frac{1}{1 + e^{-x}} \quad \textbf{Eq. 26}$$

$$\textit{Tan - Sigmoid:} \quad y = \frac{e^n - e^{-n}}{e^n + e^{-n}} \quad \textbf{Eq. 27}$$

Table 5: Various Transfer Functions used in MATLAB ANN program (Hagan, Demuth, & Beale, 1996)

Name	Input/Output Relation	Icon	MATLAB Function
Hard Limit	$a = 0 \quad n < 0$ $a = 1 \quad n \geq 0$		hardlim
Symmetrical Hard Limit	$a = -1 \quad n < 0$ $a = +1 \quad n \geq 0$		hardlims
Linear	$a = n$		purelin
Saturating Linear	$a = 0 \quad n < 0$ $a = n \quad 0 \leq n \leq 1$ $a = 1 \quad n > 1$		satlin
Symmetric Saturating Linear	$a = -1 \quad n < -1$ $a = n \quad -1 \leq n \leq 1$ $a = 1 \quad n > 1$		satlins
Log-Sigmoid	$a = \frac{1}{1 + e^{-n}}$		logsig
Hyperbolic Tangent Sigmoid	$a = \frac{e^n - e^{-n}}{e^n + e^{-n}}$		tansig
Positive Linear	$a = 0 \quad n < 0$ $a = n \quad 0 \leq n$		poslin
Competitive	$a = 1 \quad \text{neurons with max } n$ $a = 0 \quad \text{all other neurons}$		compet

3.3.5. Bringing it All Together

Individual PEs will learn once we give them memory. This memory will be used to store weights and biases from past trials. This memory will allow the PE to see which weight factors produced the best results and which produced the worst results. It will be able to make an educated guess

as to what weighting factor will produce the best results. An example of this is for image recognition. If we have pictures of numerous animals and an image that is a cat has an output of a dog then the weights need to be adjusted in order for the output to match the input. The weights will change, always learning from the known target data, until all images can be correctly linked to their respective animal.

For the problems we are examining, multiple PEs will be necessary. A set of PEs, sometimes referred to as a set of nodes, will be linked together in one 'layer' of the neural network. Each input signal will be transferred to each node of the first layer. There, the PE's will maintain their aforementioned tasks, and sum these inputs and create an output value. The complexity of the problem will dictate the number of PEs as well as the number of layers of PEs. For these problems there will always be an input layer and an output layer. The input layer stores the matrix of input data. This layer will have as many nodes as it has inputs. This could be any whole number greater than zero. The output layer stores the target output data. The target output data is what the ANN is attempting to map the input layer to. Again, the output layer will have a set number of nodes, and this number will depend on how many output values one is trying to match; it too can be any positive whole number. The layers that are in-between the input and output layers will map the input data to the target data. These layers typically cannot be 'seen' by the user, and are thus referred to as hidden layers. However, even though these layers cannot be seen by the user, they can still be manipulated and analyzed by them. This is one of the most important parts of creating an efficient ANN, and will be discussed in more detail in a later section. The hidden layers are the area where most of the computations within the ANN will take place. The hidden layer's nodes are where the transfer functions are used and also where the

weighting functions are created and changed over time. A hidden layer's node's output usually will be distributed to each sequential hidden layer's node's input. When this is the case, the network is said to be 'fully connected'. However, there are problems where it may be advantageous to only map certain inputs to certain outputs. If this is the case, a fully connected network should not be used, and a partially connected network should be used in its place. The network that we are using in this study is fully connected.

Deciding on the number of nodes in each hidden layer is very important in creating an efficient ANN. If there are too few nodes, the network will have a difficult time creating the correct pattern to solve the problem. Too few of nodes will lead to the network not being able to prioritize data, and this will affect creating the weighting factors. Moreover, the network will not be able to remember long enough to establish a successful ANN. If there are too many nodes then the network will have difficulties in creating generalizations, may become 'over trained', and may lose its prediction capabilities.

Layers of the neural network can be connected in one of two ways. The network is called a 'feedforward' network when the outputs from the hidden layer's nodes can only travel towards the inputs of the next hidden layer. A network is labeled as a 'feedbackward' network if the outputs of a hidden layer are transferred to the inputs of that particular hidden layer or a subsequent layer. Typically, feedbackward networks will converge faster because they can update the weighting factors after each output from the hidden layers whereas the feedforward network will only update weights after the input signals have propagated through the entire

ANN. The feedbackward networks will also converge faster because each node at each layer can be analyzed and adjusted. An example of a feedbackward network is shown below in Figure 41. With the feedforward network, the weights are only adjusted based on the final results. Thus, it is not known what hidden layers are performing well and which are performing poorly. The feedforward network has to continue to train numerous weighting factors that could be ruled out by the feedbackward network, and this leads to a longer training time.

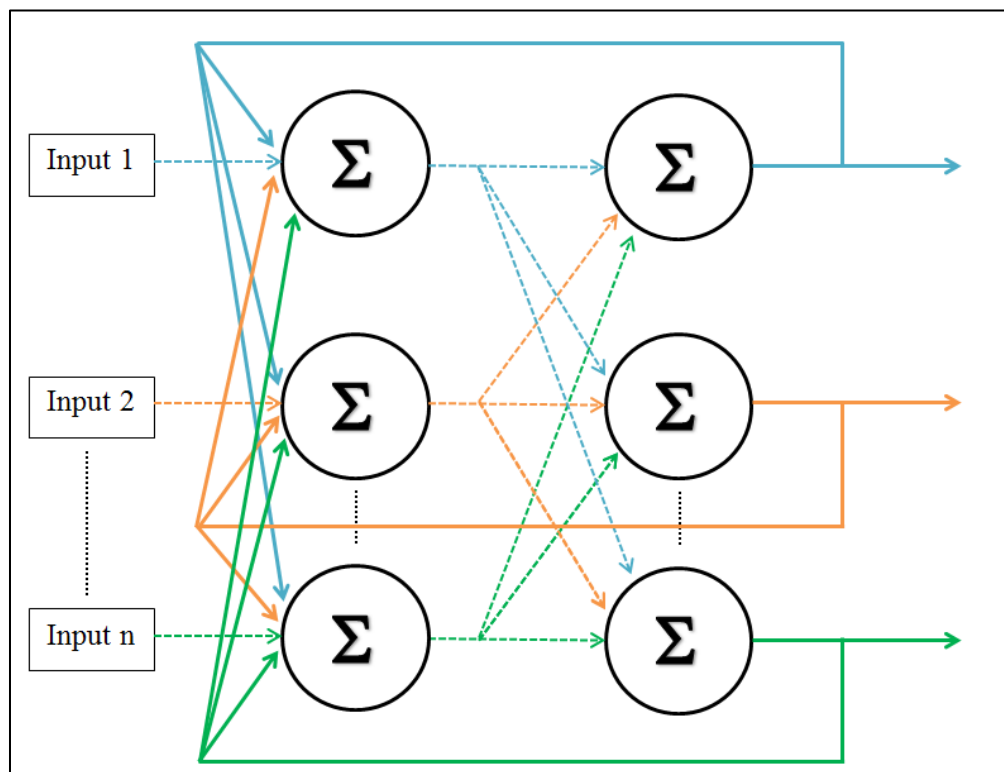


Figure 41: Schematic of a Feedback Network adapted from Nelson and Illingworth.

From the above summary one can see that traditional programming capabilities are not necessary for the manipulation of an ANN. One needs an understanding of how transfer functions, nodes,

hidden layers, and weighting factors can be manipulated in order to optimize the ANN. More importantly than traditional programming skills is the user's statistical analysis ability in creating input data and validating output data.

3.4. How is Information Stored within the Network?

Conventional computers store information in one location. ANNs store information throughout the network. Individual nodes have information, much of it cause and effect information, which they can use to optimize the system. Much like a human's memory, the ANN's memory is associative. For instance, a teacher who has graded a student's reports with sloppy handwriting will be able to read the report easily by the end of the semester because of their associative memory. The same is true when one hears a familiar tune. It is easy to hum or to sing along. ANNs will store its data associatively; this means it will look for familiarities or the closest match between data and that is how it will learn. The reason that associative memory is important to an ANN is that one can store a large amount of patterns via this associative memory and also whenever additional data is input into the system the ANN does not have to relearn all of the information over again; it will just use this additional information alongside the past information. The associative memory is why an ANN will be able to quickly solve a problem once it is successfully trained. It will simply look for a familiar 'tune' and from that information is able to generate a solution to the problem. This is why ANNs are said to be great at interpolating between data but are very poor at extrapolating data. As long as the new input data is within the data that the NN was trained, the network should be able to accurately predict the

output. However, if the input data is outside the range of the trained data set, then the ANN will have a lower chance of making an accurate prediction.

3.4.1. Learning

As mentioned previously, learning occurs when the weighting factor is being changed. The changes in the predicted output as the weighting factor is changed will be stored in the network's memory. The network will know which weighting factors worked best and which worked least. From this information the network will be able to determine which parts of the data should have the highest weights and which should have the lowest. These weights will then continue to be fine-tuned until the difference between the predicted output and target output is minimized.

3.4.2. Self-Organization

Self-organization refers to the network's ability to adjust multiple nodes at one time. By adjusting these nodes in specific patterns, non-linear problems that do not have a direct algorithm as an answer can be solved. The network can develop its own simple efficient rules much like heuristics in psychology.

3.4.3. Training

Training of a neural network can be lumped into two main categories: supervised and unsupervised. With supervised learning the input is given with an expected output. A simple example is number pattern recognition. One could show the computer the number '8' (this would

be the input). Then, one could tell the computer that the input is associated with the output of an '8'. The network, after trained, would be able to be given a number or possibly a set of numbers from a check or bank statement, and then be able to store these numbers quickly. This would aid a company who was attempting to digitize old documents. Instead of simply scanning the documents so they could be viewable on the computer, they could analyze them with a neural network. The NN could be set up in a way that would store all of the information that was being scanned into a spreadsheet where it could be further analyzed.

This study will focus on two separate ANNs that each are trained under supervision. The first ANN will be the forward solution. This model will take reservoir and well data such as thickness, porosity, permeability, fracture spacing, flowrate, and wellbore length and map it to the corresponding output data which is in the form of pressure transient data. The second ANN will be of more use to other users and will be the inverse solution. In this case the input would consist of the pressure transient data along with known parameters such as flowrate and wellbore length. The input data would then be mapped to reservoir data such as thickness, fracture spacing, porosity, and permeability. Thus, a user who has a similar well structure and performed a pressure drawdown test could input their data into the ANN and nearly instantaneously would receive information about their reservoir.

Unsupervised training is also used in neural networks. For this form of training the input data is given a score based on how well it performs. A robot used unsupervised training to find the best way to throw a ball. The robot would attempt to throw the ball using a wide range of techniques.

Based on how far the ball went, the robot would get a score. A far throw resulted in a high score and a short throw resulted in a low score. The thought was that a typical throwing motion would produce the best results and this is what the robot would learn. However, the robot learned many unconventional ways in addition to the conventional way to throw the ball that resulted in the ball traveling a large distance.

3.4.4. Generalization

It is important for the neural network to always remain generalized. ANNs can often become ‘over trained’ in which they will stop learning from the input/output pairs and actually begin to memorize them. Whenever the ANN begins to memorize it loses its ability to interpolate and instead needs to extrapolate to get an answer. As mentioned previously, ANNs are inefficient at extrapolation, and thus an overtrained network will lead to inaccurate predictions. A generalized network will be able to be given a new set of input data, which it has never seen, and accurately predict the output data associated with it. This is the objective of this study.

3.5. Neural Network Learning Rate and Learning Laws

After discussing how a neural network learns and remembers it is important to discuss how the learning rate, training techniques, and learning laws will affect the NN’s performance.

3.5.1. Learning Rate

The learning rate is a measurement of how fast the network converges. One wants the learning rate to be fast, however, like many things one needs to find a balance between a fast learning rate and an accurate network. Typically, the faster a network converges, the less accurate it will be. Moreover, a slow converging network leads to larger training time and a higher chance of overtraining. Thus, the learning rate is desired to be fast to decrease training time, but slow enough that the network is accurate.

3.5.2. Learning Laws

Some of the more common learning laws/rules are summarized below. There are endless patterns that can be created when developing an ANN, however only a set number of learning laws/rules exist. Some of these laws have been active for 60 years. They obviously have been improved throughout time, but their basic concepts will be quickly summarized below.

Hebb's Rule: Possibly the oldest rule used in neural networks. In 1949 in the book titled *The Organization of Behavior*, Hebb's stated "When an axon of cell A is near enough to excite a cell B and **repeatedly** and persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased". Thus, a neuron's interaction with another neuron can be strengthened merely through repeated interaction. This is seen in ANNs when the weight is changed and strengthened to inputs that have a heightened connection to the output.

Delta Rule: The weighting factors of the system are continuously altered based on the difference (delta) between the expected output and the predicted output. The delta rule is also referred to as the least mean square learning rule as well as the Widrow-Hoff Learning Rule

Gradient Descent Rule: The weighting factors of the system are modified in an amount proportional to the first derivative of the error with respect to the weight. The goal of this learning rule is to minimize the first derivative, meaning that the error is constant.

Kohonen's Learning Law: Kohonen was a Finnish professor who used an unsupervised learning technique to train robots. This method of learning was inspired by biological systems.

Back Propagation Learning: Back propagation is the most commonly used generalization of the delta rule. It consists of both a feedforward and a feedbackward phase. This learning rule will be used in our study, and will be discussed in more detail in the section below.

3.6. Back Propagation Network

Back Propagation Neural Networks (BPNN) are a very popular, effective, and easy to learn network for complex multi-layered networks. BPNN's main application is in forecasting and predicting, which makes it an excellent candidate for this study. The pivotal part of the BPNN is the presence of a feedforward and a feedbackward network. The feedforward is the first step in

the BPNN and it works like a typical feedforward network. The input information is propagated throughout the entire network and the difference between the target and predicted output is computed. In the feedbackward phase, the recurring difference computation is now performed in a backwards direction. By incorporating the feedbackward network into this system the network will be able to calculate the error at each node in each layer. The weighting factor will be changed based on the product of the input signal times the error. Thus, at each node the input, target output, and expected output must be stored. In feedforward networks this is difficult to do because the model is unsure about the input and weighting factor used by previous layers. One cannot reassign weights in an accurate fashion because the model does not know which nodes to assign 'praise' for good performance and which nodes to assign 'blame' for poor performance. The network's ability to run backwards will make this task possible. This is not how biology does things, and without the motivation of Hopfield who stated that NNs should be created to solve problems first not to model the brain, this great advancement may never have been developed. Running the network in reverse will allow the network to determine which nodes are best connected to the incoming data. Thus, by using two passes, the backpropagation network can initially calculate the error in the entire system and then can change the weights at each node so that this error is decreased.

This technique is used for predictions because it can produce generalizations rather well. Some of the concerns with this technique are: a large amount of input and target data is needed to train the system, a substantial amount of time is needed in order to successfully train this large abundance of data, and this BPNN can oscillate or converge to local minima rather than the

absolute minima. Software packages developed today have checks to help ensure that the latter does not occur, but it is important for the user to realize that these sources of error do exist.

Chapter 4: Development Stages

Following the literature search, the commercial software package as well as the artificial neural network (ANN) toolbox were explored. Once a familiarity was established for these two packages they were implemented together to create the final ANN(s). Once the ANNs were created, they were placed into a graphical user interface (GUI) so that they could easily be used by future users. The following is a summary of the work that was completed in order to complete the tasks at hand.

4.1. Forward Solution

4.1.1. Forward Solution Introduction

For this research the first goal of the project was to take a reservoir of known properties along with well information for a multilateral well and then to generate a pressure transient curve for that particular well-reservoir combination using an ANN. This idea is portrayed in the diagram below. The reservoir and well properties were randomized and many sets of training data were created in order to adequately train the neural network. An example of training data can be found in Appendix E at the end of this paper. Recall that this process was completed for three different well structures that were discussed and shown in Figure 2 of the introduction section of this paper. The data for these well-reservoir combinations was generated using a numerical model. Essentially the forward solution developed will match the numerical model's results. This idea is especially important for the discussion of the inverse solution, and will be elaborated on further at that point. Once the data was generated it was then fed into the ANN. At this point the structure of the ANN was changed in order to create the best results. Later, the input/output data

was scaled and the addition of parameters known as functional links was made in order to generate more accurate results. Once the ANN had an acceptable error range it was saved and input into a GUI for future use.

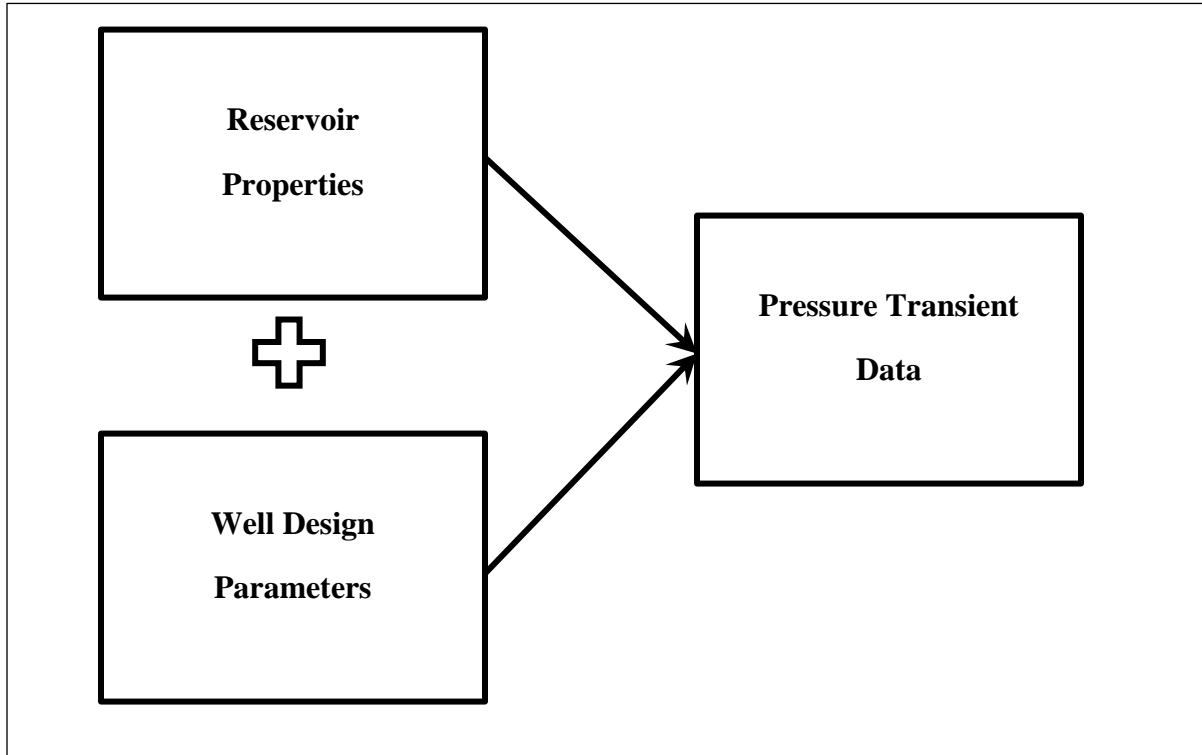


Figure 42: Forward Solution Diagram

4.1.2. Forward Solution Data Generation

In order to create an ANN that is well trained over a large range an immense amount of data that spans this large range had to be created for the ANN. It would not be efficient for an individual to manually construct each case, as in assign reservoir and well properties to the commercial software package, and then run these cases and store the results. This would be a long and tedious process that would take months to complete, but this is not the only reason it should be avoided. The well-reservoir combinations that are wanted for this project need to be unique and

random. The flowrate, main wellbore length, porosities, permeabilities, reservoir thickness, etc. all need to change each run. Random processes are especially difficult for humans to employ successively. As a society, we tend to favor whole numbers, trends, and values that make sense in our minds. In order to add randomness to the data generation, a random number simulator was used to alter all of the design properties each run. In this way, combinations of porosity and permeability that may not have been imagined by an individual, can and will be tested. There were checks placed on all data generated to ensure that the properties were plausible for a dual-porosity tight gas well. Some checks that were implemented included verifying that the fracture permeability was greater than the matrix permeability as well as ensuring that the fracture porosity was less than the matrix porosity. Recall that the fracture porosity is calculated using the bulk volume of the reservoir and not simply the fracture volume. Because the volume occupied by natural fractures is considerably less than the matrix volume, the fracture porosity will always be less than the matrix porosity. Well properties such as flowrate also had to be examined and altered for each run. The flowrate was changed based on the reservoir's porosity, permeability, and thickness. The larger these values were, the larger the possible flowrate could be, and vice-versa.

Initially 500 data sets were created. Upon testing these data sets it was made clear that there were gaps in the ANN's predictability. In order to combat this, additional data sets were created in areas that the ANN seemed to have a lack of predictability. This process was continued adding 10 to 20 data sets at a time. However, after a certain number of data sets were created (~1300), the predictability of the ANN again began to decline. The ANN had too many data sets and was beginning to over train, as was mentioned in a previous section. To combat this overtraining,

three separate data sets consisting of 280, 560, and 1120 trials were created. Each set was individually tested and the results showed that the ANN with 560 data sets had the best initial results for the inverse solution and the ANN with 1120 data sets had the best initial results for the forward solution.

Once the data sets were created the ranges for each parameter were plotted to ensure a random distribution. Many of the parameters, as mentioned in the third chapter, span multiple orders of magnitude. Upon initial inspection of the training sets the entire data range seemed to be well represented. However, once the reservoir parameters, namely the matrix and fracture permeability, were plotted on a semi-log plot, gaps in the training data were noticed. The random number generator used worked very well for all parameters that did not span multiple orders of magnitude. In order to overcome the limitation of the random number generator, the ranges of the parameters needed to be made smaller. However, in order to have a well performing and more applicable ANN this would not be an option. In order to overcome this dilemma, the matrix permeability data range was split into many smaller ranges, each covering half of an order of magnitude. The fracture permeability was then split into four separate ranges that each spanned roughly one order of magnitude, and would be altered for each matrix permeability range. The remaining reservoir properties were then varied using the random number generator. Typical plots of a property's distribution are shown in Figure 43.

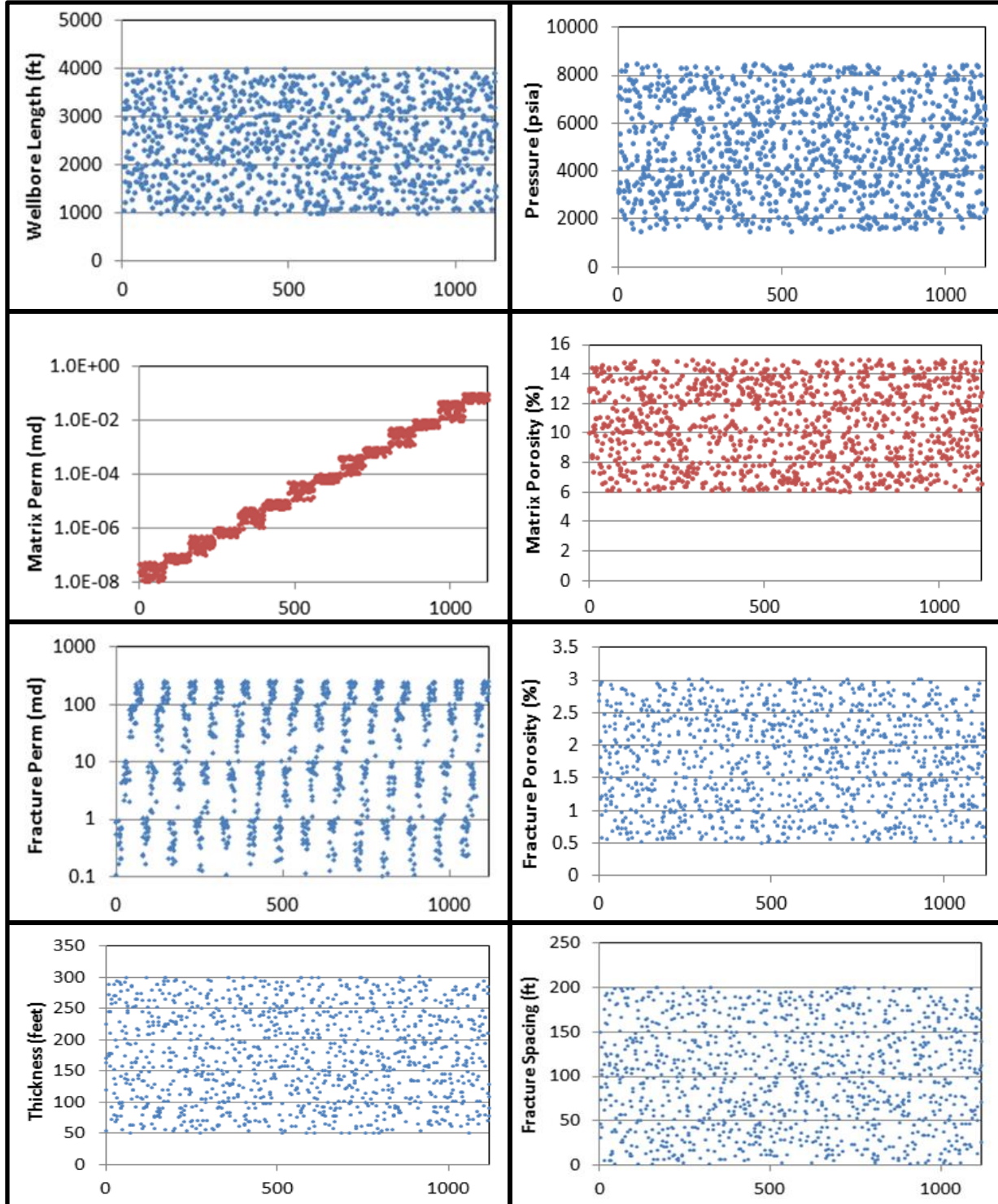


Figure 43: Reservoir/Well Data Distribution

4.1.3. Forward Solution ANN Training

Once the data for the ANN was created, it could be input into the ANN. For the forward solution the input data included: main wellbore length, individual lateral length, number of laterals, initial pressure and temperature, flowrate, reservoir thickness, matrix and fracture porosity and permeability, and fracture spacing along with the addition of functional links. The outputs of the ANN were the pressure transient data. Initially when creating the forward solution ANN, over 250 data points were included in the output solution. It was quickly discovered that these extra data points, while helping to smooth out the appearance of the PT data, made creating an accurate solution for all data sets very difficult. The fewer number of output data points, the easier the ANN is to train. After multiple trial and error attempts it was decided to only have eight output data points. The pressure data predicted was at initial time of the drawdown test, after one day, then at 10, 20, 30, 40, 50, and 60 days. While there are fewer output data points chosen for the solution, these eight data points are much more accurate than the 250+ data points used in previous runs. The accuracy gained by having fewer output data points outweighed the loss in smoothness of the PT data, and that is why less data points were sought during the creation of the final ANN. A summary of the inputs and outputs for the forward solution for all three wellbore structures is shown in the table below.

Table 6: Forward Solution Input/Output Summary

Wellbore Structure	Inputs	Functional Links	Outputs	Data Scaling
1L	$P_i, k_f, k_m, \phi_f, \phi_m, FS, h, Q, \text{Wellbore}$	$P_{60}, P_i - P_{60}, k_f * h, k_f * \phi_f, P_i^2, P_{60}^2, FS^2, h * \phi_m, \text{sqrt}(Q), \frac{k_f}{k_m}, Q * \left(\frac{P_i}{P_{60}}\right)$	$P_i, P_1, P_{10}, P_{20}, P_{30}, P_{40}, P_{50}, P_{60}$	$\log(INPUT)$
2L	$P_i, k_f, k_m, \phi_f, \phi_m, FS, h, Q, \text{Well}$	$P_{60}, P_i - P_{60}, k_f * h, k_f * \phi_f, h * \phi_m, \text{sqrt}(Q), FS^2, P_i^2, P_{60}^2, \frac{k_f}{k_m}, Q * \left(\frac{P_i}{P_{60}}\right)$	$P_i, P_1, P_{10}, P_{20}, P_{30}, P_{40}, P_{50}, P_{60}$	$\log(INPUT)$
3L	$P_i, k_f, k_m, \phi_f, \phi_m, FS, h, Q, \text{Well}$	$P_{60}, P_i - P_{60}, k_f * h, k_f * \phi_f, h * \phi_m, \text{sqrt}(Q), FS^2, P_i^2, P_{60}^2, \frac{k_f}{k_m}, Q * \left(\frac{P_i}{P_{60}}\right)$	$P_i, P_1, P_{10}, P_{20}, P_{30}, P_{40}, P_{50}, P_{60}$	$\log(INPUT)$

In order to train the forward solution ANN many intermediate steps were taken prior to the final framework being discovered. Due to the time constraint of these runs each run was only ran one time. As mentioned previously, during the creation of the ANN the weights and biases of individual neurons and their connections are changed in a somewhat random fashion. The training and learning rules implemented have checks built into them so that the ANN does not converge at a local minimum, however, it can happen at times. Because of this randomness, it was discovered that averaging the results of three separate runs and then comparing these averaged results was a much better strategy to accept or reject certain ANN structures or

functional links. However, that strategy was not used for the forward solution due to the aforementioned time constraints.

4.1.3.1 ANN Structure

The first step in creating a suitable ANN is developing the structure of the ANN. At first it is unknown which ANN structure will lead to the best results given a certain set of input/output data. A few basic strategies that were mentioned previously are used to quickly converge on the best structure. Keep in mind that the input/output data has not been modified at this point. In hindsight it would have been more efficient to remove some of the output data points in order to have less training time for the ANN. However, this is a unique case. When the reservoir parameters are being predicted one does not want to remove any of those parameters to cause the training time to decrease because certain structures will promote the prediction of certain parameters and weaken the prediction capability of others. Moreover, there are relatively few output reservoir points compared to the 250+ pressure transient data points in this forward solution, and removing them would not yield a considerable decrease in training time.

The basic strategies as discussed previously relate to the complexity of the problem. The more complex the problem, the more complex the ANN structure. For all cases a single hidden layer was always tried first. Due to the simplicity of this ANN, the training time was significantly reduced versus more complex ANN structures. Next, the number of neurons in the hidden layer was varied from low to high. One typical rule of thumb is that the number of neurons needed within a network should roughly be equal to one half of the summation of the number of input

and output data points. It was quickly discovered that this rule of thumb was more of a starting point and not an end result, and typically a sensitivity study was performed for the number of neurons in order to discover what number of neurons led to the smallest error within the network. Large changes in the number of neurons were made initially to discover the trend. Later, smaller changes in the number of neurons were used to fine tune the ANN. If the ANN had insufficient results at this point then an additional hidden layer would be added and this process would be repeated. This process was continued until the ANN error hit a minimum value. One may think that a more complex neural network will always have better predictability, but the truth to the matter is that when the ANN structure is made more complex the relationship between the input and output variables also becomes more complex. Complex networks are used to pick out subtle nuances in the data that may not be distinguishable any other way, and are thus not needed for simpler problems. Keeping a network simple essentially means that the larger more prominent connections will always dominate. If these connections are sufficient to create an accurate ANN, then there is no need to create a more complex network.

After the number of hidden layers and neurons were altered it was discovered that a separate strategy had to be implemented in order to create a more accurate ANN. Something had to be added to the network in order to enhance its predictability. This often came in the form of functional links. Note that once the best ANN structure was discovered for a particular training set the structure would not be changed. It would be extremely tedious and time intensive to vary all of the parameters involved in making an ANN each time one of those parameters was changed. Thus, once the best structure of the ANN was discovered it was kept constant. At times

the structure would be altered to ensure that the results being generated were the best possible, but generally it was left constant after this point.

Note that during the process a sensitivity study was performed on the transfer and training functions in order to decide which functions would work best with the network in this study. The transfer functions that were used most commonly in this study were ‘tansig’, ‘logsig’, and ‘purelin’, which are shown in Table 5. Typically, if one’s data is compressed between -1 and +1, tansig should be the first choice as a transfer function, especially for the first hidden layer, because this non-linear function also spans the range of -1 to +1. If one’s input data is scaled from zero to one, then logsig is typically used. However, this is simply a rule of thumb and is not always the case, as will be seen when examining the final structure of the created ANN. The order and the combination patterns of transfer functions are massive even when considering a relatively simple ANN structure. Initially each transfer function within MATLAB was used in a single layer neural network. The results showed that tansig, logsig, and purelin were superior to all other transfer functions, and this is why they were chosen to be used in the study. Multiple combinations of the transfer functions were used in all of the networks created. Typically only the two non-linear transfer functions were used in each network because this produced the best results.

Lastly, the training/learning function was also altered during this process. Once again a sensitivity study was performed on a majority of the training/learning functions. It was quickly discovered that the two training functions that produced superior results were ‘trainscg’ and

‘trainrp’. Both of these training functions were developed with speed and accuracy in mind. ‘scg’ stands for scaled conjugate gradient and was developed to combat the typical issue of training functions: their success was highly dependent on user inputs which no theoretical basis for choosing the correct input existed. The scaled conjugate gradient converges much faster than most other back propagation training functions and is not as sensitive on user input (Meiller, 1993). The user input was varied for this training function and it was discovered that the default values gave the best results, and they were kept constant throughout the remainder of the study. Secondly, the ‘rp’ in trainrp stands for resilient propagation, and this function performs a direct adaptation of the change in weight based on gradient information at each point. This training function’s main goal was to avoid obscure results generated by the gradient’s behavior (Riedmiller & Braun, 1993). Thus, if the gradient changes drastically the training function still can make accurate predictions whereas previous training functions struggled with this. Again, the input data for this training function could be altered, but it was discovered that the default values worked the best for our particular network.

Once the training functions were chosen they too needed to be optimized. Convergence criteria are very important parameters for any training function and sensitivity studies were performed on all of the valid convergence methods. The number of epochs or iterations is the first convergence criteria. It is important to give the network enough iterations in order to successfully converge and often times a maximum number is only used in order to stop a network that will never converge. Thus, once the typical number of iterations was discovered the convergence criterion was set to a value five times as large as this. Time can also be used as a convergence criterion but because the time may run while the neural network is not (i.e. the

computer is in sleep mode) it was decided to only use the total number of iterations as a convergence criterion and not the time. Next, the performance of the ANN can be used as convergence criteria. As the iterations increase, the ANN will begin to approach the solution. At a certain point the change in accuracy from one iteration to the next will become very small. There are two forms of convergence criteria that can be established from this one phenomenon. The first convergence criterion is that if the gradient error, or change of the error between runs, reaches a certain value the network will stop training and the ANN can be saved at this time. Thus, if the change from one run to the next is extremely small for subsequent runs one assumes that the ANN will not improve further and to avoid overtraining, will stop training at that time. Secondly, another convergence criterion for an ANN occurs when the gradient does not improve between subsequent iterations. Essentially each network structure has a minimum error value that it will converge to. Typically, this error value will not be low enough for the ANN to be considered well trained, and thus one needs to have another check so that the network does not continue to run without improving. Once the ANN reaches a minimum training error, all subsequent iterations will produce results with a higher error. Thus, one needs to select the number of subsequent iterations that the error is not decreased as a convergence criterion. If the number selected is too small the ANN may converge to a local minimum. If the number selected is too large the ANN will become overtrained and its predictability will be severely limited. The diagram below shows this phenomenon and may help better explain this process. A sensitivity study was performed for this convergence criterion in order to ensure the best results for each ANN structure. Lastly, the final convergence criterion is the error between the expected and predicted output. One wants to select the appropriate error that is not too high, leading to a poorly trained network, and is not too low, leading to an overtrained network. This value is

difficult to predict because it depends on parameters such as the total number of outputs as well as the individual well structure. These convergence values need to be altered as the ANN structure changes to ensure that the ANN is neither being overtrained or poorly trained.

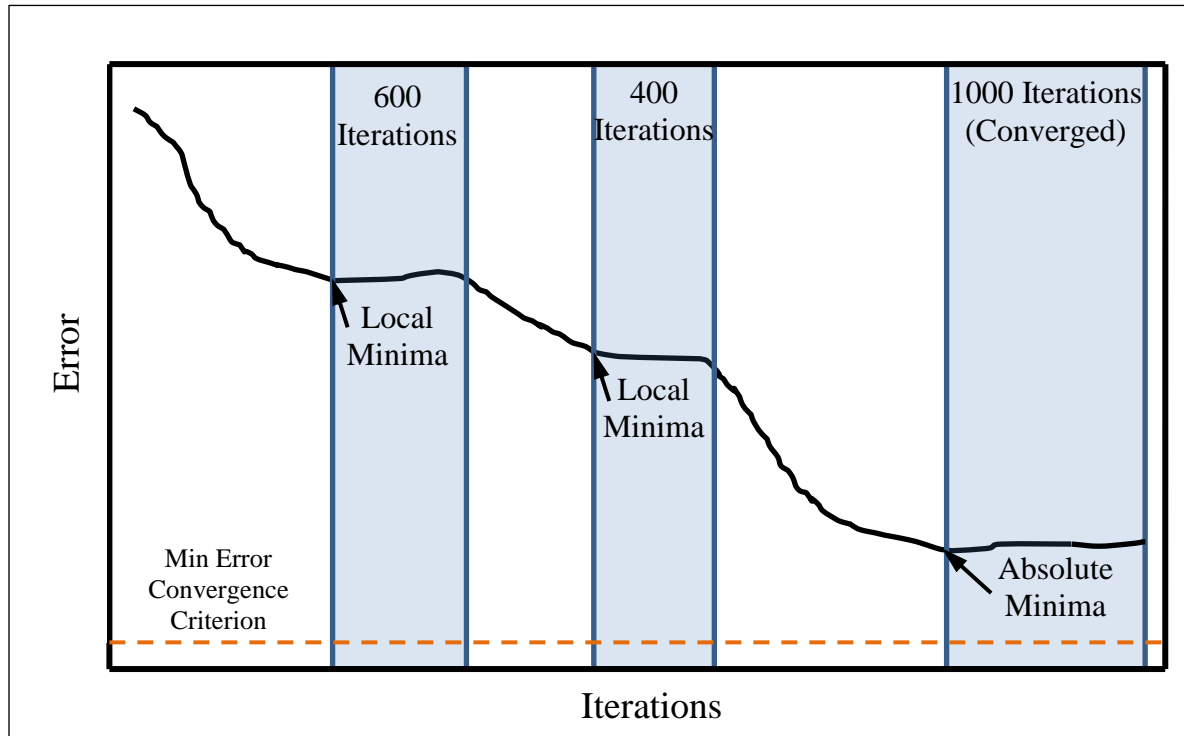


Figure 44: Error vs. Iterations Plot.

4.1.3.2. ANN Input/Output Data Manipulation

The second step in training a supervised neural network is to begin manipulating the input and output data. Simple checks can be performed to discover which input parameters help or hurt the ANN. One can simply remove certain input data and train the ANN and compare the results to a control case. Often times certain reservoir parameters seemed to confuse the neural network and led to a higher error. It was discovered that a smaller number of input parameters often led to

better predictions because the strong connections between the input and output data were not clouded by the weak connections of certain input parameters.

The second way that input and output data could be manipulated is through scaling. Recall that the input and output data are typically compressed between values of -1 and +1. Having matrix permeability at values as low as 10^{-8} md but flowrate values as large as 50,000,000 SCF/D will lead to values on the same order of magnitude to not be distinguishable from one another. This scaling would lead to higher issues if negative values were employed into the system as well. For example, if one set 10^{-8} to -1 and 50,000,000 to +1, then two different values of pressure being 5000 psia and 6000 psia would now have values of .0002 and .00024, respectively. Moreover, the values of matrix perm which vary from 10^{-8} to 10^{-2} md would also be scaled, leading to a limitation of predictability. To combat these issues, individual values can be skewed by multiplying or dividing by a constant, or the values can be scaled in a non-linear fashion using the natural logarithm function or raising parameters to a certain power. What typically worked best was scaling either the entire input data set, entire output data set, or both using the natural log function. In addition, the matrix permeability was also typically multiplied by a scalar ranging from one-billion to ten-billion. Note, that scaling the data can be performed prior to optimizing the structure of the ANN. But as with all of these optimization strategies, the scaling of data should be periodically revisited and reassessed to ensure that the scaling technique being utilized produces optimum results.

4.1.3.2. Addition of Functional Links

The structures of an ANN as well as the presentation of the input/output data have limited effects on the ANN's performance. Typically in order for an ANN to converge to an acceptable level of error the use of functional links in either the input or output data must be utilized. Functional links are user inputs that are implemented as an attempt to change the weights/biases of the network or to amplify the effect of a particular input. Recall the sensitivity study performed on the reservoir properties in Chapter 2. This study was also used to generate functional links that were expected to have a strong connection to the output data. The reservoir or well properties that had the largest impact on the PT curve were often used separately or in conjunction with one another as functional links, and these combinations typically led to the best predictions.

When selecting functional links one must study the error in the neural network. If the initial pressure is not being predicted correctly, then more weight needs to be added to this parameter. One could attempt to add a functional link of P_i^2 or some other scaled value to enhance the predictability of the initial pressure. If the PT curves are consistently decreasing at higher rates than the expected PT data then perhaps the thickness of the reservoir or the matrix permeability's weights are too small, and need to be amplified. Also, one can use well test analysis concepts to develop functional links. The general form of the diffusivity equation for a dry gas reservoir is:

$$\nabla \cdot \left[\frac{P}{\mu_g Z} \nabla P \right] = \frac{\phi C_t P}{kZ} \frac{\partial P}{\partial t}$$

One may be able to use the concepts of this equation to generate functional links that will decrease the error of the ANN. The range of functional links that can be input into a system is

limitless and the effectiveness of these functional links will vary significantly. This is typically a trial and error procedure, but one should understand which input variables have the highest impact on the output results. Starting with these parameters to create functional links may lead to a well-trained ANN in a relatively short amount of time.

After the correct functional links are implemented, the ANN is now ready for its final training. Before this is completed however, another sweep of the ANN structure and input/output data representation should be performed to ensure that the best ANN is being created.

Note that ANN optimization can be completed in any order. Functional links can be created first, followed by ANN structure optimization, and finalized with manipulation of input/output data. However, once the first step is completed it is typically not continuously altered but is periodically checked. This limits the amount of ANN scenarios that need to be tested/analyzed before testing/analyzing the next ANN setup. In this case thousands of runs were needed in order to optimize all of the neural networks for each of the wellbore structures. However, once an ANN was created that could successfully predict the forward solution to a particular wellbore structure, converting that ANN to work with the other two wellbore structures was relatively simple.

A thorough summary of each forward solution is presented in the Table in Appendix A. The results of these final forward solutions will be discussed in more detail in Chapter 5.

4.2. Inverse Solution

4.2.1. Inverse Solution Introduction

The completion of the forward solution ANN led to a better understanding of how ANNs operate as well as what are the best ways to structure and enhance an ANN's predictability. However, for the inverse solution, since well design properties and pressure transient data were provided as the input, and the output was the reservoir properties, a different ANN structure would need to be created to solve this separate problem. This relationship is portrayed in Figure 45 below. Note that due to the overwhelming similarities between the processes of the generating an ANN for the forward or the inverse solution that this section will focus mainly on summarizing results rather than explaining the methodology once again. For a reference to the methodology on how the inverse solution was solved please see the Forward Solution sections above.

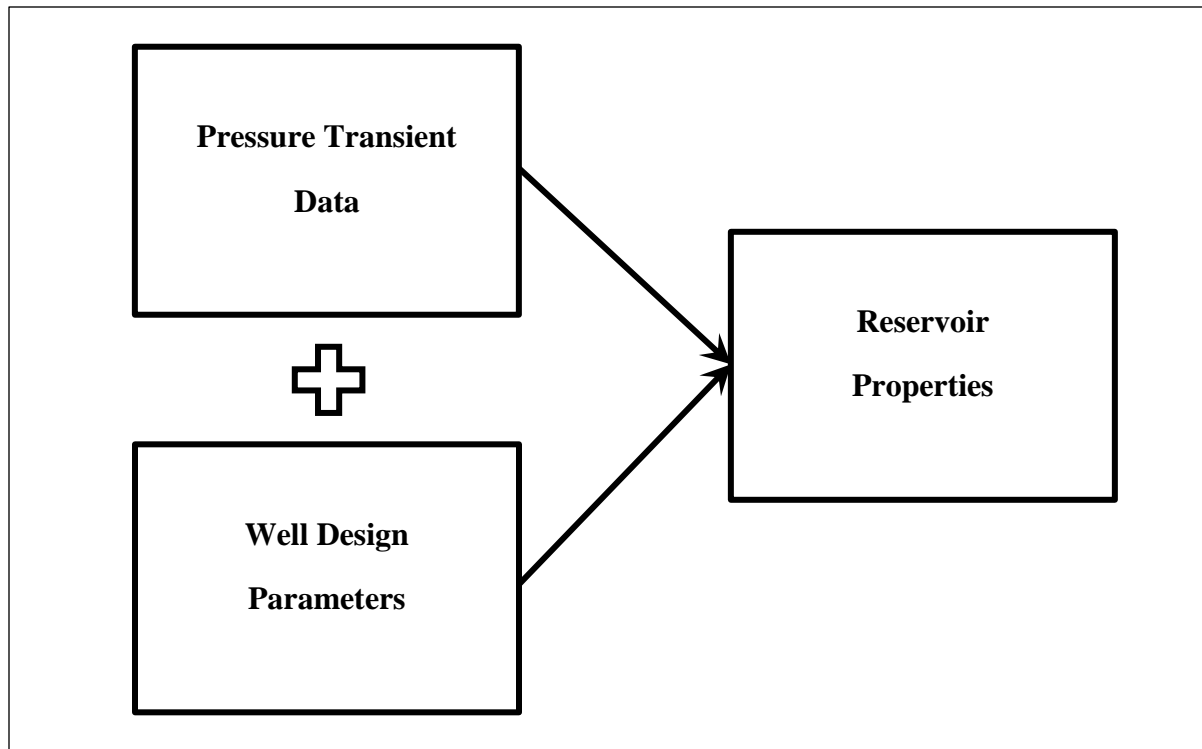


Figure 45: Inverse Solution Diagram

4.2.2. Data Generation

The data generated for the forward solution were repurposed and used within the inverse solution. The only difference is the input and output parameters changed. These changes can be done in one of two ways. Either the data can be transferred between Excel workbooks and then resaved, or the data manipulation can be done entirely in MATLAB. Typically, it is better to perform the data manipulation in MATLAB because then the data is never altered, just manipulated. When changing the data in Excel, one can make an irreparable mistake, and at that point the data set would have to be reproduced in its entirety.

As mentioned previously, a numerical model is being implemented to produce the data that will be used in the training of our system. This is important to note because when well tests are performed, a numerical model can be used to history match the pressure transient curve. The history match will provide one with a set of reservoir properties that can be used in further modeling procedures. One main issue with this process is that it can take a considerable amount of time to perform the history match. Also, numerous reservoir combinations exist that will give a similar PT curve. Thus, one's solution when performing a history match is not unique and could actually be a poor representation of the reservoir. By providing the ANN with data taken from a numerical model a tool will be created that will be able to perform 'history matching' very quickly. Moreover, because the ANN will create its own set of algorithms to connect the input to the output data, there is a larger possibility that the output results, while still being a unique solution, will give a more accurate representation of the reservoir. The user could then take the values produced by the ANN, along with their respected average error, and feed them into a more accurate model. This will lead to a more accurate representation of the reservoir they are working in relative to solely performing a history match.

4.2.3. Inverse Solution Training ANN

Once the Inverse Solution data was organized it could be implemented into the actual ANN. The inputs consisted of PT data from initial pressure to pressure after 60 days along with wellbore design parameters. The PT data was varied often to assess which data points were necessary and which merely clouded the input/output relationships. Input data from the first few days was initially focused upon. If the ANN could be trained with data from the first days rather than all 60 days then this would lead to lower expenses for the operating company, making our tool more

desirable. However, it was discovered that the entire 60 days of PT data were needed in order to create an ANN with an acceptable error. However, pressure points at every hour or even every day were unnecessary. The ANN performed best with limited PT input data. Thus, the final input data to the ANN consisted of initial pressure and pressures at 20, 40, and 60 days. The wellbore properties used in the input were the main wellbore length, individual lateral length, and flowrate. The initial output consisted of matrix and fracture permeability and porosity, thickness, fracture spacing, and reservoir temperature. The same training strategy that was used for the forward network was used for the inverse network.

It was discovered that the entire range of reservoir properties could not be accurately predicted no matter the ANN structure, input/output scaling, or functional links added. To combat this issue reservoir thickness was removed from the output and placed in the input. The reservoir thickness is a readily attainable property from well log analysis, and recall that from previous sections was proven to have a large impact on the reservoir's performance. Simply adding the thickness into the input did not significantly improve the predictability of the ANN and more changes had to be made. Next, the output/input data was separated into 'low matrix permeability' and 'high matrix permeability' cases. The low matrix permeability spanned from 10^{-8} md to 10^{-4} md while the high matrix permeability case spanned from 10^{-4} md to 10^{-1} md. Again, the ANN results were improved, but not to an acceptable level. It was discovered that the matrix and fracture porosity, fracture spacing, and reservoir temperature had the lowest training error while the matrix and fracture permeability had the highest training error. In order to overcome this issue it was decided to remove the matrix and fracture permeability from the ANN altogether, and only predict the matrix and fracture porosity, fracture spacing, and reservoir temperature.

After the error for these predicted parameters was minimized, a new ANN would be created with only the matrix and fracture permeability in the output. The difference would be that the input would now include the original PT data, the well design properties, **and** the previously calculated reservoir properties. This new ANN structure is shown in the diagram below. This method was implemented and the results were immediately improved. The addition of the reservoir properties and functional links created with these properties led to a much better prediction of the matrix and fracture permeability.

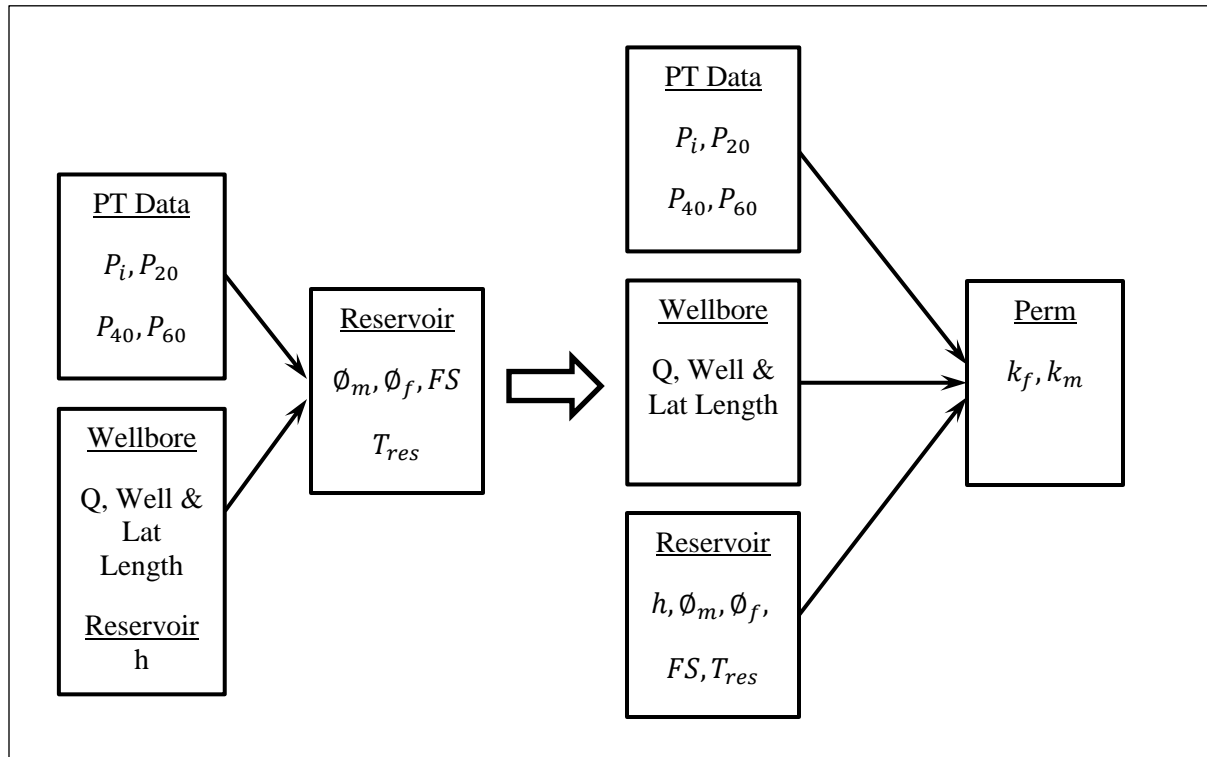


Figure 46: Inverse Solution ANN Final Structure

Splitting up the output data in this way helped the fit considerably, but there were some complications. An ANN that is trained without splitting up the outputs is typically more desirable because it is more adept with its predictions. Also, each wellbore structure will now

require four separate ANNs in order to span the entire training data range. The ANNs created will be: Reservoir Predictions (High Perm), Permeability Predictions (High Perm), Reservoir Predictions (Low Perm), and Permeability Predictions (Low Perm). Recall that for the forward solution there was only one ANN for each wellbore structure. The increase in number of ANNs will lead to an increase in the complexity of the GUI created at the end of this project. The table below shows the input, output, functional links, and data scaling for all ANNs of the inverse solution for each wellbore structure.

Table 7: Inverse Solution Input/Output Summary

Wellbore Structure	ANN	Inputs	Functional Links	Outputs	Data Scaling Implemented
1L	Reservoir: High k	$P_i, P_{20}, P_{40}, P_{60}, h, Q, Well, Lat$	$\frac{Q}{P_i}, P_i - P_{40}, P_{40} - P_{60}$	$\phi_m, \phi_f, FS, T_{res}$	$\log(INPUT), \log(Output)$
	High Perm	$P_i, P_{20}, P_{40}, P_{60}, \phi_m, \phi_f, FS, T_{res}, h, Q, Well, Lat$	$P_i - P_{40}$	k_m, k_f	$\log(Output), k_m * (10^9)$
	Reservoir: Low k	$P_i, P_{20}, P_{40}, P_{60}, h, Q, Well, Lat$	$\frac{Q}{P_i}, P_i - P_{40}, P_{40} - P_{60}$	$\phi_m, \phi_f, FS, T_{res}$	$\log(INPUT), \log(Output)$
	Low Perm	$P_i, P_{20}, P_{40}, P_{60}, \phi_m, \phi_f, FS, T_{res}, h, Q, Well, Lat$	$P_i - P_{40}$	k_m, k_f	$\log(Output), k_m * (10^{10})$
2L	Reservoir: High k	$P_i, P_{20}, P_{40}, P_{60}, h, Q, Well, Lat$	$\frac{Q}{P_i}, P_i - P_{40}, P_{40} - P_{60}$	$\phi_m, \phi_f, FS, T_{res}$	$\log(INPUT), \log(Output)$
	High Perm	$P_i, P_{20}, P_{40}, P_{60}, \phi_m, \phi_f, FS, T_{res}, h, Q, Well, Lat$	$P_i - P_{40}$	k_m, k_f	$\log(Output), k_m * (10^9)$
	Reservoir: Low k	$P_i, P_{20}, P_{40}, P_{60}, h, Q, Well, Lat$	$\frac{Q}{P_i}, P_i - P_{40}, P_{40} - P_{60}$	$\phi_m, \phi_f, FS, T_{res}$	$\log(INPUT), \log(Output)$
	Low Perm	$P_i, P_{20}, P_{40}, P_{60}, \phi_m, \phi_f, FS, T_{res}, h, Q, Well, Lat$	$P_i - P_{40}$	k_m, k_f	$\log(Output), k_m * (10^{10})$
3L	Reservoir: High k	$P_i, P_{20}, P_{40}, P_{60}, h, Q, Well, Lat$	$\frac{Q}{P_i}, P_i - P_{40}, P_{40} - P_{60}$	$\phi_m, \phi_f, FS, T_{res}$	$\log(INPUT), \log(Output)$
	High Perm	$P_i, P_{20}, P_{40}, P_{60}, \phi_m, \phi_f, FS, T_{res}, h, Q, Well, Lat$	$P_i - P_{40}$	k_m, k_f	$\log(Output), k_m * (10^9)$
	Reservoir: Low k	$P_i, P_{20}, P_{40}, P_{60}, h, Q, Well, Lat$	$\frac{Q}{P_i}, P_i - P_{40}, P_{40} - P_{60}$	$\phi_m, \phi_f, FS, T_{res}$	$\log(INPUT), \log(Output)$
	Low Perm	$P_i, P_{20}, P_{40}, P_{60}, \phi_m, \phi_f, FS, T_{res}, h, Q, Well, Lat$	$P_i - P_{40}$	k_m, k_f	$\log(Output), k_m * (10^{10})$

4.3. Creation of Graphical User Interface (GUI)

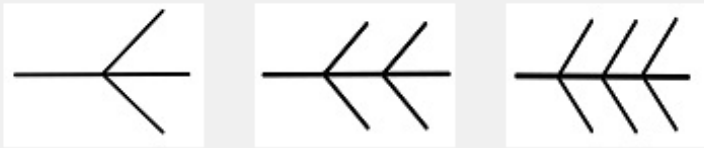
Once the ANNs had been verified, it was time to create a system in which they could be more easily used. To do this a graphical user interface, or GUI, was created. The GUI would be a ‘black box’ to the user. This means that the user would not be able to see how the calculations are occurring and would only be able to input data and see the calculated results.

For the Inverse Solution, the GUI would require PT data and wellbore design data as well as the thickness of the reservoir. For reference, see Table 6. The user would also have to input whether their reservoir was believed to be in the ‘high perm’ or ‘low perm’ region. Once this data was inputted, the GUI would calculate all necessary functional links, place the input data in the correct order, execute the ANN, and display the results. Again, for the inverse solution, the GUI would have to access four separate ANNs in order to create answers for the entire training range of data. The use of global variables was used to transfer these values from one ANN to the next within the MATLAB GUI. The Inverse solution GUI is displayed below in Figure 47.

Inverse Solution
PT and Well Input Data

	User Inputs
Number of Laterals (2, 4, or 6)	2
Expected km Value (md)	1.0000e-03
Pi (psia)	4000
P(psia) @ 20 Days	3750
P(psia) @ 40 Days	3600
P(psia) @ 60 Days	3500
Thickness (ft)	200
Flowrate (MMSCF/D)	25
Main Wellbore Length (ft)	3000
Lateral Length (ft)	1500

Inverse ANN's



Predicted Reservoir Parameters

	ANN Predicted Values
Matrix Porosity (%)	14.5632
Fracture Porosity (%)	4.1114
Fracture Spacing (ft)	32.7288
Reservoir Temp (F)	461.5638
Fracture Perm (md)	10.8658
Matrix Perm (md)	0.0021

Figure 47: Inverse Solution GUI

The forward solution input would consist of the reservoir and wellbore data as shown in Table 6. The GUI would then calculate all functional links, place the input data in the correct order, execute the ANN, and display the results. The results were displayed in both a tabulated format as well as a graphical format. The Forward Solution GUI is shown below in Figure 48.

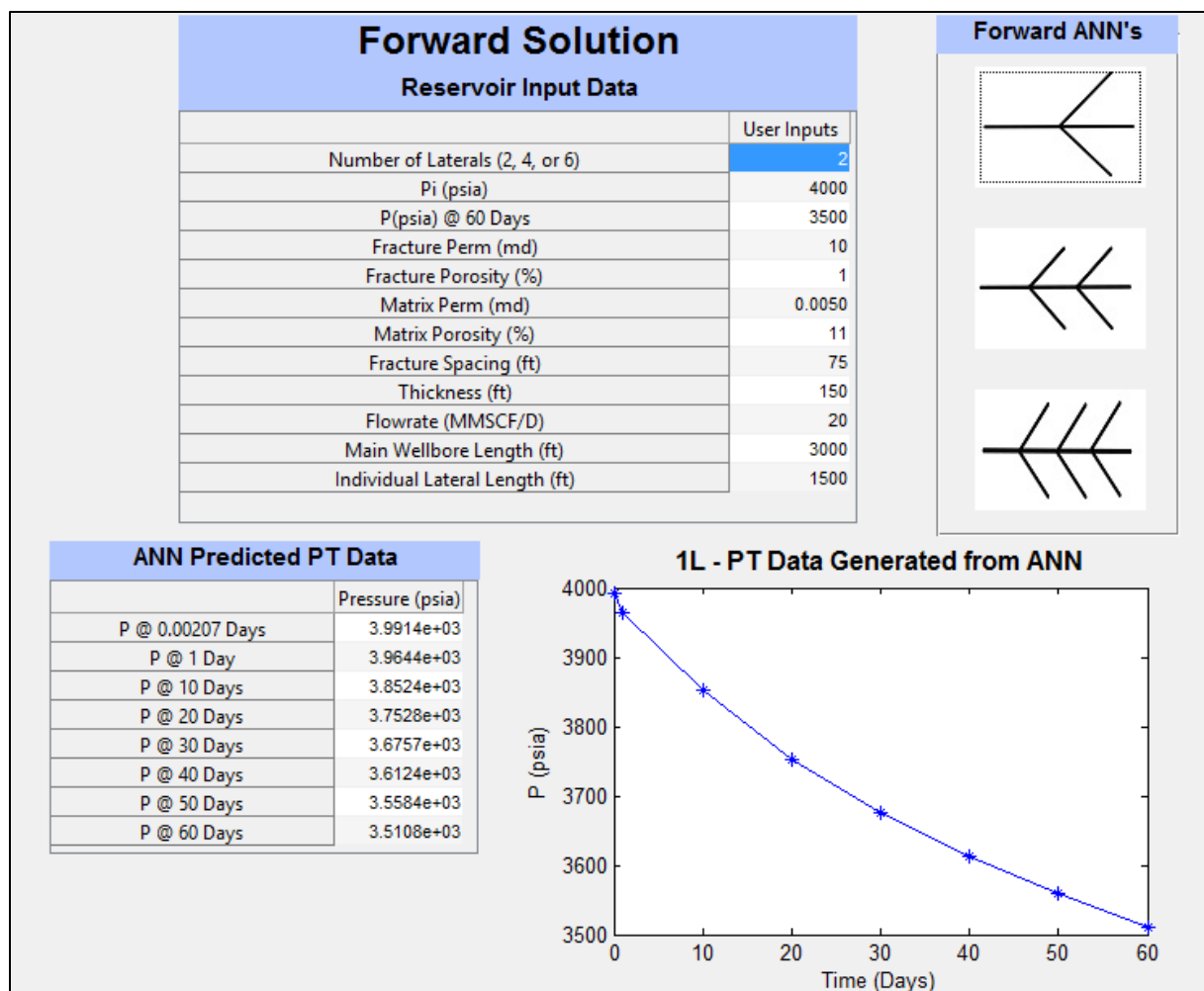


Figure 48: Forward Solution GUI

Chapter 5: Discussion of Results

In this final chapter the results of this study will be discussed. The results generated for all forward cases and all inverse cases are nearly identical, and to avoid repeating, only the ‘single’ lateral geometry will be discussed in this section. The results of the dual-lateral and tri-lateral wellbore geometries are displayed in Appendix B and C, respectively.

5.1. Forward Solution

Figures 49 and 50 display the PT data created by the forward solution ANN. The true data is also displayed on these figures so one can easily compare the results. Figure 49 shows the four best PT predictions whereas Figure 50 shows PT predictions that represent the average predictability of this system. However, in order to get a better understanding of the results the error between the predicted and target PT data will now be discussed.

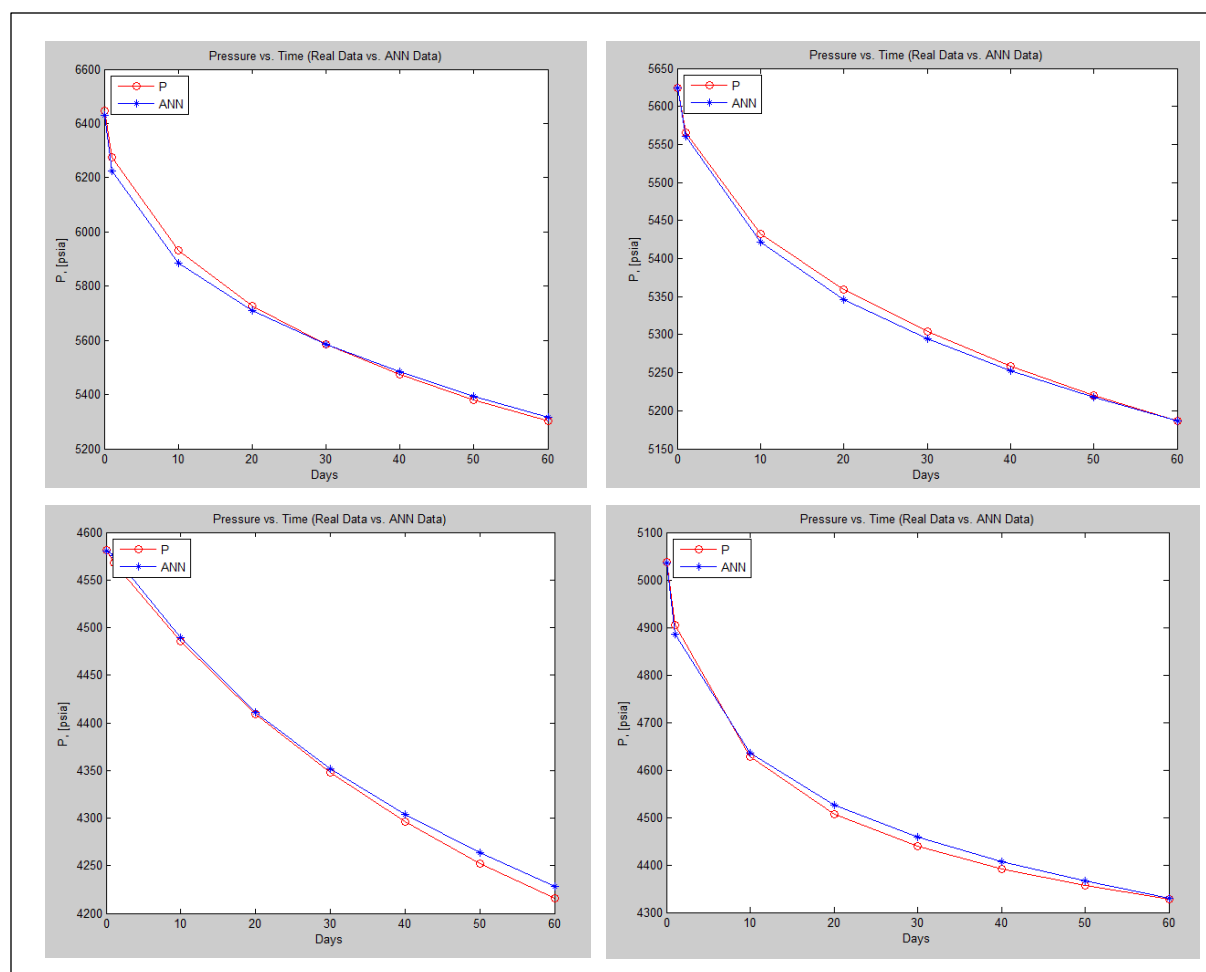


Figure 49: Best Forward Solution PT data fits

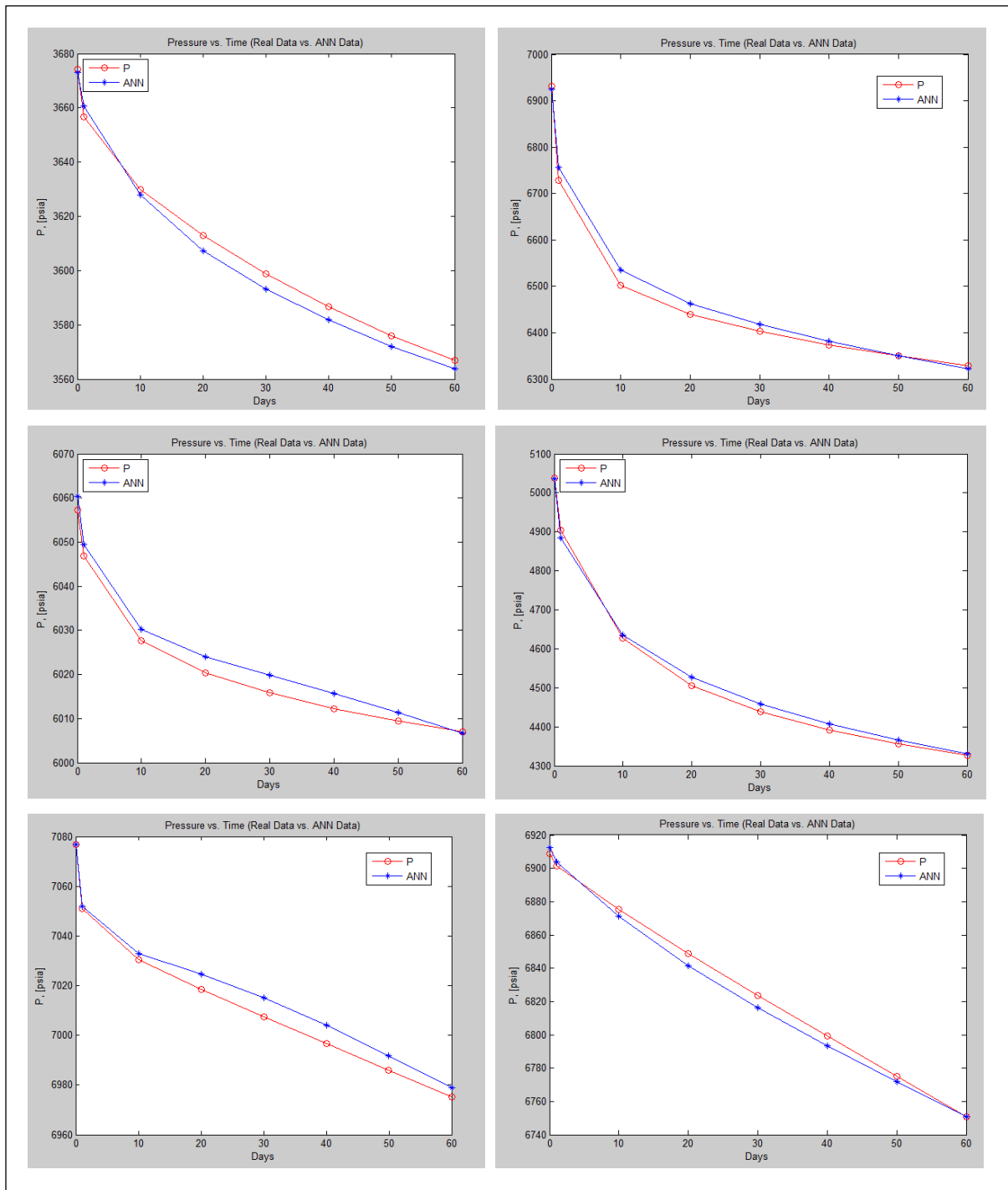


Figure 50: Forward Solution Average PT Data Fits

Figure 51 below shows both the percentage error and absolute error for every testing data set. By examining Figure 51 one can see that the majority of the testing sets have an error less than two

percent and an absolute error less than 80 psia. A histogram of the error is shown in Figure 52. This figure shows that nearly all results, whether it be training data, validation data, or testing data, have an absolute error less than ± 27 psia. The few points outside of this range are anomalies in the results or outliers. Moreover, one can see that the ‘zero-error’ line passes almost directly through the center of the histogram. This indicates that the ANN is very well trained and the majority of predictions deviate very little from the target results. Lastly, Table 8 summarizes the forward solution error. When examining all of the testing data the average error between the prediction and target values is 1.29%. This translates to about a 0.16% difference per data point, or an average error less than 8 psia. The goal of this project was to have an ANN that could predict PT data within 10-15 psia, and these results show that this has been accomplished. The training data performs slightly better than the testing error, as was expected.

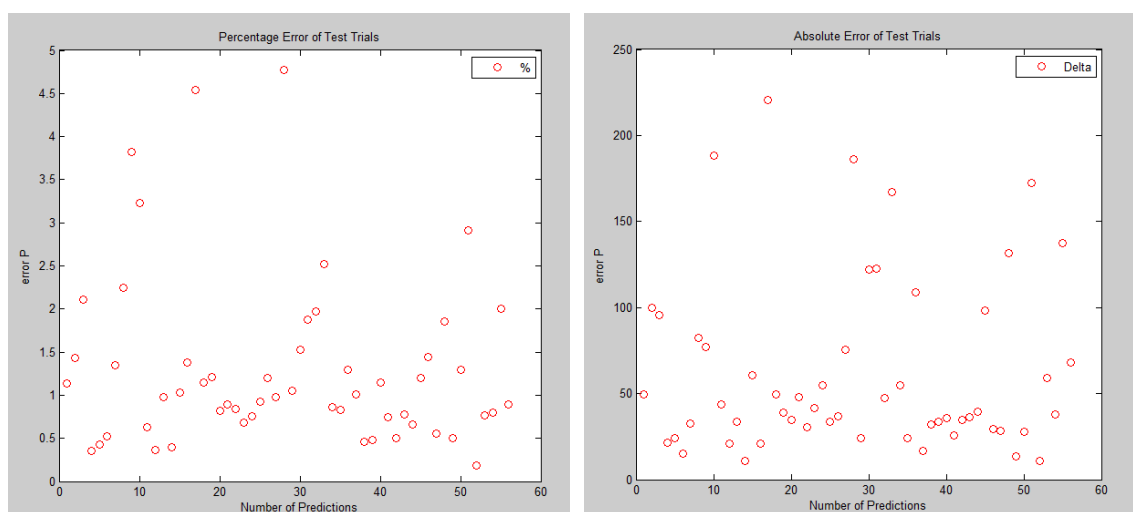


Figure 51: Forward Solution Testing Error

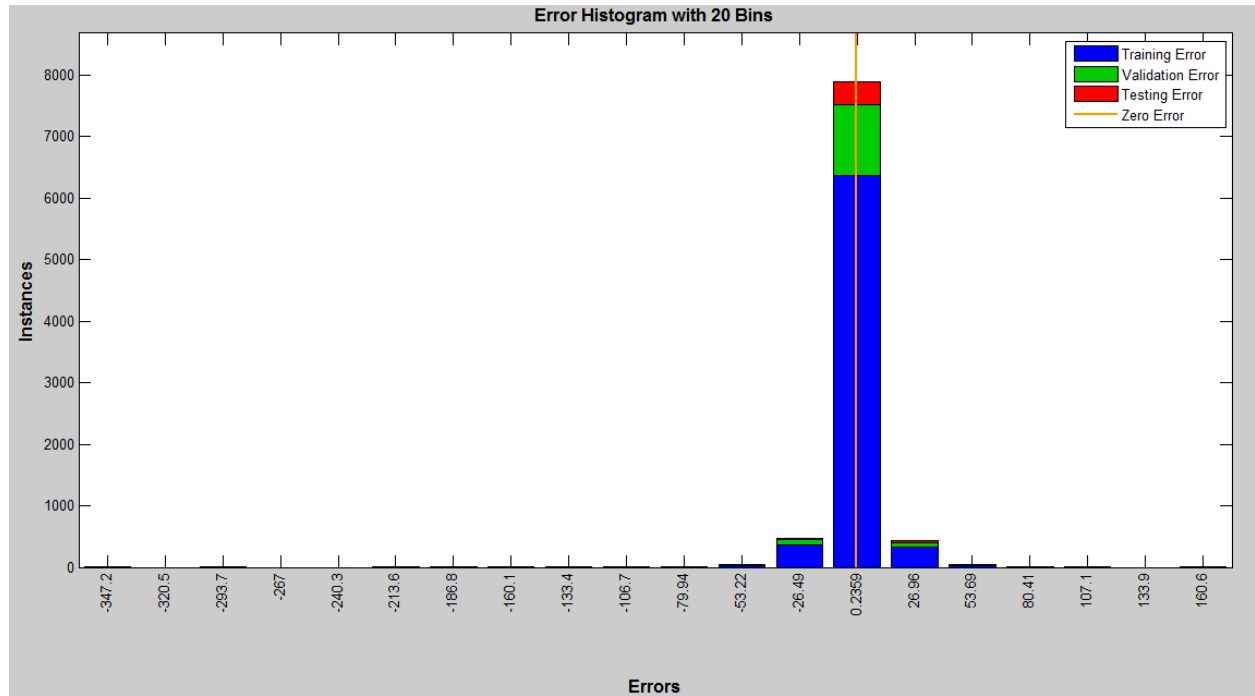


Figure 52: Forward Solution Error Histogram

Table 8: Forward Solution Error Table

	Error	Average (Individual Run)	Average (Individual Data Point)
Testing Errors	% Error	1.29%	0.16%
	Delta Error	61.89	7.74
Training Errors	% Error	1.24%	0.15%
	Delta Error	50.78	6.35

5.2. Inverse Solution

Next, the results of the Inverse Solution for the ‘1L’ geometry case will be discussed.

5.2.1. High Matrix Permeability Reservoir

The results for the high matrix permeability reservoir will be discussed first. Recall that high matrix permeabilities spanned the range 10^{-4} to 10^{-1} md.

5.2.1.1. Matrix and Fracture Porosity, Fracture Spacing, and Reservoir Temperature

As stated previously, the first goal of the inverse solution was to predict the reservoir properties from PT data and wellbore characteristics. Figure 53 displays the training data for the reservoir properties of ϕ_m , ϕ_f , FS , and T_{res} . It is clear that the trained data spans the same ranges as the target data.

Next, Figure 54 displays the testing results. Training data is often a good indication of the ANN’s performance, but the testing data/error should always be used in conjunction with the training data in order to verify the validity of the ANN. From Figure 54 one can see that the ANN seems to have the ability to predict the reservoir property’s values. To further quantify the ANN’s predictability both the testing and training errors have been compiled in Table 9. In order to determine the validity of the testing results, the permeability ANN had to be developed. Once the permeability ANN had been developed then a study of the entire high perm ANN could be

conducted. This study will be explained at the conclusion of the ‘matrix and fracture permeability’ section below.

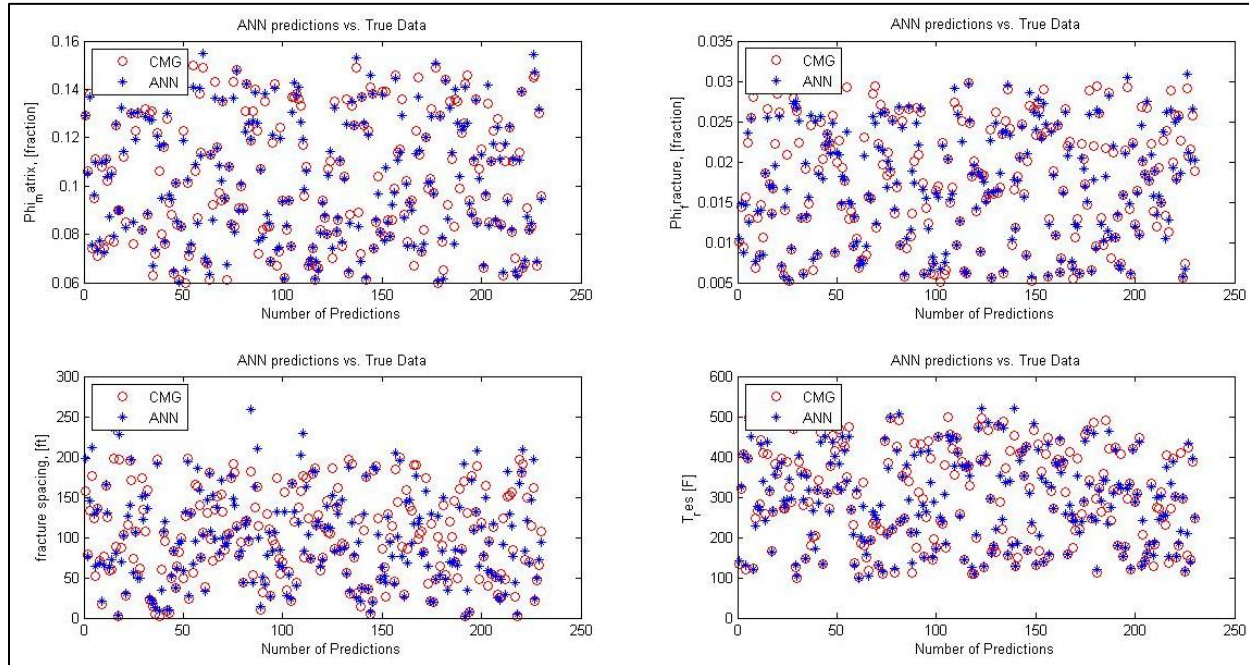


Figure 53: High Matrix Perm: Reservoir Training Results

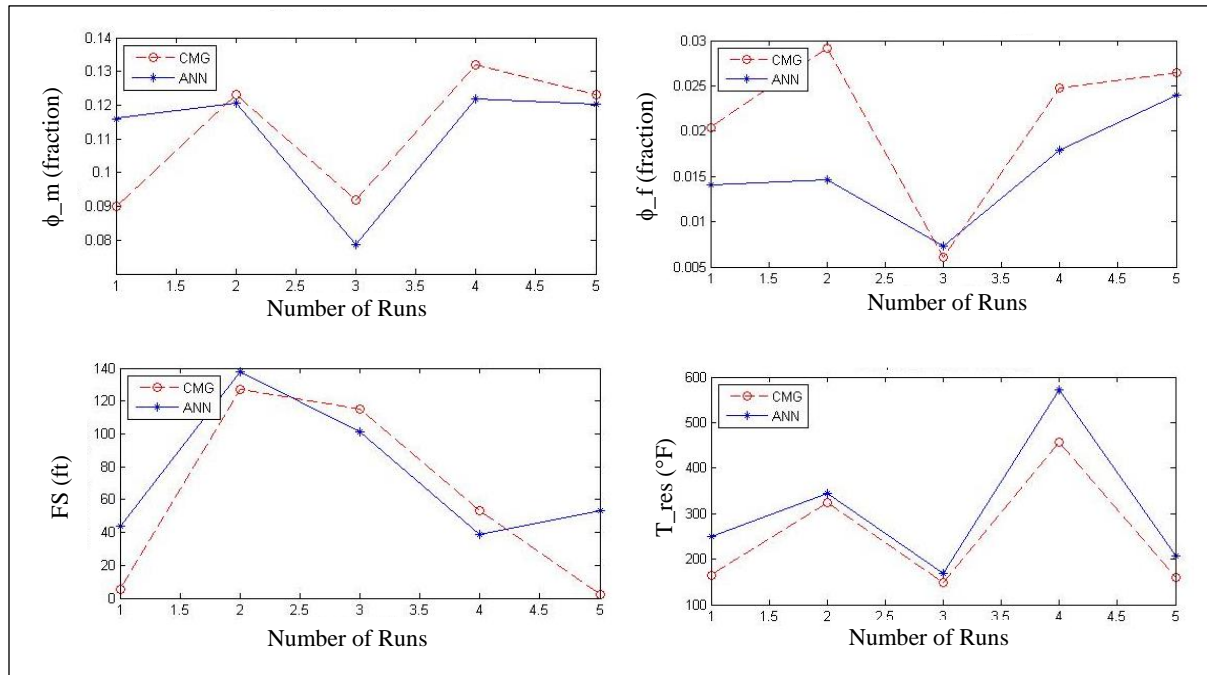


Figure 54: High Matrix Perm: Reservoir Test Results

Table 9: High Matrix Perm: Reservoir Properties Error Table

	Training Average Error (%)	Test Average Error (%)
ϕ_f	5.42	11.43
ϕ_m	2.72	10.99
FS	17.49	27.14
T_res	6.54	25.27

5.2.1.2. Matrix and Fracture Permeability

With the conclusion of the reservoir properties calculations, the matrix and fracture permeability could now be calculated. Again, Figure 55 displays the training data of the ANN. One can see that the ANN's training values span the range for both the matrix and fracture permeability. Figure 56 displays the testing results of the ANN. There are some fluctuations between the predictions and target output, but overall it appears as if the ANN is well suited at predicting matrix and fracture permeability. Table 10 summarizes the test and training error results. Once the high perm ANN was fully created, the results were validated using the methodology explained earlier.

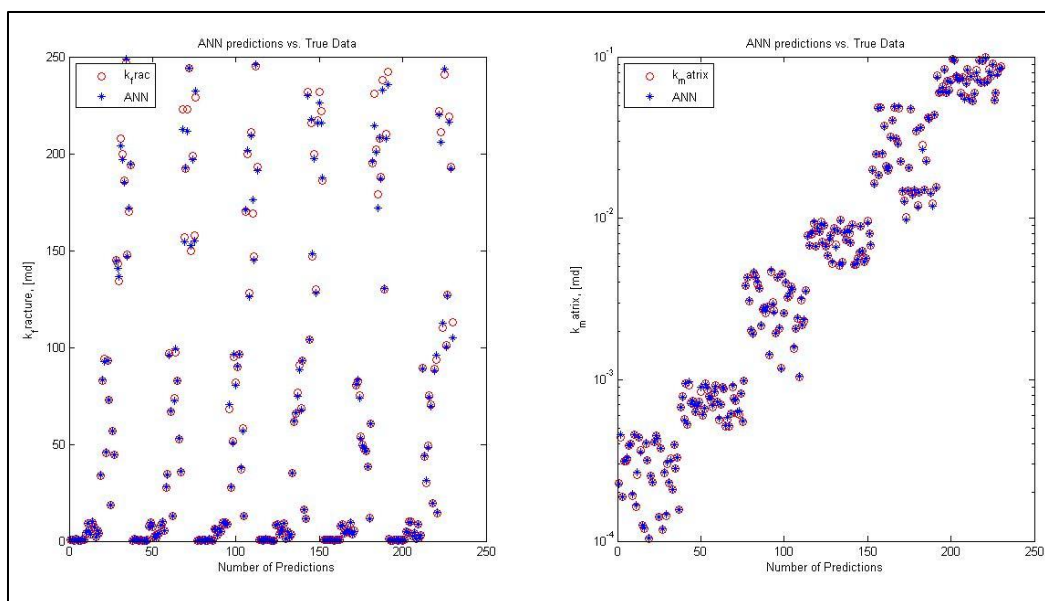


Figure 55: High Matrix Perm: Permeability Training Results

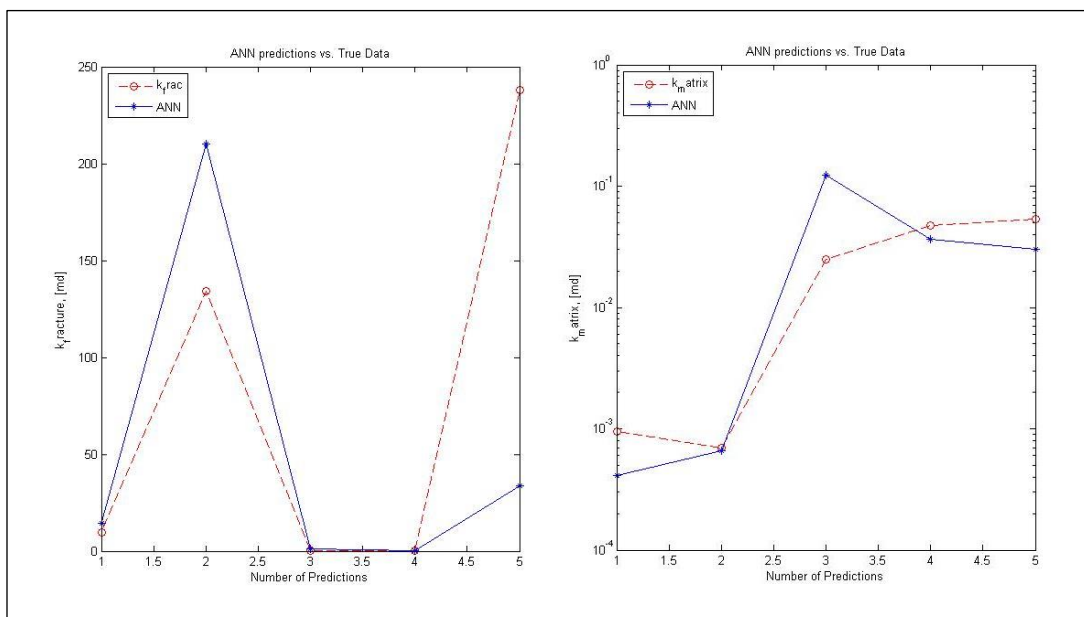


Figure 56: High Matrix Perm: Permeability Testing Results

Table 10: High Matrix Perm: Error Table

	Training Average Error (%)	Testing Average Error (%)
k_f	1.46	70.01
k_m	1.28	104.50

In order to verify that the high matrix permeability ANN was functioning correctly, the test data had to be verified. As one can see from the testing results, the ANN data did not match the target data exactly, but the trend of the data was followed. The inverse ANN is basically a history match tool. Because history matching does not provide one with a unique solution, the ANN also will not provide a unique solution. Thus, even though the ANN does not manage to match the target data exactly, its predicted values may generate a similar PT curve as the target data would have. If this is the case, then our ANN will be well trained. If the ANN's predicted properties do not generate a similar PT curve then the results are erroneous and the ANN is not fully trained.

Table 11 below shows a sample set of test data (target) and the ANN generated data. One can see that the values of the properties differ slightly. The target data and the ANN generated data were both placed into a numerical model and PT curves were generated for each. The results are displayed in Figure 57. One can see that the two curves are in good agreement with one another and the two curves never differ by more than 10 psia. This indicates that the ANN is well trained. Note the green curve that has been plotted on Figure 57. This curve was generated by plugging in the ANN generated data into the Forward Solution ANN. This was done to verify that the forward solution could predict new data given to it that it had never seen before. Because the

‘ANN’ curve and the ‘Forward ANN’ Curve have good agreement, it can be stated that the forward solution is also well trained.

Table 11: Target and ANN data generated by high matrix permeability ANN

	Target	ANN
ϕ_m (%)	9	11.5526
ϕ_f (%)	1.36	1.0903
FS (ft)	2.6	11.5352
Tres (°F)	165	250.8011
kf (md)	5.48	5.5425
km (md)	0.000401	0.000127
Pi (psia)	7854.07	7854.07
h (ft)	105	105
Q (MMscf/D)	28.00	28.00
Main Wellbore (ft)	1170	1170
Individual Lateral (ft)	590	590

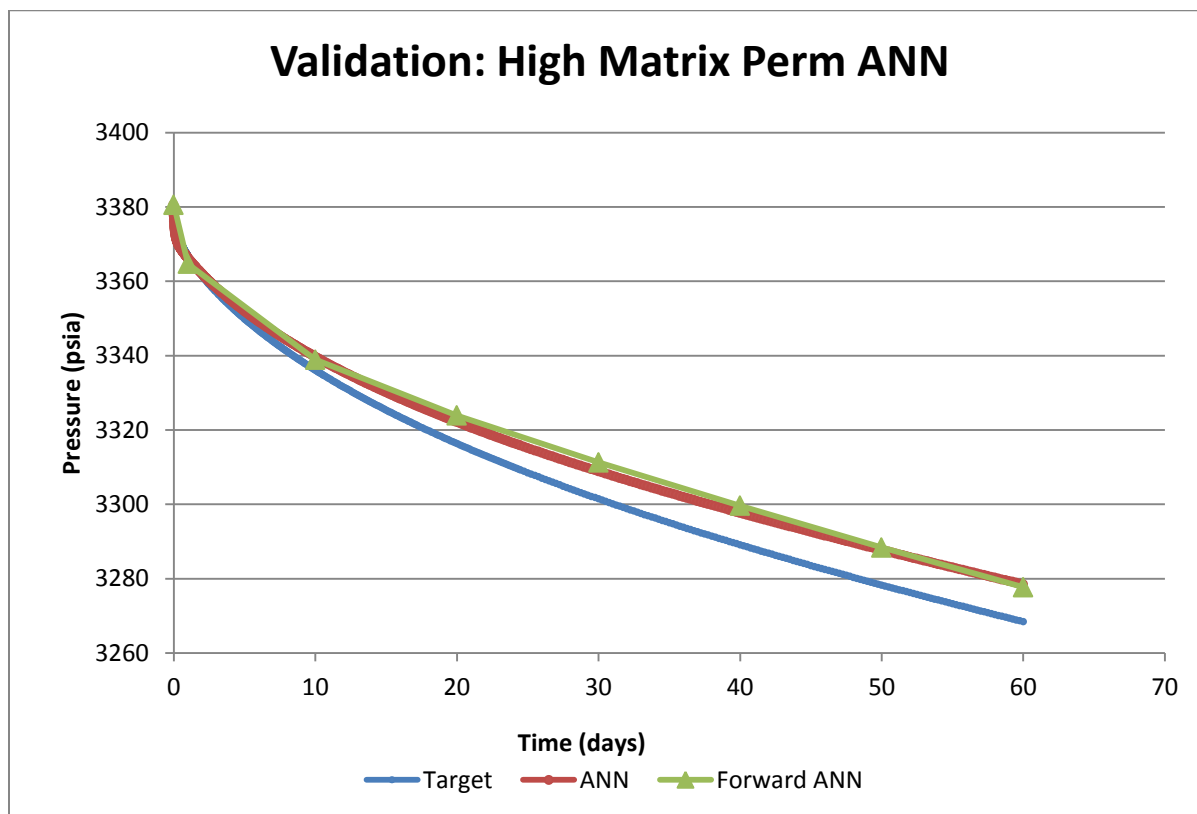


Figure 57: Validation Test for high matrix permeability ANN

5.2.2. Low Permeability Reservoir

The results for the low matrix permeability reservoir will be discussed next. Recall that high matrix permeabilities spanned the range 10^{-8} to 10^{-4} md.

5.2.2.1. Matrix and Fracture Porosity, Fracture Spacing, and Reservoir Temperature

As stated previously, the first goal of the inverse solution was to predict the reservoir properties from PT data and wellbore characteristics. Figure 58 displays the training data for the reservoir properties of ϕ_m , ϕ_f , FS , and T_{res} . It is clear that the trained data spans the same ranges as the target data.

Next, Figure 59 displays the testing results. Training data is often a good indication of the ANN's performance, but the testing data/error should always be used in conjunction with the training data in order to verify the validity of the ANN. From Figure 59 one can see that the ANN seems to have the ability to predict the reservoir property's values. To further quantify the ANN's predictability both the testing and training errors have been compiled in Table 12.

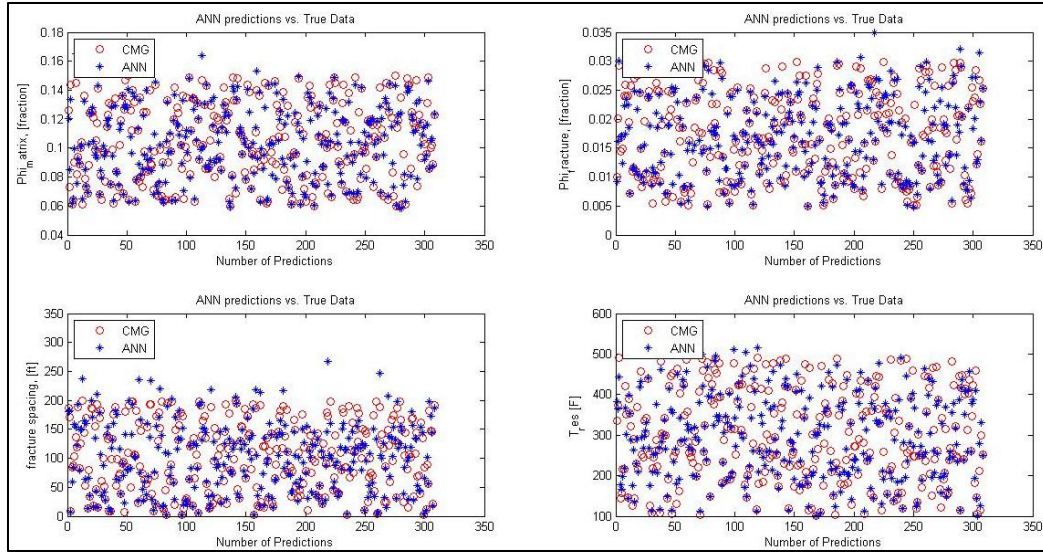


Figure 58: Low Matrix Perm: Reservoir Training Results

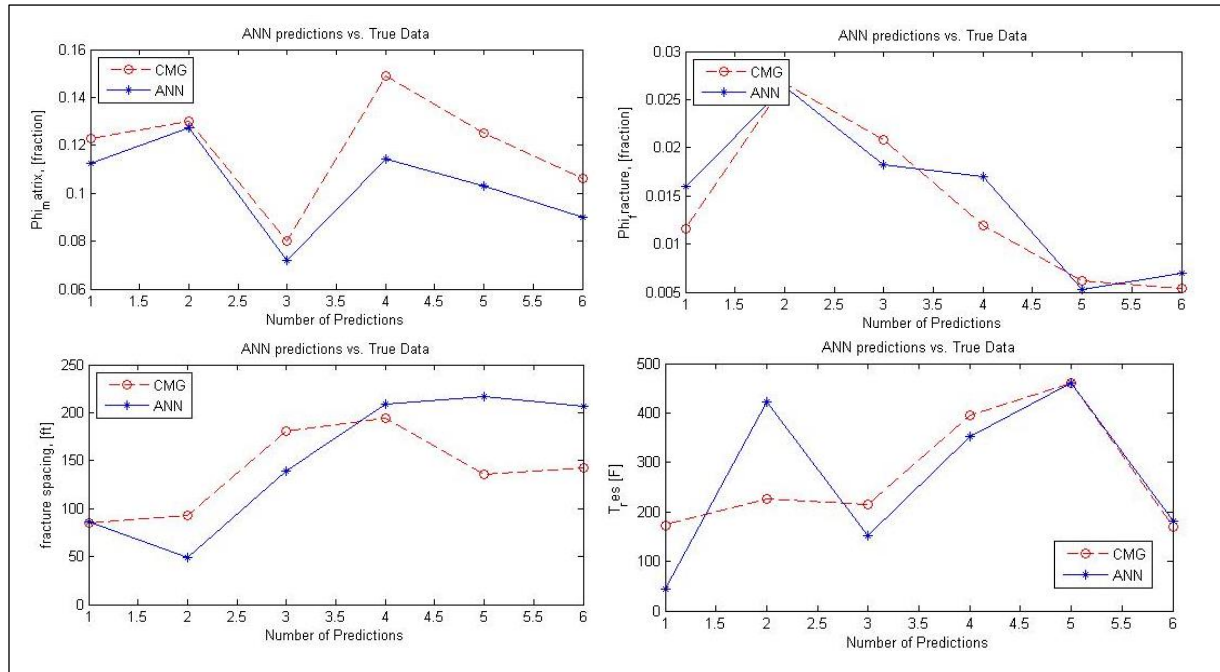


Figure 59: Low Matrix Perm: Reservoir Testing Results

Table 12: Low Matrix Perm: Reservoir Properties Error Table

	Training Average Error (%)	Test Average Error (%)
ϕ_f	10.56	22.97
ϕ_m	5.09	12.84
FS	20.39	30.47
T_res	8.83	27.37

Once again, the reservoir data will be tested in the same manner as the high permeability case. This test will be shown at the conclusion of the ‘Matrix and Fracture Permeability’ section below.

5.2.2.2. Matrix and Fracture Permeability

With the conclusion of the reservoir properties calculations, the matrix and fracture permeability could be calculated. Again, Figure 60 displays the training data of the ANN. One can see that the ANN’s training values span the range for both the matrix and fracture permeability. Figure 61 displays the testing results of the ANN. There are some deviations between the testing and training data, and this error will have to be analyzed to ensure it falls within the ‘acceptable’ error range. Table 13 summarizes the test and training error results. Again, this ANN will be verified by comparing the target data’s PT curve to the ANN generated data’s PT curve. If the two are in good agreement, then the ANN is said to be well trained.

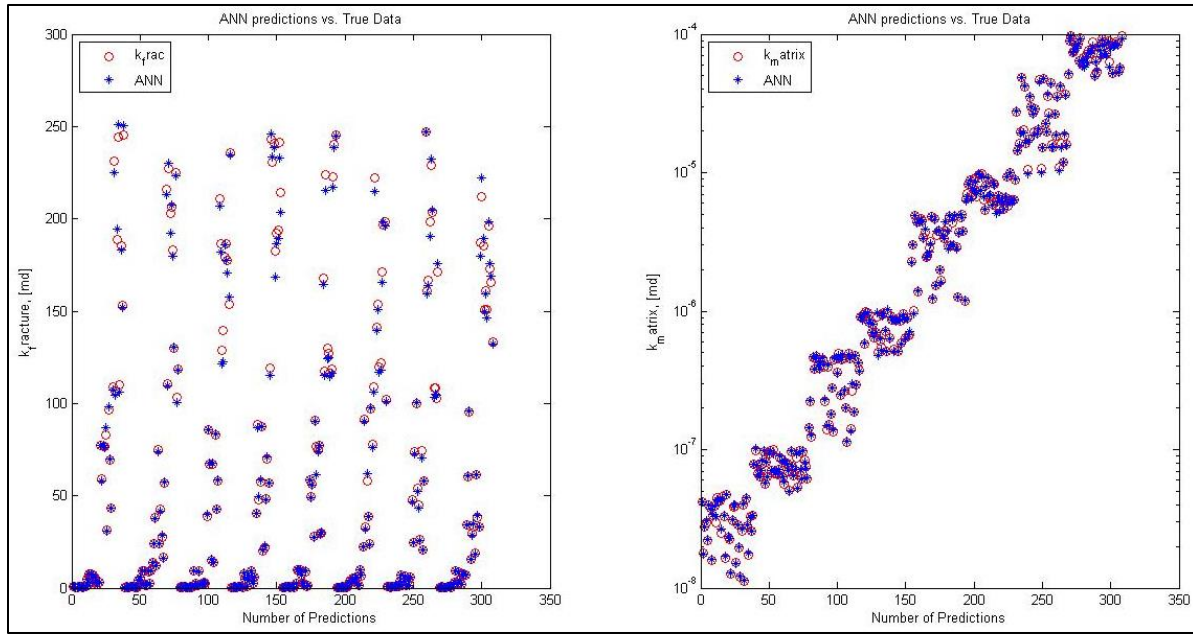


Figure 60: Low Matrix Perm: Permeability Training Results

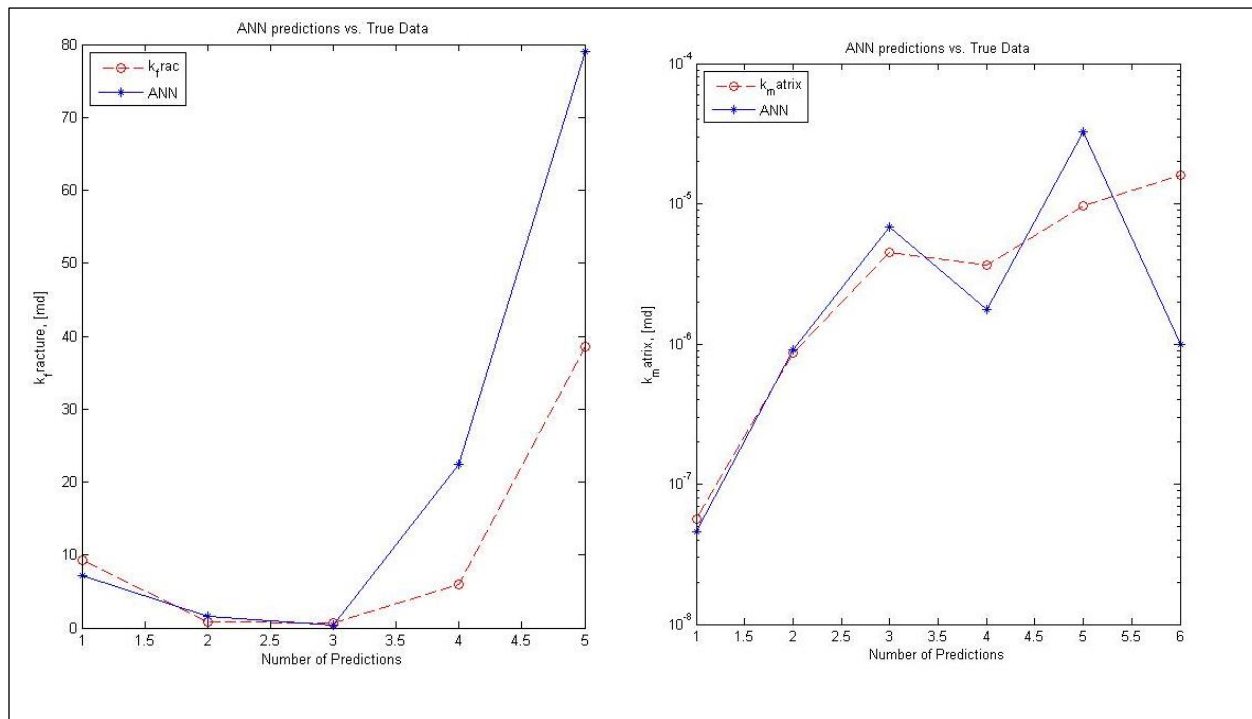


Figure 61: Low Matrix Perm: Permeability Test Results

Table 13: Low Matrix Perm: Error Table

	Training Average Error (%)	Test Average Error (%)
k_f	2.05	53.08
k_m	2.37	76.52

In order to verify that the low matrix permeability ANN was functioning correctly, the test data had to be verified. As one can see from the testing results, the ANN data did not match the target data exactly, but the trend of the data was followed. The inverse ANN is basically a history match tool. Because history matching does not provide one with a unique solution, the ANN also will not provide a unique solution. Thus, even though the ANN does not manage to match the target data exactly, its predicted values may generate a similar PT curve as the target data would have. If this is the case, then our ANN will be well trained. If the ANN's predicted properties do not generate a similar PT curve then the results are erroneous and the ANN is not fully trained.

Table 14 below shows a sample set of test data (target) and the ANN generated data. One can see that the values of the properties differ slightly. The target data and the ANN generated data were both placed into a numerical model and PT curves were generated for each. The results are displayed in Figure 62. One can see that the two curves are in good agreement with one another and the two curves never differ by more than 10 psia. Note the scale of the y-axis. This indicates that the ANN is well trained. Note the green curve that has been plotted on Figure 62. This curve was generated by plugging in the ANN generated data into the Forward Solution ANN. This was done to verify that the forward solution could predict new data given to it that it had never seen

before. Because the ‘ANN’ curve and the ‘Forward ANN’ Curve have good agreement, it can be stated that the forward solution is also well trained.

Table 14: Target and ANN data generated by low matrix permeability ANN

	T_Test	T_Test_ANN
ϕ_m (%)	11.6	10.91
ϕ_f (%)	0.85	0.62
FS (ft)	28.8	15.01
Tres (°F)	467.1	405.35
kf (md)	183.24	143.51
km (md)	6.10E-8	6.42E-10
Pi (psia)	3446.83	3446.83
h (ft)	230.6	230.6
Q (MMscf/D)	23.9	23.9
Main Wellbore (ft)	2120	2120
Individual Lateral (ft)	1060	1060

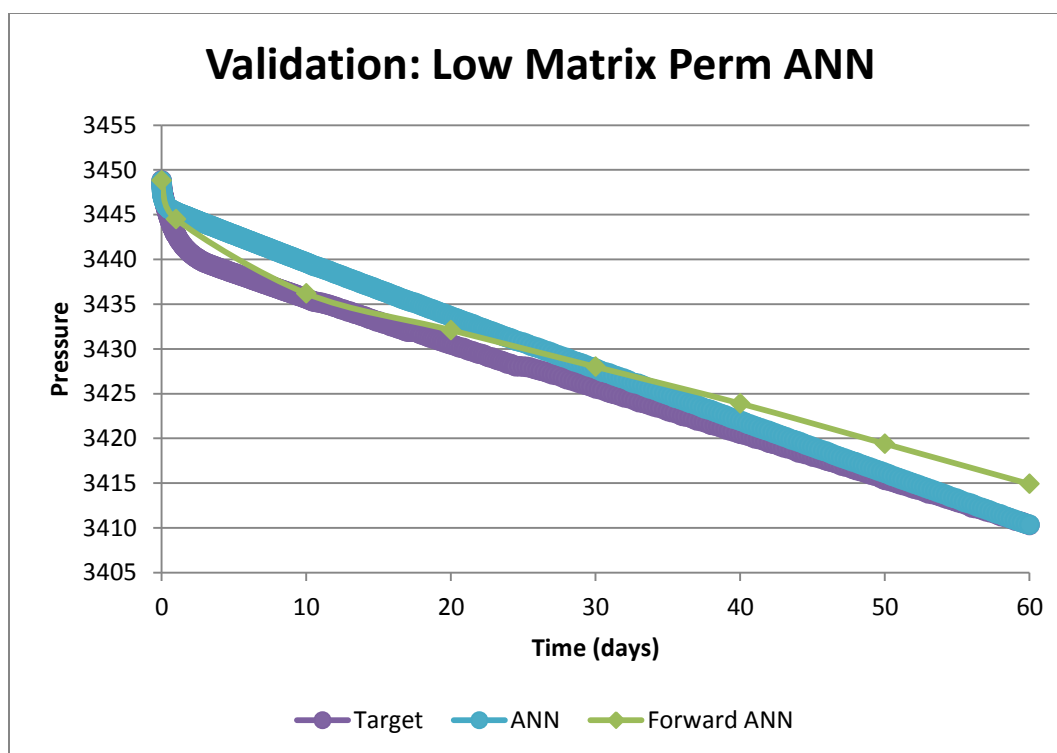


Figure 62: Validation Test for low matrix permeability ANN

Chapter 6: Conclusions

In summary, the goal of this research was to provide a tool that could analyze pressure transient data for a multi-lateral well completed within a dual-porosity reservoir. This work has provided six distinct ANNs that can be used in one of two ways. Either reservoir data and wellbore design can be input into any of the three forward ANNs to produce the corresponding pressure transient data, or the wellbore design along with pressure transient data can be input into any of the three inverse ANNs to produce the reservoir properties. This entire process has been included within a GUI for easy accessibility to future users.

This work has shown that it is possible to create an ANN as a pressure transient analysis package. The steps to creating such a tool have been shown in Chapters 3 and 4. Through testing it was discovered that the best training functions for this tool were either ‘trainsecg’ or ‘trainrp’. The best transfer functions for this tool were the non-linear transfer functions ‘tansig’ and ‘logsig’. This research has shown that complex problems require complex solutions. The networks created in this study all had three hidden layers each consisting of 50+ neurons each. However, only altering the ANN structure, as mentioned in Chapter 4, will not provide adequate results to the problem at hand. The implementation of scaling data as well as functional links must be used to obtain the best possible results from the network.

The following conclusions have been made:

1. This work has shown the ability of a Back Propagation ANN to be used as a pressure transient analysis tool.
2. ANNs can successfully predict reservoir properties of dual-porosity systems with an advanced wellbore completion technique.
3. Certain properties of a reservoir will have a higher impact on the reservoir's performance and pressure transient behavior. These properties' effects on reservoir behavior are summarized in Chapter 2. The properties that demonstrate the largest impact on the reservoir should be the first implemented functional links into one's network.
4. One should understand what level of accuracy is needed for the prediction of each property. The sensitivity study in Chapter 2 showed that changing a reservoir property does not always lead to a distinguishable change in the reservoir's performance.
5. The addition of input data into an ANN will not always yield more accurate results.
6. The conjugate training function (trainscg) and resilient propagation training functions (trainrp) were each found to give superior results as compared to other training functions.
7. Non-linear transfer functions (tansig, logsig) were found to provide the most accurate ANNs
8. The larger the amount of output variables, the more difficult it is to obtain accurate results for every variable.
9. ANNs do not have to be entirely inclusive, and multiple ANNs can be used to produce accurate results as was the case with the inverse solution.
10. For best results, this ANN should only be used with the reservoir properties listed in Table 3

11. Understanding Artificial Neural Networks will lead one to develop better ANN structures. However, most ANN rules are only rules of thumb, and should not be followed blindly.

- a. Some rules of thumb developed from this research are: The pressure values from the drawdown test are the most influential parameters on the convergence of the ANN. Taking the difference between these pressure values increased the fit of the ANN significantly.

Suggestions for Future Work:

1. Additional wellbore structures should be analyzed using this method
 - a. Varying the Wellbore:Lateral Angle (30° , **45°** , 60°)³
 - b. Vary the Geometry of the Wellbore:Lateral (**L:L**, L:2L, L:3L, etc.)³
2. Incorporate Matrix and Fracture Compressibility Terms
3. Incorporate Viscosity of Gas
4. Compare production results from multilateral wells in dual-porosity systems to results from hydraulically fracture wells in the same dual-porosity systems

³ In bold are the Wellbore:Lateral values used in this study.

Bibliography

- Alajmi, M. N. (2003). *The Development of an Artificial Neural Network As a Pressure Transient Analysis Tool for Applications in Double-Porosity Reservoirs*. State College: Pennsylvania State University.
- Almousa, T. S. (2013). *DEVELOPMENT AND UTILIZATION OF INTEGRATED ARTIFICIAL EXPERT SYSTEMS FOR DESIGNING MULTI-LATERAL WELL CONFIGURATIONS, ESTIMATING RESERVOIR PROPERTIES AND FORECASTING RESERVOIR PERFORMANCE*. State College: Pennsylvania State University.
- Artun, E. F. (2008). *Optimized Design of Cyclic Pressure Pulsing in Naturally Fractured Reservoirs Using Neural Network Based Proxy Models*. State College: Pennsylvania State University.
- Beale, M. H., Hagan, M. T., & Demuth, H. B. (2013, August). *Neural Network Toolbox - User's Guide*. Retrieved from MathWorks.
- Bodipati, K. (2011). *Numerical Model Representation Of Multi-stage Hydraulically Fractured Horizontal Wells Located In Shale Gas Reservoirs Using Neural Networks*. State College: Pennsylvania State University.
- Bustin, A. M., & Bustin, R. (2012, May 5). Importance of rock properties on the producibility of gas shales. *International Journal of Coal Geology*, 103, 132-147.
doi:10.1016/j.coal.2012.04.012

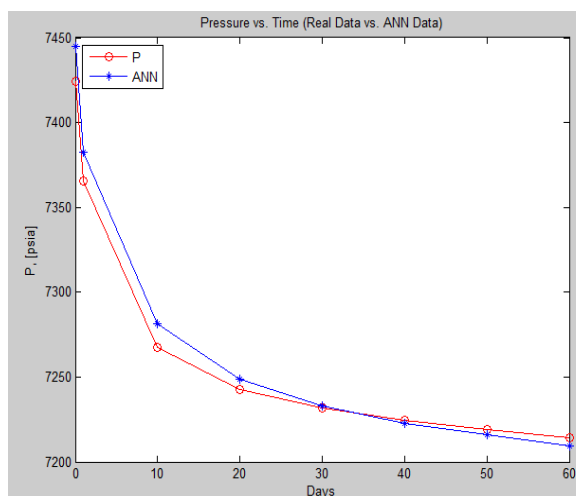
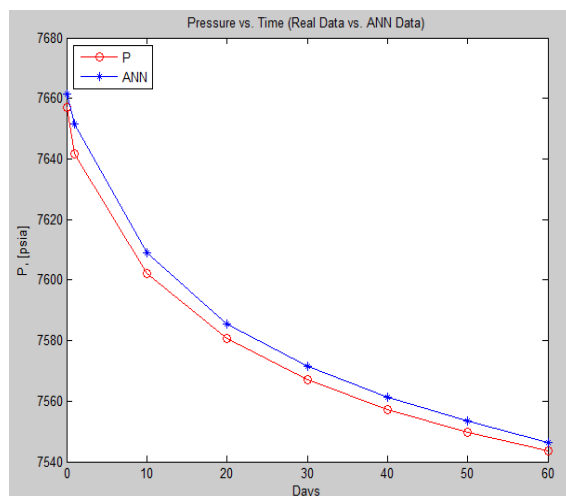
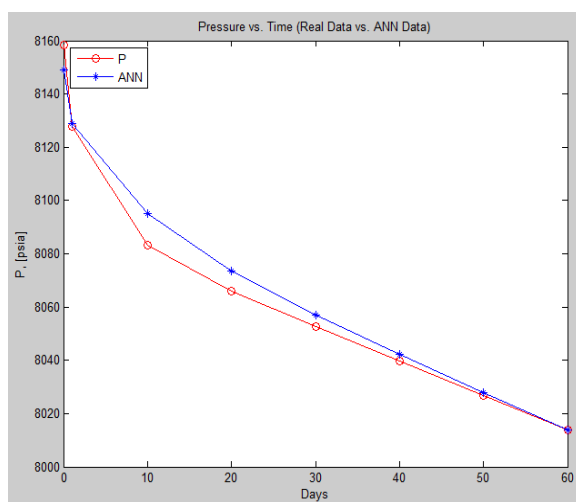
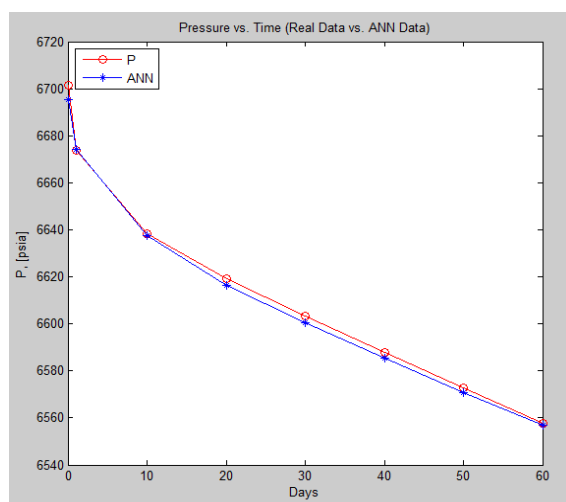
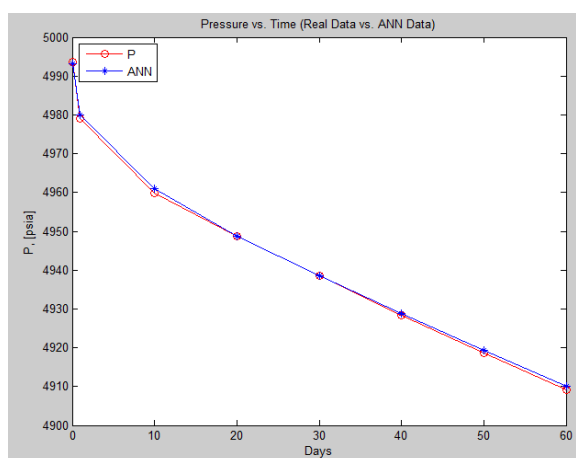
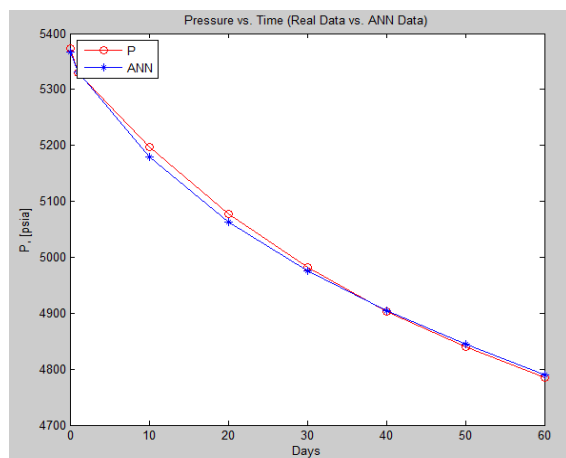
- Callard, J. G., & Schenewerk, P. A. (1995). Reservoir Performance History Matching Using Rate/Cumulative Type-Curves. *SPE Annual Technical Conference & Exhibition*, (pp. 947-956). Dallas.
- Computer Modelling Group Ltd. (2012). *User's Guide IMEX Advanced Black Oil/Gas Reservoir Simulator*. Calgary, Alberta Canada: Reprise License Manager.
- Department of Environmental Protection. (2011). *Oil and Gas Well Drilling and Production in Pennsylvania*.
- Dong, X. (2003). *Characterization of Coalbed Methane Reservoirs from Pressure Transient Data: An Artificial Neural Network Approach*. State College: Pennsylvania State University.
- Ezekwe, N. (2011). *Petroleum reservoir engineering practice*. Upper Saddle River, NJ: Prentice Hall.
- Gotka, B., & Ertekin, T. (1999). Implementation of a Local Grid Refinement Technique in Modeling Slanted, Undulating Horizontal and Multi-Lateral Wells. *SPE Annual Technical Conference and Exhibition*, (pp. 1-10). Houston.
- Hagan, M. T., Demuth, H. B., & Beale, M. (1996). *Neural Network Design*. Boston: PWS Publishing Company.
- Hyun Ahn, C. (2012). *Modeling Of Hydraulic Fracture Network Propagation In Shale Gas Reservoirs*. State College: Pennsylvania State University.
- Jenkins, C. D., DeGolyer and MacNaughton, & Boyer II, C. M. (2008). Coalbed- and Shale-Gas Reservoirs. *JPT - SPE*, 92-99.

- Josh, M., Esteban, L., Delle Piane, C., Sarout, J., Dewhurst, D., & Clennel, M. (2012, February 7). Laboratory characterisation of shale properties. *Journal of Petroleum Science and Engineering*, 88(89), 107-124. doi:10.1016/j.petrol.2012.01.023
- Khattirat, K. (2004). *Development of an Artificial Neural Network as a Pressure Transient Analysis Tool for Application in a Hydraulically Fractured Reservoir*. State College: Pennsylvania State University.
- Larson, & Falvo. (2009). *Elementary Linear Algebra 6th edition*. New York: Houghton Mifflin Harcourt Publishing Company.
- Lee, D. S., Herman, J. D., Elsworth, D., Kim, H. T., & Lee, H. S. (2011, February 16). A Critical Evaluation of Unconventional Gas Recovery from the. *KSCE Journal of Civil Engineering*, 15(4), 679-687. doi:10.1007/s12205-011-0008-4
- Liu, B., Kelkar, M. G., Gang, T., & Dixon, T. N. (2007). Efficient History Matching Through Production Data Selection. *SPE Production and Operations Symposium*, (pp. 1-12). Oklahoma City.
- Meiller, M. F. (1993). A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning. *Neural Networks*, 525-533.
- Nagel, N. B., Sanchez-Nagel, M. A., Zhang, F., Garcia, X., & Lee, B. (2013, March 7). Coupled Numerical Evaluations of the Geomechanical Interactions Between a Hydraulic Fracture Stimulation and a Natural Fracture System in Shale Formations. *Rock Mech Rock Eng*, 46, 581-609. doi:10.1007/s00603-013-0391-x

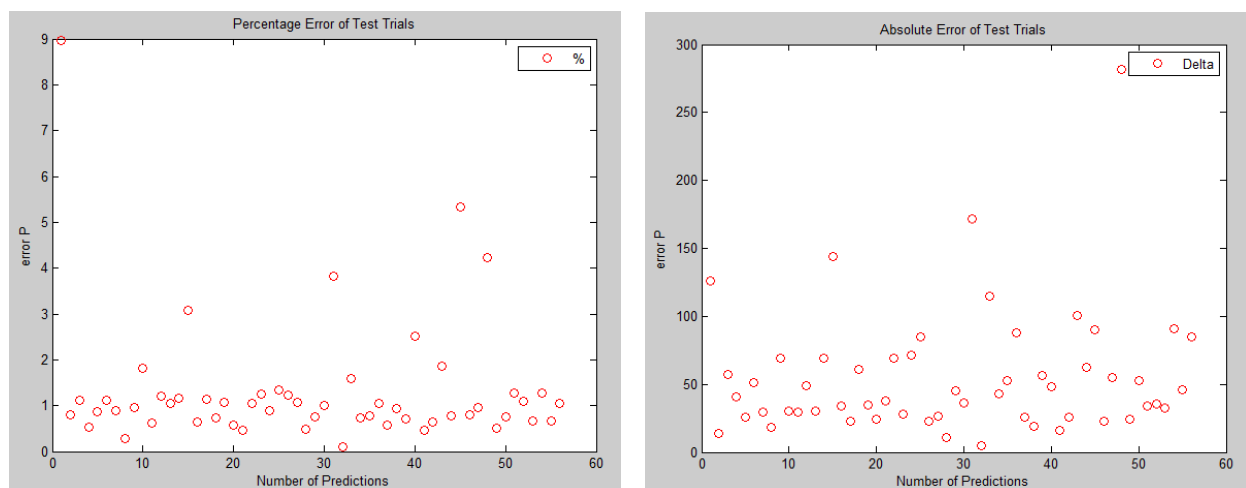
- Nelson, M. M., & Illingworth, W. T. (1994). *A Practical Guide to Neural Nets*. New York: Addison-Wesley Publishing Company.
- Riedmiller, M., & Braun, H. (1993). A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm. *Neural Networks, 1993., IEEE International Conference* (pp. 586-591). San Francisco: IEEE.
- Saxena, A. (2012). *TYPE CURVES FOR PRODUCTION TRANSIENT ANALYSIS OF MULTILATERAL WELLS IN NATURALLY FRACTURED SHALE GAS RESERVOIRS*. State College: Pennsylvania State University.
- Siripatrachai, N. (2011). *Alternate Representations for Numerical Modeling of Multi-Stage Hydraulically Fractured Horizontal Wells in Shale Gas Reservoirs*. State College: Pennsylvania State University.
- Suglyarma, H., Tochikawa, T., Peden, J., & Nicoll, G. (1997). The Optimal Application of Multi-Lateral/Multi-Branch Completions. *SPE Asia Pacific Oil and Gas Conference*, (pp. 135-148). Kuala Lumpur, Malaysia.
- Tavassoli, Z., Carter, J. N., & King, P. R. (2004). Errors in History Matching. *SPE Journal*, 352-361.
- Toktabolat, Z. (2012). *Characterization of Sealing and Partially Communicating Faults in Dual-Porosity Gas Reservoirs Using Artificial Neural Network*. State College: Pennsylvania State University.
- Walton, I., & McLennan, J. (2013). The Role of Natural Fractures in Shale Gas Production. *InTech*, 327-356. doi:<http://dx.doi.org/10.5772/56404>

- Warren, J. E., & Root, P. J. (1962). The Behavior of Naturally Fractured Reservoirs. *Society of Petroleum Fall Meeting*, (pp. 245-255). Los Angeles.
- Watson, A. T., Lane, H. S., & Gatens III, J. M. (1990). History Matching with Cumulative Production Data. *JPT*, 96-100.
- Zhou, Q. (2013). *DEVELOPMENT AND APPLICATION OF AN ARTIFICIAL EXPERT SYSTEM FOR THE PRESSURE TRANSIENT ANALYSIS OF LATERAL WELL CONFIGURATION*. State College: Pennsylvania State University.
- Zou, J., Han, Y., & So, S.-S. (2008). Overview of Artificial Neural Networks. *Methods in molecular biology*, 458(1064-3745), 15.

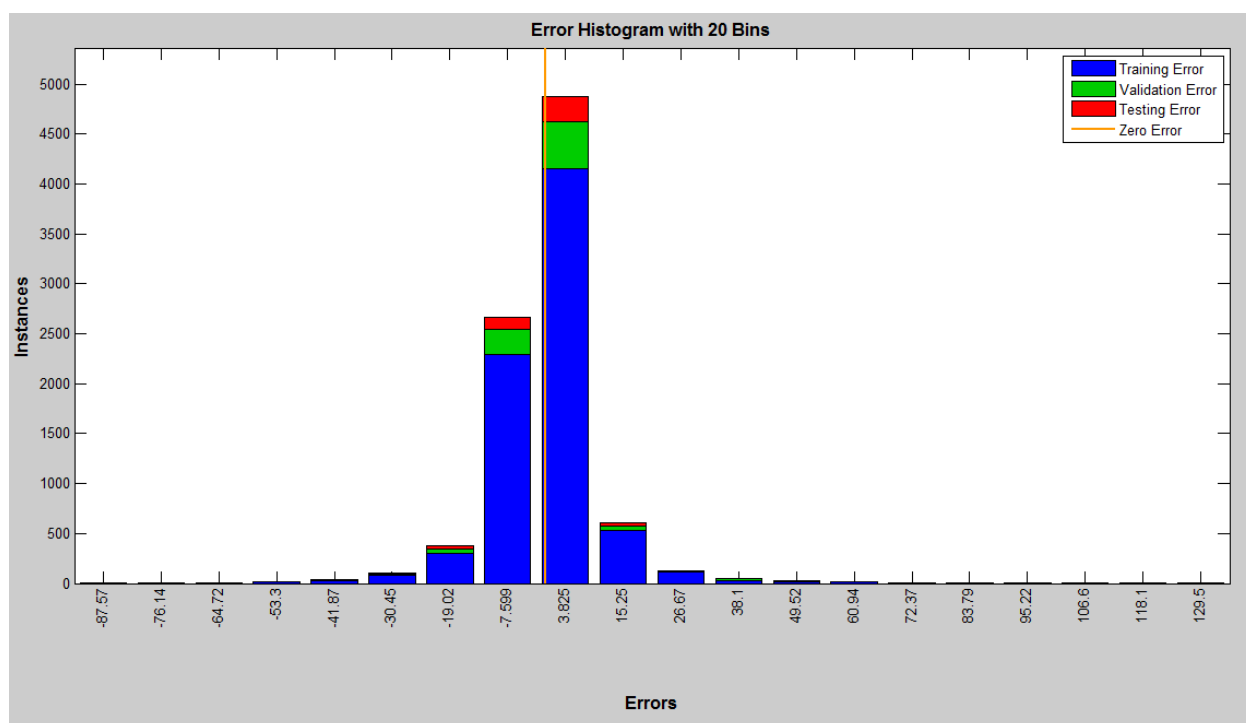
Appendix B: 2L Forward/Inverse Solution Results



3L – Forward Solution – Typical PT plots



2L – Forward Solution – Percentage and Delta Error

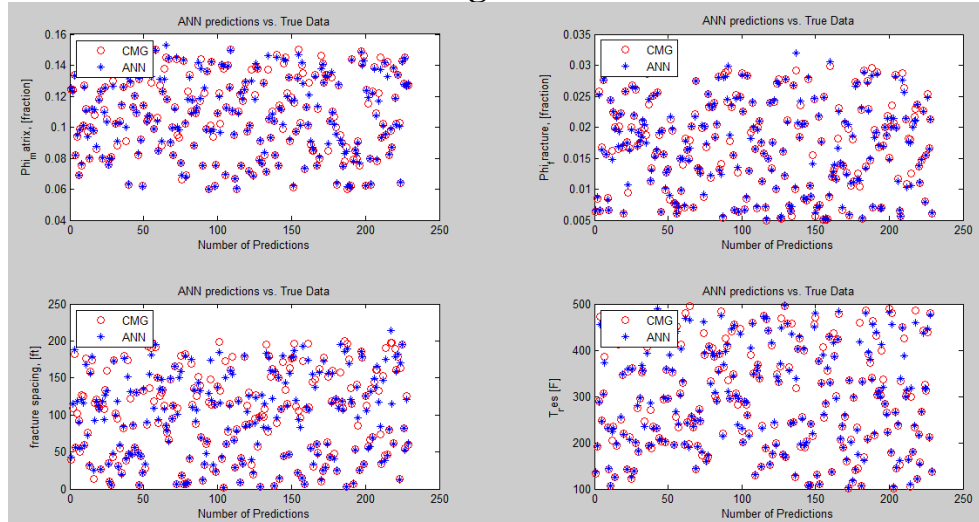


2L – Forward Solution – Error Histogram

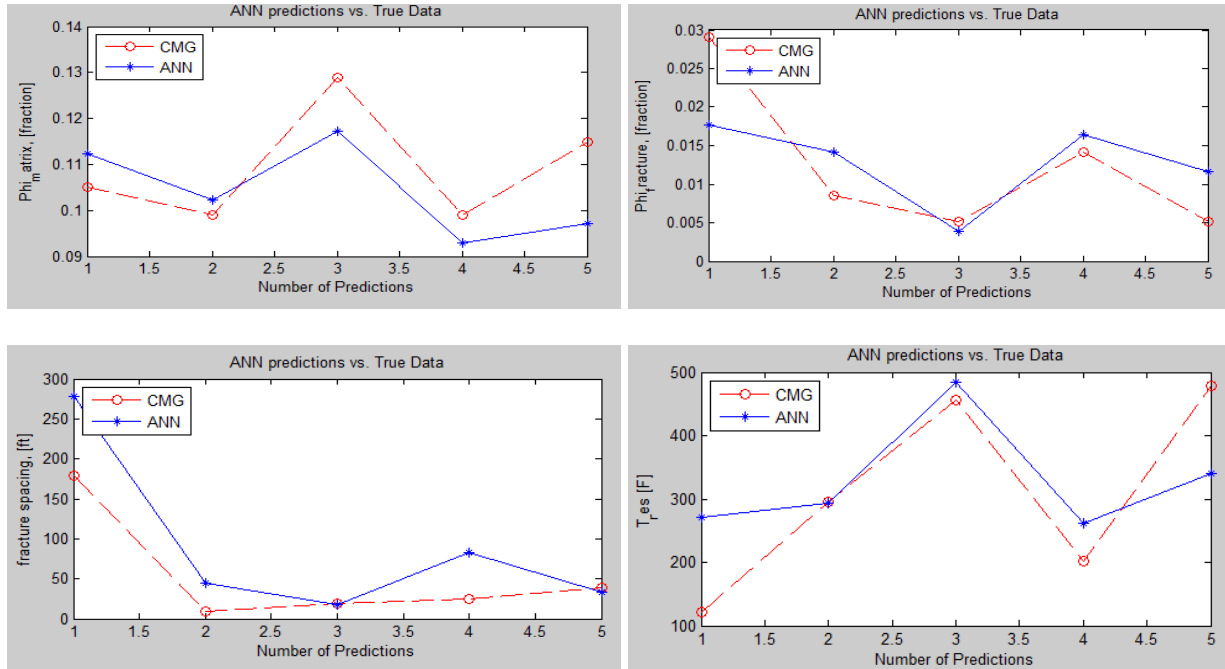
	Error	Average (Individual Run)	Average (Individual Data Point)
Testing Errors	% Error	1.31%	.16%
	Delta Error	54.92	6.86
Training Errors	% Error	1.31%	.16%
	Delta Error	51.55	6.44

2L – Forward Solution – Error Table

2L – Inverse – High k: Reservoir ANN



2L – Inverse – High k: Reservoir: Training

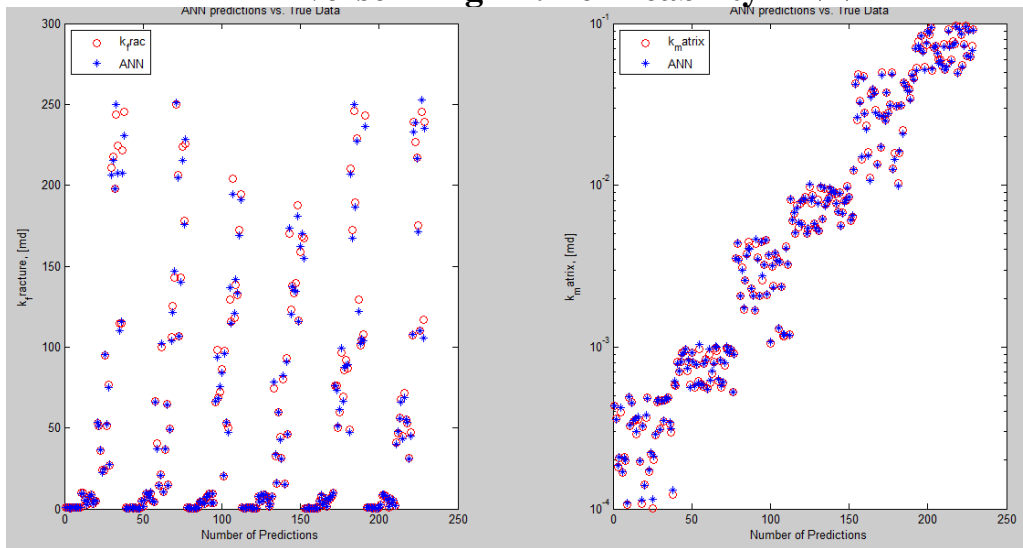


2L – Inverse – High k: Reservoir: Testing

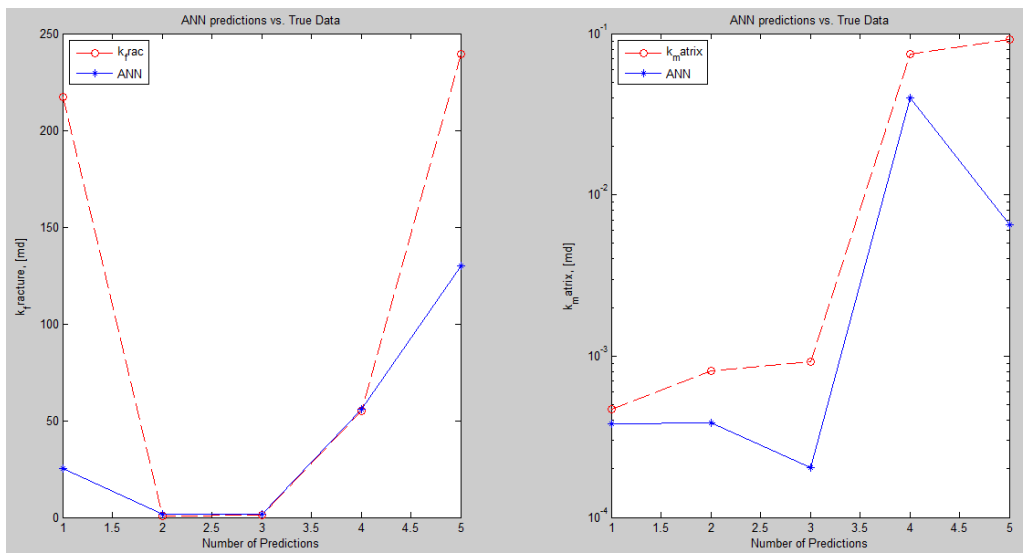
2L – Inverse – High k: Reservoir: Error Table

	Training Average Error (%)	Test Average Error (%)
ϕ_f	2.27	13.8
ϕ_m	1.38	27.6
FS	6.23	40.2
T_{res}	1.89	22.9

2L – Inverse – High k: Permeability ANN



2L – Inverse – High k: Permeability: Training

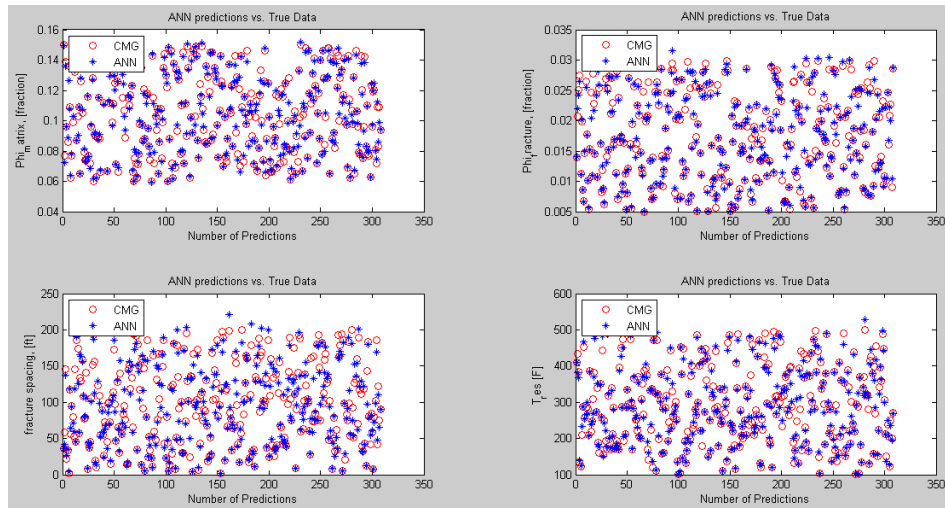


2L – Inverse – High k: Permeability: Testing

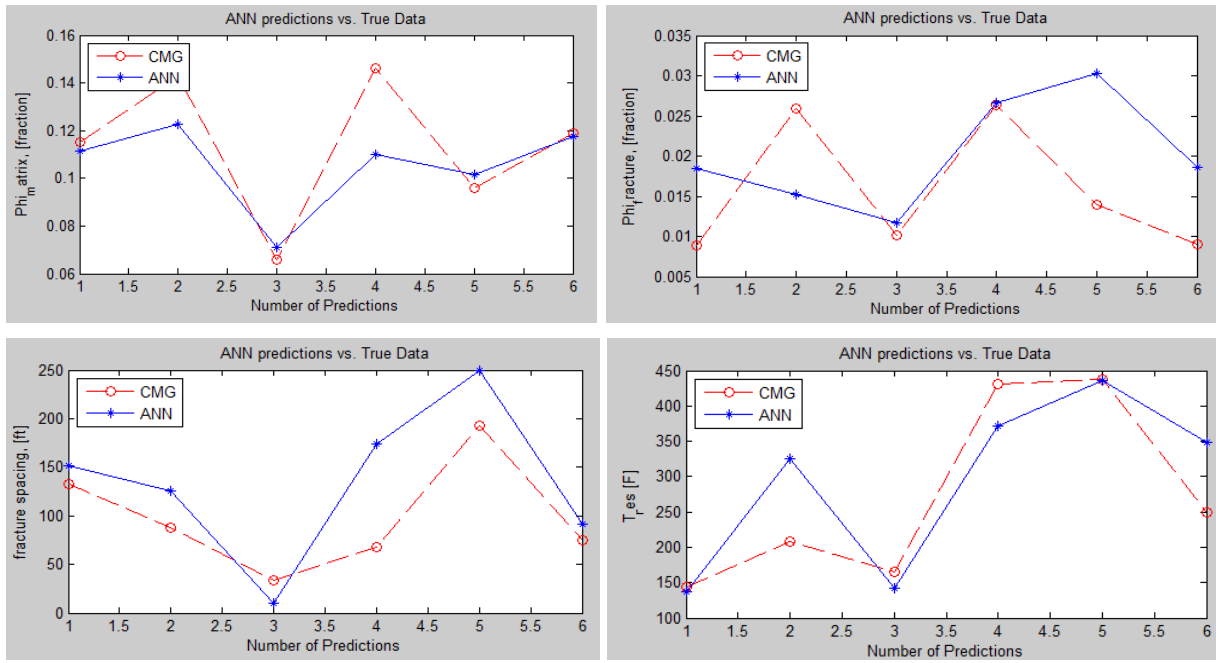
2L – Inverse – High k: Permeability: Error Table

	Training Average Error (%)	Testing Average Error (%)
k_f	2.49	51.72
k_m	2.23	57.60

2L – Inverse – Low k: Reservoir ANN



2L – Inverse – Low k: Reservoir: Training

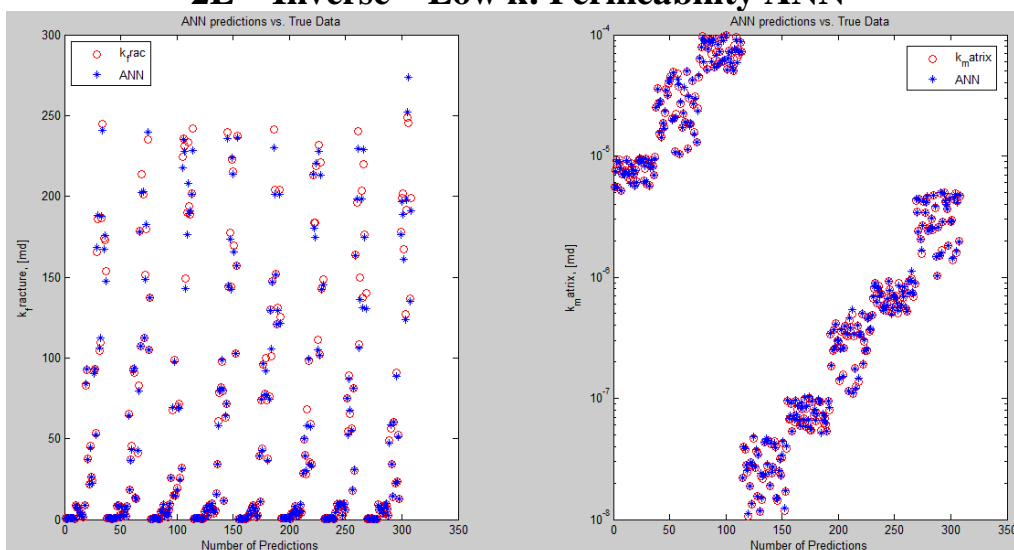


2L – Inverse – Low k: Reservoir: Testing

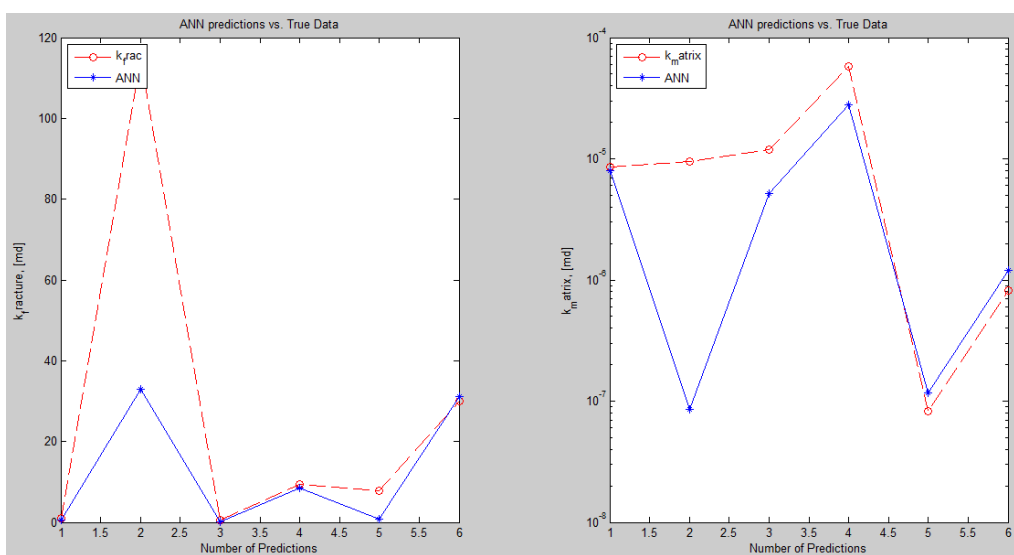
2L – Inverse – Low k: Reservoir: Error Table

	Training Average Error (%)	Test Average Error (%)
ϕ_f	3.15	22.1
ϕ_m	2.16	9.5
FS	10.00	43.9
T_{res}	3.39	16.6

2L – Inverse – Low k: Permeability ANN



2L – Inverse – Low k: Permeability: Training

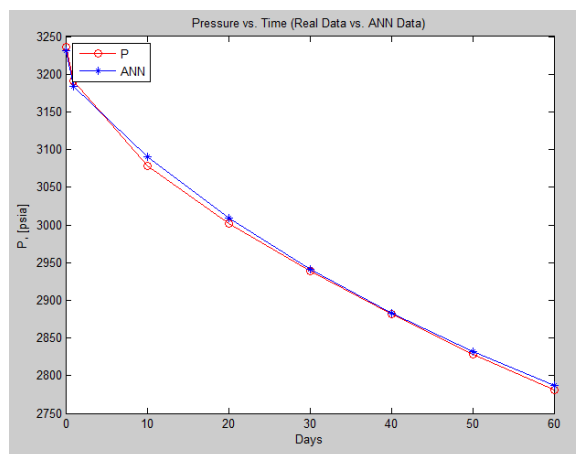
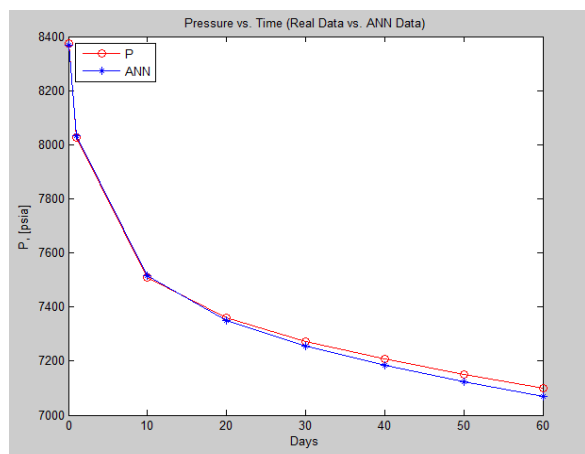
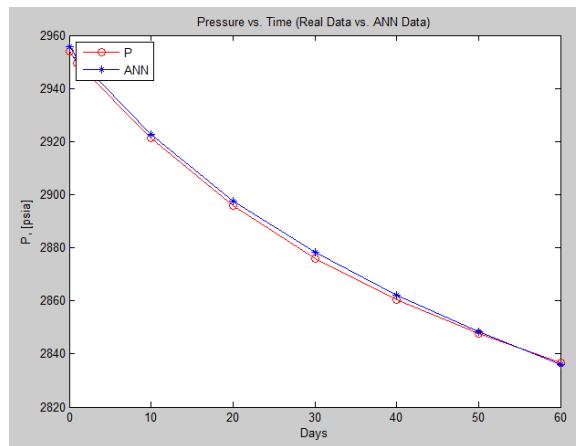
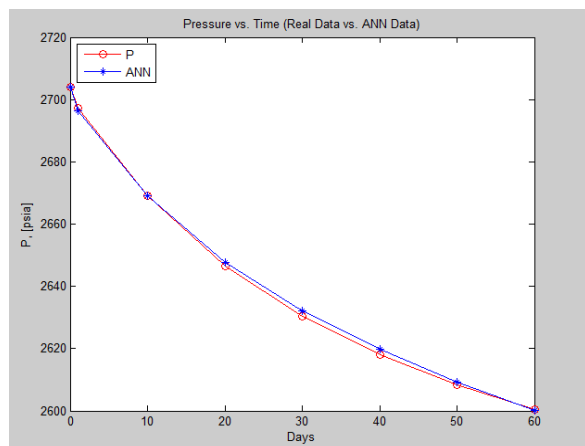
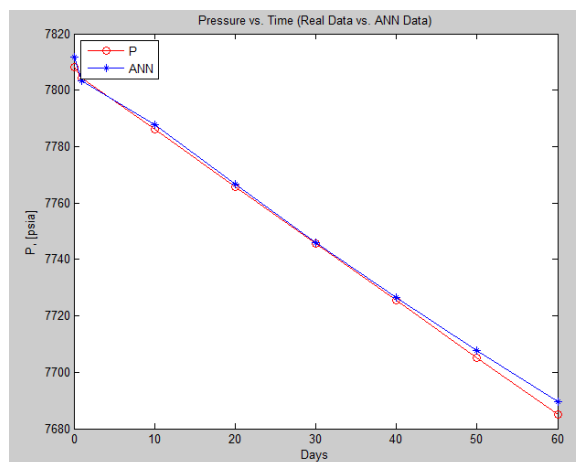
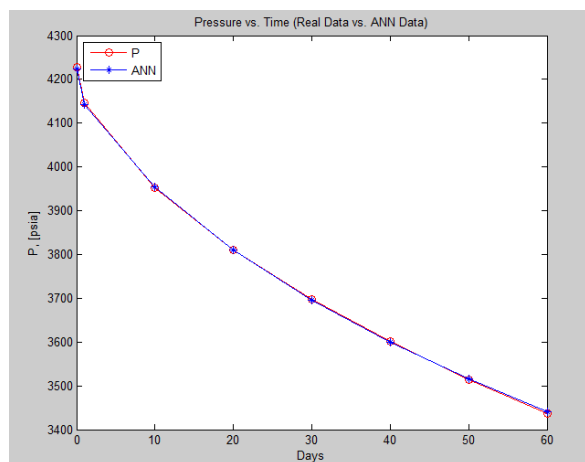


2L – Inverse – Low k: Permeability: Testing

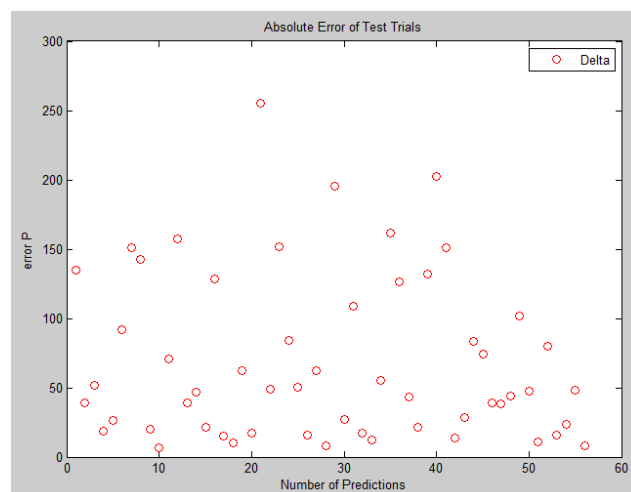
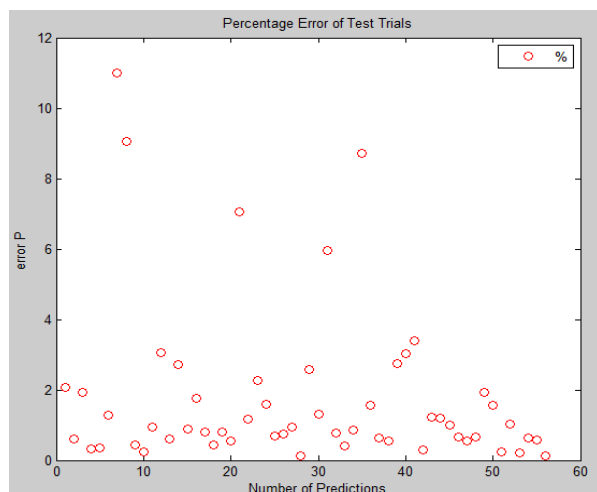
2L – Inverse – Low k: Permeability: Error Table

	Training Average Error (%)	Testing Average Error (%)
k_f	2.45	49.3
k_m	2.34	49.9

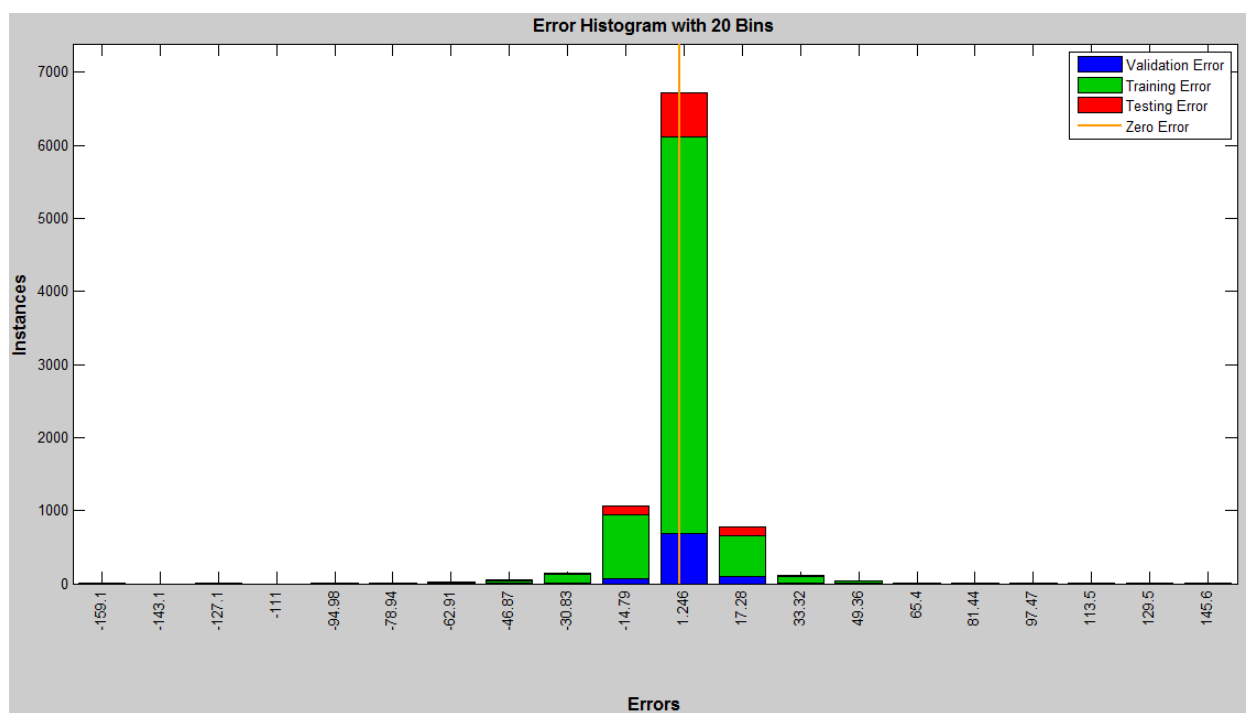
Appendix C: 3L Forward/Inverse Solution Results



3L – Forward Solution – Typical PT plots



3L – Forward Solution – Percentage and Delta Error

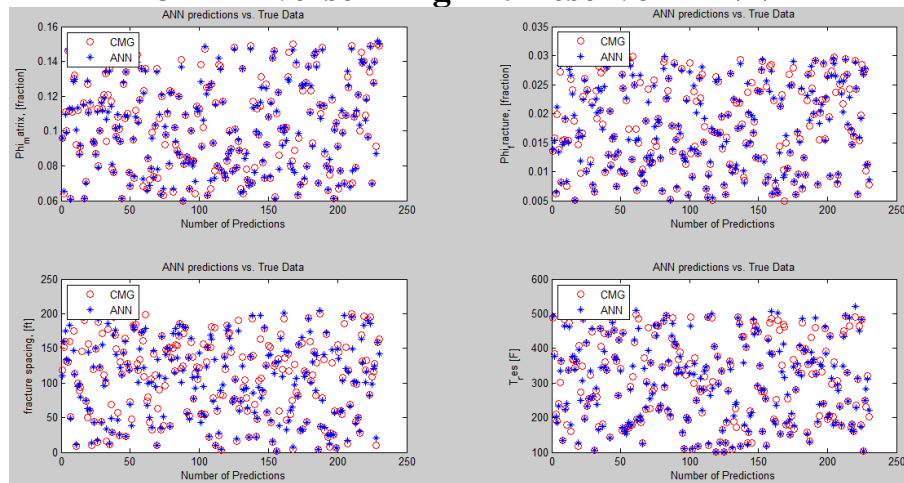


3L – Forward Solution – Error Histogram

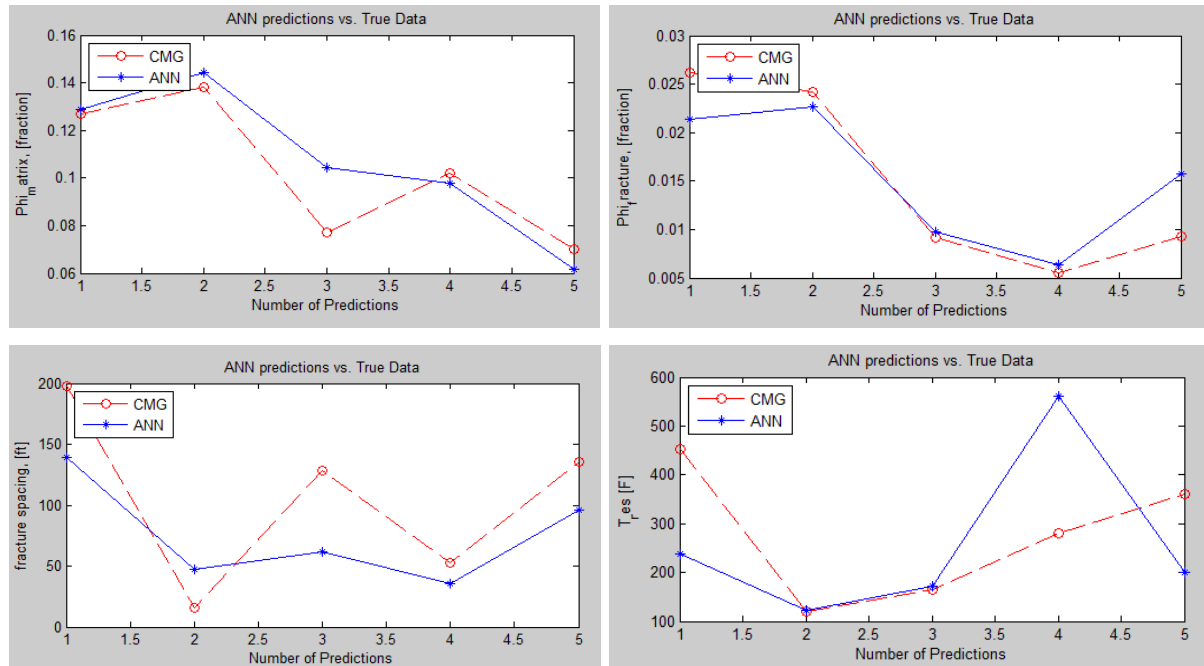
	Error	Average (Individual Run)	Average (Individual Data Point)
Testing Errors	% Error	1.61%	0.20%
	Delta Error	66.29	8.29
Training Errors	% Error	1.21	0.15
	Delta Error	51.57	6.45

3L – Forward Solution – Error Table

3L – Inverse – High k: Reservoir ANN



3L – Inverse – High k: Reservoir: Training

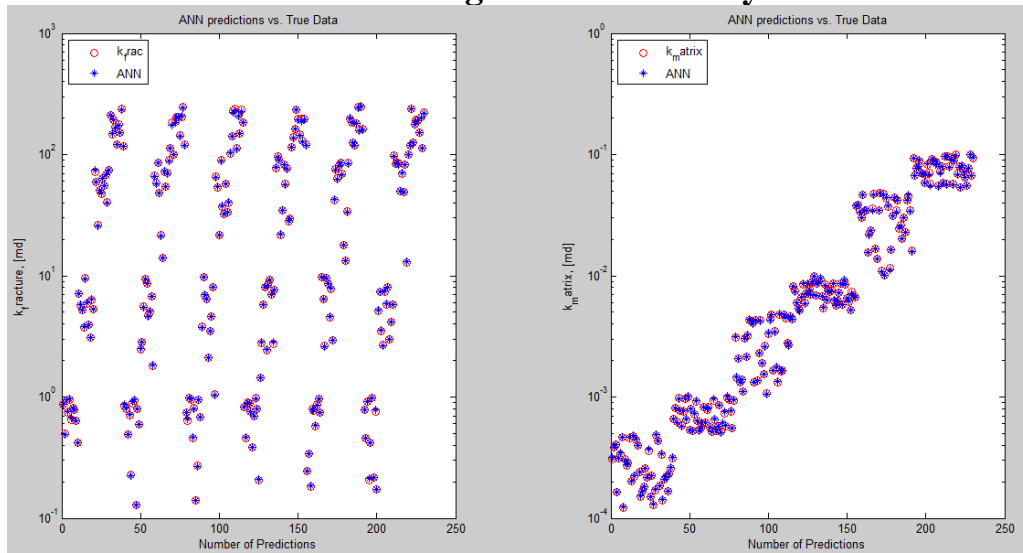


3L – Inverse – High k: Reservoir: Testing

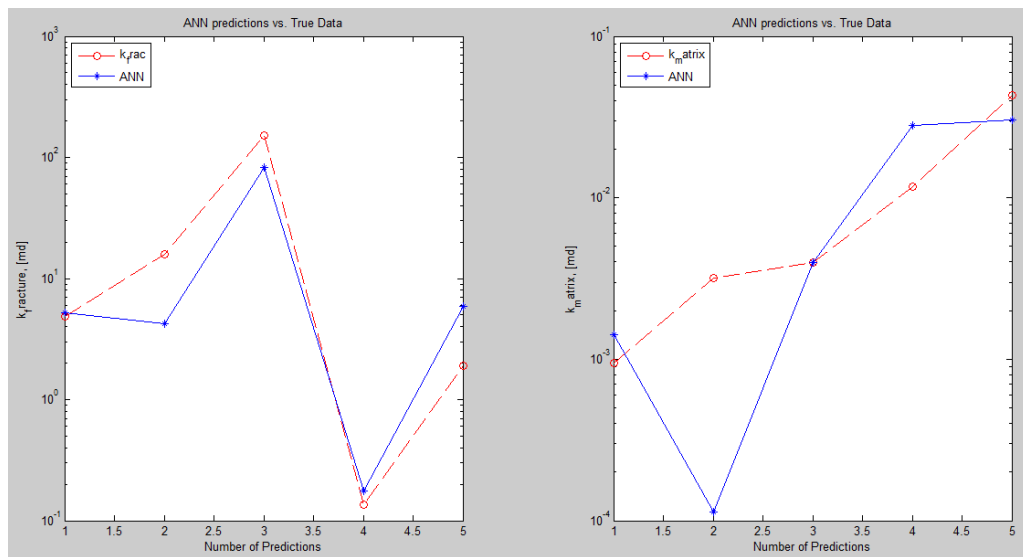
3L – Inverse – High k: Reservoir: Error Table

	Training Average Error (%)	Test Average Error (%)
ϕ_f	3.86	19.59
ϕ_m	1.78	8.96
FS	10.16	42.20
T_{res}	4.09	29.80

3L – Inverse – High k: Permeability ANN



3L – Inverse – High k: Permeability: Training

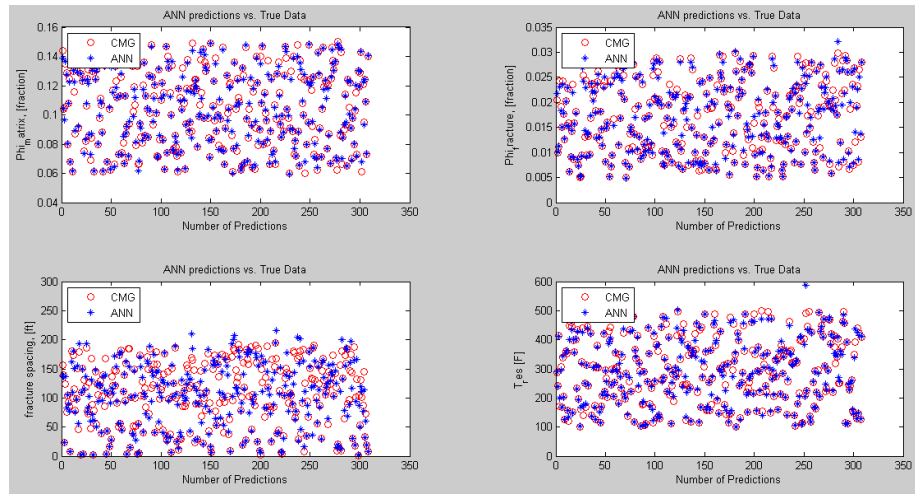


3L – Inverse – High k: Permeability: Testing

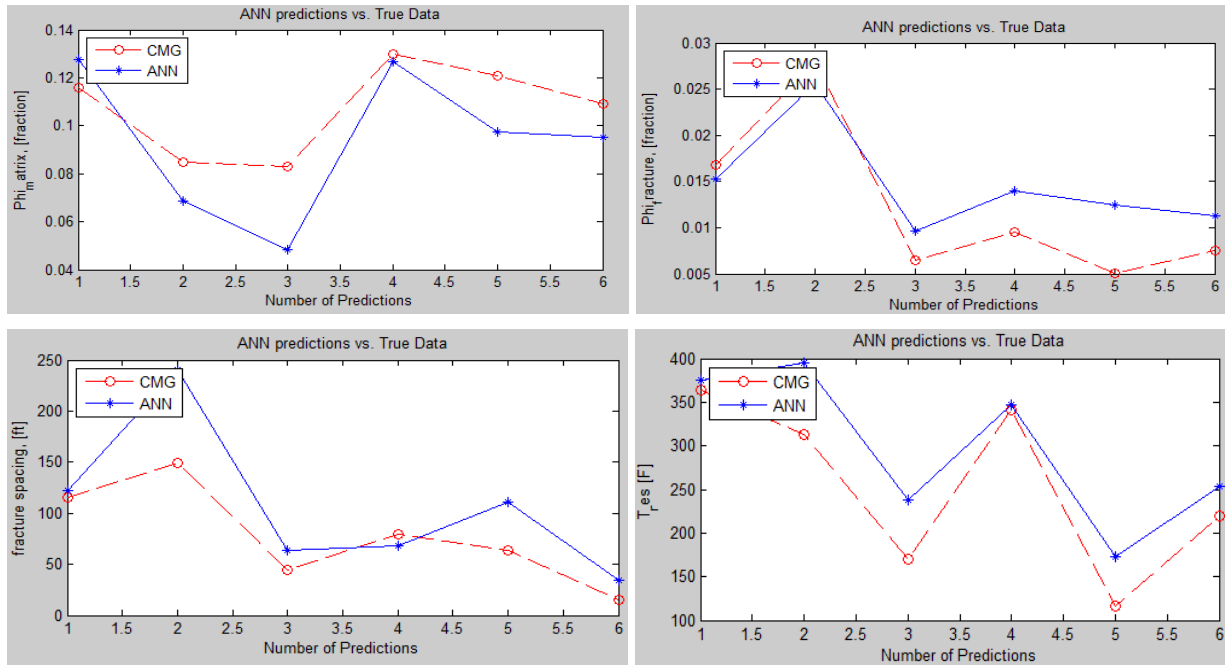
3L – Inverse – High k: Permeability: Error Table

	Training Average Error (%)	Testing Average Error (%)
k_f	0.97	51.85
k_m	0.95	63.38

3L – Inverse – Low k: Reservoir ANN



3L – Inverse – Low k: Reservoir: Training

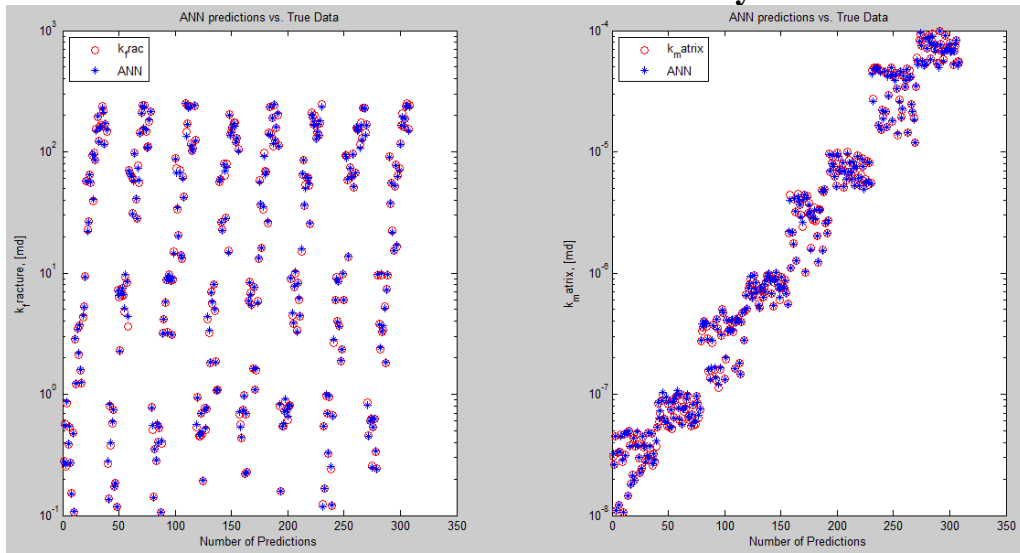


3L – Inverse – Low k: Reservoir: Testing

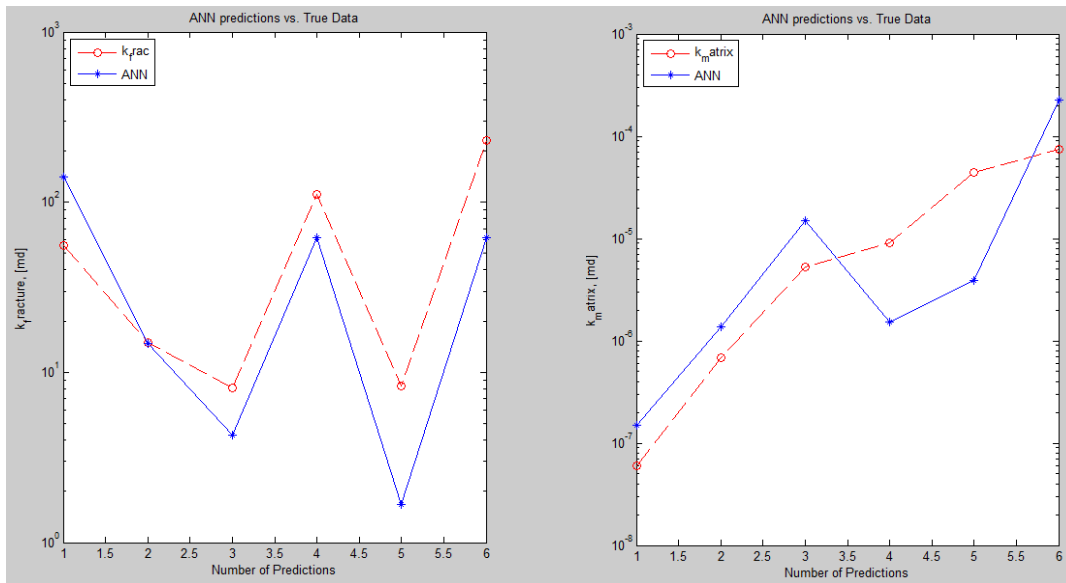
3L – Inverse – Low k: Reservoir: Error Table

	Training Average Error (%)	Test Average Error (%)
ϕ_f	3.16	29.74
ϕ_m	1.62	17.64
FS	11.26	38.47
T_{res}	2.80	15.72

3L – Inverse – Low k: Permeability ANN



3L – Inverse – Low k: Permeability: Training



3L – Inverse – Low k: Permeability: Testing

3L – Inverse – Low k: Permeability: Error Table

	Training Average Error (%)	Testing Average Error (%)
k_f	2.41	66.43
k_m	2.39	85.83

Appendix D: Example MATLAB Code Generating Data for CMG Analysis

```

function crunmaker()
%% Performing a drawdown test with constant Production varying from 10 Mcf
%% to 30 MMcf for 60 days
clc
clear all
rng shuffle

%number of runs
nc=560;
%Arbitrary length value
L=500;
%main well bore length (varies from 1000' to 4000')
range(1,1) = 2*L;   range(1,2) = 8*L;
%number of laterals (2, 4, or 6)
range(2,1) = 2;     range(2,2) = 2;

%% Mainbore length
INPUT(1,:) = range(1,1) + (range(1,2)-range(1,1)).*rand(nc,1);

%% Number of Laterals
INPUT(2,:) = range(2,1) + (range(2,2)-range(2,1)).*rand(nc,1);

%% Length of Laterals (2, 4, or 6) decided by INPUT(2,:) above
for i=3:8
    INPUT(i,:) =round(INPUT(1,:)./(2));
end

%% Vertical Location of Well; Placing the well in the center of the reservoir
INPUT(9,:) = 85;
%% lateral spacing for lateral "1" from the "surface" block.
%Only need lateral spacing for odd number laterals
for i=10
    INPUT(i,:) = INPUT(1,:)/(2);
end
%% lateral spacing for lateral "2"
for i=11
    INPUT(i,:) = 0;
end
%% lateral spacing for lateral 3:
% Only need lateral spacing for odd number laterals
for i=12
    INPUT(i,:) = 0;
end
%% lateral spacing for lateral 4:
for i=13
    INPUT(i,:) = 0;
end
%% lateral spacing for lateral 5:
% Only need lateral spacing for odd number laterals
for i=14
    INPUT(i,:) = 0;
end
end

```

```

%% lateral spacing for lateral 6:
for i=15
    INPUT(i,:) = 0;
end

%% ensuring sum of spacings stays within reservoir limit (5100')
% 5100 = 5100 ft = Reservoir Limit
% 170 = 170 blocks in horizontal/vertical directions

for i=1:size(INPUT,2)
    TT =
INPUT(10,i)+INPUT(11,i)+INPUT(12,i)+INPUT(13,i)+INPUT(14,i)+INPUT(15,i);
    if TT>5100
        for j=10:15
            INPUT(j,i) = INPUT(j,i)-170;
            if INPUT(j,i)<0
                INPUT(j,i)=INPUT(j,i)*-1;
            end
        end
    end
    if TT>INPUT(1,i)
        for j=10:15
            INPUT(j,i) = INPUT(j,i)-((TT-INPUT(1,i))/7);
            if INPUT(j,i)<0
                INPUT(j,i)=INPUT(j,i)*-1;
            end
        end
    end
end

end

%% Rounding inputs 1:15 to the nearest whole number
for i=1:15
    %Number of Laterals
    if i==2
        INPUT(i,:)=round(INPUT(i,:));
        %Vertical Placement of Well
    elseif i==9
        INPUT(i,:)=round(INPUT(i,:));
        %All remaining inputs
    else
        INPUT(i,:)= roundn(INPUT(i,:),1);
    end
end

end

%% Placing zeros where there are no laterals
for i=1:size(INPUT,2)
    hh=8;h=15;
    if INPUT(2,i)<6
        for k=1:(6-INPUT(2,i))
            INPUT(hh,i)=0;
            INPUT(h,i)=0;
            hh=hh-1; h=h-1;
        end
    end
end

```

```

end

%% Direction of laterals, if they exist.
% (-1 = lateral facing ~North, +1 = lateral facing ~South)
INPUT(16,:)=-1;
INPUT(17,:)=1;
INPUT(18,:)=-1;
INPUT(19,:)=1;
INPUT(20,:)=-1;
INPUT(21,:)=1;

%% Placing zeros where there are no laterals
for i=1:size(INPUT,2)
    hh=21;
    if INPUT(2,i)<6
        for k=1:(6-INPUT(2,i))
            INPUT(hh,i)=0;
            hh=hh-1;
        end
    end
end

end

%% Range for Bottom Hole Pressure
INPUT(22,:)=1500+(8500-1500).*rand(nc,1);
INPUT(22,:)=round(INPUT(22,+)/0.01)*0.01;

%% Fracture Perm
INPUT(23,:)= 0.1+(250-0.1).*rand(nc,1); %Kxf max = 1 Min = 0.0001
a = [INPUT(23,:)]; % Work with this array (k_frac).
A_1 = zeros(size(a)); % Make another array to fill up for varying
B_1 = zeros(size(a)); % fracture perm
C_1 = zeros(size(a));
D_1 = zeros(size(a));
E_1 = zeros(size(a));
F_1 = zeros(size(a));
G_1 = zeros(size(a));
H_1 = zeros(size(a));
I_1 = zeros(size(a));
J_1 = zeros(size(a));
K_1 = zeros(size(a));
L_1 = zeros(size(a));
M_1 = zeros(size(a));
N_1 = zeros(size(a));
O_1 = zeros(size(a));
P_1 = zeros(size(a));
Q_1 = zeros(size(a));
R_1 = zeros(size(a));
S_1 = zeros(size(a));
T_1 = zeros(size(a));
U_1 = zeros(size(a));
V_1 = zeros(size(a));
W_1 = zeros(size(a));
X_1 = zeros(size(a));
Y_1 = zeros(size(a));
Z_1 = zeros(size(a));
A_2 = zeros(size(a));

```

```

B_2 = zeros(size(a));
C_2 = zeros(size(a));
D_2 = zeros(size(a));
E_2 = zeros(size(a));
F_2 = zeros(size(a));
G_2 = zeros(size(a));
H_2 = zeros(size(a));
I_2 = zeros(size(a));
J_2 = zeros(size(a));
K_2 = zeros(size(a));
L_2 = zeros(size(a));
M_2 = zeros(size(a));
N_2 = zeros(size(a));
O_2 = zeros(size(a));
P_2 = zeros(size(a));
Q_2 = zeros(size(a));
R_2 = zeros(size(a));
S_2 = zeros(size(a));
T_2 = zeros(size(a));
U_2 = zeros(size(a));
V_2 = zeros(size(a));
W_2 = zeros(size(a));
X_2 = zeros(size(a));
Y_2 = zeros(size(a));
Z_2 = zeros(size(a));
A_3 = zeros(size(a));
B_3 = zeros(size(a));
C_3 = zeros(size(a));
D_3 = zeros(size(a));

for ii = 1:round(length(a)/(56))
    A_1(ii)=0.1+(1-0.1).*rand(1);
end
for ii = round(length(a)/(56)+1):round(2*length(a)/(56))
    B_1(ii)=1+(10-1).*rand(1);
end
for ii = round(2*length(a)/(56)+1):round(3*length(a)/(56))
    C_1(ii)=10+(100-10).*rand(1);
end
for ii = round(3*length(a)/(56)+1):round(4*length(a)/(56))
    D_1(ii)=100+(250-100).*rand(1);
end
for ii = round(4*length(a)/(56)+1):round(5*length(a)/(56))
    E_1(ii)=0.1+(1-0.1).*rand(1);
end
for ii = round(5*length(a)/(56)+1):round(6*length(a)/(56))
    F_1(ii)=1+(10-1).*rand(1);
end
for ii = round(6*length(a)/(56)+1):round(7*length(a)/(56))
    G_1(ii)=10+(100-10).*rand(1);
end
for ii = round(7*length(a)/(56)+1):round(8*length(a)/(56))
    H_1(ii)=100+(250-100).*rand(1);
end
for ii = round(8*length(a)/(56)+1):round(9*length(a)/(56))
    I_1(ii)=0.1+(1-0.1).*rand(1);
end

```

```

for ii = round(9*length(a)/(56)+1):round(10*length(a)/(56))
    J_1(ii)=1+(10-1).*rand(1);
end
for ii = round(10*length(a)/(56)+1):round(11*length(a)/(56))
    K_1(ii)=10+(100-10).*rand(1);
end
for ii = round(11*length(a)/(56)+1):round(12*length(a)/(56))
    L_1(ii)=100+(250-100).*rand(1);
end
for ii = round(12*length(a)/(56)+1):round(13*length(a)/(56))
    M_1(ii)=0.1+(1-0.1).*rand(1);
end
for ii = round(13*length(a)/(56)+1):round(14*length(a)/(56))
    N_1(ii)=1+(10-1).*rand(1);
end
for ii = round(14*length(a)/(56)+1):round(15*length(a)/(56))
    O_1(ii)=10+(100-10).*rand(1);
end
for ii = round(15*length(a)/(56)+1):round(16*length(a)/(56))
    P_1(ii)=100+(250-100).*rand(1);
end
for ii = round(16*length(a)/(56)+1):round(17*length(a)/(56))
    Q_1(ii)=0.1+(1-0.1).*rand(1);
end
for ii = round(17*length(a)/(56)+1):round(18*length(a)/(56))
    R_1(ii)=1+(10-1).*rand(1);
end
for ii = round(18*length(a)/(56)+1):round(19*length(a)/(56))
    S_1(ii)=10+(100-10).*rand(1);
end
for ii = round(19*length(a)/(56)+1):round(20*length(a)/(56))
    T_1(ii)=100+(250-100).*rand(1);
end
for ii = round(20*length(a)/(56)+1):round(21*length(a)/(56))
    U_1(ii)=0.1+(1-0.1).*rand(1);
end
for ii = round(21*length(a)/(56)+1):round(22*length(a)/(56))
    V_1(ii)=1+(10-1).*rand(1);
end
for ii = round(22*length(a)/(56)+1):round(23*length(a)/(56))
    W_1(ii)=10+(100-10).*rand(1);
end
for ii = round(23*length(a)/(56)+1):round(24*length(a)/(56))
    X_1(ii)=100+(250-100).*rand(1);
end
for ii = round(24*length(a)/(56)+1):round(25*length(a)/(56))
    Y_1(ii)=0.1+(1-0.1).*rand(1);
end
for ii = round(25*length(a)/(56)+1):round(26*length(a)/(56))
    Z_1(ii)=1+(10-1).*rand(1);
end
for ii = round(26*length(a)/(56)+1):round(27*length(a)/(56))
    A_2(ii)=10+(100-10).*rand(1);
end
for ii = round(27*length(a)/(56)+1):round(28*length(a)/(56))
    B_2(ii)=100+(250-100).*rand(1);
end

```

```

for ii = round(28*length(a)/(56)+1):round(29*length(a)/(56))
    C_2(ii)=0.1+(1-0.1).*rand(1);
end
for ii = round(29*length(a)/(56)+1):round(30*length(a)/(56))
    D_2(ii)=1+(10-1).*rand(1);
end
for ii = round(30*length(a)/(56)+1):round(31*length(a)/(56))
    E_2(ii)=10+(100-10).*rand(1);
end
for ii = round(31*length(a)/(56)+1):round(32*length(a)/(56))
    F_2(ii)=100+(250-100).*rand(1);
end
for ii = round(32*length(a)/(56)+1):round(33*length(a)/(56))
    G_2(ii)=0.1+(1-0.1).*rand(1);
end
for ii = round(33*length(a)/(56)+1):round(34*length(a)/(56))
    H_2(ii)=1+(10-1).*rand(1);
end
for ii = round(34*length(a)/(56)+1):round(35*length(a)/(56))
    I_2(ii)=10+(100-10).*rand(1);
end
for ii = round(35*length(a)/(56)+1):round(36*length(a)/(56))
    J_2(ii)=100+(250-100).*rand(1);
end
for ii = round(36*length(a)/(56)+1):round(37*length(a)/(56))
    K_2(ii)=0.1+(1-0.1).*rand(1);
end
for ii = round(37*length(a)/(56)+1):round(38*length(a)/(56))
    L_2(ii)=1+(10-1).*rand(1);
end
for ii = round(38*length(a)/(56)+1):round(39*length(a)/(56))
    M_2(ii)=10+(100-10).*rand(1);
end
for ii = round(39*length(a)/(56)+1):round(40*length(a)/(56))
    N_2(ii)=100+(250-100).*rand(1);
end
for ii = round(40*length(a)/(56)+1):round(41*length(a)/(56))
    O_2(ii)=0.1+(1-0.1).*rand(1);
end
for ii = round(41*length(a)/(56)+1):round(42*length(a)/(56))
    P_2(ii)=1+(10-1).*rand(1);
end
for ii = round(42*length(a)/(56)+1):round(43*length(a)/(56))
    Q_2(ii)=10+(100-10).*rand(1);
end
for ii = round(43*length(a)/(56)+1):round(44*length(a)/(56))
    R_2(ii)=100+(250-100).*rand(1);
end
for ii = round(44*length(a)/(56)+1):round(45*length(a)/(56))
    S_2(ii)=0.1+(1-0.1).*rand(1);
end
for ii = round(45*length(a)/(56)+1):round(46*length(a)/(56))
    T_2(ii)=1+(10-1).*rand(1);
end
for ii = round(46*length(a)/(56)+1):round(47*length(a)/(56))
    U_2(ii)=10+(100-10).*rand(1);
end

```

```

for ii = round(47*length(a)/(56)+1):round(48*length(a)/(56))
    V_2(ii)=100+(250-100).*rand(1);
end
for ii = round(48*length(a)/(56)+1):round(49*length(a)/(56))
    W_2(ii)=0.1+(1-0.1).*rand(1);
end
for ii = round(49*length(a)/(56)+1):round(50*length(a)/(56))
    X_2(ii)=1+(10-1).*rand(1);
end
for ii = round(50*length(a)/(56)+1):round(51*length(a)/(56))
    Y_2(ii)=10+(100-10).*rand(1);
end
for ii = round(51*length(a)/(56)+1):round(52*length(a)/(56))
    Z_2(ii)=100+(250-100).*rand(1);
end
for ii = round(52*length(a)/(56)+1):round(53*length(a)/(56))
    A_3(ii)=0.1+(1-0.1).*rand(1);
end
for ii = round(53*length(a)/(56)+1):round(54*length(a)/(56))
    B_3(ii)=1+(10-1).*rand(1);
end
for ii = round(54*length(a)/(56)+1):round(55*length(a)/(56))
    C_3(ii)=10+(100-10).*rand(1);
end
for ii = round(55*length(a)/(56)+1):round(56*length(a)/(56))
    D_3(ii)=100+(250-100).*rand(1);
end

INPUT(23,:)=A_1+B_1+C_1+D_1+E_1+F_1+G_1+H_1+I_1+J_1+K_1+L_1+M_1+N_1+O_1+P_1+Q_1+R_1+S_1+T_1+U_1+V_1+W_1+X_1+Y_1+Z_1+A_2+B_2+C_2+D_2+E_2+F_2+G_2+H_2+I_2+J_2+K_2+L_2+M_2+N_2+O_2+P_2+Q_2+R_2+S_2+T_2+U_2+V_2+W_2+X_2+Y_2+Z_2+A_3+B_3+C_3+D_3;

%% Matrix Perm
INPUT(24,:)= 1E-8+(1E-1-1E-8).*rand(nc,1); %Kxm max = 0.1 Min = 0.000000001

%% Changing Fracture Perm
% iteration approach - look at one value at a time!
a = [INPUT(24,:)]; % Work with this array (k_matrix).
A = zeros(size(a)); % Make another array to fill up for varying
B = zeros(size(a)); % matrix perm
C = zeros(size(a));
D = zeros(size(a));
E = zeros(size(a));
F = zeros(size(a));
G = zeros(size(a));
H = zeros(size(a));
I = zeros(size(a));
J = zeros(size(a));
K = zeros(size(a));
L = zeros(size(a));
M = zeros(size(a));
N = zeros(size(a));
A_1 = zeros(size(a)); % Make another array to fill up for varying
B_1 = zeros(size(a)); % fracture perm
C_1 = zeros(size(a));

```

```

D_1 = zeros(size(a));
A_2 = zeros(size(a));
B_2 = zeros(size(a));
C_2 = zeros(size(a));
D_2 = zeros(size(a));
A_3 = zeros(size(a));
B_3 = zeros(size(a));
C_3 = zeros(size(a));
D_3 = zeros(size(a));

%% Ranges of fracture perm
for ii = 1:(length(a)/14)
    A(ii)=1E-8+(5E-8-1E-8).*rand(1);
end
for ii = (round(length(a)/14)+1):1:(round(2*length(a)/14))
    B(ii)=5E-8+(1E-7-5E-8).*rand(1);
end
for ii = (round(2*length(a)/14)+1):1:(round(3*length(a)/14))
    C(ii)=1E-7+(5E-7-1E-7).*rand(1);
end
for ii = (round(3*length(a)/14)+1):1:(round(4*length(a)/14))
    D(ii)=5E-7+(1E-6-5E-7).*rand(1);
end
for ii = (round(4*length(a)/14)+1):1:(round(5*length(a)/14))
    E(ii)=1E-6+(5E-6-1E-6).*rand(1);
end
for ii = (round(5*length(a)/14)+1):1:(round(6*length(a)/14))
    F(ii)=5E-6+(1E-5-5E-6).*rand(1);
end
for ii = (round(6*length(a)/14)+1):1:(round(7*length(a)/14))
    G(ii)=1E-5+(5E-5-1E-5).*rand(1);
end
for ii = (round(7*length(a)/14)+1):1:(round(8*length(a)/14))
    H(ii)=5E-5+(1E-4-5E-5).*rand(1);
end
for ii = (round(8*length(a)/14)+1):1:(round(9*length(a)/14))
    I(ii)=1E-4+(5E-4-1E-4).*rand(1);
end
for ii = (round(9*length(a)/14)+1):1:(round(10*length(a)/14))
    J(ii)=5E-4+(1E-3-5E-4).*rand(1);
end
for ii = (round(10*length(a)/14)+1):1:(round(11*length(a)/14))
    K(ii)=1E-3+(5E-3-1E-3).*rand(1);
end
for ii = (round(11*length(a)/14)+1):1:(round(12*length(a)/14))
    L(ii)=5E-3+(1E-2-5E-3).*rand(1);
end
for ii = (round(12*length(a)/14)+1):1:(round(13*length(a)/14))
    M(ii)=1E-2+(5E-2-1E-2).*rand(1);
end
for ii = (round(13*length(a)/14)+1):1:(round(14*length(a)/14))
    N(ii)=5E-2+(1E-1-5E-2).*rand(1);
end

%Creating New flowrate array
INPUT(24,:) = A+B+C+D+E+F+G+H+I+J+K+L+M+N;

```



```

%Check to ensure K_xm < K_xf
for i=1:nc
    if INPUT(24,i)>=INPUT(23,i)
        INPUT(24,i)=(unidrnd(50,1,1)/100)*INPUT(23,i);
    end
end

%CHECK that matrix permeability is less than fracture permeability
for i=1:nc
    if INPUT(23,i)<INPUT(24,i)
        disp('fracture permeability in x-direction is less than matrix
permeability in the x-direction')
        pause
    elseif INPUT(30,i)<INPUT(33,i)
        disp('fracture permeability in y-direction is less than matrix
permeability in the y-direction')
        pause
    end
end

%% Matrix Porosity
INPUT(25,:)=.06+ (.15-.06).*rand(nc,1);
INPUT(25,:)=round(INPUT(25,+)/0.001)*0.001;

%% Fracture Porosity
INPUT(26,:)=.005+ (.03-.005).*rand(nc,1);
INPUT(26,:)=round(INPUT(26,+)/0.0001)*0.0001;

%% Fracture Spacing
INPUT(27,:)=1+(200-1).*rand(nc,1);
INPUT(27,:)=round(INPUT(27,+)/0.1)*0.1;
for i=1:nc
    if INPUT(27,i)>200
        INPUT(27,i)=185;
    end
end

%% Range for Reservoir Temperature
INPUT(28,:)=100+(500-100).*rand(nc,1);
INPUT(28,:)=round(INPUT(28,+)/0.1)*0.1;

%% Range for constant Production value (scf) [1 MMcf to 50 MMcf]
INPUT(29,:)=1000000+(50000000-1000000).*rand(nc,1);
INPUT(29,:)=round(INPUT(29,+)/0.1)*0.1;

%% Varying the Thickness of the Reservoir
INPUT(30,:)=50+(300-50).*rand(nc,1);
INPUT(30,:)=round(INPUT(30,+)/.1)*.1;

%% Changing Flowrates depending on the characteristics of the well to
%% Avoid wasted trials
% iteration approach - look at one value at a time!
x = [INPUT(22,:)]; % Work with this array (BHP).

```

```

y = [INPUT(25,:)] ; % Work with this array (matrix porosity).
z = [INPUT(30,:)] ; % Work with this array (thickness).

T = zeros(size(x)); % Make another array to fill up for each
U = zeros(size(y)); % varying parameter
V = zeros(size(z));

%% Ranges of Pressure and their coinciding production rates
for ii = 1:length(x)
    if x(ii)<8500 && x(ii)>=5000
        T(ii) = 1000000+(50000000-1000000).*rand(1);
        %         if y(ii)<.10
        %             U(ii) = 250000+(5000000-250000).*rand(1);
        %             T(ii) = 0;
        %         end

    elseif x(ii) < 5000 && x(ii)>=4000
        T(ii) = 1000000+(40000000-1000000).*rand(1);
        if y(ii)<.10
            U(ii) = 1000000+(35000000-1000000).*rand(1);
            T(ii)=0;
            if z(ii)<100
                U(ii) = 1000000+(30000000-1000000).*rand(1);
                T(ii)=0;
            end
        end
    end

    elseif x(ii) < 4000 && x(ii)>=3000
        T(ii) = 1000000+(35000000-1000000).*rand(1);
        if y(ii)<.10
            U(ii) = 1000000+(30000000-1000000).*rand(1);
            T(ii)=0;
            if z(ii)<100
                U(ii) = 1000000+(25000000-1000000).*rand(1);
                T(ii)=0;
            end
        end
    end

    elseif x(ii) < 3000 && x(ii)>=1000
        T(ii) = 1000000+(25000000-1000000).*rand(1);
        if y(ii)<.10
            U(ii) = 1000000+(20000000-1000000).*rand(1);
            T(ii)=0;
            if z(ii)<100
                U(ii) = 1000000+(15000000-1000000).*rand(1);
                T(ii)=0;
            end
        end
    end

    T(ii)=round(T(ii)/0.01)*0.01;
    U(ii)=round(U(ii)/0.01)*0.01;
end

%Creating New flowrate array

```

```

INPUT(29,:)=T+U;

%% check for sum of spacings
for i=1:size(INPUT,2)
    i;
    TT =
INPUT(12,i)+INPUT(13,i)+INPUT(14,i)+INPUT(15,i)+INPUT(16,i)+INPUT(17,i);
%ensuring sum of spacings stays within limit
    INPUT(1,i);
    if TT>5100
        disp('Sum of lateral lengths greater than reservoir extent')
        pause
    end
end
end

%% Creating CMB Batch file to run
forCMG='CMGbatch_file.bat';

fidbat=fopen(forCMG,'wt');

%% Code that CMG will read in.
for i=1:size(INPUT,2)
    temp = ['run' num2str(i) '.dat'];
    fid = fopen(temp,'w');

    fprintf(fid, '**$=====INPUT/OUTPUT CONTROL=====\n\n');
    fprintf(fid, 'RESULTS SIMULATOR IMEX 201211\n\n');

    fprintf(fid, 'INUNIT *FIELD \n');
    fprintf(fid, 'WSRF WELL 1\n');
    fprintf(fid, 'WSRF GRID TIME\n');
    fprintf(fid, 'WSRF SECTOR TIME\n');
    fprintf(fid, 'OUTSRF WELL LAYER NONE \n');
    fprintf(fid, 'OUTSRF RES ALL \n');
    fprintf(fid, 'OUTSRF GRID SO SG SW PRES OILPOT BPP SSPRES WINFLUX \n');
    fprintf(fid, 'WPRN GRID 0 \n');
    fprintf(fid, 'OUTPRN GRID NONE \n');
    fprintf(fid, 'OUTPRN RES NONE \n');
    fprintf(fid, '**$ Distance units: ft \n');
    fprintf(fid, 'RESULTS XOFFSET      0.0000\n');
    fprintf(fid, 'RESULTS YOFFSET      0.0000\n');
    fprintf(fid, 'RESULTS ROTATION      0.0000 **$ (DEGREES) \n');
    fprintf(fid, 'RESULTS AXES-DIRECTIONS 1.0 -1.0 1.0 \n');

    fprintf(fid, '**$=====Definition of fundamental cartesian
grid=====\n\n');
    fprintf(fid, 'GRID VARI 170 170 1 \n\n');
    fprintf(fid, 'KDIR DOWN \n');

```

```

fprintf(fid, 'DI IVAR \n');
fprintf(fid, ' 170*30 \n');
fprintf(fid, 'DJ JVAR \n');
fprintf(fid, ' 170*30 \n');
fprintf(fid, 'DK ALL \n');
fprintf(fid, ' 28900*%d \n', INPUT(30,i));
fprintf(fid, 'DTOP \n');
fprintf(fid, ' 28900*5000 \n');
fprintf(fid, 'DUALPOR \n');
fprintf(fid, 'SHAPE WR \n');

fprintf(fid, '**$ 0 = null block, 1 = active block \n');
fprintf(fid, 'NULL *MATRIX CON          1 \n');
fprintf(fid, '**$ 0 = null block, 1 = active block \n');
fprintf(fid, 'NULL *FRACTURE CON          1 \n');

fprintf(fid, 'POR *MATRIX CON          %d \n', INPUT(25,i));
fprintf(fid, 'POR *FRACTURE CON          %d \n', INPUT(26,i));
fprintf(fid, 'PERMI *MATRIX CON          %d \n', INPUT(24,i));
fprintf(fid, 'PERMI *FRACTURE CON          %d \n', INPUT(23,i));
fprintf(fid, 'PERMJ *MATRIX CON          %d \n', INPUT(24,i));
fprintf(fid, 'PERMJ *FRACTURE CON          %d \n', INPUT(23,i));
fprintf(fid, 'PERMK *MATRIX CON          %d \n', INPUT(24,i));
fprintf(fid, 'PERMK *FRACTURE CON          %d \n', INPUT(23,i));
fprintf(fid, 'DJFRAC CON          %d \n', INPUT(27,i));
fprintf(fid, 'DIFRAC CON          %d \n', INPUT(27,i));
fprintf(fid, 'DKFRAC CON          %d \n', INPUT(27,i));
fprintf(fid, '**$ 0 = pinched block, 1 = active block \n');
fprintf(fid, 'PINCHOUTARRAY CON          1 \n');
fprintf(fid, 'CPOR FRACTURE 0.0000001 \n');
fprintf(fid, 'CPOR MATRIX 0.0000001 \n');
fprintf(fid, 'MODEL GASWATER \n');
fprintf(fid, 'TRES %d \n', INPUT(28,i));
fprintf(fid, 'PVTG EG 1 \n');
fprintf(fid, '**$          p          Eg          visg \n');
fprintf(fid, '14.696      4.80783  0.0128675 \n');
fprintf(fid, '367.05      123.677  0.0131685 \n');
fprintf(fid, '719.403     249.101  0.0136271 \n');
fprintf(fid, '1071.76     380.05   0.0142057 \n');
fprintf(fid, '1424.11     514.731  0.0148938 \n');
fprintf(fid, '1776.46     650.658  0.0156807 \n');
fprintf(fid, '2128.82     784.996  0.0165507 \n');
fprintf(fid, '2481.17     915.068  0.0174841 \n');
fprintf(fid, '2833.52     1038.79  0.0184597 \n');
fprintf(fid, '3185.88     1154.89  0.0194577 \n');
fprintf(fid, '3538.23     1262.79  0.0204617 \n');
fprintf(fid, '3890.59     1362.49  0.0214591 \n');
fprintf(fid, '4242.94     1454.32  0.0224406 \n');
fprintf(fid, '4595.29     1538.83  0.0234002 \n');
fprintf(fid, '4947.65     1616.62  0.0243339 \n');
fprintf(fid, '5300.00     1688.33  0.0252398 \n');
fprintf(fid, 'GRAVITY GAS 0.5537 \n');
fprintf(fid, 'REFPW 3500 \n');
fprintf(fid, 'DENSITY WATER 62.2554 \n');
fprintf(fid, 'BWI 1.00934 \n');
fprintf(fid, 'CW 2.87223e-006 \n');

```

```

fprintf(fid,'VWI 0.47184 \n');
fprintf(fid,'CVW .000000001 \n');
fprintf(fid,'ROCKFLUID \n');
fprintf(fid,'RPT 1 \n');
fprintf(fid,'**$      Sw      krw      Pcgw \n');
fprintf(fid,'SWT \n');
fprintf(fid,'      0      0      0.0 \n');
fprintf(fid,'      1      1      0 \n');
fprintf(fid,'**$      Sg      krg \n');
fprintf(fid,'SGT \n');
fprintf(fid,'      0      0 \n');
fprintf(fid,'      1      1 \n');
fprintf(fid,'INITIAL \n');
fprintf(fid,'VERTICAL DEPTH_AVE WATER_GAS EQUIL NOTRANZONE \n');
fprintf(fid,'\n');
fprintf(fid,'REFDEPTH 5080 \n');
fprintf(fid,'REFPRES %d \n', INPUT(22,i));
fprintf(fid,'DWGC 10000 \n');
fprintf(fid,'GOC_PC 0 \n');
fprintf(fid,'WOC_PC 0 \n');
fprintf(fid,'NUMERICAL \n');

fprintf(fid,'RUN \n');
fprintf(fid,'DATE 2013 1 1 \n');
fprintf(fid,'**$ \n');
fprintf(fid,'WELL ''Open_Well_30_ft'' \n');
fprintf(fid,'PRODUCER ''Open_Well_30_ft'' \n');
fprintf(fid,'OPERATE MAX STG %d CONT\n', INPUT(29,i));
fprintf(fid,'OPERATE MIN BHP 14.7 STOP \n');
fprintf(fid,'**$      rad geofac wfrac skin \n');
fprintf(fid,'GEOMETRY K 0.25 0.37 1. 0. \n');
fprintf(fid,'PERF GEOA ''Open_Well_30_ft'' \n');
fprintf(fid,'**$ UBA      ff Status Connection \n');

%% determining the well block connected to the surface
%5100 = 5100ft = reservoir width/length
%30 = 30ft = individual grid block size
%170 = 170 grid blocks in x- and y-direction
%42.4264 = diagonal length of 30x30 block
ii=round(((5100-INPUT(1,i))/2)/30);
cnt=0;

for j=ii:(ii+round(INPUT(1,i)/30))

    if j==ii
        fprintf(fid,'      %0.0f %0.0f 1      1. OPEN FLOW-TO %s
REFLAYER \n',ii,INPUT(9,i),'SURFACE');
    else
        cnt=cnt+1;
        fprintf(fid,'      %0.0f %0.0f 1      1. OPEN FLOW-TO %d
\n',j,INPUT(9,i),cnt);
    end
end

```

```

%% Placing laterals. Focusing primarily on the even Inputs. i.e 2, 4 and
6.
a=0;
if INPUT(2,i)==1

    if INPUT(16,i)==1
        u = (170-INPUT(9,i))*30;
    else u=INPUT(9,i)*30;
    end

    if INPUT(3,i)>u
        INPUT(3,i)=u;
%trying to ensure that the lateral length stays within reservoir limits
    end

    for j=1:round(INPUT(3,i)/42.4264)
        a=a+1;
        cnt=cnt+1;
        if j==1

            fprintf(fid, '      %0.0f %0.0f 1          1.  OPEN    FLOW-TO
%d
\n', (ii+round(INPUT(10,i)/30)+a), INPUT(9,i)+(INPUT(16,i)*a), round(INPUT(10,i)
/30)+1);

            else
                fprintf(fid, '      %0.0f %0.0f 1          1.  OPEN    FLOW-TO
%d \n', (ii+round(INPUT(10,i)/30)+a), INPUT(9,i)+(INPUT(16,i)*a), cnt);
            end
        end
    end

a=0;
if INPUT(2,i)>1

    for iii=1:INPUT(2,i)
        if iii==1

            if INPUT(16,i)==1
                u = (170-INPUT(9,i))*30;
            else u=INPUT(9,i)*30;
            end

            if INPUT(3,i)>u
                INPUT(3,i)=u;
            end

            aaa=round(INPUT(10,i)/30);
            for j=1:round(INPUT(3,i)/42.4264)
                a=a+1;
                cnt=cnt+1;
                if j==1
                    fprintf(fid, '      %0.0f %0.0f 1          1.  OPEN
FLOW-TO  %d \n', (ii+aaa+a), INPUT(9,i)+(INPUT(16,i)*a), aaa+1);
                else

```

```

                                fprintf(fid, '      %0.0f %0.0f 1      1.  OPEN
FLOW-TO  %d \n', (ii+aaa+a), INPUT(9,i)+(INPUT(16,i)*a), cnt);
                                end
                                end
                                end
                                if iii==2
                                    a=0;

                                    if INPUT(17,i)==1
                                        u = (170-INPUT(9,i))*30;
                                    else u=INPUT(9,i)*30;
                                    end

                                    if INPUT(4,i)>u
                                        INPUT(4,i)=u;
                                    end

                                    aaa=round((INPUT(10,i)+INPUT(11,i))/30);
                                    for j=1:round(INPUT(4,i)/42.4264)
                                        a=a+1;
                                        cnt=cnt+1;
                                        if j==1
                                            fprintf(fid, '      %0.0f %0.0f 1      1.  OPEN
FLOW-TO  %d \n', (ii+aaa+a), INPUT(9,i)+(INPUT(17,i)*a), aaa+1);
                                            else
                                            fprintf(fid, '      %0.0f %0.0f 1      1.  OPEN
FLOW-TO  %d \n', (ii+aaa+a), INPUT(9,i)+(INPUT(17,i)*a), cnt);
                                            end
                                        end
                                    end
                                end
                                if iii==3
                                    a=0;

                                    if INPUT(18,i)==1
                                        u = (170-INPUT(9,i))*30;
                                    else u=INPUT(9,i)*30;
                                    end

                                    if INPUT(5,i)>u
                                        INPUT(5,i)=u;
                                    end

                                    aaa=round((INPUT(10,i)+INPUT(11,i)+INPUT(12,i))/30);
                                    for j=1:round(INPUT(5,i)/42.4264)
                                        a=a+1;
                                        cnt=cnt+1;
                                        if j==1
                                            fprintf(fid, '      %0.0f %0.0f 1      1.  OPEN
FLOW-TO  %d \n', (ii+aaa+a), INPUT(9,i)+(INPUT(18,i)*a), aaa+1);
                                            else
                                            fprintf(fid, '      %0.0f %0.0f 1      1.  OPEN
FLOW-TO  %d \n', (ii+aaa+a), INPUT(9,i)+(INPUT(18,i)*a), cnt);
                                            end
                                        end
                                    end
                                end
                                end
                                end

```

```

if iii==4
    a=0;

    if INPUT(19,i)==1
        u = (170-INPUT(9,i))*30;
    else u=INPUT(9,i)*30;
    end

    if INPUT(6,i)>u
        INPUT(6,i)=u;
    end

aaa=round((INPUT(10,i)+INPUT(11,i)+INPUT(12,i)+INPUT(13,i))/30);
for j=1:round(INPUT(6,i)/42.4264)
    a=a+1;
    cnt=cnt+1;
    if j==1
        fprintf(fid, '      %0.0f %0.0f 1          1.  OPEN
FLOW-TO  %d \n', (ii+aaa+a), INPUT(9,i)+(INPUT(19,i)*a), aaa+1);
    else
        fprintf(fid, '      %0.0f %0.0f 1          1.  OPEN
FLOW-TO  %d \n', (ii+aaa+a), INPUT(9,i)+(INPUT(19,i)*a), cnt);
    end
end
end

if iii==5
    a=0;

    if INPUT(20,i)==1
        u = (170-INPUT(9,i))*30;
    else u=INPUT(9,i)*30;
    end

    if INPUT(7,i)>u
        INPUT(7,i)=u;
    end

aaa=round((INPUT(10,i)+INPUT(11,i)+INPUT(12,i)+INPUT(13,i)+INPUT(14,i))/30);
for j=1:round(INPUT(7,i)/42.4264)
    a=a+1;
    cnt=cnt+1;
    if j==1
        fprintf(fid, '      %0.0f %0.0f 1          1.  OPEN
FLOW-TO  %d \n', (ii+aaa+a), INPUT(9,i)+(INPUT(20,i)*a), aaa+1);
    else
        fprintf(fid, '      %0.0f %0.0f 1          1.  OPEN
FLOW-TO  %d \n', (ii+aaa+a), INPUT(9,i)+(INPUT(20,i)*a), cnt);
    end
end
end

```



```

        if iii==6
            a=0;

aaa=round((INPUT(10,i)+INPUT(11,i)+INPUT(12,i)+INPUT(13,i)+INPUT(14,i)+INPUT(
15,i))/30);

        if INPUT(21,i)==1
            u = (170-INPUT(9,i))*30;
        else u=INPUT(9,i)*30;
        end

        if INPUT(9,i)<25
            if INPUT(21,i)==-1
                if INPUT(8,i)>u
                    INPUT(8,i)=u;
                    if aaa>27
                        INPUT(8,i)=round(u/2);
                    end
                end
            elseif INPUT(21,i)==1
                if aaa>27
                    if INPUT(8,i)>u
                        INPUT(8,i)=round(u/2);
                    end
                end
            end
        end

        if INPUT(9,i)>=25
            if INPUT(21,i)==1

                if INPUT(8,i)>u
                    INPUT(8,i)=u;
                    if aaa>27
                        INPUT(8,i)=round(INPUT(8,i)/4);
                    end
                end
            elseif INPUT(25,i)==-1
                if aaa>27
                    if INPUT(8,i)>u
                        INPUT(8,i)=round(u/2);
                    end
                end
            end
        end

        for j=1:round(INPUT(8,i)/42.4264)
            a=a+1;
            cnt=cnt+1;
            if j==1
                fprintf(fid,'          %0.0f %0.0f 1          1.  OPEN
FLOW-TO  %d \n',(ii+aaa+a),INPUT(9,i)+(INPUT(21,i)*a),aaa+1);
            else

```

```

                                fprintf(fid, '      %0.0f %0.0f 1      1.  OPEN
FLOW-TO  %d \n', (ii+aaa+a), INPUT(9,i)+(INPUT(21,i)*a),cnt);
                                end
                                end
                                end

                                end
                                end

                                %% Dates that CMG will record a data point. Doing a drawdown test with
                                % constant production for 60 days, as stated above.
                                fprintf(fid, 'DATE 2013 1  1.00207 \n');
                                fprintf(fid, 'DATE 2013 1  1.01389 \n');
                                fprintf(fid, 'DATE 2013 1  1.02778 \n');
                                fprintf(fid, 'DATE 2013 1  1.04167 \n');
                                fprintf(fid, 'DATE 2013 1  1.05556 \n');
                                fprintf(fid, 'DATE 2013 1  1.06944 \n');
                                fprintf(fid, 'DATE 2013 1  1.08333 \n');
                                fprintf(fid, 'DATE 2013 1  1.09722 \n');
                                fprintf(fid, 'DATE 2013 1  1.11111 \n');
                                fprintf(fid, 'DATE 2013 1  1.12500 \n');
                                fprintf(fid, 'DATE 2013 1  1.13889 \n');
                                fprintf(fid, 'DATE 2013 1  1.15278 \n');
                                fprintf(fid, 'DATE 2013 1  1.16667 \n');
                                fprintf(fid, 'DATE 2013 1  1.18056 \n');
                                fprintf(fid, 'DATE 2013 1  1.19444 \n');
                                fprintf(fid, 'DATE 2013 1  1.20833 \n');
                                fprintf(fid, 'DATE 2013 1  1.22222 \n');
                                fprintf(fid, 'DATE 2013 1  1.23611 \n');
                                fprintf(fid, 'DATE 2013 1  1.25000 \n');
                                fprintf(fid, 'DATE 2013 1  1.26389 \n');
                                fprintf(fid, 'DATE 2013 1  1.27778 \n');
                                fprintf(fid, 'DATE 2013 1  1.29167 \n');
                                fprintf(fid, 'DATE 2013 1  1.30556 \n');
                                fprintf(fid, 'DATE 2013 1  1.31944 \n');
                                fprintf(fid, 'DATE 2013 1  1.33333 \n');
                                fprintf(fid, 'DATE 2013 1  1.34722 \n');
                                fprintf(fid, 'DATE 2013 1  1.36111 \n');
                                fprintf(fid, 'DATE 2013 1  1.37500 \n');
                                fprintf(fid, 'DATE 2013 1  1.38889 \n');
                                fprintf(fid, 'DATE 2013 1  1.40278 \n');
                                fprintf(fid, 'DATE 2013 1  1.41667 \n');
                                fprintf(fid, 'DATE 2013 1  1.43056 \n');
                                fprintf(fid, 'DATE 2013 1  1.44444 \n');
                                fprintf(fid, 'DATE 2013 1  1.45833 \n');
                                fprintf(fid, 'DATE 2013 1  1.47222 \n');
                                fprintf(fid, 'DATE 2013 1  1.48611 \n');
                                fprintf(fid, 'DATE 2013 1  1.50000 \n');
                                fprintf(fid, 'DATE 2013 1  1.51389 \n');
                                fprintf(fid, 'DATE 2013 1  1.52778 \n');
                                fprintf(fid, 'DATE 2013 1  1.54167 \n');
                                fprintf(fid, 'DATE 2013 1  1.55556 \n');
                                fprintf(fid, 'DATE 2013 1  1.56944 \n');
                                fprintf(fid, 'DATE 2013 1  1.58333 \n');
                                fprintf(fid, 'DATE 2013 1  1.59722 \n');
                                fprintf(fid, 'DATE 2013 1  1.61111 \n');

```

```
fprintf(fid, 'DATE 2013 1 1.62500 \n');
fprintf(fid, 'DATE 2013 1 1.63889 \n');
fprintf(fid, 'DATE 2013 1 1.65278 \n');
fprintf(fid, 'DATE 2013 1 1.66667 \n');
fprintf(fid, 'DATE 2013 1 1.68056 \n');
fprintf(fid, 'DATE 2013 1 1.69444 \n');
fprintf(fid, 'DATE 2013 1 1.70833 \n');
fprintf(fid, 'DATE 2013 1 1.72222 \n');
fprintf(fid, 'DATE 2013 1 1.73611 \n');
fprintf(fid, 'DATE 2013 1 1.75000 \n');
fprintf(fid, 'DATE 2013 1 1.76389 \n');
fprintf(fid, 'DATE 2013 1 1.77778 \n');
fprintf(fid, 'DATE 2013 1 1.79167 \n');
fprintf(fid, 'DATE 2013 1 1.80556 \n');
fprintf(fid, 'DATE 2013 1 1.81944 \n');
fprintf(fid, 'DATE 2013 1 1.83333 \n');
fprintf(fid, 'DATE 2013 1 1.84722 \n');
fprintf(fid, 'DATE 2013 1 1.86111 \n');
fprintf(fid, 'DATE 2013 1 1.87500 \n');
fprintf(fid, 'DATE 2013 1 1.88889 \n');
fprintf(fid, 'DATE 2013 1 1.90278 \n');
fprintf(fid, 'DATE 2013 1 1.91667 \n');
fprintf(fid, 'DATE 2013 1 1.93056 \n');
fprintf(fid, 'DATE 2013 1 1.94444 \n');
fprintf(fid, 'DATE 2013 1 1.95833 \n');
fprintf(fid, 'DATE 2013 1 1.97222 \n');
fprintf(fid, 'DATE 2013 1 1.98611 \n');
fprintf(fid, 'DATE 2013 1 2.00000 \n');
fprintf(fid, 'DATE 2013 1 2.08333 \n');
fprintf(fid, 'DATE 2013 1 2.16667 \n');
fprintf(fid, 'DATE 2013 1 2.25000 \n');
fprintf(fid, 'DATE 2013 1 2.33333 \n');
fprintf(fid, 'DATE 2013 1 2.41667 \n');
fprintf(fid, 'DATE 2013 1 2.50000 \n');
fprintf(fid, 'DATE 2013 1 2.58333 \n');
fprintf(fid, 'DATE 2013 1 2.66667 \n');
fprintf(fid, 'DATE 2013 1 2.75000 \n');
fprintf(fid, 'DATE 2013 1 2.83333 \n');
fprintf(fid, 'DATE 2013 1 2.91667 \n');
fprintf(fid, 'DATE 2013 1 3.00000 \n');
fprintf(fid, 'DATE 2013 1 3.08333 \n');
fprintf(fid, 'DATE 2013 1 3.16667 \n');
fprintf(fid, 'DATE 2013 1 3.25000 \n');
fprintf(fid, 'DATE 2013 1 3.33333 \n');
fprintf(fid, 'DATE 2013 1 3.41667 \n');
fprintf(fid, 'DATE 2013 1 3.50000 \n');
fprintf(fid, 'DATE 2013 1 3.58333 \n');
fprintf(fid, 'DATE 2013 1 3.66667 \n');
fprintf(fid, 'DATE 2013 1 3.75000 \n');
fprintf(fid, 'DATE 2013 1 3.83333 \n');
fprintf(fid, 'DATE 2013 1 3.91667 \n');
fprintf(fid, 'DATE 2013 1 4.08333 \n');
fprintf(fid, 'DATE 2013 1 4.25000 \n');
fprintf(fid, 'DATE 2013 1 4.41667 \n');
fprintf(fid, 'DATE 2013 1 4.58333 \n');
fprintf(fid, 'DATE 2013 1 4.75000 \n');
fprintf(fid, 'DATE 2013 1 4.91667 \n');
```

```
fprintf(fid, 'DATE 2013 1 5.08333 \n');
fprintf(fid, 'DATE 2013 1 5.25000 \n');
fprintf(fid, 'DATE 2013 1 5.41667 \n');
fprintf(fid, 'DATE 2013 1 5.58333 \n');
fprintf(fid, 'DATE 2013 1 5.75000 \n');
fprintf(fid, 'DATE 2013 1 5.91667 \n');
fprintf(fid, 'DATE 2013 1 6.08333 \n');
fprintf(fid, 'DATE 2013 1 6.25000 \n');
fprintf(fid, 'DATE 2013 1 6.41667 \n');
fprintf(fid, 'DATE 2013 1 6.58333 \n');
fprintf(fid, 'DATE 2013 1 6.75000 \n');
fprintf(fid, 'DATE 2013 1 6.91667 \n');
fprintf(fid, 'DATE 2013 1 7.08333 \n');
fprintf(fid, 'DATE 2013 1 7.25000 \n');
fprintf(fid, 'DATE 2013 1 7.41667 \n');
fprintf(fid, 'DATE 2013 1 7.58333 \n');
fprintf(fid, 'DATE 2013 1 7.75000 \n');
fprintf(fid, 'DATE 2013 1 7.91667 \n');
fprintf(fid, 'DATE 2013 1 8.08333 \n');
fprintf(fid, 'DATE 2013 1 8.25000 \n');
fprintf(fid, 'DATE 2013 1 8.41667 \n');
fprintf(fid, 'DATE 2013 1 8.58333 \n');
fprintf(fid, 'DATE 2013 1 8.75000 \n');
fprintf(fid, 'DATE 2013 1 8.91667 \n');
fprintf(fid, 'DATE 2013 1 9.08333 \n');
fprintf(fid, 'DATE 2013 1 9.25000 \n');
fprintf(fid, 'DATE 2013 1 9.41667 \n');
fprintf(fid, 'DATE 2013 1 9.58333 \n');
fprintf(fid, 'DATE 2013 1 9.75000 \n');
fprintf(fid, 'DATE 2013 1 9.91667 \n');
fprintf(fid, 'DATE 2013 1 10.08333 \n');
fprintf(fid, 'DATE 2013 1 10.25000 \n');
fprintf(fid, 'DATE 2013 1 10.41667 \n');
fprintf(fid, 'DATE 2013 1 10.58333 \n');
fprintf(fid, 'DATE 2013 1 10.75000 \n');
fprintf(fid, 'DATE 2013 1 10.91667 \n');
fprintf(fid, 'DATE 2013 1 11.08333 \n');
fprintf(fid, 'DATE 2013 1 11.20833 \n');
fprintf(fid, 'DATE 2013 1 11.33333 \n');
fprintf(fid, 'DATE 2013 1 11.45833 \n');
fprintf(fid, 'DATE 2013 1 11.58333 \n');
fprintf(fid, 'DATE 2013 1 11.83333 \n');
fprintf(fid, 'DATE 2013 1 12.08333 \n');
fprintf(fid, 'DATE 2013 1 12.33333 \n');
fprintf(fid, 'DATE 2013 1 12.58333 \n');
fprintf(fid, 'DATE 2013 1 12.83333 \n');
fprintf(fid, 'DATE 2013 1 13.08333 \n');
fprintf(fid, 'DATE 2013 1 13.33333 \n');
fprintf(fid, 'DATE 2013 1 13.58333 \n');
fprintf(fid, 'DATE 2013 1 13.83333 \n');
fprintf(fid, 'DATE 2013 1 14.08333 \n');
fprintf(fid, 'DATE 2013 1 14.33333 \n');
fprintf(fid, 'DATE 2013 1 14.58333 \n');
fprintf(fid, 'DATE 2013 1 14.83333 \n');
fprintf(fid, 'DATE 2013 1 15.08333 \n');
fprintf(fid, 'DATE 2013 1 15.33333 \n');
fprintf(fid, 'DATE 2013 1 15.58333 \n');
```

```

fprintf(fid, 'DATE 2013 1 15.83333 \n');
fprintf(fid, 'DATE 2013 1 16.08333 \n');
fprintf(fid, 'DATE 2013 1 16.33333 \n');
fprintf(fid, 'DATE 2013 1 16.58333 \n');
fprintf(fid, 'DATE 2013 1 16.83333 \n');
fprintf(fid, 'DATE 2013 1 17.08333 \n');
fprintf(fid, 'DATE 2013 1 17.33333 \n');
fprintf(fid, 'DATE 2013 1 17.58333 \n');
fprintf(fid, 'DATE 2013 1 17.83333 \n');
fprintf(fid, 'DATE 2013 1 18.08333 \n');
fprintf(fid, 'DATE 2013 1 18.54167 \n');
fprintf(fid, 'DATE 2013 1 18.87500 \n');
fprintf(fid, 'DATE 2013 1 19.20833 \n');
fprintf(fid, 'DATE 2013 1 19.54167 \n');
fprintf(fid, 'DATE 2013 1 19.87500 \n');
fprintf(fid, 'DATE 2013 1 20.20833 \n');
fprintf(fid, 'DATE 2013 1 20.54167 \n');
fprintf(fid, 'DATE 2013 1 20.87500 \n');
fprintf(fid, 'DATE 2013 1 21.20833 \n');
fprintf(fid, 'DATE 2013 1 21.54167 \n');
fprintf(fid, 'DATE 2013 1 21.87500 \n');
fprintf(fid, 'DATE 2013 1 22.20833 \n');
fprintf(fid, 'DATE 2013 1 22.54167 \n');
fprintf(fid, 'DATE 2013 1 22.87500 \n');
fprintf(fid, 'DATE 2013 1 23.20833 \n');
fprintf(fid, 'DATE 2013 1 23.54167 \n');
fprintf(fid, 'DATE 2013 1 23.87500 \n');
fprintf(fid, 'DATE 2013 1 24.20833 \n');
fprintf(fid, 'DATE 2013 1 24.54167 \n');
fprintf(fid, 'DATE 2013 1 24.87500 \n');
fprintf(fid, 'DATE 2013 1 25.20833 \n');
fprintf(fid, 'DATE 2013 1 25.50000 \n');
fprintf(fid, 'DATE 2013 1 26.00000 \n');
fprintf(fid, 'DATE 2013 1 26.50000 \n');
fprintf(fid, 'DATE 2013 1 27.00000 \n');
fprintf(fid, 'DATE 2013 1 27.50000 \n');
fprintf(fid, 'DATE 2013 1 28.00000 \n');
fprintf(fid, 'DATE 2013 1 28.50000 \n');
fprintf(fid, 'DATE 2013 1 29.00000 \n');
fprintf(fid, 'DATE 2013 1 29.50000 \n');
fprintf(fid, 'DATE 2013 1 30.00000 \n');
fprintf(fid, 'DATE 2013 1 30.50000 \n');
fprintf(fid, 'DATE 2013 1 31.00000 \n');
fprintf(fid, 'DATE 2013 1 31.50000 \n');
fprintf(fid, 'DATE 2013 2 1.00000 \n');
fprintf(fid, 'DATE 2013 2 1.50000 \n');
fprintf(fid, 'DATE 2013 2 2.00000 \n');
fprintf(fid, 'DATE 2013 2 2.50000 \n');
fprintf(fid, 'DATE 2013 2 3.00000 \n');
fprintf(fid, 'DATE 2013 2 3.50000 \n');
fprintf(fid, 'DATE 2013 2 4.00000 \n');
fprintf(fid, 'DATE 2013 2 4.50000 \n');
fprintf(fid, 'DATE 2013 2 5.00000 \n');
fprintf(fid, 'DATE 2013 2 5.50000 \n');
fprintf(fid, 'DATE 2013 2 6.00000 \n');
fprintf(fid, 'DATE 2013 2 6.50000 \n');
fprintf(fid, 'DATE 2013 2 7.00000 \n');

```

```

fprintf(fid, 'DATE 2013 2 7.50000 \n');
fprintf(fid, 'DATE 2013 2 8.00000 \n');
fprintf(fid, 'DATE 2013 2 8.50000 \n');
fprintf(fid, 'DATE 2013 2 9.00000 \n');
fprintf(fid, 'DATE 2013 2 9.50000 \n');
fprintf(fid, 'DATE 2013 2 10.00000 \n');
fprintf(fid, 'DATE 2013 2 10.50000 \n');
fprintf(fid, 'DATE 2013 2 11.00000 \n');
fprintf(fid, 'DATE 2013 2 11.50000 \n');
fprintf(fid, 'DATE 2013 2 12.00000 \n');
fprintf(fid, 'DATE 2013 2 12.50000 \n');
fprintf(fid, 'DATE 2013 2 13.00000 \n');
fprintf(fid, 'DATE 2013 2 13.50000 \n');
fprintf(fid, 'DATE 2013 2 14.00000 \n');
fprintf(fid, 'DATE 2013 2 14.50000 \n');
fprintf(fid, 'DATE 2013 2 15.00000 \n');
fprintf(fid, 'DATE 2013 2 15.50000 \n');
fprintf(fid, 'DATE 2013 2 16.00000 \n');
fprintf(fid, 'DATE 2013 2 16.50000 \n');
fprintf(fid, 'DATE 2013 2 17.00000 \n');
fprintf(fid, 'DATE 2013 2 17.50000 \n');
fprintf(fid, 'DATE 2013 2 18.00000 \n');
fprintf(fid, 'DATE 2013 2 18.50000 \n');
fprintf(fid, 'DATE 2013 2 19.00000 \n');
fprintf(fid, 'DATE 2013 2 19.50000 \n');
fprintf(fid, 'DATE 2013 2 20.00000 \n');
fprintf(fid, 'DATE 2013 2 20.50000 \n');
fprintf(fid, 'DATE 2013 2 21.00000 \n');
fprintf(fid, 'DATE 2013 2 21.50000 \n');
fprintf(fid, 'DATE 2013 2 22.00000 \n');
fprintf(fid, 'DATE 2013 2 22.50000 \n');
fprintf(fid, 'DATE 2013 2 23.00000 \n');
fprintf(fid, 'DATE 2013 2 23.50000 \n');
fprintf(fid, 'DATE 2013 2 24.00000 \n');
fprintf(fid, 'DATE 2013 2 24.50000 \n');
fprintf(fid, 'DATE 2013 2 25.00000 \n');
fprintf(fid, 'DATE 2013 2 25.50000 \n');
fprintf(fid, 'DATE 2013 2 26.00000 \n');
fprintf(fid, 'DATE 2013 2 26.50000 \n');
fprintf(fid, 'DATE 2013 2 27.00000 \n');
fprintf(fid, 'DATE 2013 2 27.50000 \n');
fprintf(fid, 'DATE 2013 2 28.00000 \n');
fprintf(fid, 'DATE 2013 2 28.50000 \n');
fprintf(fid, 'DATE 2013 3 1.00000 \n');
fprintf(fid, 'DATE 2013 3 1.50000 \n');
fprintf(fid, 'DATE 2013 3 2.00000 \n');

fclose(fid);
% Calling CMG IMEX 2012 to run this batch file
fprintf(fidbat, '%s', 'call "C:\Program Files
(x86)\CMG\IMEX\2012.11\Win_x64\EXE\mx201211.exe" -f run');
fprintf(fidbat, num2str(i));
fprintf(fidbat, '%s\n', '.dat');

end

```

```
fclose(fidbat);  
% Saving the files within a sole INPUT text file that can be accessed at  
% any time  
save INPUT.txt INPUT -ASCII -TABS
```

Appendix E: Example Training Data

Pi (psia)	P(60)	kf (md)	km (md)	ϕ m (%)	ϕ f (%)	FS (ft)	Tres (F)	h (ft)	Q (MMcf/D)	Main Wellbore Length (ft)	Individual Lateral Length (ft)	# of Laterals
5011.5	4597.3	56.5	1.66E-06	8.9	0.56	89.5	279.9	139.1	49.074	2940	1470	2
7328.3	6993.3	28.2	3.63E-06	13	1.09	116.1	377.5	166.6	38.959	2300	1150	2
6963.4	6308.8	90.8	3.34E-06	10.8	1.92	146.4	357.3	87.3	35.320	1800	900	2
7056.9	7005.3	76.7	3.79E-06	12.5	1.19	172.7	353.5	163.8	3.749	2730	1370	2
5733.4	5614.9	74.8	3.95E-06	11.5	2.01	182.4	437.5	150.6	9.010	2620	1310	2
7555.1	7379.5	77.4	2.97E-06	7.1	1.49	117.6	260.5	262.3	34.324	2480	1240	2
5990.4	5712.6	29.6	4.78E-06	13.9	1.84	189.7	208.5	201.4	26.225	3890	1940	2
2774.1	2666.9	29.8	2.97E-06	8.5	1.26	72.2	427.1	126	16.263	3440	1720	2
2866.7	2228.5	167.7	4.32E-06	10.8	0.62	138.1	151.1	63.6	21.720	1450	730	2
3249.5	3092.0	117.2	2.97E-06	12.8	2.78	141.4	435.3	182	27.361	2410	1200	2
2919.2	2622.0	224.1	4.83E-06	7.8	0.86	183.9	103.4	125.2	15.794	3700	1850	2
2129.6	1809.5	129.8	2.93E-06	12.9	1.09	111.9	232.3	74	16.833	2480	1240	2
7913.6	7896.5	127.2	1.26E-06	7.3	2.93	74.6	489.1	105.7	1.934	3780	1890	2
5634.2	5469.2	116.3	3.71E-06	9.4	2.96	83.8	348.8	87.5	17.683	3040	1520	2
2398.7	2378.9	118.8	4.71E-06	6.1	2.45	33.4	214.1	186	7.974	2930	1470	2
6698.5	6559.0	222.5	4.98E-06	6.4	2	78.7	180.4	182.1	33.590	3710	1860	2
7357.0	7145.4	240.2	3.75E-06	9.7	1.56	167.1	385.2	267.7	35.662	3580	1790	2
7413.4	7263.7	245.2	1.18E-06	11.8	0.97	148.7	421.6	289.4	30.132	1490	750	2
4240.8	4108.1	0.3	6.51E-06	9.6	1.07	51.6	178.1	169.6	4.586	1730	860	2
7927.1	7721.0	0.4	7.04E-06	12.2	2.06	32.9	237.7	173.9	9.503	1660	830	2
5473.0	2789.0	0.3	8.27E-06	14.8	1.28	109.7	448.9	79.8	26.403	1330	660	2
5434.4	5184.5	0.2	8.78E-06	11.3	0.92	45.3	182.3	226.8	19.416	3580	1790	2
2232.0	1999.5	0.7	6.69E-06	6.2	1.65	34.9	307.6	159.5	11.653	2030	1020	2
6133.5	2636.8	0.3	7.75E-06	8	2.52	175.1	331.3	63.5	46.855	2970	1490	2
7835.9	7196.3	0.9	7.51E-06	11	2.58	55.8	128.4	223.7	40.644	1060	530	2
5469.7	5291.3	0.2	9.12E-06	7.2	0.98	118.4	245.5	275.1	11.940	3650	1820	2
6182.9	6156.0	1.0	6.72E-06	6.3	0.8	78.6	272.8	249.8	3.684	3580	1790	2
7037.2	6721.1	0.2	8.09E-06	8.1	1.34	149.2	272.5	269.7	16.183	3480	1740	2
6477.6	6321.5	3.4	9.58E-06	13.5	2.91	9.5	403.4	110.8	19.302	2110	1050	2
3094.5	2931.7	4.5	9.66E-06	14.3	2.08	114.3	202.2	179	23.531	3790	1900	2
1900.4	1744.5	3.9	7.42E-06	12.8	2.43	138.8	290.3	87.6	9.236	3350	1680	2
6181.7	6069.2	5.2	9.9E-06	13.6	1.08	74.1	401.6	125.1	12.321	2260	1130	2
3841.8	3514.3	5.3	6.34E-06	6.6	2.87	99.1	435	110.2	27.719	1790	890	2
5396.1	5356.2	5.9	5.37E-06	6	0.76	126.6	348.3	65.9	1.058	2720	1360	2
6529.7	6164.4	2.5	8.17E-06	7.6	0.67	94.1	281.7	116.9	23.688	3170	1590	2
2931.6	2868.0	1.4	9E-06	7.5	1.59	36.7	104.9	245.7	6.275	1260	630	2
4179.4	3650.1	9.8	6.63E-06	12.6	1.15	142.3	357	70.1	20.207	3720	1860	2
4859.9	4825.7	6.6	6.86E-06	10.8	2.15	31.8	237.1	120.8	5.900	3340	1670	2
5991.0	5970.4	26.9	9.57E-06	13	2.68	52.2	433.3	53.4	1.363	3080	1540	2
2782.8	2396.5	87.9	8.39E-06	10.7	2.66	106.2	257.1	56.8	24.913	2170	1090	2
6256.8	6150.8	22.6	5.93E-06	8	2.69	21	154.2	212.5	33.939	2680	1340	2
5865.9	5708.7	91.6	8.39E-06	8.7	2.63	125.9	360.6	265.5	34.760	3530	1760	2
7927.9	7925.9	32.9	8.25E-06	10.5	1.28	32.2	145.9	263.7	3.368	3310	1650	2
4286.4	4119.7	58.0	5.71E-06	11	2.06	91.3	395	116.3	20.950	3220	1610	2
2157.3	2102.7	38.8	6.29E-06	10	1.61	178.5	420.4	283.3	9.243	1290	650	2
5715.7	5219.5	24.0	6.78E-06	6.9	2.46	173.1	446.1	134.2	37.258	2290	1140	2
2507.5	2487.8	97.0	5.42E-06	14.4	2.07	92.3	364.3	235.9	5.771	1620	810	2
5585.0	5201.7	78.0	6.58E-06	10.2	2.41	197.4	444.2	101.7	20.749	3940	1970	2
8048.6	8044.6	109.2	6.18E-06	6.6	2.25	112.8	329.2	245.7	1.935	2300	1150	2
4208.4	4147.9	222.0	5.4E-06	13.3	0.8	45.6	115.3	197.8	28.176	3810	1910	2

Pi (psia)	P(60)	kf (md)	km (md)	ϕ m (%)	ϕ f (%)	FS (ft)	Tres (F)	h (ft)	Q (MMcf/D)	Main Wellbore Length (ft)	Individual Lateral Length (ft)	# of Laterals
6151.0	6111.6	141.3	6.61E-06	14.9	1.85	66.7	205.5	70.6	3.927	1100	550	2
4805.2	4681.2	153.7	9.33E-06	14.5	2.99	139.6	258.3	205.3	21.476	2040	1020	2
1916.6	1785.3	119.9	5.69E-06	11.7	2.32	89.4	363.7	112.7	17.367	3770	1890	2
6494.4	6297.9	121.8	9.99E-06	10.1	2.09	40.1	355.3	80.8	27.655	3530	1770	2
8209.8	8117.7	171.3	6.36E-06	9.1	1.76	39.1	154.2	116.7	18.026	1030	510	2
3761.8	3655.0	196.8	6.26E-06	8.8	2.11	109.3	350.8	221.2	24.394	1170	580	2
7893.0	7748.4	198.3	6.42E-06	7.5	1.06	80.4	213.5	74.8	12.467	1940	970	2
7081.3	6736.1	102.0	8.82E-06	13.1	2.28	184.7	486.3	121.7	22.857	1970	990	2
7818.2	7380.1	0.7	2.75E-05	14.4	1.38	179.5	222.6	242.4	25.691	2360	1180	2
8071.0	7629.0	0.7	1.44E-05	11.8	2.63	84	138.2	66.8	16.311	3720	1860	2
1978.8	1884.7	0.5	1.63E-05	7	2.73	84.5	305.6	259.1	6.579	2370	1180	2
1680.5	1540.7	1.0	1.97E-05	9.1	0.88	3	390.6	78.8	6.739	3050	1530	2
5971.5	5673.6	0.7	4.86E-05	13.8	2.44	190.6	141.9	250.1	19.916	2320	1160	2
2755.3	2639.0	0.2	4.77E-05	6.7	1.02	66.6	157	143.3	2.415	2160	1080	2
7586.3	7378.5	0.8	2.04E-05	11.2	2.15	113.9	334.6	189.1	17.552	3330	1670	2
3309.7	3254.7	1.0	4.13E-05	9.3	0.89	162.2	484.1	185.2	2.887	2720	1360	2
7954.3	7764.2	0.9	1.62E-05	6.4	1.94	75.3	156.5	242.2	24.406	3480	1740	2
2390.2	1795.8	0.6	1.05E-05	9.6	0.79	110.4	118.9	164.2	13.088	1030	510	2
6868.1	6446.0	3.7	1.75E-05	14.9	0.82	149	269	299.2	47.018	1310	660	2
8088.8	7556.3	7.3	3.5E-05	6.4	2.01	186.6	228	201.2	45.050	1290	640	2
5634.4	4436.9	1.2	2.97E-05	6.6	0.75	128.3	488.3	85.3	31.709	2360	1180	2
7015.9	6200.8	3.5	2.69E-05	12.5	0.58	177.8	256.7	170	44.539	3420	1710	2
3068.6	2602.3	6.3	1.96E-05	7.9	0.78	117.6	321.7	105.1	27.877	3160	1580	2
1763.3	1630.7	1.2	2.91E-05	14.1	0.84	99.9	263.1	219.4	9.534	2530	1260	2
7949.1	7744.6	4.8	1.92E-05	9.8	1.6	150.8	392.6	201.6	16.901	1020	510	2
1878.0	1716.0	2.4	4.42E-05	10.1	1.63	9.6	459.9	94.2	17.888	3320	1660	2
7702.9	7081.7	1.7	2.05E-05	9.7	0.51	147.6	247.3	125.7	28.427	3450	1720	2
4456.1	4336.9	5.9	4.71E-05	8.6	2.25	142	246.6	281.9	21.229	2570	1290	2
5950.3	5854.1	47.7	1.08E-05	11.9	0.52	24.2	120	163.2	27.523	2370	1190	2
3212.2	3062.4	24.7	1.51E-05	15	1.42	109.8	461.5	139.5	17.288	2800	1400	2
4826.4	4774.5	73.8	4.76E-05	10.6	0.66	66.6	134.7	99	5.861	1070	530	2
1565.5	1446.9	99.7	2.26E-05	11.8	1.21	99.8	300.5	180.7	15.317	1030	510	2
4863.9	4396.1	54.2	1.98E-05	12	2.88	74.7	249.8	61.8	39.256	2040	1020	2
3214.3	3068.6	45.2	3.54E-05	11.6	2.68	70.1	251.5	57.3	13.196	3260	1630	2
2374.1	2191.7	26.5	2.68E-05	7.3	1.11	173.6	171	163.6	13.490	1610	810	2
8187.1	8102.6	74.7	4.45E-05	10.8	1.72	43.9	193.1	258.4	37.916	2160	1080	2
7784.7	7053.7	20.4	1.59E-05	9.4	1.7	198.1	446.1	90.4	28.119	3810	1910	2
6974.3	6974.5	58.2	1.53E-05	11.6	2.81	4	235.3	270.8	3.284	2870	1440	2
2785.7	2787.9	246.9	2.65E-05	10.6	2.93	4.9	357.7	283.3	1.037	3870	1940	2
7768.9	7199.4	161.1	1.85E-05	14.2	1.81	187.7	443.5	110	30.872	3400	1700	2
5458.0	5256.4	166.5	3.55E-05	6.9	2.82	124.3	417.3	211.4	38.546	1260	630	2
3960.9	3767.1	198.2	1.09E-05	11.3	0.82	140.8	204.7	157.1	19.164	2670	1340	2
6463.4	6363.1	228.9	4.14E-05	6.7	1.99	116.9	151	243.3	25.049	2140	1070	2
6846.5	6770.4	203.4	1.62E-05	10.7	1.86	24	465.9	113.2	17.978	2450	1230	2
5053.3	5025.8	108.2	1.2E-05	13	1.19	86.3	312.1	193.8	6.277	3380	1690	2
5131.5	5083.6	108.4	1.83E-05	14.5	2.58	23.9	241.6	240.8	30.846	3850	1920	2
4946.2	4655.3	102.6	3.69E-05	6.4	1.19	149.5	467.6	216	34.274	1380	690	2
1936.2	1770.9	171.1	1.6E-05	12.5	1.08	121.6	261.4	189.9	19.715	1760	880	2
5770.8	5452.2	0.3	5.16E-05	14.4	0.86	59	387.4	56.4	3.320	1560	780	2
6614.0	5818.7	0.5	8.93E-05	12.1	1.86	166.5	166.8	117.7	18.919	2230	1120	2

Pi (psia)	P(60)	kf (md)	km (md)	ϕ m (%)	ϕ f (%)	FS (ft)	Tres (F)	h (ft)	Q (MMcf/D)	Main Wellbore Length (ft)	Individual Lateral Length (ft)	# of Laterals
6707.7	5584.1	0.7	9.76E-05	8.3	0.9	94.8	121	135.6	35.745	1560	780	2
8236.6	7599.6	0.5	7.44E-05	6.7	2.9	174.5	341.3	260.3	42.902	2680	1340	2
2989.8	2305.7	0.5	7.55E-05	15	1.99	130.2	191.6	54	11.548	3340	1670	2
2372.0	2166.2	0.2	8.66E-05	6	1.8	32.6	206.8	68	2.737	2380	1190	2
7057.0	6954.0	0.6	9.26E-05	11.4	1.29	187.5	314.7	145.5	3.583	2870	1430	2
5063.5	1980.7	0.7	8.52E-05	11.9	1.75	125.1	153.9	61.3	47.515	2010	1000	2
1995.1	1662.9	0.8	6.42E-05	6	2.57	197.4	156.5	145.2	10.616	1950	970	2
4410.5	3735.6	0.2	9.39E-05	13.5	1.4	166.8	237.5	241.7	28.348	3700	1850	2
3389.7	3292.8	2.1	5.97E-05	6.1	1.15	149.4	267	283.9	11.840	3150	1570	2
5086.0	5015.4	3.2	6.81E-05	9.6	0.54	23.7	482.2	264.3	18.942	2810	1410	2
2119.1	2060.7	9.5	6.12E-05	8	2.66	104.2	253.9	158.9	9.508	3930	1970	2
5937.9	5273.3	1.8	7.18E-05	6.1	2.97	144.1	423.2	97.7	29.012	2290	1140	2
3582.0	3487.1	8.8	6.51E-05	7.9	1.37	15	124	163.1	24.796	3180	1590	2
4092.5	3817.3	4.1	7.62E-05	14.8	2.85	80.9	322.3	117.2	34.962	3400	1700	2
5562.4	5333.1	2.2	7.29E-05	14.3	1.65	15.6	248.1	289.5	49.203	1600	800	2
5011.6	4593.0	7.3	7.71E-05	14.4	2.05	107.1	242.2	107.2	30.594	1370	690	2
3950.9	3663.4	7.8	8.76E-05	8.6	2.07	101.1	445	122.4	28.539	2320	1160	2
4810.6	4793.5	6.9	5.06E-05	12.5	2.97	167.4	115.4	268.2	3.606	2380	1190	2
3004.5	2520.0	34.4	6.43E-05	12.1	2.9	170	416.3	79.3	33.269	1380	690	2
3137.7	3072.7	60.5	5.29E-05	11.7	2.04	34.5	447.4	78.2	12.011	1910	950	2
7720.5	7710.2	95.4	8.06E-05	11.4	0.85	107.5	241	290.9	4.262	1480	740	2
4605.9	4459.5	15.5	8.01E-05	14.7	1.04	143.6	401.5	181.7	13.725	2290	1150	2
3237.8	3206.7	29.5	7.1E-05	7.4	0.61	68.7	414.2	163.4	5.461	3330	1670	2
7500.9	7254.5	33.3	9.23E-05	12.3	0.54	70.9	313.8	180.5	49.586	2000	1000	2
3439.0	3416.0	19.0	8.75E-05	11.3	2.2	38.3	403.5	251	11.840	1530	770	2
6703.0	6055.8	61.2	7.65E-05	9.3	0.8	161.5	437.5	91.1	25.791	3110	1560	2
2999.3	2524.2	38.5	7.19E-05	11.7	0.72	135.3	262	77	19.041	1500	750	2
6367.5	6306.2	32.9	8.91E-05	10.6	2.54	2.4	346.1	296.2	39.002	3880	1940	2
2779.7	2719.1	187.0	6.36E-05	10.3	2.31	83.7	423.7	159.7	13.455	3140	1570	2
3355.0	3333.1	212.2	8.28E-05	9.6	2.37	11.8	182	157.6	9.740	2250	1130	2
7394.3	7258.7	185.6	5.25E-05	13.1	2.85	89.7	459.5	248.7	46.884	2620	1310	2
3624.9	3566.0	151.1	8.49E-05	14.9	1.22	55.8	145.2	75.5	8.592	3340	1670	2
4006.0	3911.0	161.2	8.48E-05	8.6	1.39	181.8	106.2	70.7	3.178	3000	1500	2
7945.7	7852.0	150.7	8.71E-05	10.3	0.78	33.1	364.3	209.5	36.661	2590	1300	2
7577.9	7282.2	196.4	8.48E-05	10	2.68	147.9	211.1	128.2	29.160	1650	820	2
3042.3	2932.3	173.1	5.36E-05	10.2	2.27	145.2	115.2	64.2	5.705	2140	1070	2
2668.5	2611.6	165.8	5.68E-05	8.9	1.62	21.9	300.3	125.9	18.513	1860	930	2
2347.1	2341.2	133.1	9.73E-05	12.3	2.53	191.1	253	262.9	1.816	2340	1170	2
7098.2	6579.5	0.5	0.000228	12.9	1.01	157.1	132.9	299	48.837	2710	1350	2
4233.6	3425.8	0.5	0.000439	10.6	1.44	78.4	319	67.3	21.303	2410	1210	2
8296.0	7925.9	0.4	0.000188	13.8	0.94	132.6	406.6	249.3	36.317	3550	1780	2
7609.4	7561.4	1.0	0.000312	12.6	0.7	122.5	465.2	228.3	5.675	2400	1200	2
7526.8	7216.3	0.5	0.000312	7.4	1.56	177.1	121.2	234.2	21.093	2560	1280	2
1822.4	1386.2	0.2	0.000319	9.5	2.23	123.7	396.5	170.5	8.216	1670	840	2
3987.9	3660.7	0.4	0.000388	11.1	1.29	51.5	499.6	237.2	18.529	1630	810	2
2216.2	2100.1	0.9	0.000401	8.5	1.76	29.6	327.5	156.1	10.458	2660	1330	2
7757.8	7673.0	0.7	0.000105	7.1	2.55	135.8	481.2	279.8	8.623	2770	1390	2
2773.6	2700.4	0.8	0.000191	7.6	2.81	70.9	121.8	149.3	7.942	3860	1930	2
6757.8	6740.4	4.2	0.000458	10.8	0.68	16.9	247.4	234.7	6.705	3860	1930	2

Pi (psia)	P(60)	kf (md)	km (md)	ϕ m (%)	ϕ f (%)	FS (ft)	Tres (F)	h (ft)	Q (MMcf/D)	Main Wellbore Length (ft)	Individual Lateral Length (ft)	# of Laterals
6019.4	5855.7	9.3	0.000163	7.4	1.3	75.9	265.8	128.9	20.202	3910	1950	2
1628.1	1521.6	5.5	0.000258	11.1	0.84	164.2	282.9	197.2	11.093	3000	1500	2
3186.8	3182.4	3.4	0.000437	10.2	1.46	125.2	441.4	196.7	1.237	3830	1910	2
5627.4	5532.1	9.9	0.000367	7.8	1.06	58.7	408.5	156.8	16.706	3080	1540	2
1919.3	1898.0	7.3	0.000126	10.6	1.86	60.7	258	258.3	5.604	1370	690	2
7443.7	7405.6	2.3	0.00012	7.7	2.65	198.4	406.3	267.4	5.834	3650	1820	2
7043.2	7020.9	8.4	0.000417	12.5	1.72	88.4	314.2	139.6	4.439	3820	1910	2
7865.6	7563.2	5.5	0.000401	9	1.36	2.6	164.8	105.4	28.032	1170	590	2
5270.4	4161.1	4.0	0.000317	9	1.67	195.5	264.3	62.8	38.636	1860	930	2
7379.0	7171.8	33.5	0.000103	14.1	2.22	87.7	383.4	145.9	34.021	3530	1760	2
3030.7	2984.2	82.7	0.000254	11.2	2.85	103.3	356.5	248.6	21.699	2680	1340	2
6795.4	6786.5	50.7	0.000282	8.5	2.61	28.1	477.2	157.5	2.257	2630	1320	2
2387.0	2317.4	94.0	0.000231	14.4	0.69	116.4	396.6	211.4	17.835	1570	790	2
7495.8	7355.6	45.7	0.00041	14	0.59	194.9	271.5	277	36.621	3280	1640	2
3596.9	3578.6	93.3	0.000451	13	1.4	171.3	271.2	125.1	2.688	1590	800	2
6250.0	6164.0	72.8	0.000414	7.6	2.09	73.9	387.8	210.5	26.146	2320	1160	2
6235.1	5564.8	18.7	0.000142	12.3	0.52	106.6	325.1	83.2	44.662	1320	660	2
2341.3	2300.3	56.8	0.000375	13	0.91	53.5	288.5	74	7.704	1730	870	2
3278.4	3233.2	44.4	0.000118	14.5	2.79	62.1	467.7	97.1	8.337	2710	1360	2
3328.7	3194.1	144.6	0.000263	8.2	2.68	174	126.7	192.6	28.715	3490	1750	2
7264.3	7013.5	142.9	0.000144	14.7	2.63	134	100.2	200.9	49.163	1320	660	2
7653.9	7503.9	133.6	0.0003	13.2	2.24	107.8	387.2	179.1	45.579	3500	1750	2
2588.6	2548.6	207.7	0.000229	9.1	1.85	123.6	272.4	285.4	15.917	1150	580	2
2736.0	2717.3	199.8	0.000324	8.8	1.31	158.9	363.1	178.5	3.878	1520	760	2
2020.4	2016.3	185.8	0.000208	13	0.61	22.2	279.5	216.8	4.162	3390	1690	2
3885.8	3881.6	246.7	0.000393	13.1	1.91	20.9	145.9	186	3.136	2110	1050	2
2969.4	2936.2	148.2	0.000281	6.3	1.31	13.6	357.1	145.9	9.062	1850	930	2
5574.3	5318.4	170.0	0.000328	7.2	1.48	5.3	288.8	119.3	40.559	3200	1600	2
7617.4	7557.9	194.4	0.000157	12.2	2.45	196.8	181.1	247.3	10.947	2360	1180	2
5114.2	4731.3	0.4	0.000671	10.6	0.65	2.7	202.4	247.7	42.768	3110	1550	2
5436.1	5204.0	1.0	0.000791	8	2.95	59.5	204.3	255.5	36.451	2940	1470	2
3723.2	3670.0	0.9	0.000567	11.6	0.85	105.7	318.4	103	4.026	3560	1780	2
7474.4	7029.2	0.9	0.000945	7.6	1.9	95	341.1	265.6	46.954	1940	970	2
5561.6	5483.0	0.2	0.000527	12.8	2.49	8.3	459.5	126.8	3.660	3450	1730	2
6321.7	6095.1	0.6	0.000927	9.3	2.04	5.9	134	295.3	34.462	2960	1480	2
6711.7	6127.0	0.3	0.000723	8.8	2.2	49.7	486.7	222.6	47.994	3330	1670	2
8414.0	7711.7	0.4	0.000735	6.4	2.34	83	309.6	84.1	25.672	3800	1900	2
6698.4	4165.0	0.2	0.000703	10.1	2.23	37.2	270.5	85	49.415	2170	1080	2
3739.7	3471.7	0.5	0.000629	8.4	2.15	64.4	377.8	133.1	16.919	3030	1520	2
6457.2	6245.8	7.5	0.000691	6.1	2.58	82.3	454	228.8	43.305	1290	650	2
1736.5	1556.4	9.6	0.000727	8.3	0.87	99.1	415.6	59.5	13.658	2530	1260	2
7466.7	7354.1	7.3	0.000843	12.3	1.99	48.7	317.3	195.1	30.315	1760	880	2
4658.8	4486.5	2.5	0.000602	6	2.27	126.6	468.8	217.7	33.715	3880	1940	2
5445.5	4854.9	1.5	0.000661	10.1	1.45	197.9	327.1	82.7	18.915	1330	670	2
3484.5	3381.8	3.5	0.000942	7.1	1.93	23.3	435.7	72.7	7.692	2330	1170	2
4173.3	4015.6	3.9	0.000868	7.4	1.77	130.8	425	100.1	17.928	3910	1950	2
7282.3	7244.5	8.2	0.000791	15	2.92	55.5	349.6	164.9	12.331	3180	1590	2
5906.3	5844.9	9.9	0.000742	13.2	2.21	48.3	254.3	62.8	5.789	2040	1020	2
7219.6	7143.7	5.3	0.000673	10.6	1.29	164.6	473.2	182.5	13.594	2680	1340	2

Pi (psia)	P(60)	kf (md)	km (md)	ϕ m (%)	ϕ f (%)	FS (ft)	Tres (F)	h (ft)	Q (MMcf/D)	Main Wellbore Length (ft)	Individual Lateral Length (ft)	# of Laterals
4526.1	4497.0	27.8	0.00083	8.7	1.38	162.9	268	199.5	7.305	1350	670	2
8018.9	7949.6	34.8	0.00092	13.8	0.95	139.9	231.8	145.4	16.663	3800	1900	2
1890.3	1869.3	96.9	0.000733	11.4	1.31	74.9	135.9	68.9	3.710	3730	1870	2
7848.5	7812.4	66.7	0.000563	9	2.86	50.7	239	202.8	13.657	2720	1360	2
3176.1	3104.8	13.0	0.0007	14.9	1.05	106.3	307.3	155.7	22.265	2660	1330	2
2114.2	2104.9	73.8	0.0009	6.9	0.68	38.6	101.2	286.7	7.043	2750	1380	2
7414.7	7401.7	97.6	0.000876	6.8	0.78	130.9	184.4	269.7	6.027	1380	690	2
6308.0	6276.0	82.6	0.000514	6.1	0.78	179.6	336.6	214.8	6.434	1750	870	2
4385.1	4354.9	52.6	0.000562	11.3	2.31	98.1	180.5	125.1	6.749	2070	1040	2
4060.0	3949.6	35.6	0.000514	9.2	0.85	71.2	148.9	181.4	33.413	2490	1250	2
4843.4	4791.6	223.3	0.000616	14.3	1.56	134.3	116.6	179.9	19.290	2190	1100	2
3144.6	3127.0	157.4	0.000899	11.6	1.55	150.9	192	143.7	5.283	3170	1590	2
7004.8	6819.1	192.1	0.000765	9.7	2.84	76.5	234.3	126.3	48.782	3520	1760	2
4707.5	4672.0	196.0	0.000549	9.4	2.94	153.3	242.7	266.1	17.573	1760	880	2
6583.5	6549.4	223.2	0.000737	13.5	1.18	132.2	229.6	126.7	8.292	1750	880	2
1589.0	1579.1	244.4	0.000616	9.5	2.22	84.8	131.8	227	6.279	2830	1420	2
8110.3	7970.2	150.2	0.000607	6.1	2.71	103.8	217.5	215.5	45.253	2160	1080	2
2646.0	2627.3	199.4	0.000824	10.8	1.97	105.1	109.9	213.6	11.216	1310	650	2
3665.2	3641.4	157.8	0.000548	8.4	2.61	174	437.5	109.6	3.607	1360	680	2
2812.8	2768.7	228.6	0.000982	14.3	1.83	190.8	208.9	162.4	14.446	1950	980	2
5141.4	4514.5	0.3	0.003821	8.3	2.11	179.5	384.9	289	49.936	2600	1300	2
7954.7	7794.4	0.6	0.004311	14.8	1.87	113.8	497.1	138.6	6.770	1540	770	2
4658.4	4641.4	0.4	0.003069	9.1	2.52	77.6	221.2	227.9	1.192	1810	900	2
7968.0	7576.7	0.9	0.002024	10	1.36	93.3	111.5	297.4	43.053	1640	820	2
8334.0	8305.6	0.5	0.001943	9.1	1.68	43.7	424.6	224.8	3.576	3420	1710	2
4713.8	4401.1	0.2	0.004593	13.1	2.64	120.1	490.4	284	38.155	3880	1940	2
6141.0	6038.2	0.2	0.004343	14.2	0.97	120.8	238.2	99.3	2.487	2130	1070	2
2657.6	2514.2	0.9	0.003895	12.4	2.51	95.9	353	179.8	21.968	3240	1620	2
3683.9	3641.1	0.6	0.003685	11.9	0.58	181.8	247.9	284.6	9.881	3840	1920	2
7839.2	7558.5	0.4	0.002144	13.1	0.93	53.2	362.8	282	45.247	3580	1790	2
5981.9	5936.5	6.3	0.002708	14	2.66	137.1	338.9	204.4	17.191	3630	1820	2
3443.9	3370.2	5.7	0.002781	12.3	1.07	162.6	212.2	110.8	12.273	2590	1290	2
6332.1	6151.6	3.1	0.002581	7.2	1.36	113.6	113.1	252.8	38.237	2510	1260	2
2899.0	2837.9	6.3	0.002818	10.7	2.01	9.5	432.8	226.6	16.875	1230	610	2
5520.4	5267.4	5.0	0.001417	8.2	2.67	32.3	174.8	78.5	23.092	2770	1380	2
4588.5	4553.1	9.9	0.004664	10.6	1.58	2.2	164.5	149.2	7.169	1260	630	2
7019.7	6836.6	2.7	0.004253	13.5	2.09	45.4	311.5	90.8	19.576	2880	1440	2
5289.9	4893.0	8.8	0.003006	8.3	2.62	128.6	429.6	69.5	36.340	2920	1460	2
5705.8	5379.1	9.6	0.002675	8.3	1.74	110.1	297.1	111.4	38.180	1280	640	2
7118.1	6498.0	8.6	0.001951	6.9	0.61	27.9	152.6	55.6	29.422	1290	640	2
6859.5	6825.9	68.1	0.004295	11.8	1.43	97.2	249.6	195.9	13.829	3390	1690	2
5834.2	5797.6	27.6	0.002071	14.2	1.23	90.8	438.1	119	7.876	1570	790	2
4832.4	4586.0	51.6	0.001179	6.7	0.61	174.1	396.5	104.6	26.786	2130	1060	2
5415.9	5262.7	95.2	0.004528	7.3	0.75	73.4	145.2	169	32.545	1370	680	2
6955.6	6855.3	81.8	0.002573	12.4	0.6	55.4	197.2	204.3	41.450	2410	1200	2
4593.5	4586.8	89.8	0.003986	9.4	1.64	70	175.2	141.1	1.589	1370	690	2
3399.7	3374.4	96.5	0.00321	6.2	1.69	156	451.4	283.7	14.220	1440	720	2
6737.9	6600.4	36.8	0.003295	8.2	0.51	34.3	241.2	162.7	28.795	2230	1120	2
3328.2	3317.2	58.1	0.003793	9.4	2.23	56.9	232.5	224.9	7.460	3050	1530	2
4235.0	4206.1	12.8	0.003622	7.5	0.6	20.4	379.6	190.6	8.064	2920	1460	2

Pi (psia)	P(60)	kf (md)	km (md)	ϕ m (%)	ϕ f (%)	FS (ft)	Tres (F)	h (ft)	Q (MMcf/D)	Main Wellbore Length (ft)	Individual Lateral Length (ft)	# of Laterals
3977.1	3956.8	184.8	0.00287	13.7	0.66	167.1	173.3	245	14.958	1900	950	2
4313.3	4304.4	169.5	0.001563	13.7	2.04	119.1	497.9	258.1	8.581	1960	980	2
2013.8	2009.6	177.0	0.004449	13.8	1.5	44.5	455.2	252.2	6.120	1310	650	2
4440.0	4425.3	199.8	0.002074	13.6	1.68	137.4	420.8	271.5	13.759	2550	1280	2
6075.7	6063.5	128.4	0.002389	13.3	2.73	157.1	354.7	279.1	11.262	2090	1040	2
8413.9	8350.4	211.1	0.001041	11.7	0.96	176.9	446.2	212	24.766	1130	570	2
4272.5	4206.7	169.2	0.003117	13.7	2	130.7	153.3	179.3	34.180	2690	1340	2
1532.1	1517.2	147.0	0.002189	9.6	2.66	75.4	412.7	283.4	14.875	2570	1280	2
1603.8	1566.0	245.3	0.002296	10.8	1.26	172.7	258.3	232.9	21.275	1500	750	2
5804.0	5703.2	192.7	0.003569	6.7	2.5	124.1	386.7	272.7	47.621	3540	1770	2
8030.3	7382.6	0.7	0.007787	6.9	0.64	73.8	143.6	182	28.172	1390	690	2
3166.0	2399.7	0.6	0.006744	7.4	0.61	179.4	372.9	169.3	28.494	1260	630	2
7817.8	4181.8	0.1	0.007964	6.1	2.98	186.2	376.1	54.5	12.309	1050	520	2
4415.9	4009.2	0.7	0.008057	6.7	2.48	114.8	328	205.8	31.647	2110	1050	2
4369.5	4313.1	0.9	0.009219	6.6	1.82	127.8	108.9	208.9	5.512	2300	1150	2
8025.6	7957.6	0.9	0.00667	8.2	2.01	59	110.3	181.7	6.873	2810	1400	2
1515.4	1456.4	0.6	0.008765	9.4	1.39	75.1	199.4	277.2	7.176	2700	1350	2
5035.0	2019.8	0.3	0.008291	8.7	1.87	112.2	472.9	55.2	44.467	2240	1120	2
3677.7	3428.7	0.3	0.009464	8	0.88	145.3	392.1	258.2	10.057	1200	600	2
3219.8	2980.0	0.3	0.007177	10.7	1.84	105.6	495.7	139.9	8.730	1890	950	2
3983.2	3934.6	8.3	0.009062	11	1.83	105.1	124.5	216.8	21.799	3760	1880	2
1816.2	1765.1	8.8	0.006708	7	1.53	185.9	388.9	202.8	13.637	2050	1030	2
4504.7	4498.2	3.5	0.005909	13.5	1.6	85.9	467.2	176.4	2.106	3690	1850	2
6710.7	6403.0	5.9	0.00715	8.1	2.4	137.2	295.9	74.4	19.634	1060	530	2
4383.1	4282.0	5.8	0.0078	13.7	1.29	48.6	224.6	123	15.294	1440	720	2
5863.2	5822.0	9.2	0.005225	8.6	0.56	160.6	162.7	284.7	16.454	3910	1960	2
6809.2	6464.0	1.2	0.00866	11	1.6	82.7	176	62.6	16.766	3200	1600	2
6597.5	6453.5	5.0	0.006903	7.5	2.61	92.3	163.5	249.4	37.765	2090	1050	2
2350.5	1733.1	2.1	0.008372	10.4	1.61	89.3	478	80.5	24.582	1220	610	2
4305.3	4214.0	3.5	0.005115	13.6	1.68	55.6	288	178.9	28.502	3880	1940	2
1864.0	1851.3	35.0	0.009761	6.4	2.35	197.1	219.6	258.1	6.960	1410	700	2
1844.0	1825.5	61.6	0.005331	8.8	0.61	30.5	488.1	194.4	9.014	1950	980	2
7521.7	7413.6	65.8	0.00814	13.3	2.5	190.7	371.6	196.8	47.006	1580	790	2
1701.7	1682.2	76.5	0.007309	12.5	2.9	74.7	155.8	235.7	19.891	3970	1990	2
4092.9	4051.5	90.8	0.008612	14.9	2.41	30.6	414.6	177.6	24.661	3630	1810	2
5084.6	4967.8	72.8	0.008207	8.9	0.96	122.3	191.3	92.5	15.930	2350	1170	2
8356.2	8206.7	68.4	0.008053	7.2	1.66	14.8	452	191.5	38.535	1970	980	2
6624.8	6628.7	93.2	0.00707	13.7	1.18	37.8	129.1	272.9	1.102	2510	1260	2
8380.3	8208.5	16.2	0.009142	12.5	1.82	108.5	150.6	156.5	49.105	2370	1180	2
2255.3	2252.5	11.4	0.005128	6.1	2.06	175.2	409.5	226.8	1.631	3210	1610	2
5768.4	5514.3	231.7	0.005147	12.3	2.47	35.7	413.3	65.8	36.753	1820	910	2
5967.9	5877.9	104.2	0.005635	14.1	1.6	4.4	354	160.4	33.412	1600	800	2
1695.7	1680.0	216.2	0.006125	13.8	0.84	20.9	425.7	285.9	20.037	1250	620	2
4545.6	4542.6	147.2	0.008915	7.4	2.01	183.4	170	296.8	4.593	3490	1740	2
3805.0	3763.5	200.0	0.006304	8.5	2.69	189.1	323.4	271.1	29.970	1790	900	2
6274.3	6187.2	129.5	0.005357	10.4	0.52	123	220.5	252	41.313	1560	780	2
2225.9	2225.6	217.4	0.005652	8.1	2.65	47.7	465.2	251	1.651	2740	1370	2
7192.9	7132.1	231.6	0.009602	13.9	2.33	85.7	132.3	168.5	26.537	1190	600	2

Pi (psia)	P(60)	kf (md)	km (md)	ϕ m (%)	ϕ f (%)	FS (ft)	Tres (F)	h (ft)	Q (MMcf/D)	Main Wellbore Length (ft)	Individual Lateral Length (ft)	# of Laterals
7379.6	7312.8	222.4	0.008022	7.9	2.72	63	432	286.3	35.165	2270	1130	2
7739.7	7660.1	186.2	0.006766	14.2	1.28	159.8	110	164.6	33.574	2700	1350	2
5047.3	4754.4	0.8	0.019889	13.9	2.64	26.8	166.7	154.1	29.339	2770	1380	2
2547.6	2259.6	0.5	0.016387	8.6	2.32	129.4	363.1	171.4	13.072	1580	790	2
4236.0	3947.8	0.9	0.025136	7.9	2.9	99.7	138.4	58.2	12.796	3620	1810	2
4433.7	4318.9	0.8	0.048591	8.6	2.37	114.3	408.3	194.7	9.154	1970	980	2
3970.4	3642.4	0.7	0.018625	9.9	0.57	45.6	448	61.5	10.723	2910	1450	2
1559.5	1460.8	0.8	0.048438	14.6	1.29	183.9	153.1	114.1	7.775	3030	1510	2
5779.2	5729.9	0.3	0.024957	11.4	2.42	144.1	315.6	212.3	5.577	3990	2000	2
2454.8	2420.8	0.7	0.037417	11.5	1.89	199.2	271.9	185.6	5.363	3630	1820	2
4741.1	4630.4	0.8	0.021028	13.6	1.09	88.6	385.1	293	18.554	2490	1250	2
8045.7	8013.2	0.6	0.019774	8.8	2.45	28.4	393	292.9	4.059	2120	1060	2
8399.7	8160.9	7.8	0.020627	10.2	0.63	158.7	437.5	83.1	19.723	1050	520	2
6333.6	6278.9	8.6	0.032021	7.7	0.8	86.4	374.8	210.1	11.239	1230	620	2
1519.0	1316.6	4.0	0.040603	9.3	1.64	139.4	385.3	83	12.342	1570	780	2
8281.0	8150.9	4.8	0.048765	9.4	2.66	132.5	311.7	264.9	44.427	2540	1270	2
4855.6	4710.4	6.0	0.031262	6.7	2.19	190.2	157.9	207.5	34.059	2660	1330	2
5734.2	5611.4	4.7	0.029025	12.3	2.25	190.4	413	239	39.990	2450	1230	2
4288.4	4256.3	5.5	0.014028	14.5	0.71	98.4	246.9	227.7	14.181	3130	1560	2
7651.5	7554.4	9.3	0.048094	11.3	0.55	104.8	173.2	146.2	22.098	3400	1700	2
2996.1	2909.9	3.6	0.022353	7.3	1.06	108.4	216.9	246.4	16.244	1290	640	2
4986.5	4893.9	5.3	0.014636	8.2	1.29	82.2	396.4	293.3	33.975	3240	1620	2
6278.0	6281.8	80.4	0.012912	12	0.61	173.2	218.8	278	1.054	1380	690	2
8255.4	8174.4	82.5	0.0101	9.9	2.22	42	258.4	253.9	40.136	2860	1430	2
6991.7	6985.5	75.2	0.014923	10.3	2.03	21.5	288.6	257	5.454	3790	1900	2
2757.3	2731.7	53.8	0.020585	13.2	1.72	116.7	456.4	245.8	24.434	3090	1540	2
5269.5	5265.7	49.9	0.047396	9.7	1.46	19.7	464.7	170.9	1.223	2570	1280	2
2635.2	2626.9	47.5	0.014064	14.9	2.89	48.3	340.9	228.4	8.502	1690	850	2
6673.7	5989.1	46.5	0.014882	6	0.65	100.8	246.1	69.5	48.168	3680	1840	2
4018.3	3985.7	38.5	0.034551	12.9	2.2	64.4	483.6	213.5	19.898	2650	1320	2
7329.6	7209.8	12.0	0.011965	7.7	0.95	89.4	294.6	281.5	39.288	1820	910	2
4190.4	4074.0	60.7	0.014356	6.5	2.94	140.1	111.8	147.2	29.815	1890	940	2
2396.0	2393.8	194.7	0.036609	14.4	2.6	53.5	322.4	260.9	5.394	3190	1600	2
6461.4	6386.1	231.4	0.028358	9.2	1.31	43.4	264.2	150.2	18.197	2470	1230	2
4838.9	4765.8	201.7	0.015148	13.5	0.96	119.1	359.1	204	38.739	1880	940	2
6737.4	6634.3	178.9	0.022701	11.1	2.17	101.1	490.6	206.8	44.808	3220	1610	2
6023.5	5731.0	208.1	0.041921	8.3	1.08	144.9	273.5	108.2	47.515	3580	1790	2
4259.6	4224.9	188.0	0.041263	13.6	2.86	69.2	421	267.1	31.157	2130	1070	2
5722.1	5620.5	238.2	0.014366	13.5	0.8	48.8	241.9	168.2	38.046	3430	1710	2
6831.1	6825.5	130.1	0.012318	9.9	2.11	160.4	317.4	220.3	3.707	3220	1610	2
5045.7	5012.3	210.3	0.043948	12.2	1.06	85.8	422.6	63	3.745	1890	950	2
1983.2	1966.2	242.4	0.015607	12.6	1.32	168.6	180.6	116.8	7.163	1800	900	2
7209.4	6286.9	1.0	0.076143	14.2	0.71	1.6	318	91	47.472	2290	1150	2
6075.5	5672.4	0.5	0.059685	14.6	0.74	131.1	407.4	242.9	47.495	2880	1440	2
5531.5	3028.0	0.5	0.060246	9.4	1.59	7.9	154	79.5	35.758	1100	550	2
5385.2	4928.8	0.6	0.065119	8.9	2.14	175.5	178.2	256.9	34.216	1710	860	2
5179.4	5080.7	0.7	0.083448	11	2.93	110.1	296.3	231.9	7.502	1540	770	2
4402.4	3927.9	0.4	0.060105	13.6	1.41	131.8	334.3	72.4	21.830	3900	1950	2
3496.5	3481.2	0.9	0.070906	8.5	0.6	189.3	118.9	242.8	1.423	1420	710	2

Pi (psia)	P(60)	kf (md)	km (md)	ϕ m (%)	ϕ f (%)	FS (ft)	Tres (F)	h (ft)	Q (MMcf/D)	Main Wellbore Length (ft)	Individual Lateral Length (ft)	# of Laterals
4268.4	4149.9	0.6	0.06191	10.7	2.24	48.5	418.8	270.6	13.288	2020	1010	2
2335.7	1850.4	0.3	0.072078	11.5	1.22	117.5	286.6	194.8	23.473	1880	940	2
1603.8	1384.6	0.8	0.097109	6.6	1.64	77	308.7	147.1	17.862	3400	1700	2
8285.4	8246.2	4.9	0.09511	9.6	1.1	164.3	446.1	263.6	10.935	1370	690	2
5967.4	5821.9	4.4	0.075993	13.4	2.68	40.1	130.1	237.7	39.325	1600	800	2
5254.3	5203.6	10.0	0.074171	8.5	2	95.9	206.1	134.3	9.535	3710	1850	2
4903.3	4816.8	10.0	0.073728	11	1.84	54.1	142	52.8	5.956	1310	660	2
3389.7	3273.3	2.4	0.059968	12.3	2.67	112.6	158.9	121.8	18.769	3070	1530	2
3686.9	3402.8	4.2	0.074648	8.5	1.76	50.6	180.3	51.8	20.481	3150	1570	2
8116.7	7889.3	1.1	0.056351	12.8	2.22	28.5	333.6	214.8	25.351	1770	880	2
6944.2	6851.9	8.8	0.068646	11.6	1.38	24.8	129.1	295.6	46.723	2830	1420	2
4433.4	4267.0	1.7	0.084084	8.4	1.88	55.4	183.1	172.3	16.855	1800	900	2
5006.1	4875.7	3.1	0.055268	6.3	1.19	123.3	273.3	264.9	31.544	3450	1720	2
5060.7	4896.8	89.4	0.075176	11	1.93	48.2	271.6	131.9	42.866	2990	1500	2
5299.0	5018.7	43.7	0.053118	9.5	1.5	150.9	312.5	64.3	29.109	3300	1650	2
6061.4	5937.1	29.8	0.060045	12.4	2.27	152.3	250.3	82.4	18.806	1390	700	2
4976.8	4953.1	49.4	0.083038	8.1	1.33	155.5	230.3	224.1	8.452	3340	1670	2
4553.2	4521.0	75.1	0.072536	11	2.39	84.7	403	89.7	6.057	2140	1070	2
5679.5	5588.3	70.7	0.095678	7	1.13	27	136.1	279.3	33.633	3270	1640	2
5897.1	5823.8	19.7	0.06897	6.4	2.89	1.9	271.5	124.9	11.426	2090	1050	2
8485.2	8378.5	88.7	0.093871	11.4	1.99	195.5	395.8	69.3	14.185	3290	1650	2
1894.3	1876.9	93.7	0.069522	13.9	2.58	181.1	301.8	88.2	5.848	2120	1060	2
5305.5	5230.7	14.6	0.099488	6.1	1.91	29.6	377.2	283.7	25.721	2920	1460	2
7600.5	7576.5	130.3	0.083193	6.6	1.79	191.4	151.3	96.5	2.502	1230	610	2
8003.1	7895.2	221.7	0.071719	6.8	2.57	72.9	180.1	211.8	33.445	1350	680	2
8121.0	8004.0	211.2	0.083671	9.1	2.26	150.4	409	176.1	36.132	2310	1150	2
4436.2	4426.5	110.2	0.075678	8.2	0.56	70.1	297.4	96.3	1.094	2020	1010	2
4749.0	4549.7	240.9	0.091122	8.3	0.74	114.1	115.4	107.5	32.968	3820	1910	2
5341.9	5288.5	100.6	0.054105	14.5	2.91	161.4	155.5	206.3	29.648	2970	1490	2
6561.8	6487.2	127.1	0.059912	14.6	2.07	20.9	422.5	70.3	12.460	3440	1720	2
7454.4	7359.5	219.1	0.079173	6.7	2.15	46.2	144	293	40.944	3910	1950	2
6913.0	6882.6	193.5	0.085091	13	2.58	66.1	386.7	77.7	4.788	2730	1360	2
1861.5	1827.4	112.8	0.086676	9.6	1.88	108.2	246	150.4	16.053	2980	1490	2
5183.4	3606.3	0.6	4.21E-08	12.6	1	187.5	335.6	78.8	28.838	3210	1600	2
8137.6	7628.0	0.7	1.79E-08	7.3	2.01	170.7	373.8	165.9	20.910	2210	1110	2
8355.6	8222.3	0.3	2.76E-08	14.4	2.92	8.2	490.7	296.1	14.787	2640	1320	2
6799.8	5475.5	0.3	2.98E-08	6.1	1.45	85	144.3	108.2	44.119	3600	1800	2
7784.2	7716.9	0.9	2.21E-08	6.4	1.7	87.3	157.8	158.2	2.606	1120	560	2
7067.3	6435.3	0.9	3.98E-08	9	1.49	103.1	216.4	153.2	48.335	3770	1890	2
3728.3	3490.0	1.0	3.8E-08	14.5	2.41	135.7	215.3	206.2	15.818	1870	940	2
7137.9	6084.8	0.5	1.58E-08	8.2	2.6	119.4	420.5	95.5	36.949	3460	1730	2
6253.2	6181.9	0.7	3.35E-08	9.4	2.39	188.3	351.5	168.4	3.797	3160	1580	2
1731.5	1245.8	0.9	3.92E-08	8.9	2.4	83.4	146.8	84.2	14.047	2610	1310	2
4554.3	4519.0	4.8	2.97E-08	10	1.95	125.5	390.1	143.2	2.979	2270	1130	2
7592.9	6805.0	1.6	4.49E-08	10.4	0.72	158.7	352.4	165.4	36.815	1840	920	2
1904.3	1807.9	7.7	4.28E-08	6.1	2.58	199.7	322	117.8	6.937	1630	820	2
5657.9	5601.6	7.9	4.27E-08	13.6	1.1	144.2	126.8	196.2	5.588	3180	1590	2
3610.4	3515.5	3.2	2.5E-08	7.9	1.16	52.9	289.8	185.2	18.601	3870	1940	2
5685.4	5558.6	7.1	4.54E-08	7.3	2.7	168.7	144.9	109.9	7.032	1400	700	2

Pi (psia)	P(60)	kf (md)	km (md)	ϕ m (%)	ϕ f (%)	FS (ft)	Tres (F)	h (ft)	Q (MMcf/D)	Main Wellbore Length (ft)	Individual Lateral Length (ft)	# of Laterals
2156.2	2127.3	5.6	3.26E-08	6.9	1.09	13.9	229.9	68.7	1.822	1600	800	2
7338.7	7268.1	3.1	1.64E-08	8.8	2.75	80.4	369.9	216.6	13.798	3360	1680	2
7075.4	7035.2	2.1	4.77E-08	12.4	1	34.2	215.4	94.1	2.114	1420	710	2
3483.0	3347.6	3.2	2.33E-08	12.6	2.45	186.5	497.3	296.6	24.606	3320	1660	2
4502.4	4464.8	77.5	2.23E-08	12.5	1.49	185.4	457.1	287.2	5.621	1630	810	2
4750.4	4409.2	59.4	1.26E-08	14.3	1.57	161.2	378.9	159.3	28.801	1780	890	2
7636.3	7592.0	76.7	3.09E-08	12.1	0.83	18.7	251.9	249.4	24.291	3300	1650	2
6303.0	5774.4	76.9	3.85E-08	9.9	0.93	155.2	244.4	183.3	47.184	2910	1460	2
6554.1	6517.2	99.3	3.11E-08	10.5	1.97	46.3	336.4	294	21.099	1940	970	2
6102.5	5809.1	83.1	3.93E-08	11.7	2.96	193.3	304.1	123.1	22.497	3630	1810	2
7488.8	7470.1	31.4	1.52E-08	14.1	1.07	15.1	341.3	238.8	10.768	1470	730	2
6493.8	6442.7	96.5	2.84E-08	6.8	2.37	168.8	440	64.7	1.785	2220	1110	2
4659.7	4305.2	69.3	1.96E-08	8.1	2.48	181.6	251.2	83.9	18.574	3900	1950	2
4510.8	4383.3	43.0	1.21E-08	6.4	2.92	60.4	126.3	113.3	18.643	2460	1230	2
4967.0	4961.0	108.9	2.57E-08	9.8	0.89	100.9	107.4	280.6	2.706	2630	1320	2
6691.0	6619.0	231.4	3.98E-08	9.7	0.55	51.9	217.4	252.4	32.208	1970	990	2
4323.6	4224.8	106.5	1.12E-08	7.7	0.96	28.6	244.2	137.6	21.378	1790	890	2
5471.2	5224.7	188.8	4.44E-08	7.1	2.17	181.2	315.9	250.2	36.337	1400	700	2
3345.7	3269.0	244.2	2.83E-08	11.5	1.03	139.7	262.1	286.9	14.857	1160	580	2
8021.7	7643.7	110.2	1.74E-08	12.3	2.63	134	287.8	124	36.717	2270	1130	2
7364.0	7269.6	185.2	3.28E-08	13.1	2.5	8.3	222	182.8	42.329	1950	980	2
6824.3	6817.6	153.0	2.6E-08	7.5	2.48	9	264.3	184.1	2.281	2470	1240	2
6720.4	6708.5	245.4	3.33E-08	13	0.92	98.9	107.9	261.1	5.088	3430	1720	2
3885.7	3804.9	167.6	1.03E-08	6.5	0.57	137.2	422.6	262.6	13.808	3170	1580	2
8363.4	5488.9	0.2	7.89E-08	9.9	1.18	133.5	482.2	162.1	40.474	1340	670	2
3287.5	3063.1	0.9	9.88E-08	6.2	1.9	175.5	302.9	184.2	8.735	1280	640	2
3333.0	1835.4	0.5	7.75E-08	13.4	2.9	188.9	299.5	67.3	29.297	3000	1500	2
3573.5	3500.5	0.5	6.7E-08	8.4	1.41	53.3	341.8	167.6	2.223	1210	600	2
7001.1	6213.0	0.9	6.33E-08	6.6	0.57	99.3	299.7	142.8	43.047	3220	1610	2
5984.8	5898.6	0.7	7.05E-08	7.3	0.71	69.8	142.8	56.3	2.078	3640	1820	2
5843.9	5429.1	0.6	7.81E-08	10.2	2.48	74	190.7	175.4	23.946	2130	1060	2
3887.2	3861.0	0.1	9.58E-08	11.9	2.04	38.1	421.2	260.2	1.935	2660	1330	2
7294.4	6561.3	0.8	5.59E-08	13.6	2.16	91.2	445.4	132.5	31.228	1730	870	2
5503.5	5460.9	0.5	8.08E-08	6.9	2.45	93.1	104.9	200.3	3.911	3840	1920	2
3090.6	3037.9	4.3	6.37E-08	12.7	0.65	7.5	453.3	144.5	11.380	2520	1260	2
3656.2	3490.6	5.6	9.27E-08	14.7	2.95	180.3	247.3	182	20.662	2610	1300	2
3466.7	3436.3	2.8	9.87E-08	13.1	2.91	84	259.4	236.6	7.413	3230	1610	2
6943.1	6891.8	9.5	7.06E-08	15	2.89	109.7	130.1	271.4	13.560	2800	1400	2
3532.6	3491.4	9.3	9.93E-08	6.4	2	191.7	463.3	106.9	2.313	1910	960	2
7733.4	7482.9	1.6	6.68E-08	12.6	1.86	50	161	218.8	34.687	1980	990	2
7550.6	6861.4	1.2	6.75E-08	8.1	2.8	102.6	428.6	117.1	35.463	2150	1070	2
5145.9	4514.8	8.9	6.54E-08	11.2	1.61	198.3	401.9	119.6	31.156	3300	1650	2
3894.7	3860.6	2.3	7.25E-08	14	0.57	197.3	281.6	143.8	1.077	1310	650	2
7457.0	6808.2	9.1	8.77E-08	13.2	1.86	107.1	499.9	56.7	28.027	2440	1220	2
8123.7	8116.2	13.8	8.52E-08	6.3	0.8	105.4	424.5	190	1.022	2000	1000	2
2161.4	1970.8	24.0	6.24E-08	7.8	0.73	182.1	348.3	122.5	8.134	3250	1630	2
4264.6	4213.5	38.5	5.68E-08	11.2	2.28	23.2	487	278	30.306	1030	520	2
4420.2	4164.9	12.3	7.92E-08	7.7	0.84	162.2	357.1	230.2	23.569	2270	1140	2
6842.6	6787.6	74.7	8.04E-08	13.5	0.51	86.4	283.3	239.6	14.904	3950	1970	2

Pi (psia)	P(60)	kf (md)	km (md)	ϕ m (%)	ϕ f (%)	FS (ft)	Tres (F)	h (ft)	Q (MMcf/D)	Main Wellbore Length (ft)	Individual Lateral Length (ft)	# of Laterals
7645.1	7085.7	24.3	6.94E-08	8	1.63	147.3	257.1	60	17.781	2230	1120	2
5087.3	5021.4	43.0	5.06E-08	7.3	0.71	72.7	245.5	243.7	14.992	2640	1320	2
5373.5	5174.7	27.5	9.73E-08	12.5	2.9	31.4	258.7	97.8	33.352	1260	630	2
5247.2	4770.1	16.4	7.59E-08	6.2	2.3	198.5	473.4	65.3	15.441	2580	1290	2
6273.0	6165.6	56.7	6.44E-08	9.8	1.25	64.2	232.7	273.3	36.962	2160	1080	2
5876.5	5745.4	216.1	7.02E-08	12.5	1.86	56	311.5	167.7	41.083	1990	990	2
3290.0	3208.5	110.7	8.68E-08	8.4	2.33	154.4	284.6	210.1	13.534	3130	1560	2
4456.0	4198.9	227.5	5.35E-08	8.5	2.38	63.4	198.1	57.3	24.401	1260	630	2
4535.7	4200.2	202.7	9.79E-08	14	2.67	193.7	220.2	146.4	32.517	1990	1000	2
6713.4	6449.3	206.6	6.46E-08	8.9	1.77	193.7	408.4	113.7	15.099	3470	1740	2
3444.8	3408.3	183.2	6.1E-08	11.6	0.85	28.8	467.1	230.6	23.906	2120	1060	2
7732.5	7656.5	130.0	9.6E-08	8.8	2.97	39.1	102.5	145.9	19.294	2110	1050	2
3948.7	3923.8	225.0	8.35E-08	7.1	1.75	9.2	487.7	159.7	7.445	3290	1650	2
8248.4	8216.7	103.3	7.51E-08	9.8	2.63	70.3	460	298.4	16.741	3430	1720	2
5280.2	5276.4	118.7	6.18E-08	14.4	1.49	21.9	257.7	209.5	3.063	3550	1780	2
2737.2	1874.4	0.4	1.45E-07	6.8	2.96	172.7	323.8	66.9	12.645	2440	1220	2
1779.0	1165.3	0.7	2.22E-07	6.7	1.1	130.1	233	144.3	17.138	2410	1210	2
3499.2	2430.4	0.3	1.23E-07	7.5	0.53	179.3	478.6	223.6	22.154	1320	660	2
2684.1	2365.4	1.0	4.6E-07	9.9	2.97	152.5	442.2	69.8	11.295	3640	1820	2
3720.6	3019.7	0.9	3.84E-07	6.7	1.35	197.8	147.5	165.2	29.192	2310	1160	2
4248.5	2508.5	0.1	4.51E-07	9	1.86	15.4	385.1	103.9	29.761	1940	970	2
4816.2	4675.1	0.3	4.67E-07	6.3	2.26	11.4	425.4	141.6	8.351	3950	1970	2
5050.7	4352.3	0.4	4.1E-07	6.4	2.84	44.7	375.7	109.2	29.376	3060	1530	2
8083.2	7990.5	0.8	3.85E-07	10.3	1.93	2.7	461	150	9.313	3510	1750	2
2334.5	1402.1	0.5	4.38E-07	13.3	1.69	194.4	489	202.9	22.290	1510	760	2
3454.4	3163.0	9.4	4.44E-07	6.9	1.54	135.8	347	81.3	14.789	3780	1890	2
3255.3	3111.6	1.2	3.98E-07	9	2.27	75.2	452.3	242.7	22.035	2950	1480	2
7997.0	7778.0	2.4	2.21E-07	10.8	1.38	60.8	395	145.2	25.154	2690	1340	2
5903.2	5667.5	1.3	1.39E-07	9.8	0.73	89.6	476.9	177.6	20.912	2920	1460	2
2237.1	2234.4	9.1	3.96E-07	6.4	2.85	197.6	394.3	265.1	1.062	3060	1530	2
4100.1	4066.5	1.8	1.53E-07	12	2.8	59.1	181.1	162.6	5.897	3960	1980	2
3970.3	3925.4	2.2	1.82E-07	14.2	1.35	19.7	234.4	273	9.073	1380	690	2
2987.9	2926.1	2.7	2.79E-07	6.6	0.75	149.1	287.7	158	3.771	3360	1680	2
7248.5	6548.9	2.9	1.34E-07	12.8	0.74	173.3	338.3	146.2	30.981	1810	910	2
4062.5	3711.9	2.4	4.47E-07	12.4	0.93	131.1	188	124.1	20.652	3120	1560	2
3077.6	2945.9	38.7	4.57E-07	11.4	2.15	156.5	319.4	236.7	23.803	3440	1720	2
3863.4	3806.4	86.0	3.62E-07	10.2	0.56	179.3	407.6	273.1	6.029	1070	540	2
6298.1	6296.7	66.9	4.94E-07	8.6	2.44	2.3	190.7	290.5	3.667	1490	750	2
6328.1	6152.6	15.1	2.49E-07	13.6	0.91	173	315.1	119.5	8.035	3860	1930	2
5247.2	4912.9	66.9	4.44E-07	6.5	2.47	13.4	202.4	68.4	30.570	2120	1060	2
3054.7	2937.1	14.1	4.56E-07	10.1	1.47	26.9	476.2	137.4	32.263	2820	1410	2
4070.6	4007.1	82.8	2.65E-07	13.5	0.75	30.5	315.5	142.9	20.878	1740	870	2
2364.2	2347.1	10.5	4.09E-07	9.3	1.61	23.4	252.6	179	4.746	1550	780	2
3250.9	3250.0	42.6	2.02E-07	14.9	2.55	63.5	381.8	293.8	2.485	1280	640	2
1724.5	1627.3	58.0	1.13E-07	8.2	0.9	120.9	423.9	68	3.454	1630	820	2
4069.6	3919.6	210.8	4.92E-07	11.2	1.2	123.4	425.2	182.8	22.287	3410	1700	2
6652.5	6503.5	186.4	4.53E-07	13.4	0.88	18.1	313.5	67.5	23.432	2750	1370	2
2106.3	2051.5	114.8	4.19E-07	13.4	1.42	135	425.8	165.1	5.962	1480	740	2
3308.3	3275.2	128.8	1.42E-07	6.3	1.76	65.2	298.8	162.9	7.446	1490	740	2

Pi (psia)	P(60)	kf (md)	km (md)	ϕ m (%)	ϕ f (%)	FS (ft)	Tres (F)	h (ft)	Q (MMcf/D)	Main Wellbore Length (ft)	Individual Lateral Length (ft)	# of Laterals
2272.9	2219.9	139.3	2.65E-07	10.4	1.2	106.2	178.6	225.1	9.716	1260	630	2
3920.3	3814.9	179.2	1.88E-07	13.1	1.7	86.7	460	249.6	34.730	1370	690	2
2547.7	2532.6	186.3	4.77E-07	12	2.15	6	146	94.5	4.471	1120	560	2
7458.9	7034.2	177.3	2.93E-07	9.2	1.44	198	200.3	173.4	33.069	3280	1640	2
5482.4	5119.9	153.7	4.21E-07	10.1	1.93	191.8	218.6	146.4	25.998	1140	570	2
7924.6	7446.4	235.5	3.76E-07	14.7	2.53	140.2	413.7	119.1	45.872	2260	1130	2
8489.7	8171.9	0.5	9.15E-07	10.2	0.7	164	368.2	122.7	9.336	3520	1760	2
6963.5	6638.0	0.4	8.95E-07	13	0.52	172.8	279.9	225.1	11.374	2020	1010	2
8074.2	7748.6	0.9	9.49E-07	8.6	2.38	134.2	463.4	105	8.880	1490	750	2
5237.5	5229.2	0.2	5.75E-07	14.3	1.66	56.6	420	261.2	1.164	3250	1630	2
4534.0	3915.0	0.2	5.07E-07	8.6	2.14	40.5	117.2	61.9	6.332	1890	940	2
3384.1	2004.5	0.3	9.83E-07	8.7	1.55	12.9	489.8	116	27.618	1430	710	2
5583.2	4842.9	0.9	9.78E-07	10.5	1.68	14.4	179.3	62.4	22.198	2240	1120	2
6634.2	4439.5	0.6	8.27E-07	12.9	1.06	188.5	260.2	51	15.413	1080	540	2
7740.2	7595.8	1.0	8.6E-07	11.4	1.19	21	414.9	250.2	15.734	1690	850	2
8478.5	8172.4	0.3	6.98E-07	12.2	1.46	96	269.3	245.5	18.519	2570	1280	2
4300.7	4177.0	7.9	8.11E-07	8	1.67	151.4	183.4	279.4	19.684	3460	1730	2
2996.4	2935.2	4.7	6.47E-07	9.2	2.45	137.2	208.2	232.5	10.079	2460	1230	2
4960.0	4811.7	1.1	7.02E-07	12.1	2.87	30.4	474.7	157.3	21.205	3630	1820	2
7947.8	7765.3	7.3	9.12E-07	11.9	0.89	104.5	301.1	202.6	25.313	2270	1140	2
5350.0	5230.2	4.4	5E-07	9.4	1.69	117	409.8	274.7	20.248	2640	1320	2
7820.2	7695.4	9.1	6.12E-07	8.5	0.79	26.5	243	248.1	33.108	1590	790	2
4706.4	4473.7	9.2	8.94E-07	11.3	1.62	82.6	287.1	157.1	36.396	3460	1730	2
5887.1	5762.7	5.6	5.5E-07	14.2	2.33	60.7	249.4	202.7	28.758	2320	1160	2
7773.8	7666.5	2.4	9.37E-07	12.9	1.23	145.4	477.9	203.3	9.692	2880	1440	2
6013.1	5964.2	5.7	9.68E-07	13.2	1.93	193.1	352.6	204	4.278	1890	950	2
5501.8	5265.7	40.7	7.09E-07	11.9	2.3	21.9	293.7	107.8	49.304	2730	1360	2
7687.0	7665.2	88.5	5.09E-07	12.4	1.83	116.1	252	180.5	3.392	1750	870	2
2536.1	2487.9	47.8	9.79E-07	6.5	1.37	177.9	200.9	111.2	2.690	2110	1060	2
6674.8	6671.6	57.7	6.37E-07	6	2.38	80.3	173.3	285.9	2.672	1500	750	2
5656.5	4843.1	87.3	9.29E-07	8.1	0.94	75.3	483.2	54.6	44.786	1660	830	2
3542.5	3371.6	20.0	8.6E-07	14.9	2.17	123.4	366.3	166	23.614	2140	1070	2
5295.6	5249.8	22.1	5.3E-07	13.3	2.91	99	379.9	189.2	8.663	1960	980	2
3741.8	3630.6	48.6	8.65E-07	6.9	1.94	199.1	230.2	260.5	16.310	1100	550	2
8339.7	8282.5	70.0	8.63E-07	13.3	0.9	194	421.2	277.5	6.335	3910	1960	2
6261.2	6259.9	57.1	5.14E-07	14.8	1.04	46.7	487.6	273	3.305	2690	1350	2
6306.8	6308.6	119.1	8.88E-07	11.1	2.44	31.5	419.1	251	1.349	3980	1990	2
2660.1	2648.1	243.3	8.45E-07	11.2	2.05	130	119	167.2	1.779	2560	1280	2
5511.2	4700.7	230.7	5.88E-07	10.9	1.97	155.5	221.7	86.5	46.933	2660	1330	2
3248.4	3201.2	240.9	9.62E-07	9.6	2.3	153.7	281.5	62.3	2.183	3550	1780	2
7083.0	7034.6	182.7	8.99E-07	10.6	1.48	82.9	252.3	222.9	14.623	1720	860	2
4224.1	3847.0	192.1	6.39E-07	8.3	1.42	184.6	259.9	161.2	30.689	3220	1610	2
2298.7	2258.9	193.9	8.95E-07	7.8	1.66	64.8	172.2	231.7	14.829	2750	1380	2
2605.1	2503.4	241.4	6.7E-07	14.4	1.09	172.3	410.9	278.8	13.991	1080	540	2
7589.9	7543.6	212.9	7.57E-07	10.3	2.98	139.9	350.4	257.7	11.520	1820	910	2
3620.5	3509.9	214.1	6.82E-07	13.9	2.34	143	225.6	187.6	17.233	3080	1540	2
5394.1	4116.9	0.3	2.28E-06	7.2	2.75	81.8	164.1	149.3	41.474	2310	1160	2
2251.3	1344.9	0.1	3.04E-06	9.6	2.08	110.8	229.6	98	9.551	2470	1230	2
2999.0	2816.3	0.3	1.01E-06	9.8	2.03	193	435.1	212.6	9.938	3720	1860	2